

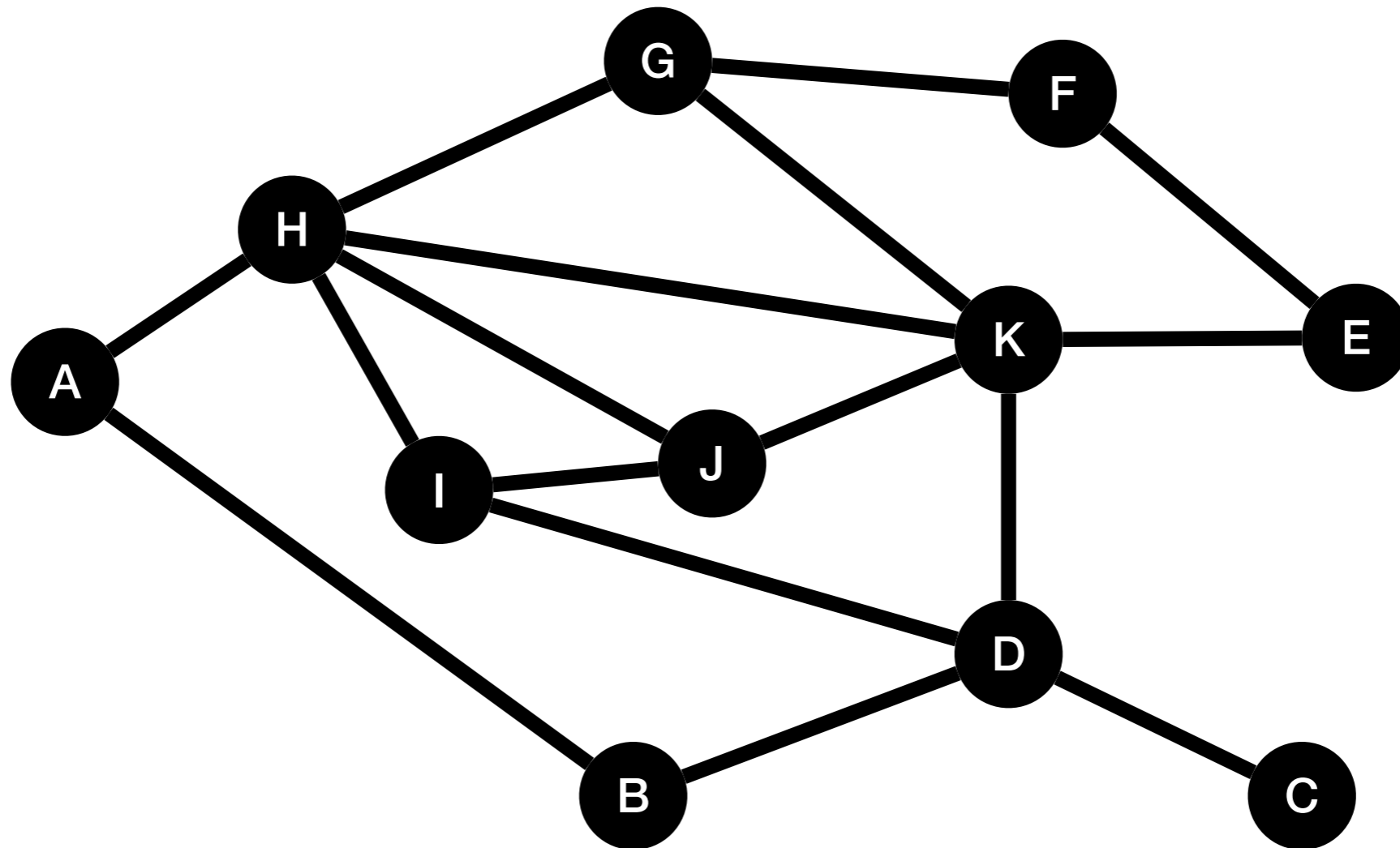
Science informatique

CM5

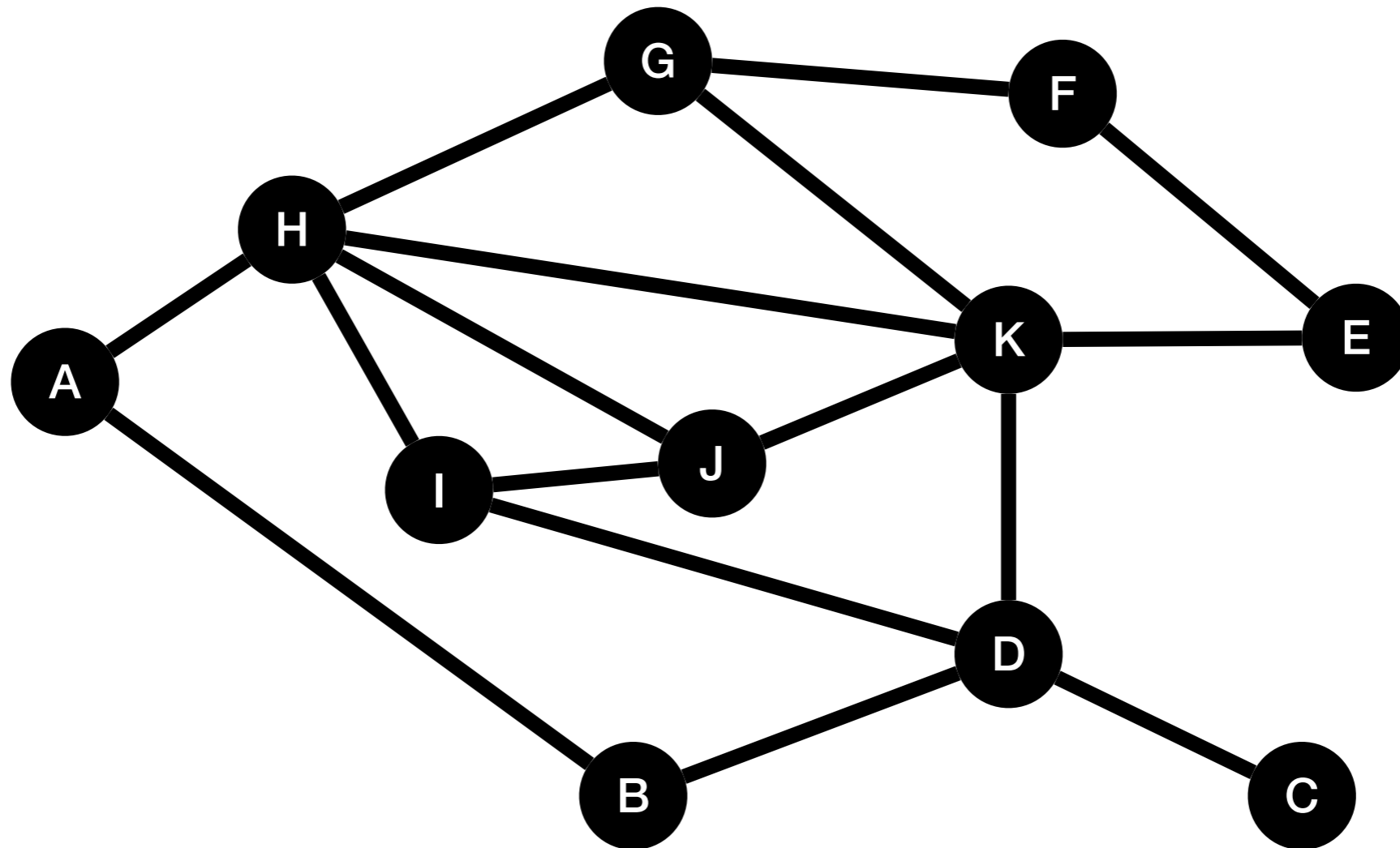
Antonio E. Porreca

aeporreca.org/scienceinfo

Algorithme de Welsh-Powell



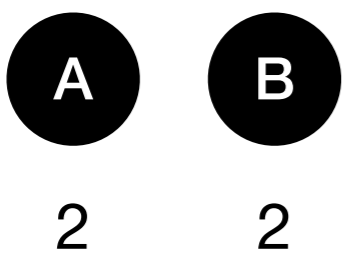
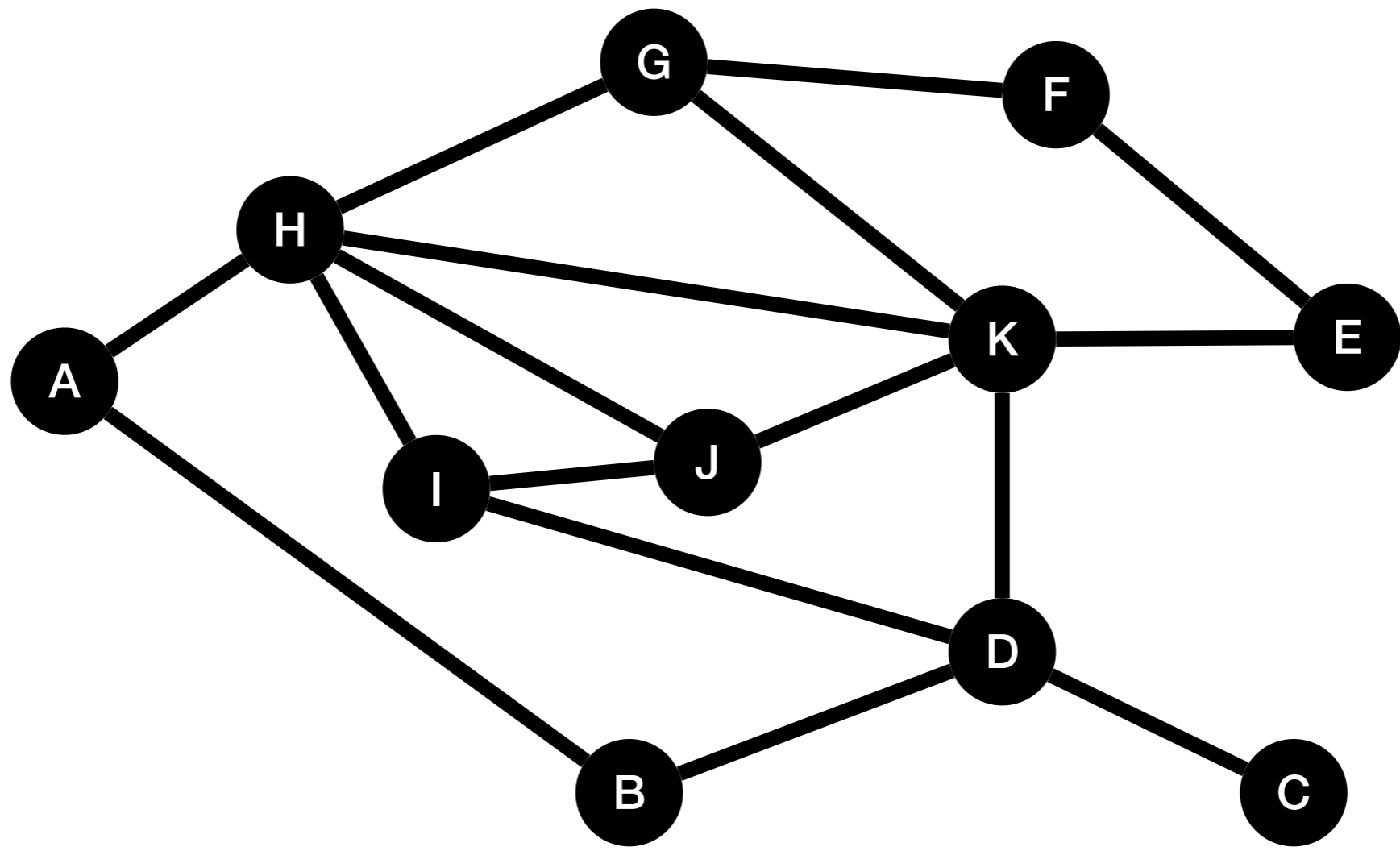
Algorithme de Welsh-Powell



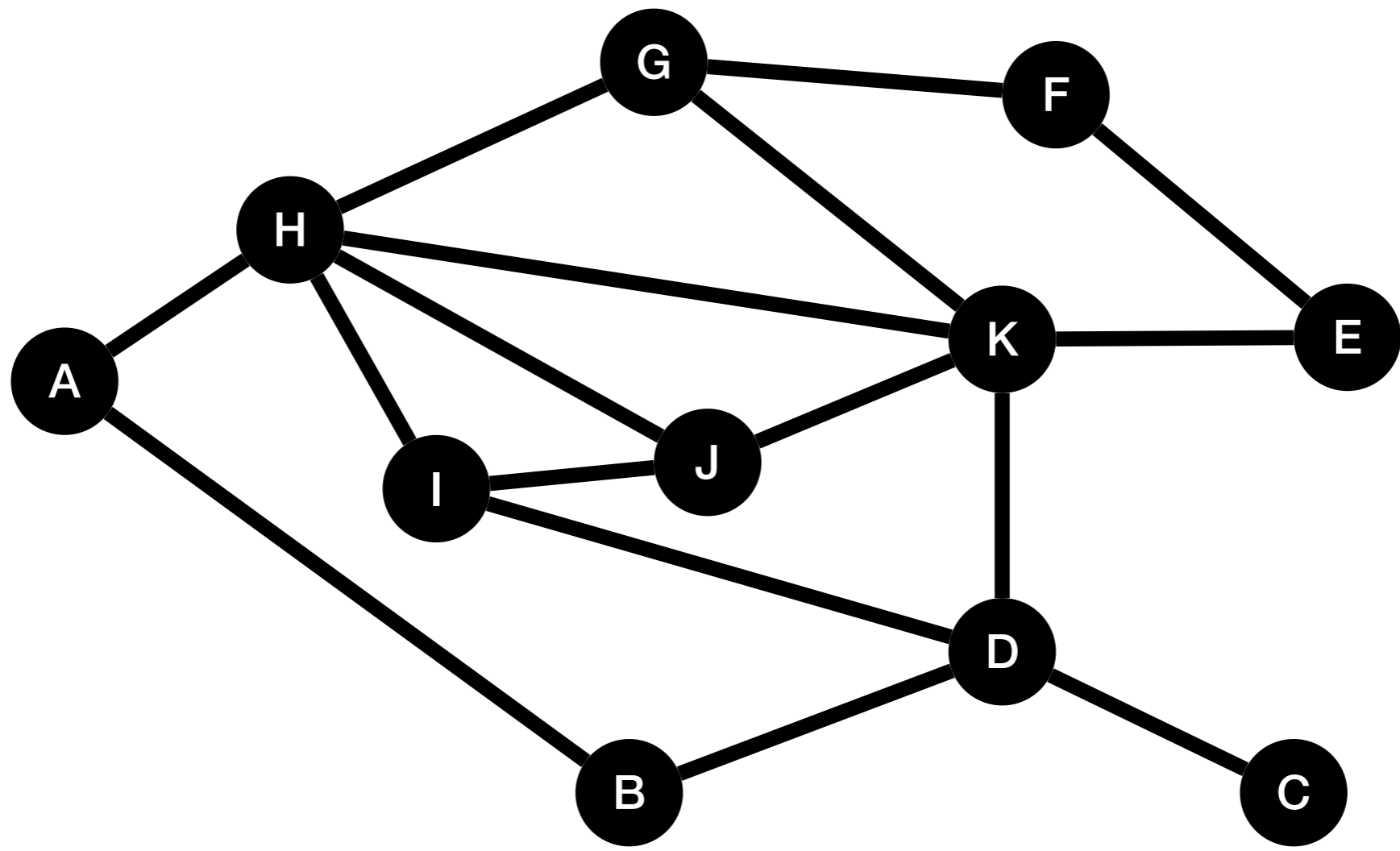
A

2

Algorithme de Welsh-Powell

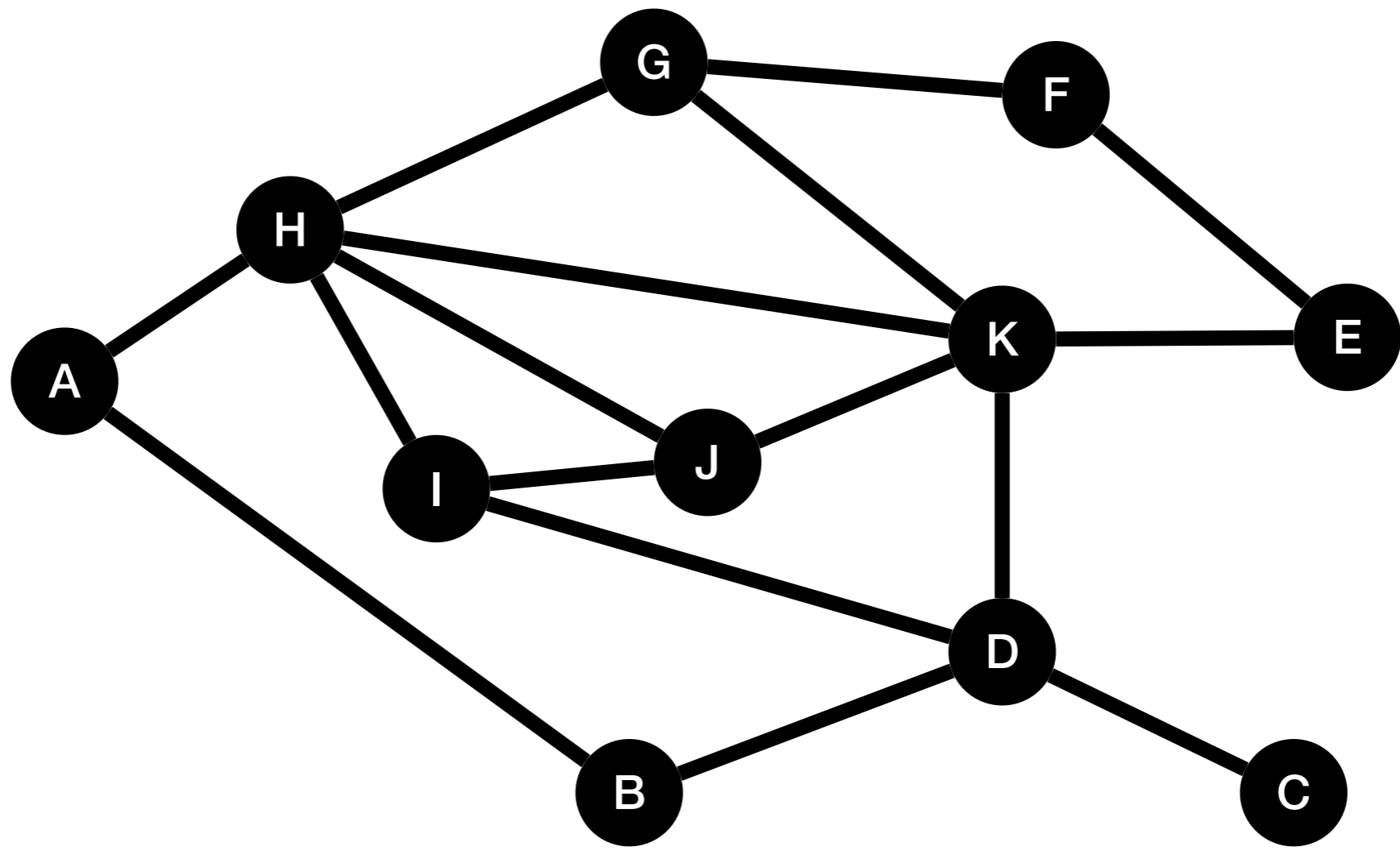


Algorithme de Welsh-Powell



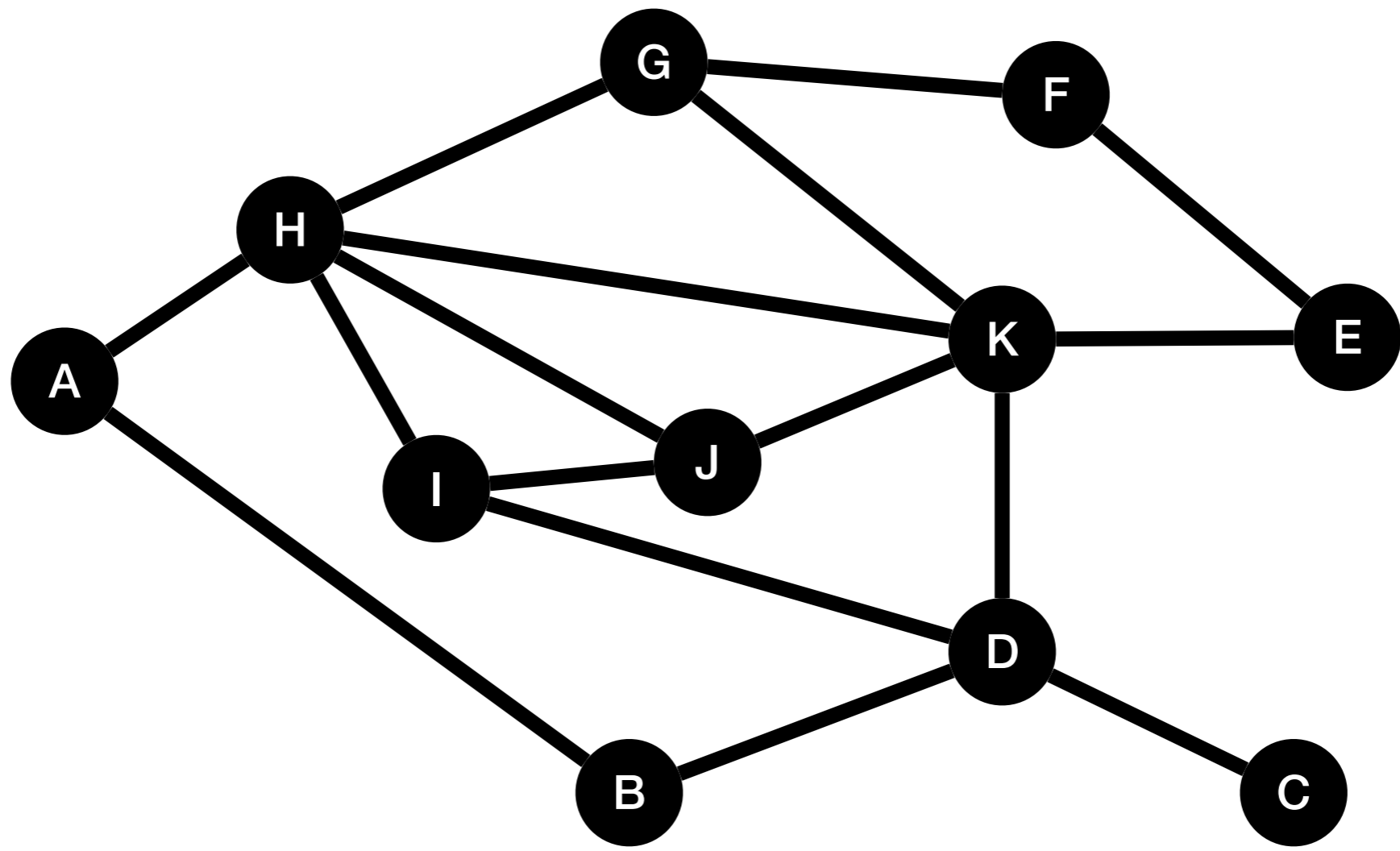
- A**
2
- B**
2
- C**
1

Algorithme de Welsh-Powell



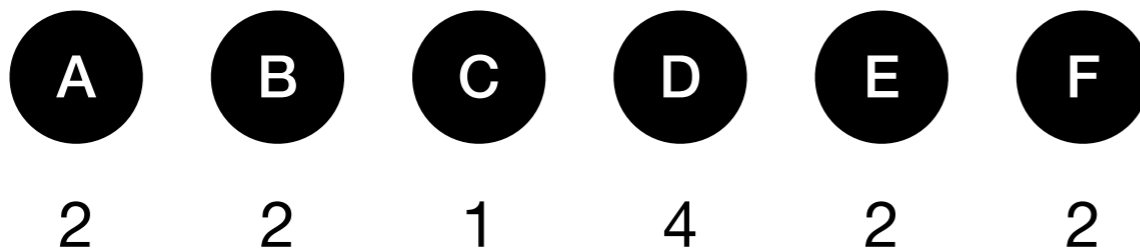
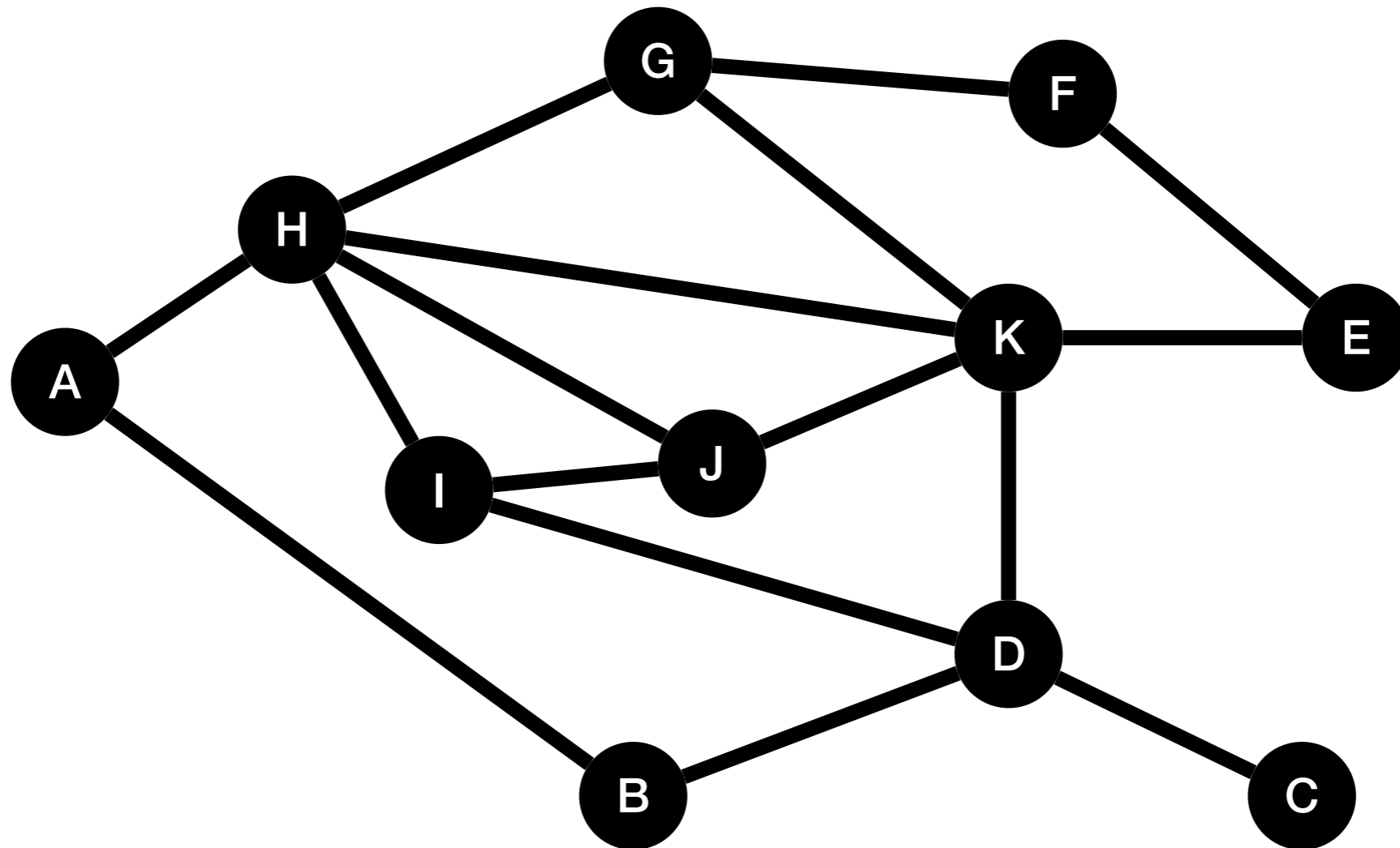
- A**
2
- B**
2
- C**
1
- D**
4

Algorithme de Welsh-Powell

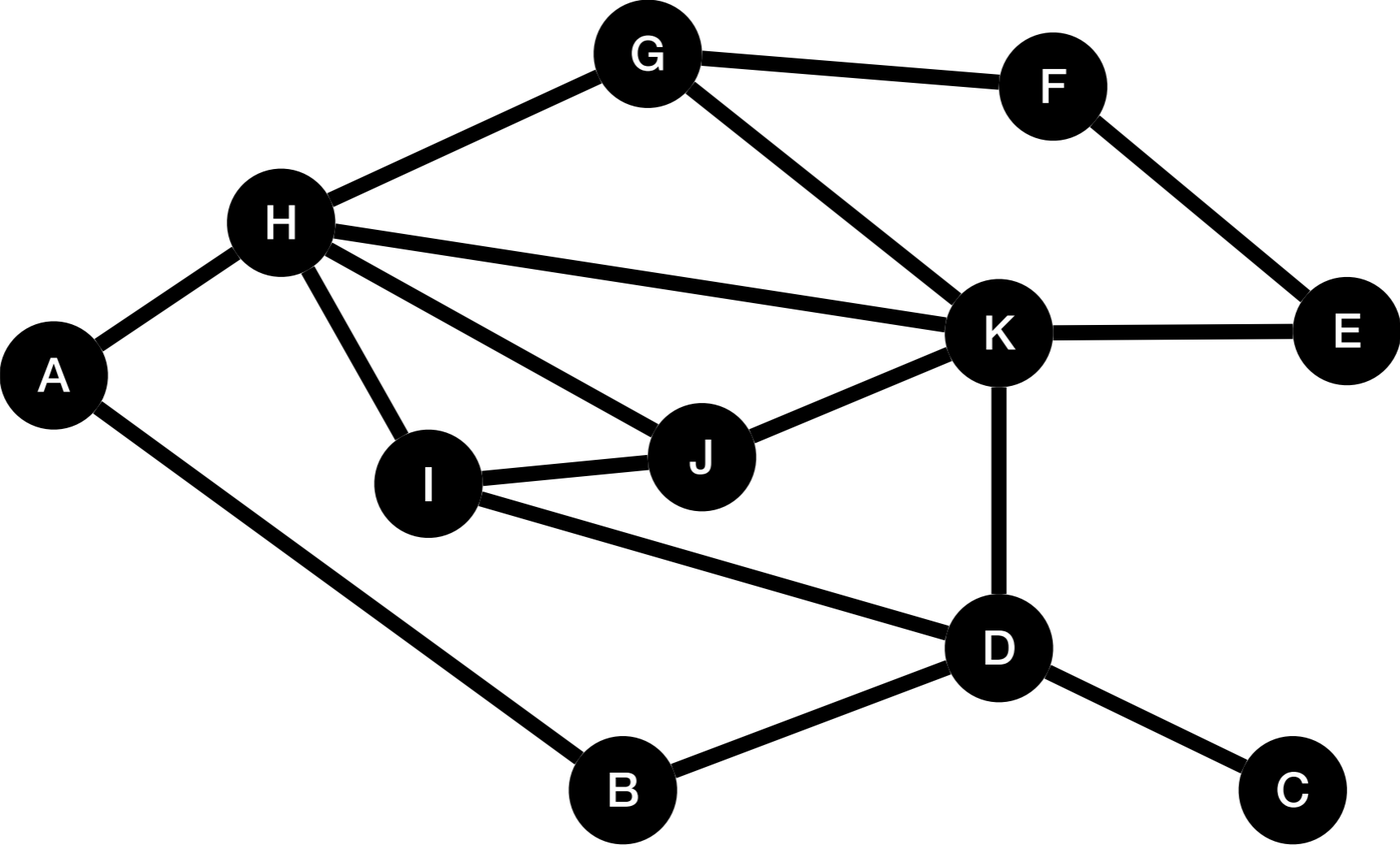


- A 2
- B 2
- C 1
- D 4
- E 2

Algorithme de Welsh-Powell

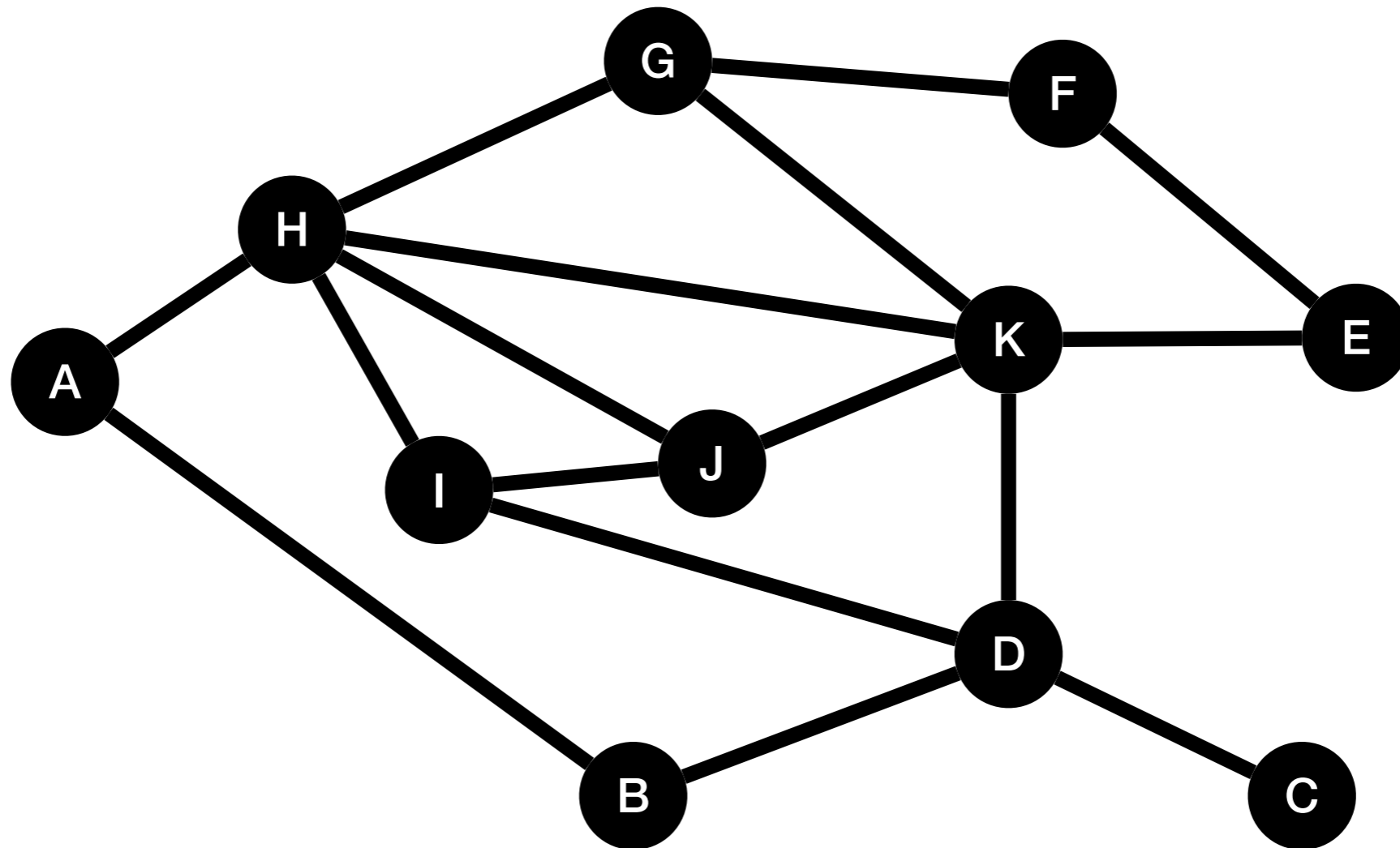


Algorithme de Welsh-Powell



- A 2
- B 2
- C 1
- D 4
- E 2
- F 2
- G 3

Algorithme de Welsh-Powell



A

2

B

2

C

1

D

4

E

2

F

2

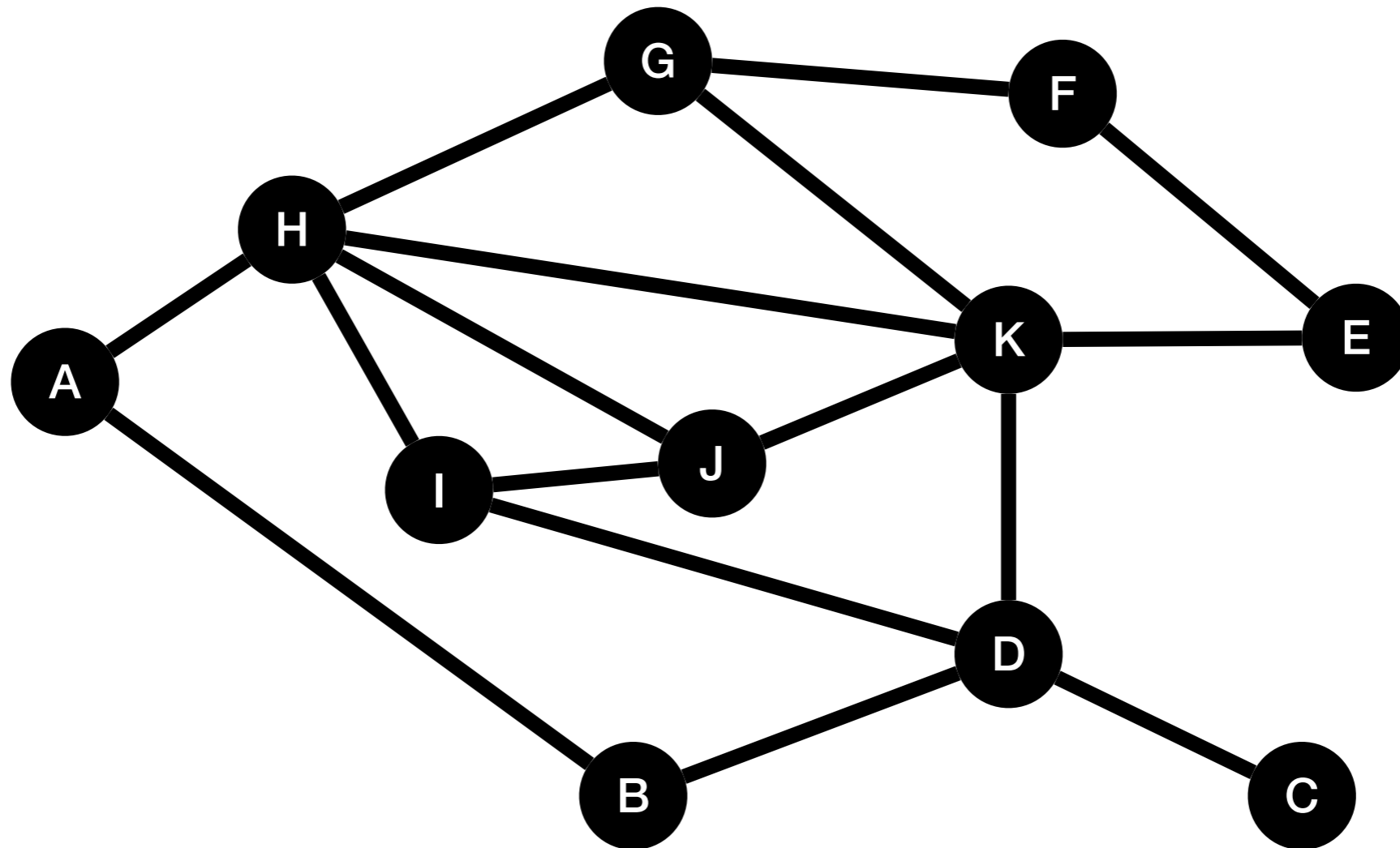
G

3

H

5

Algorithme de Welsh-Powell



A

2

B

2

C

1

D

4

E

2

F

2

G

3

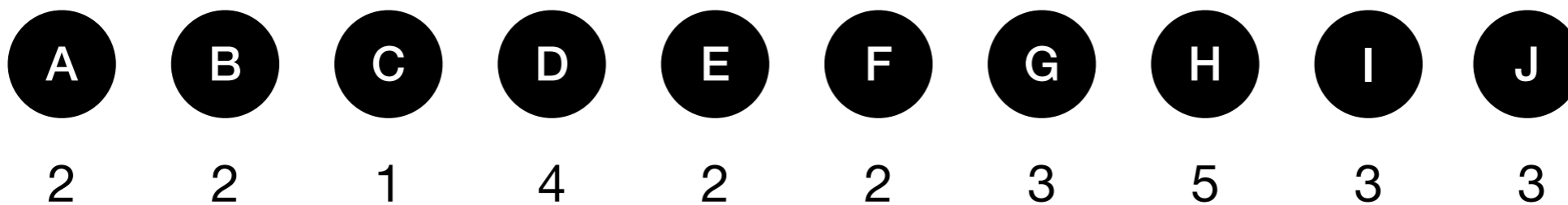
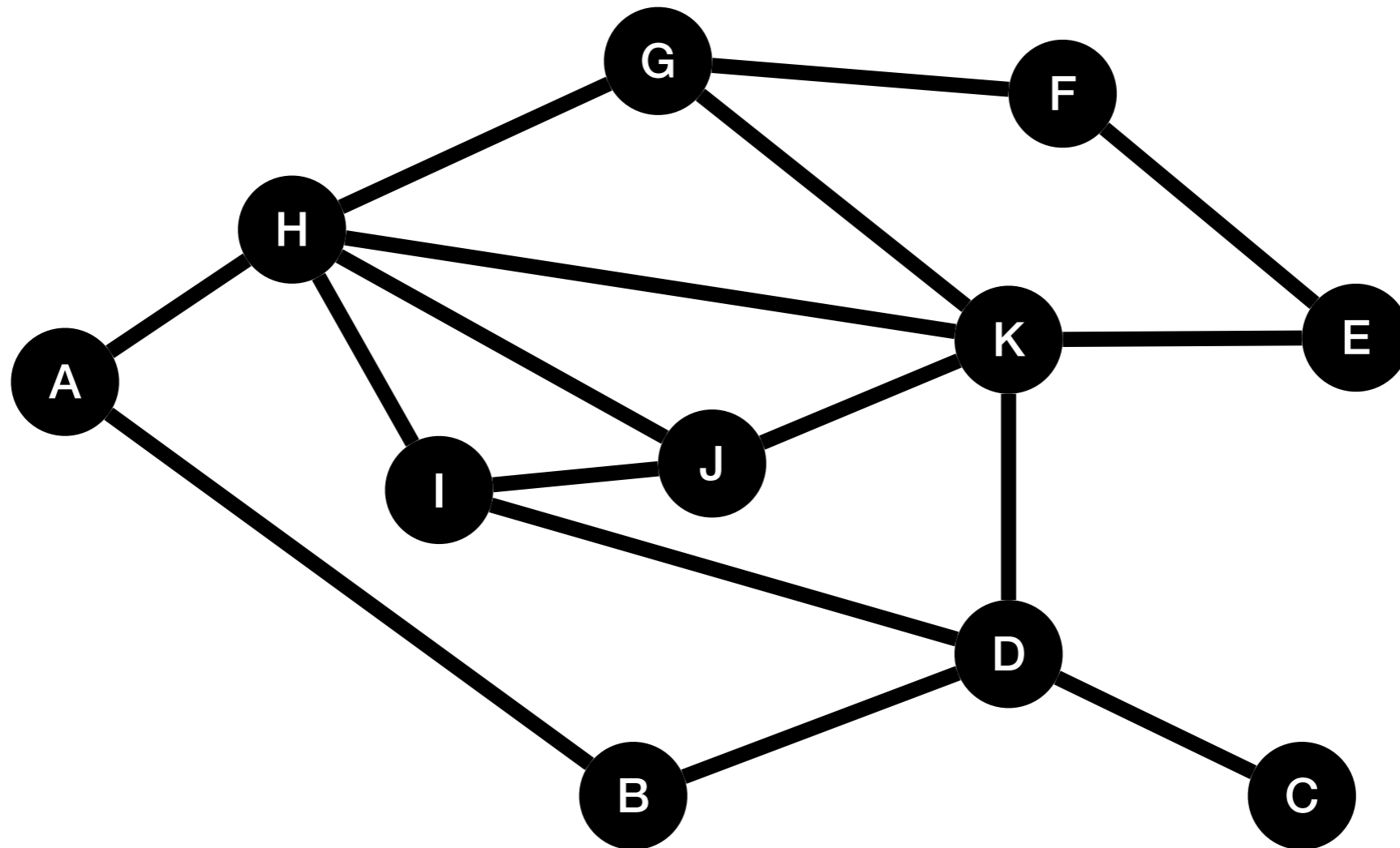
H

5

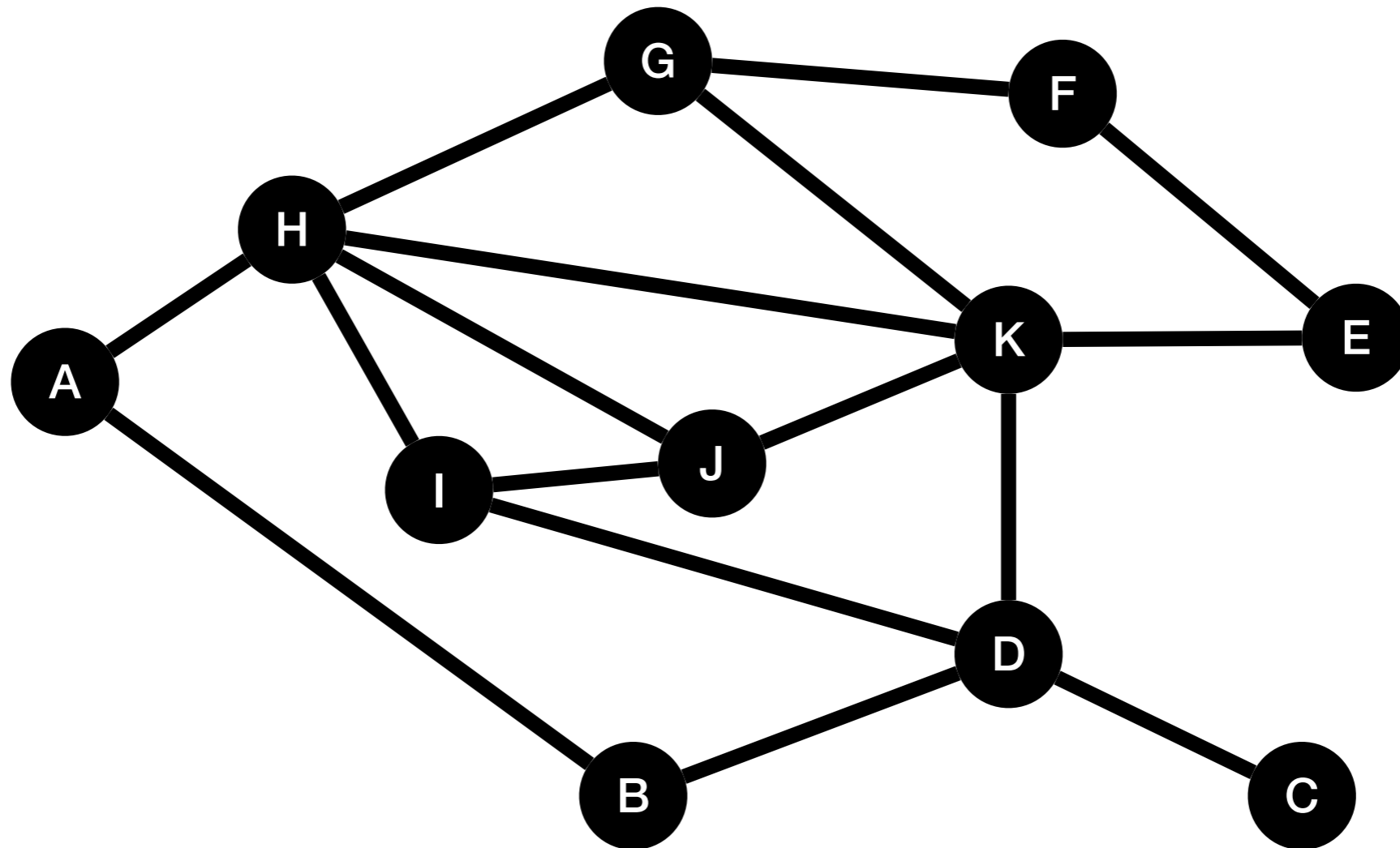
I

3

Algorithme de Welsh-Powell



Algorithme de Welsh-Powell



A

B

C

D

E

F

G

H

I

J

K

2

2

1

4

2

2

3

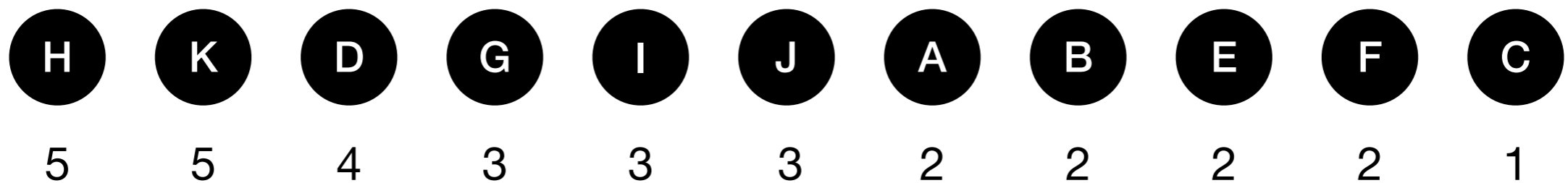
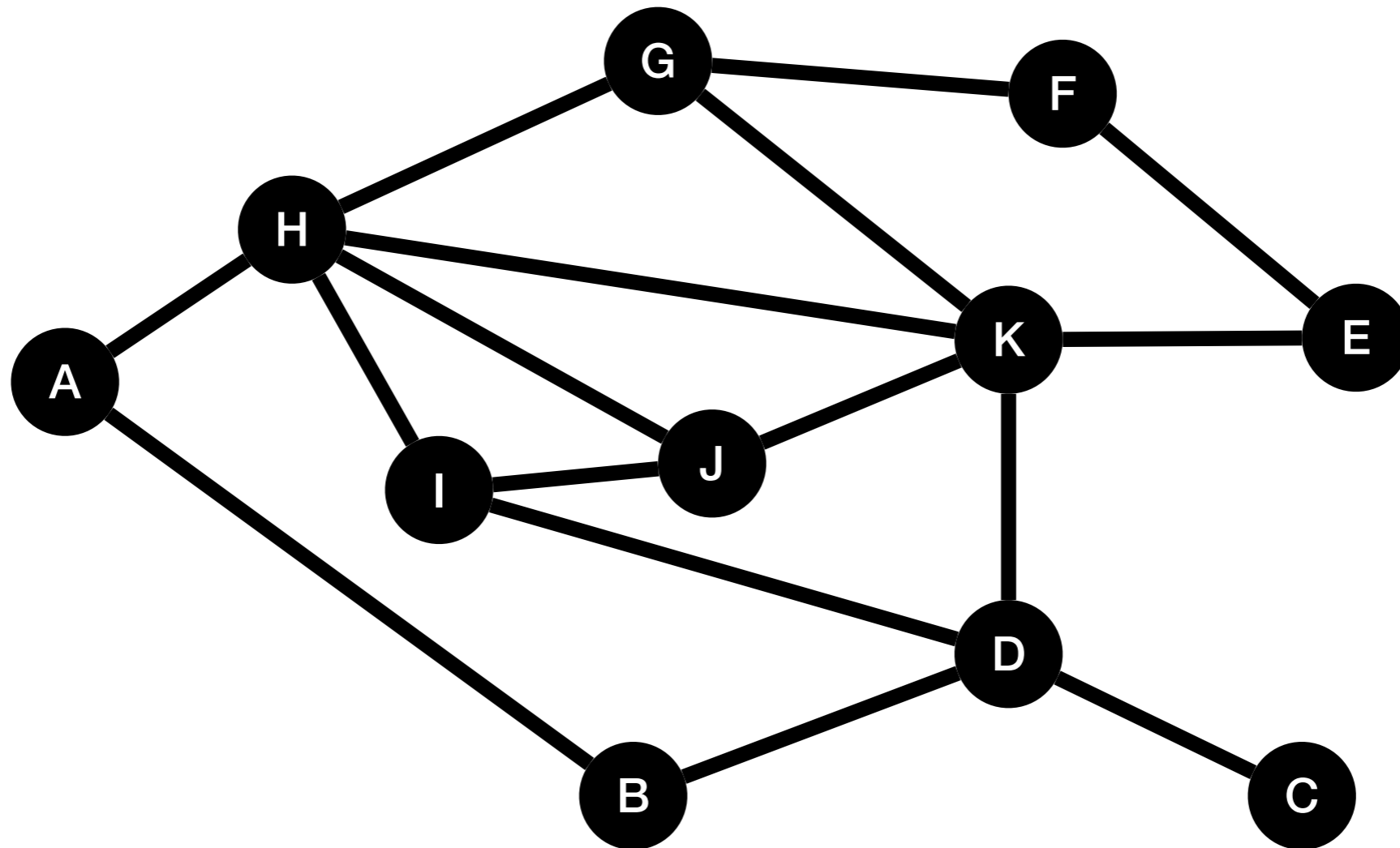
5

3

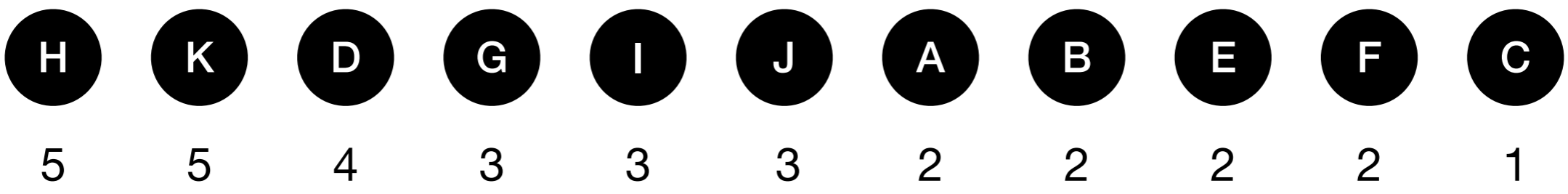
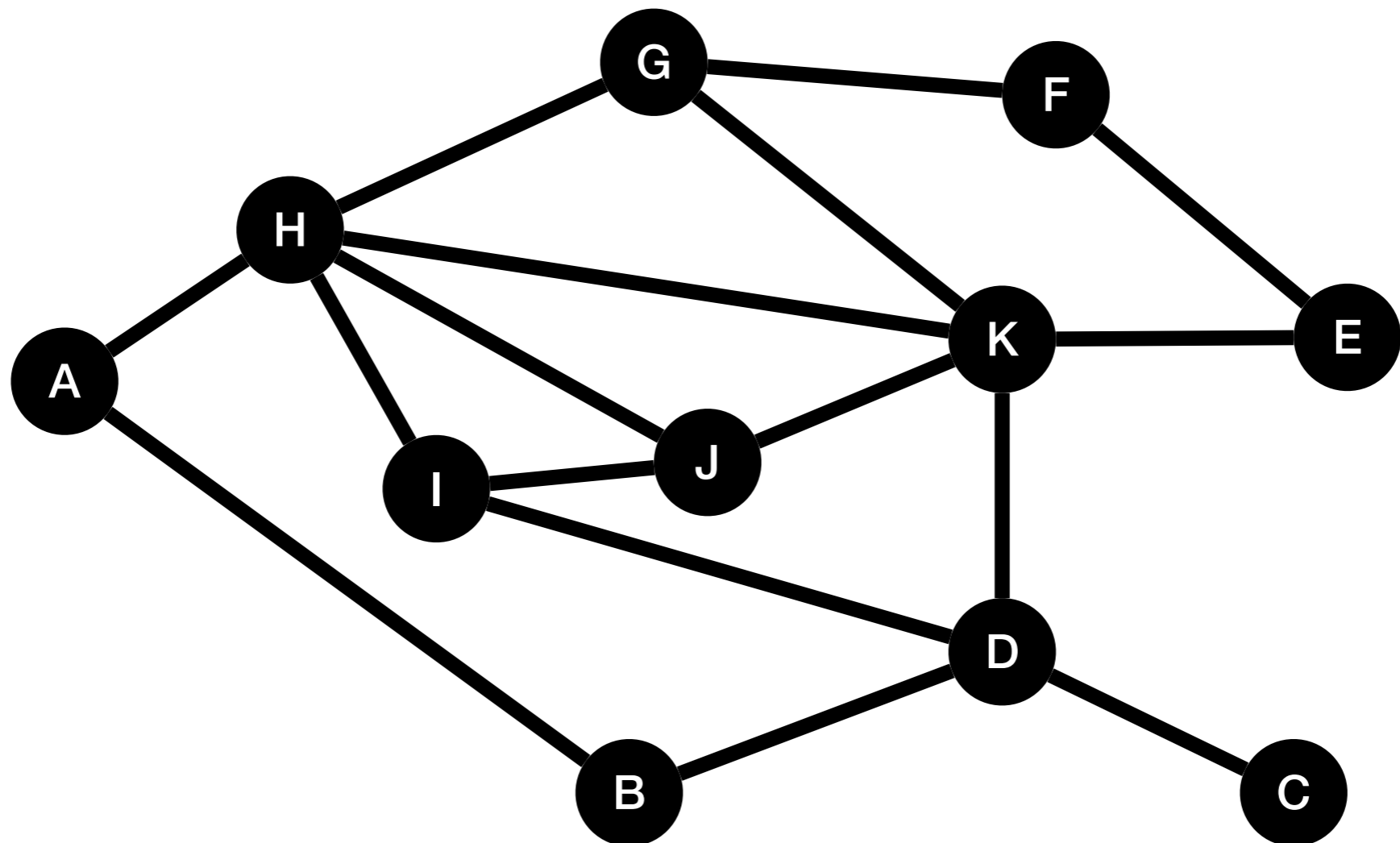
3

5

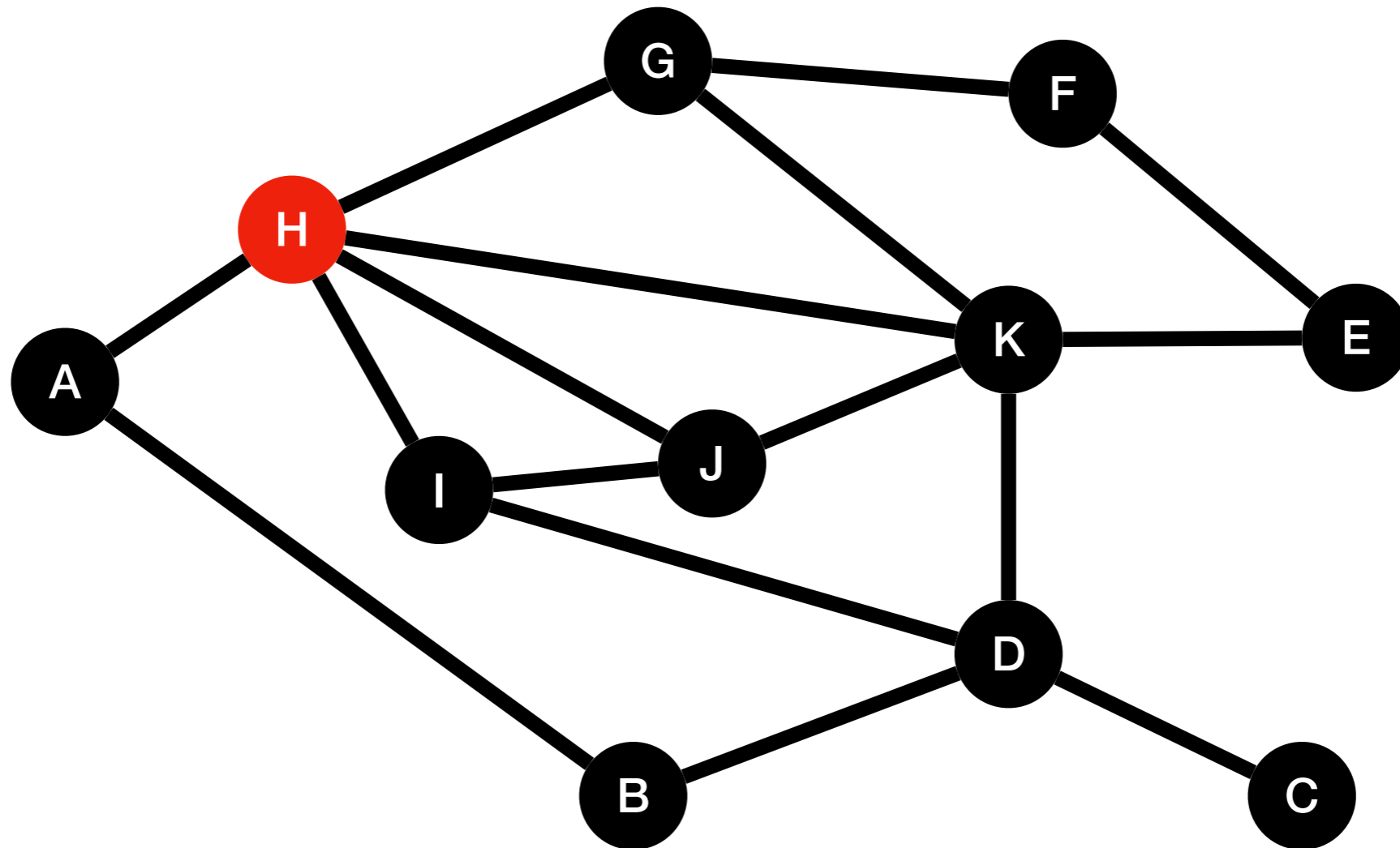
Algorithme de Welsh-Powell



Algorithme de Welsh-Powell



Algorithme de Welsh-Powell



5



5



4



3



3



3



2



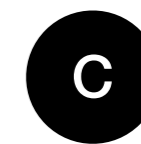
2



2

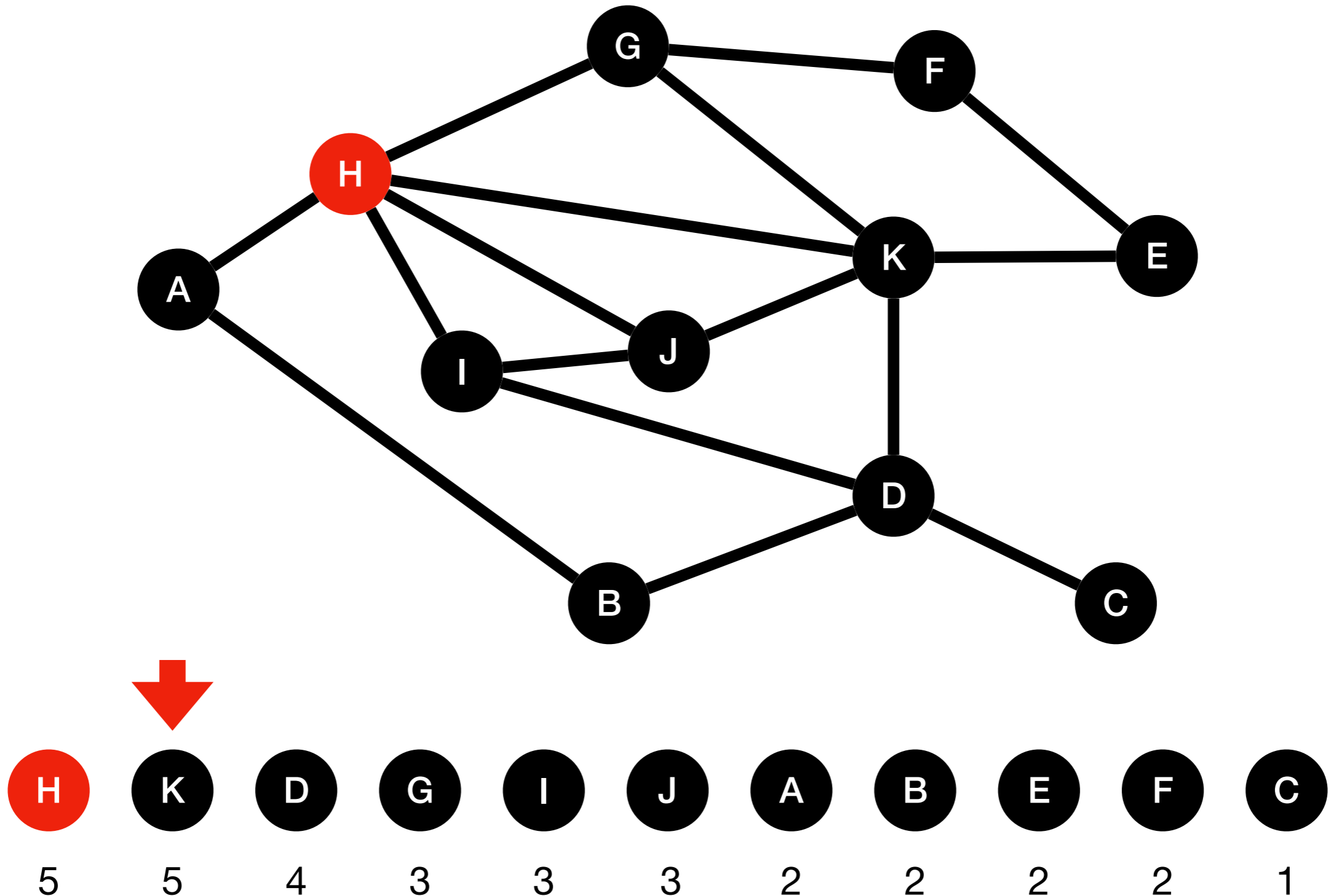


2

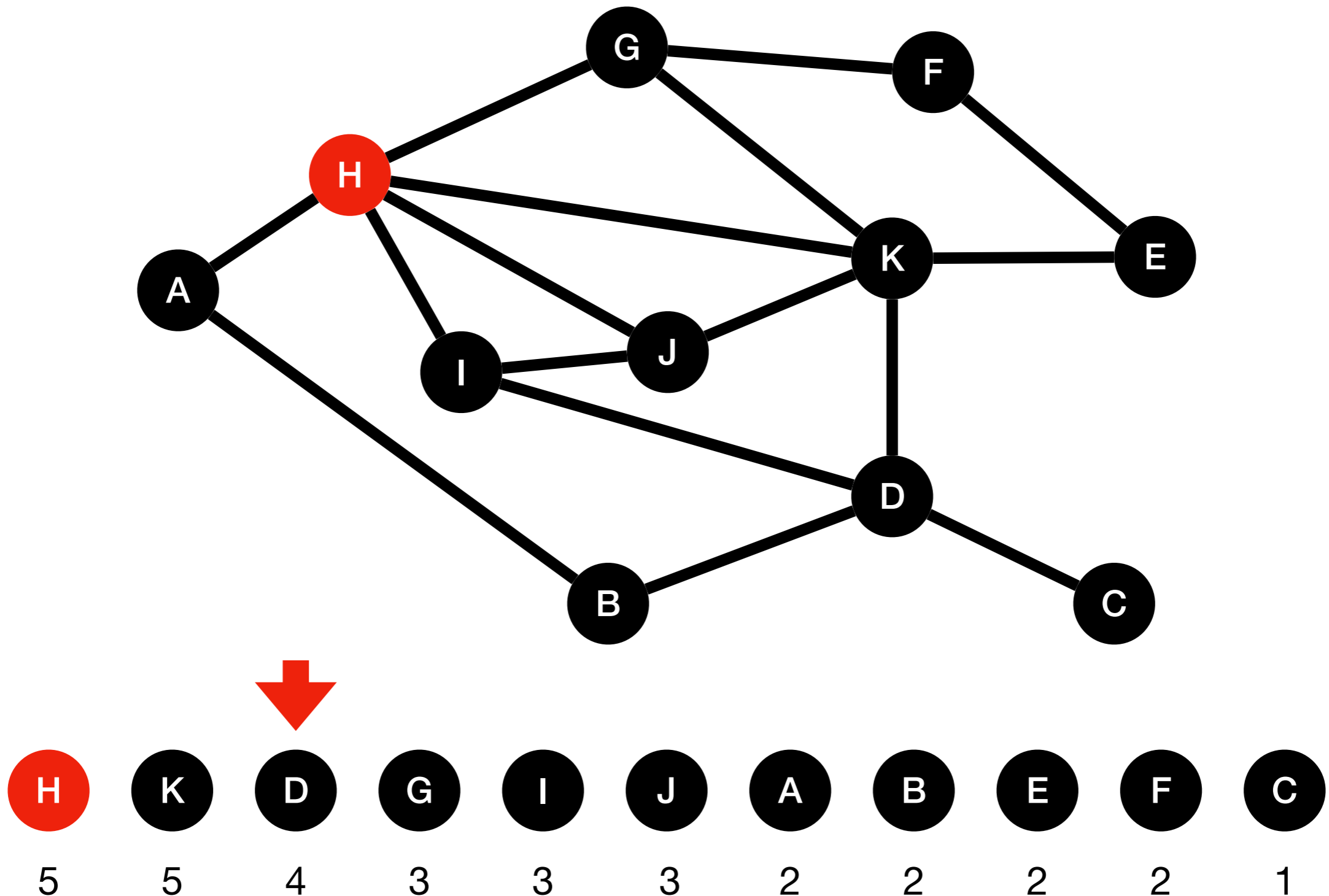


1

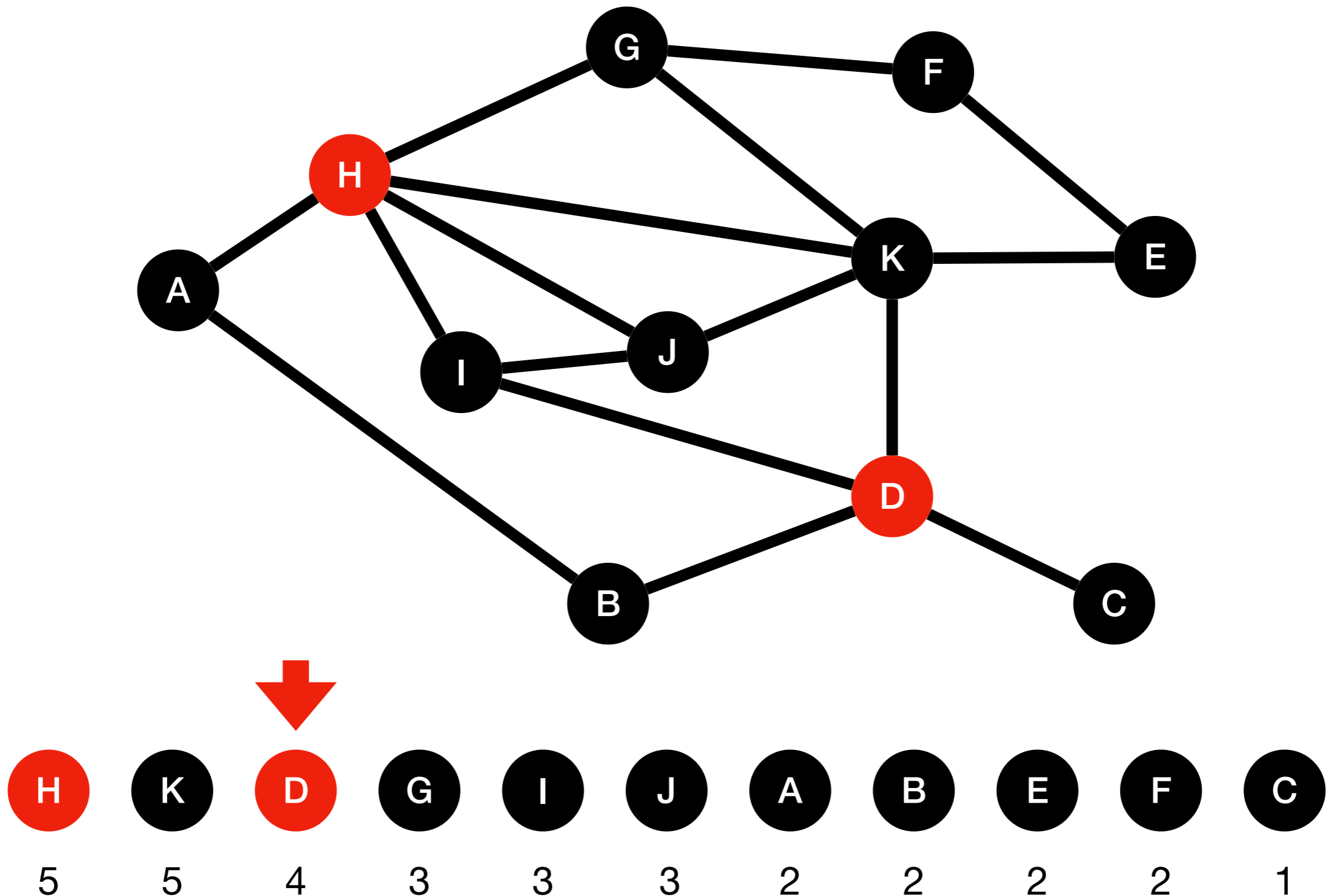
Algorithme de Welsh-Powell



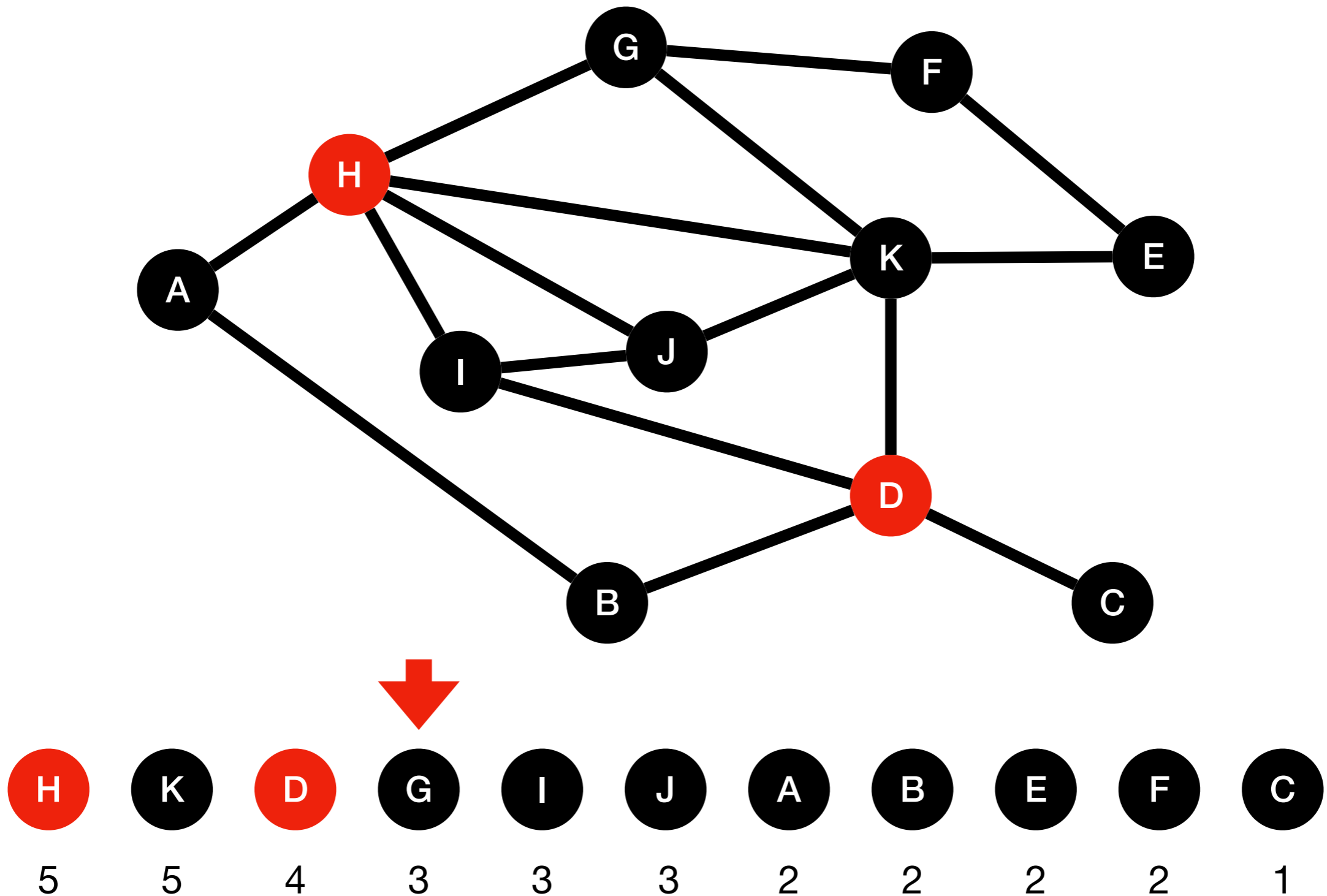
Algorithme de Welsh-Powell



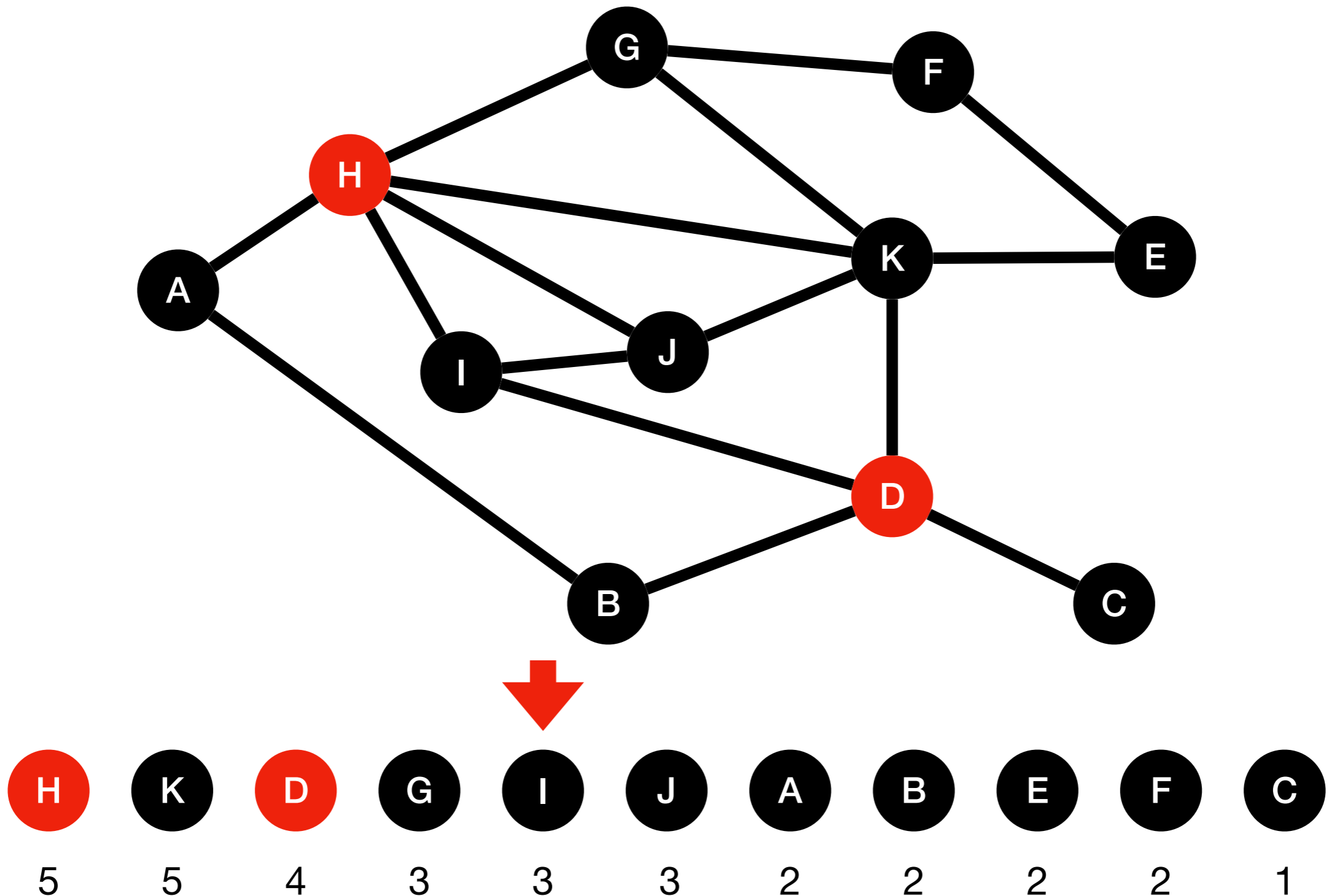
Algorithme de Welsh-Powell



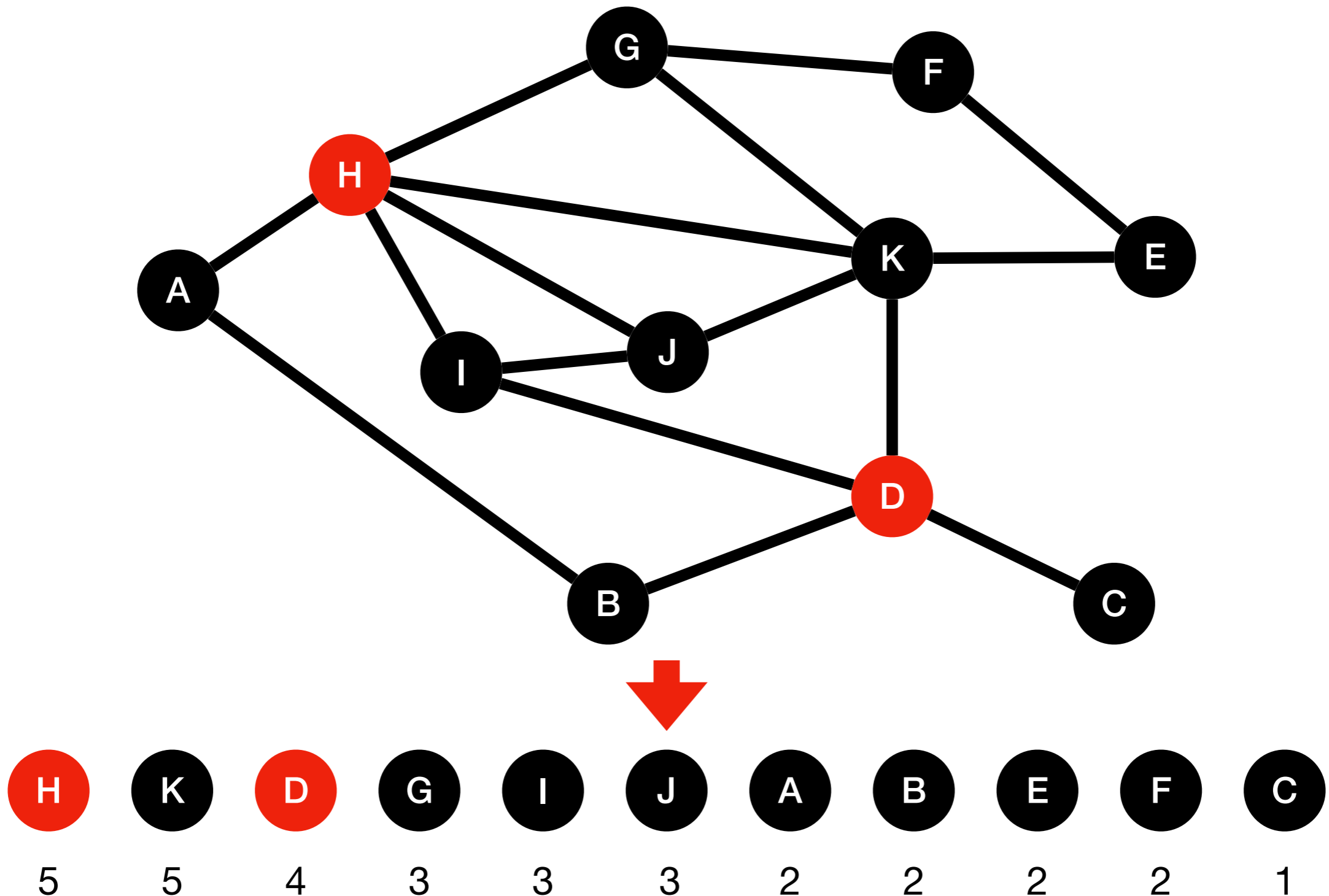
Algorithme de Welsh-Powell



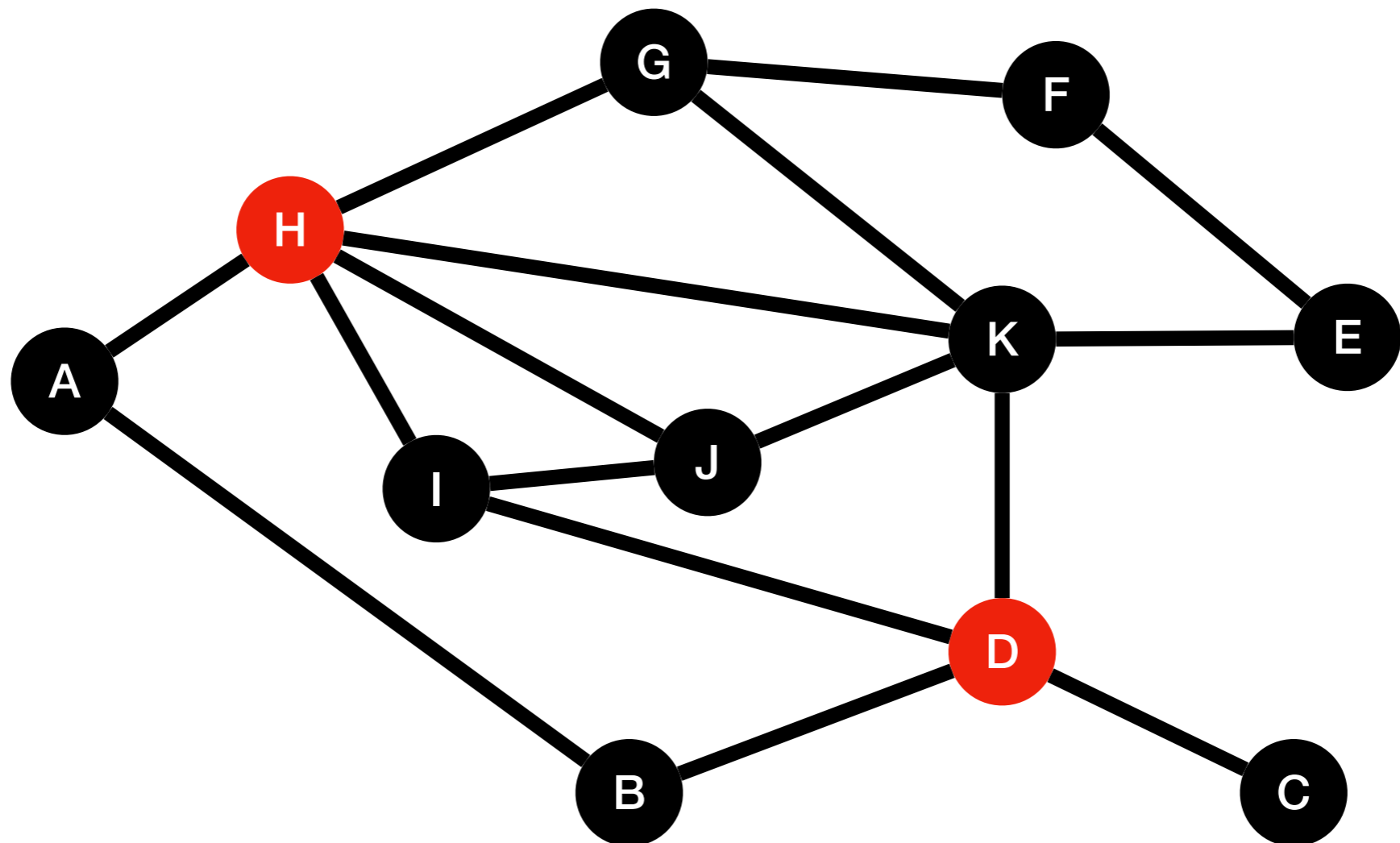
Algorithme de Welsh-Powell



Algorithme de Welsh-Powell

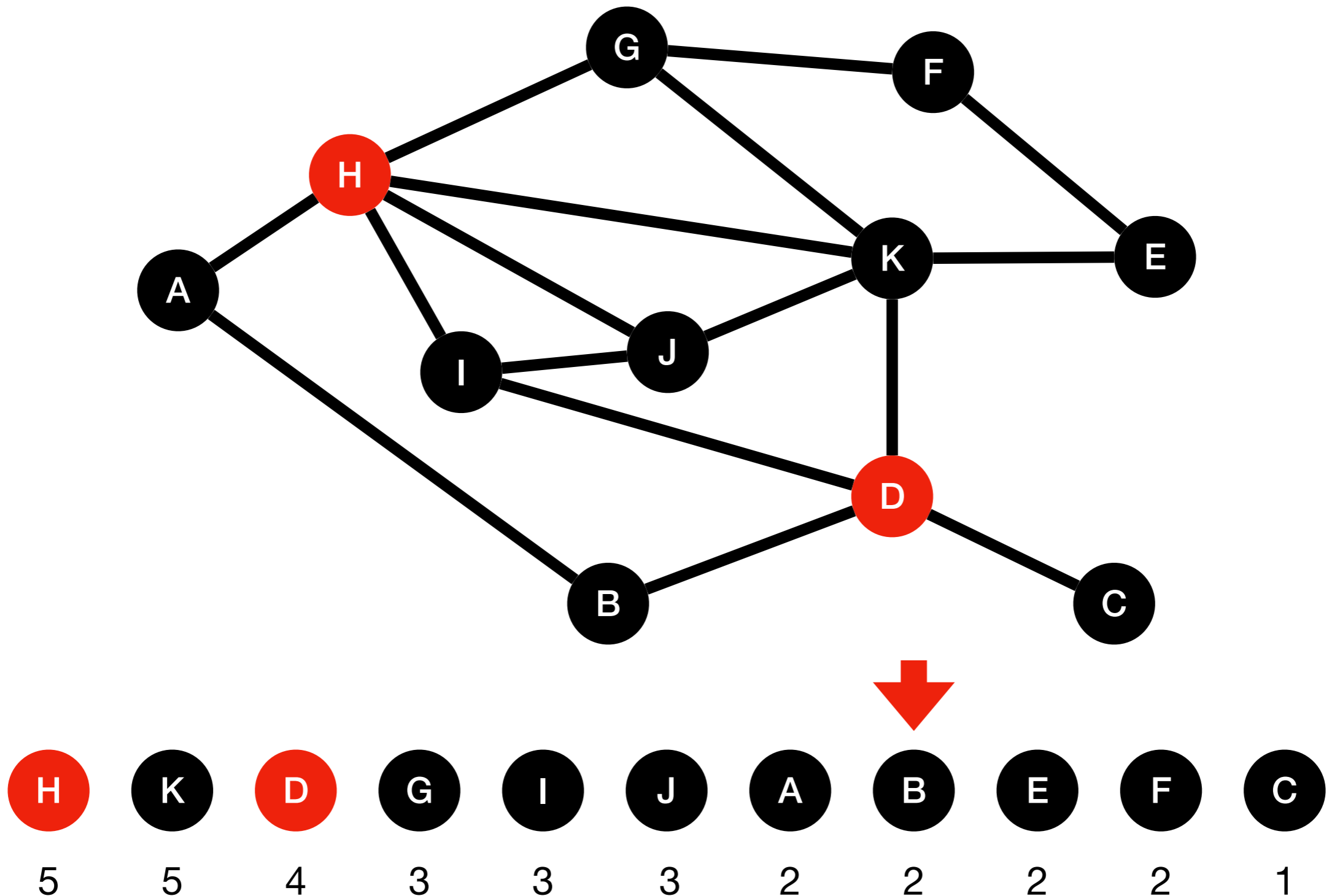


Algorithme de Welsh-Powell

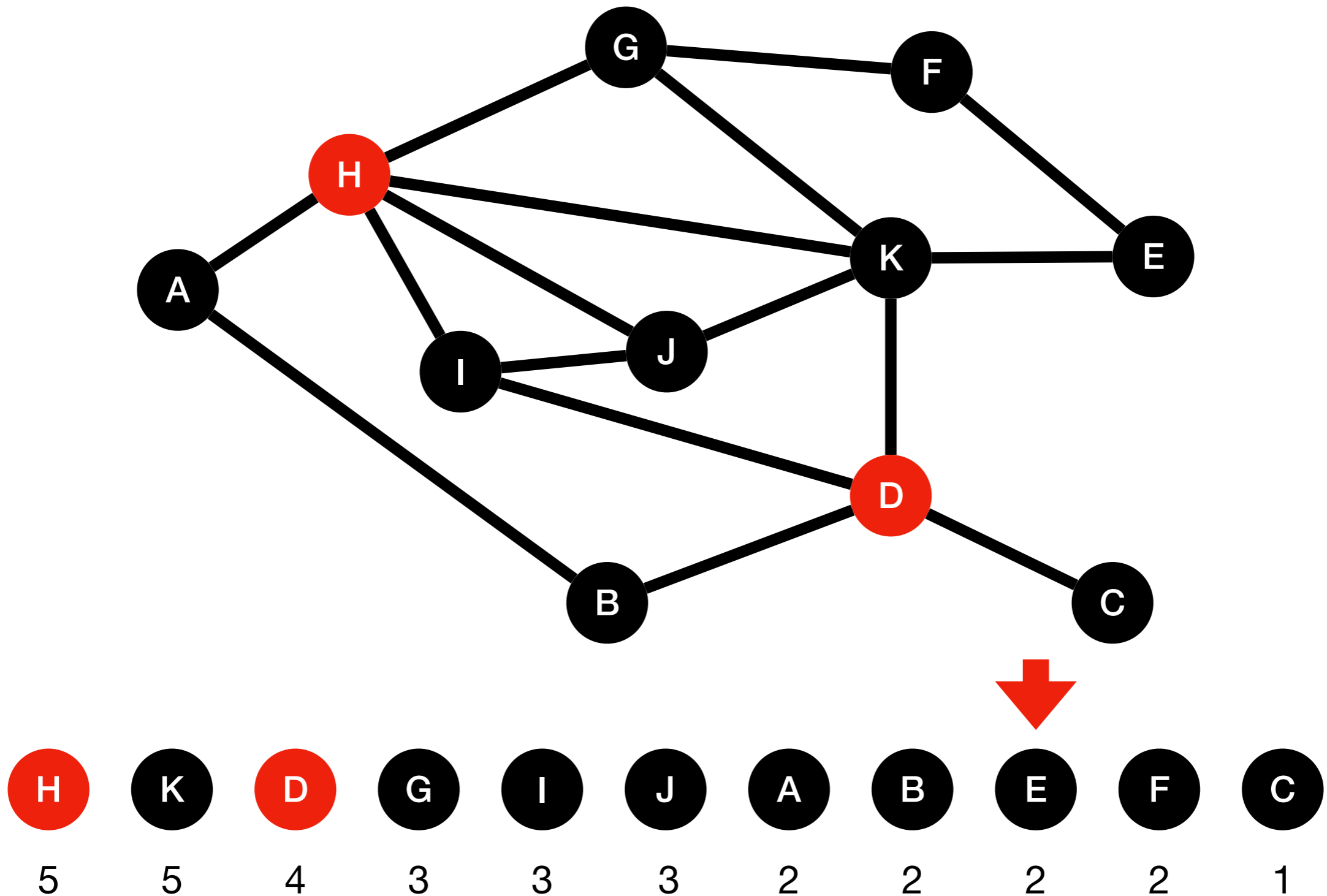


- H (5)
- K (5)
- D (4)
- G (3)
- I (3)
- J (3)
- A (2)
- B (2)
- E (2)
- F (2)
- C (1)

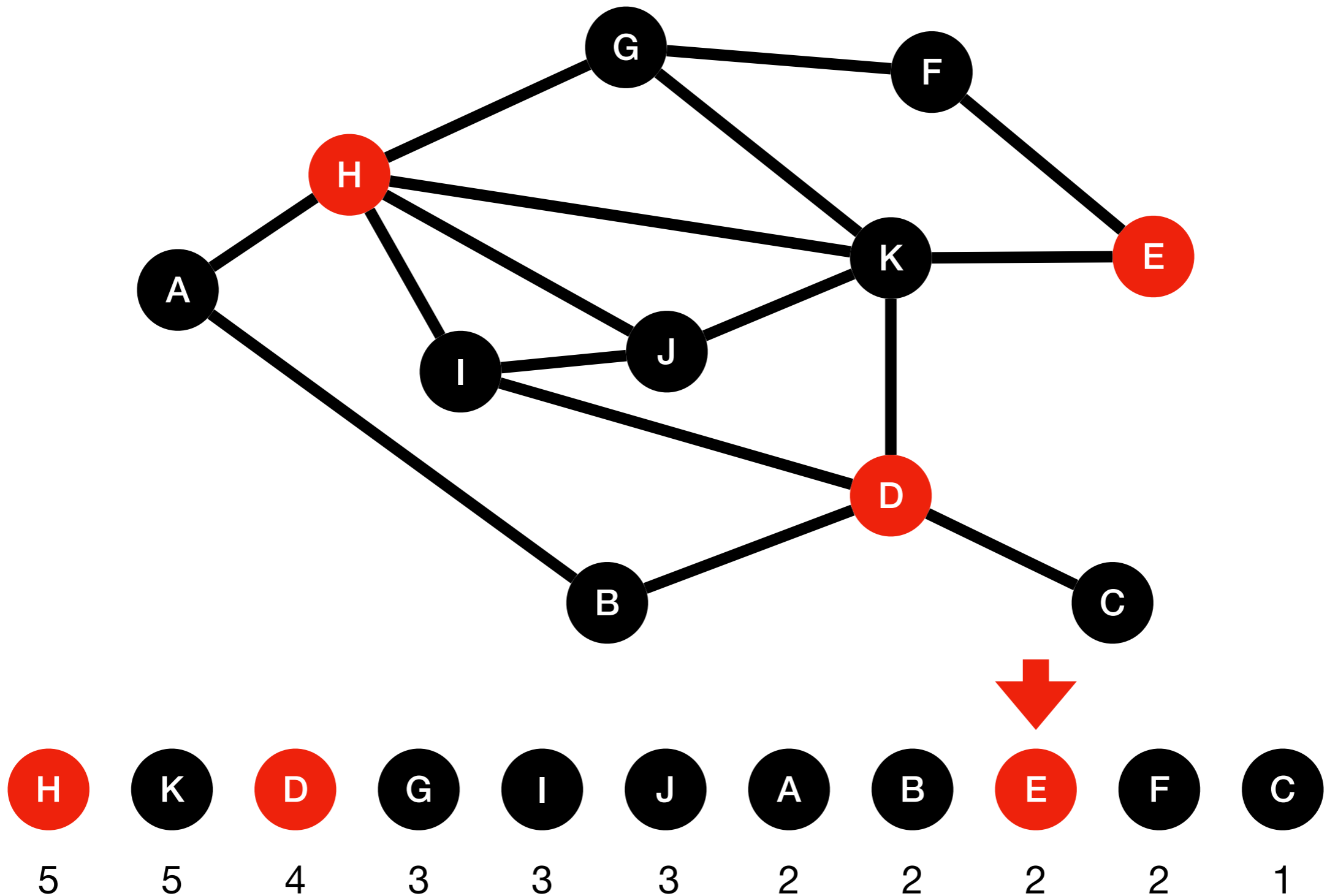
Algorithme de Welsh-Powell



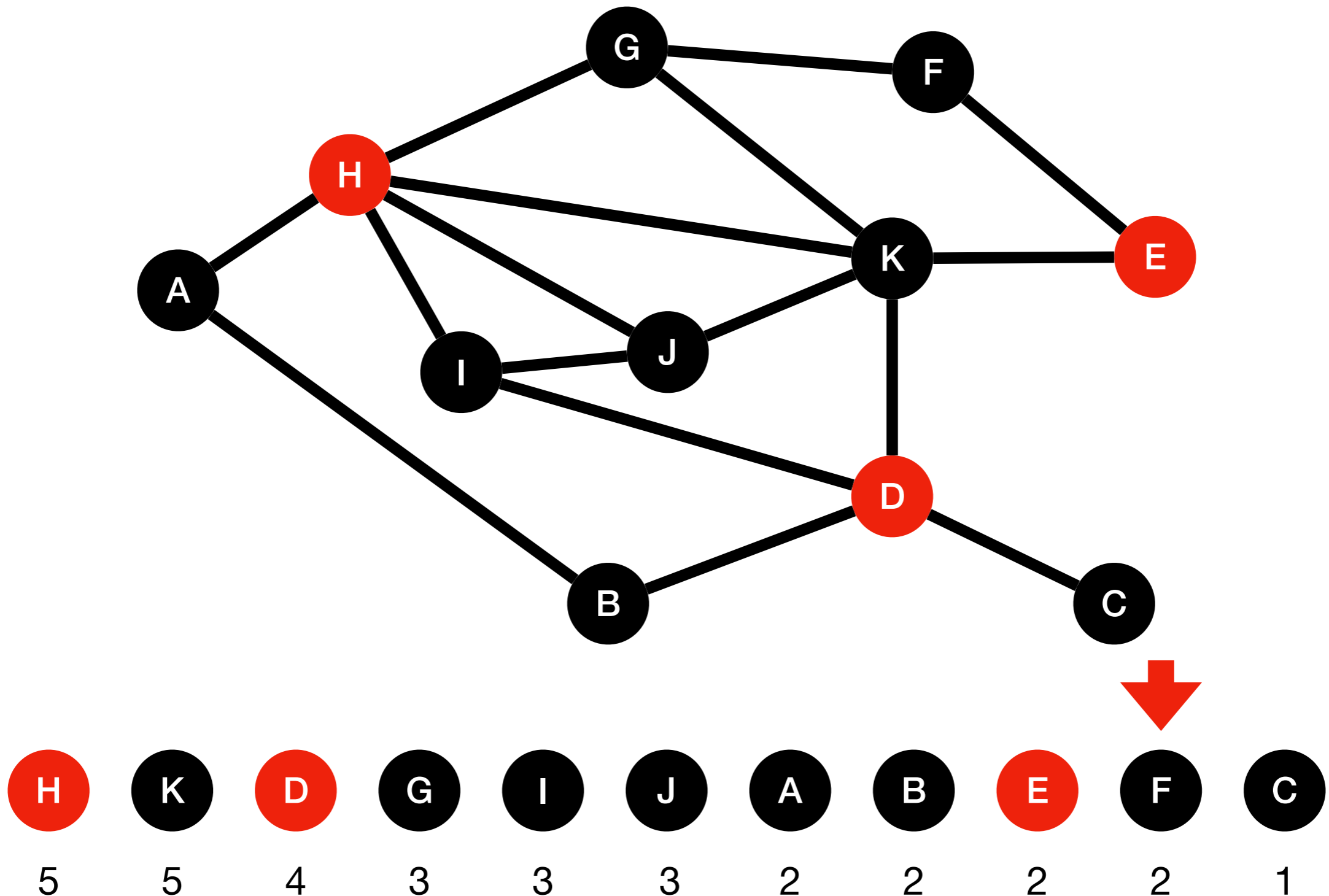
Algorithme de Welsh-Powell



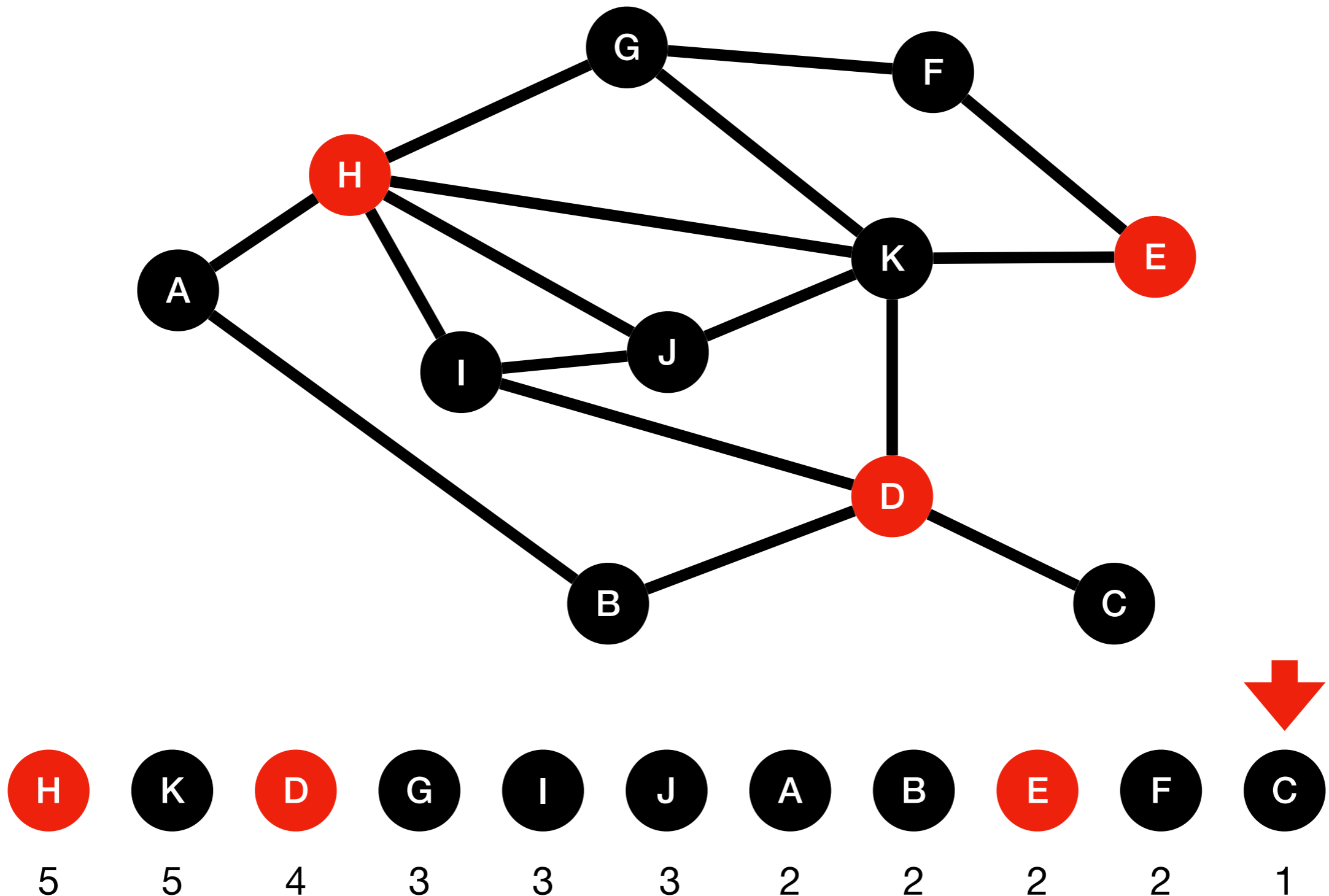
Algorithme de Welsh-Powell



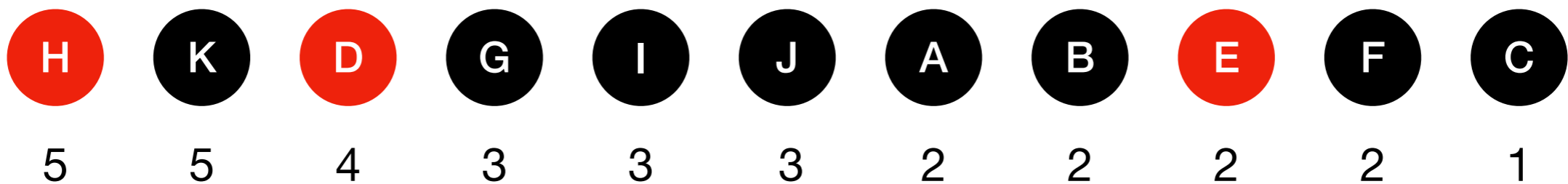
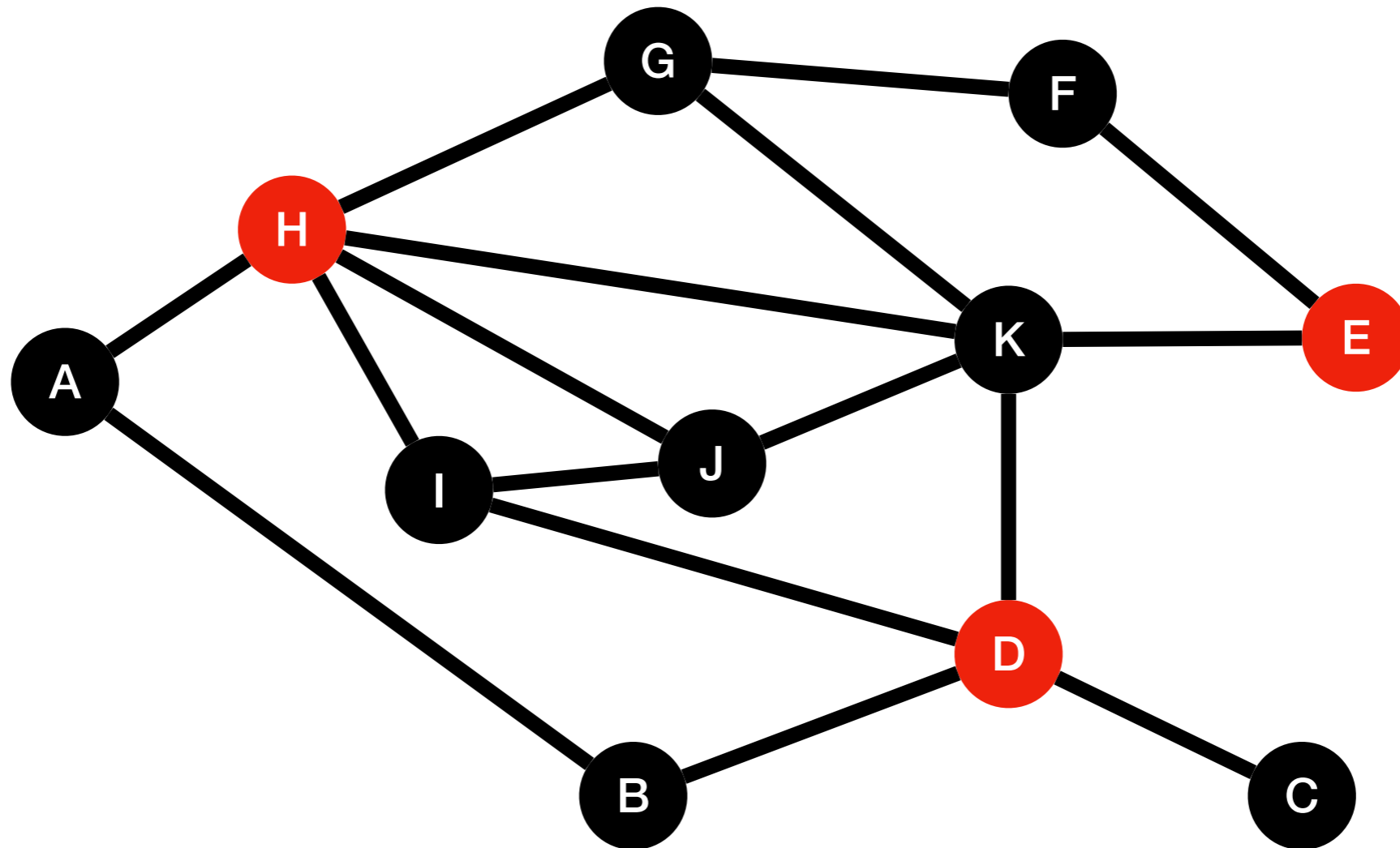
Algorithme de Welsh-Powell



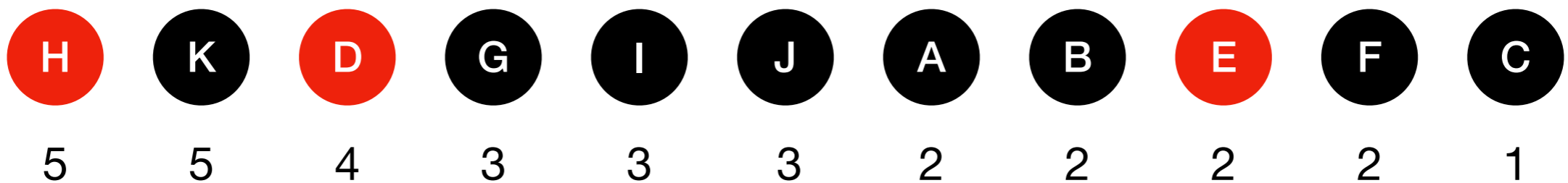
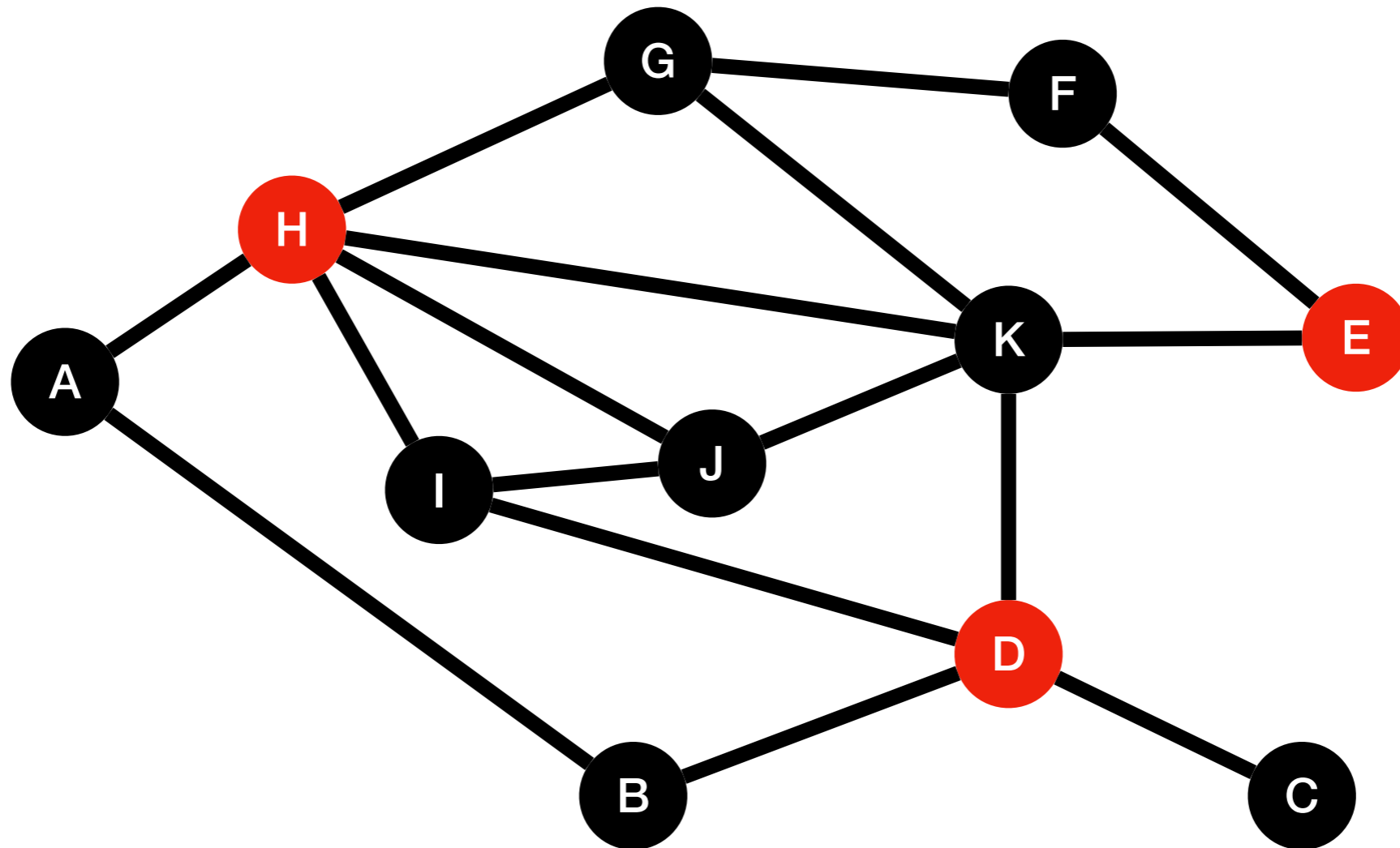
Algorithme de Welsh-Powell



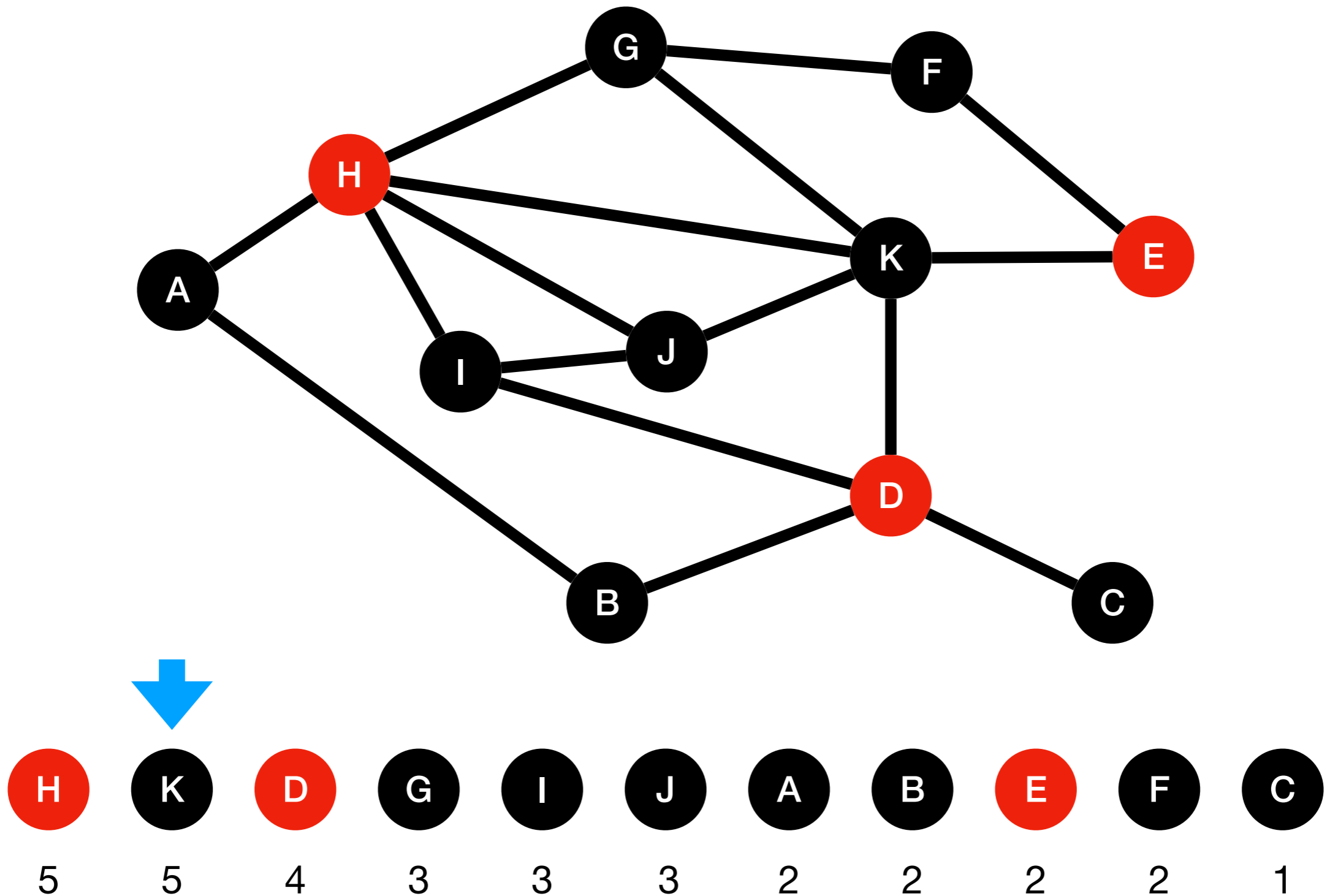
Algorithme de Welsh-Powell



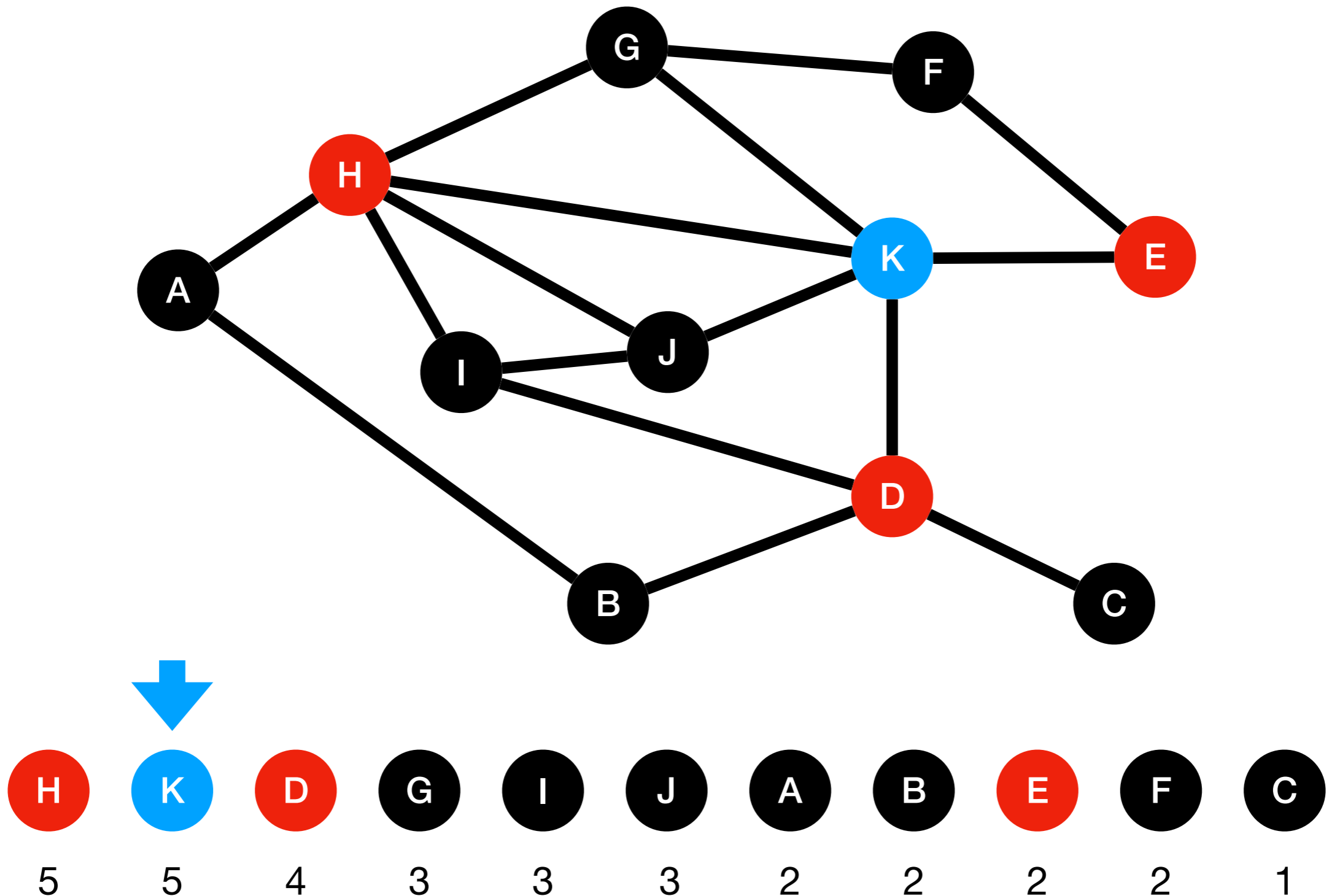
Algorithme de Welsh-Powell



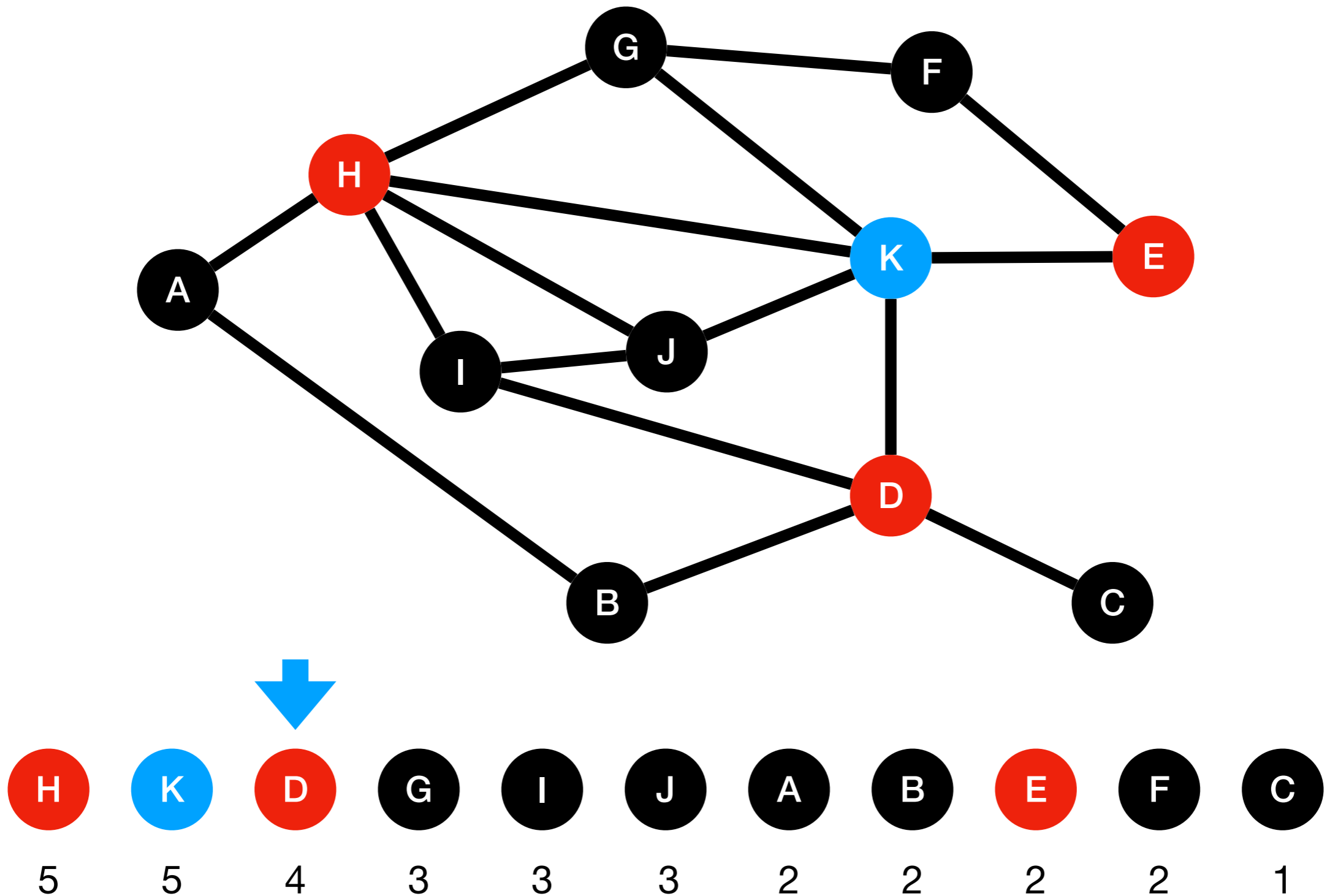
Algorithme de Welsh-Powell



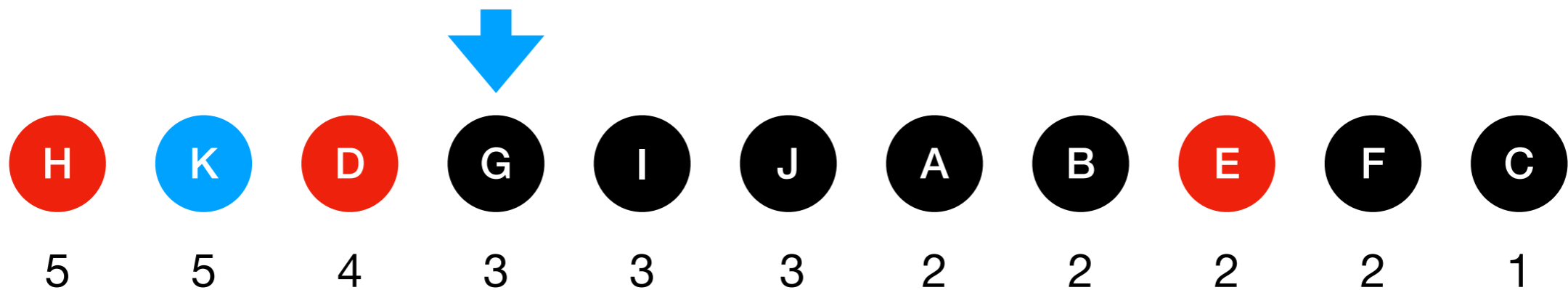
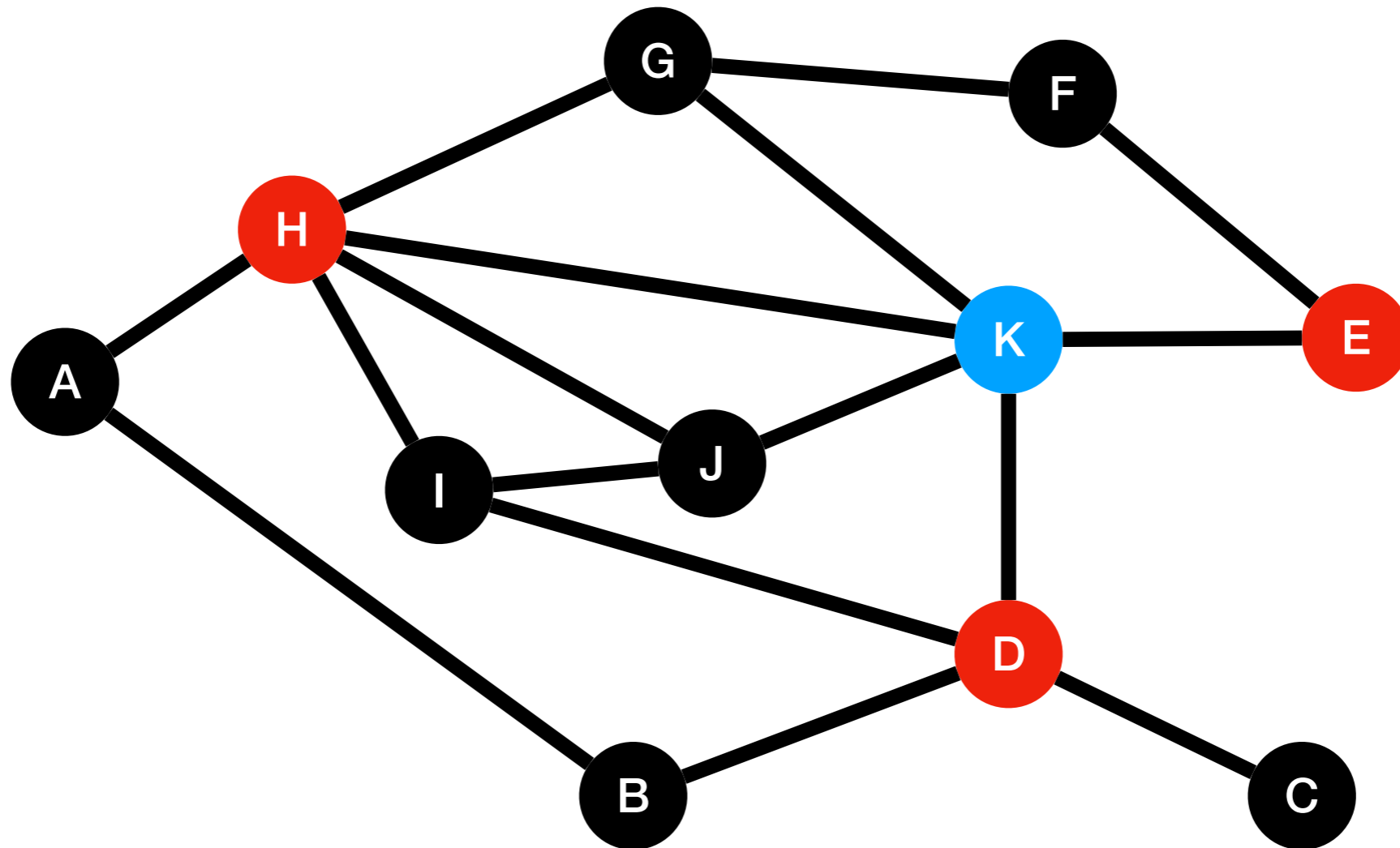
Algorithme de Welsh-Powell



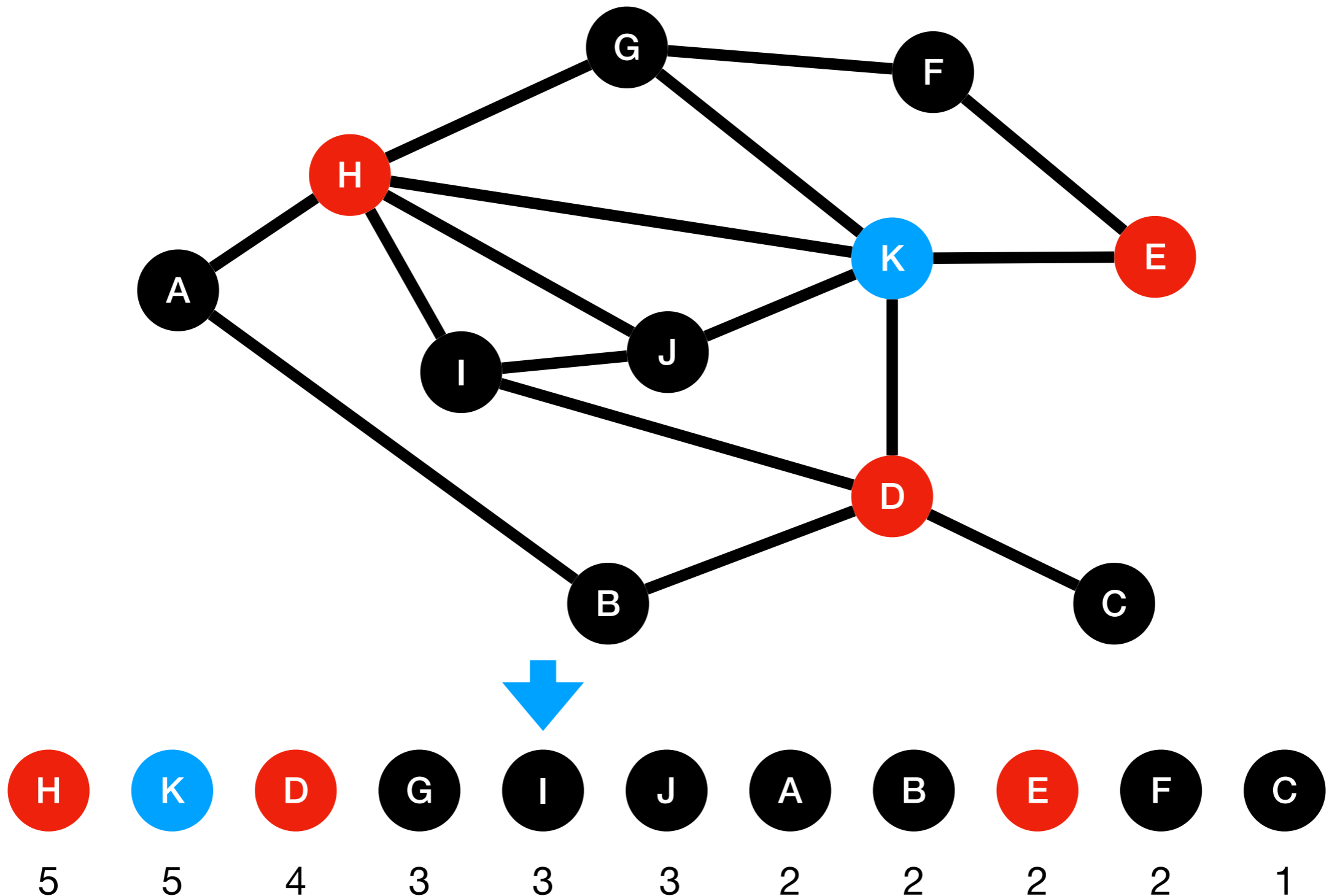
Algorithme de Welsh-Powell



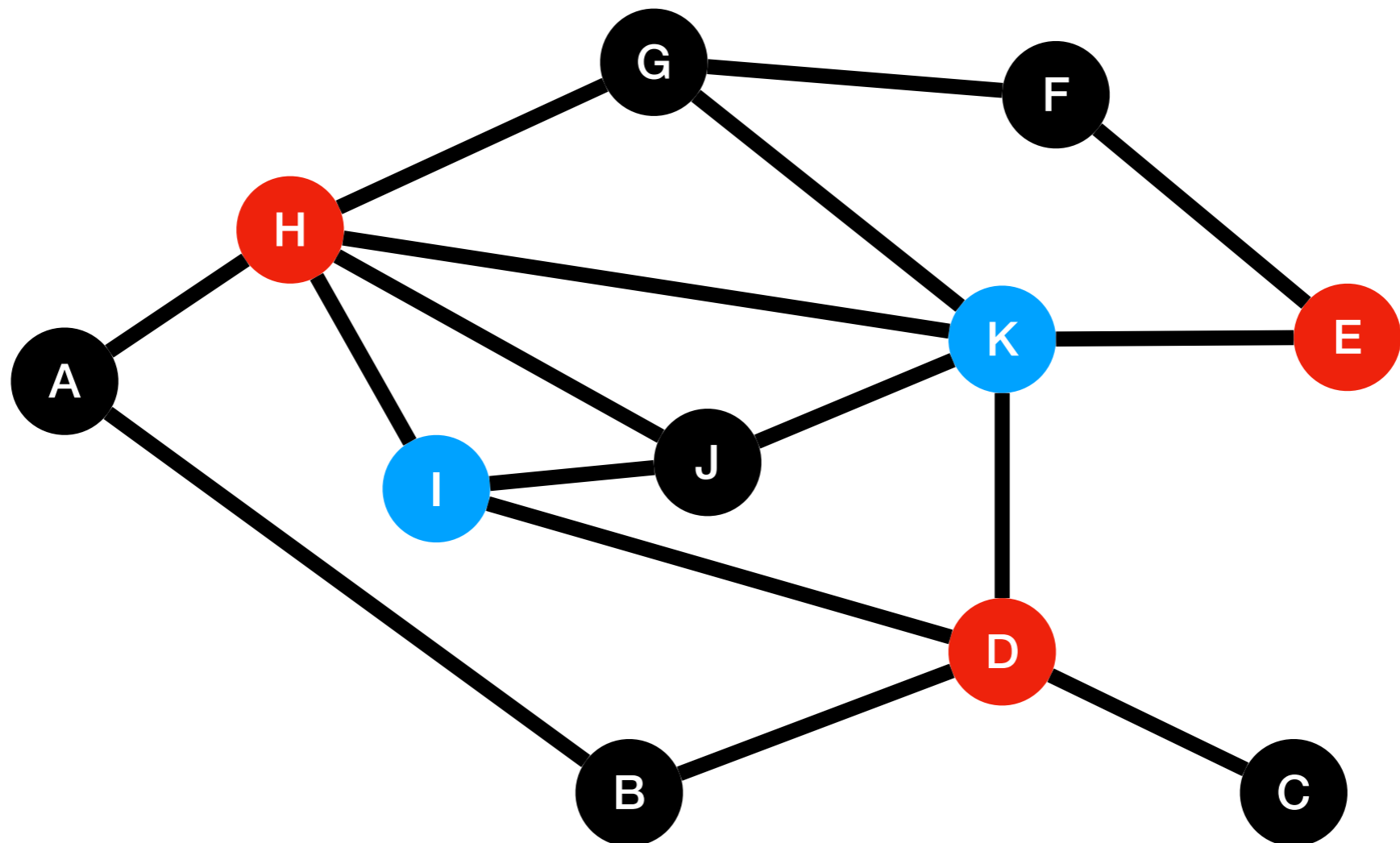
Algorithme de Welsh-Powell



Algorithme de Welsh-Powell

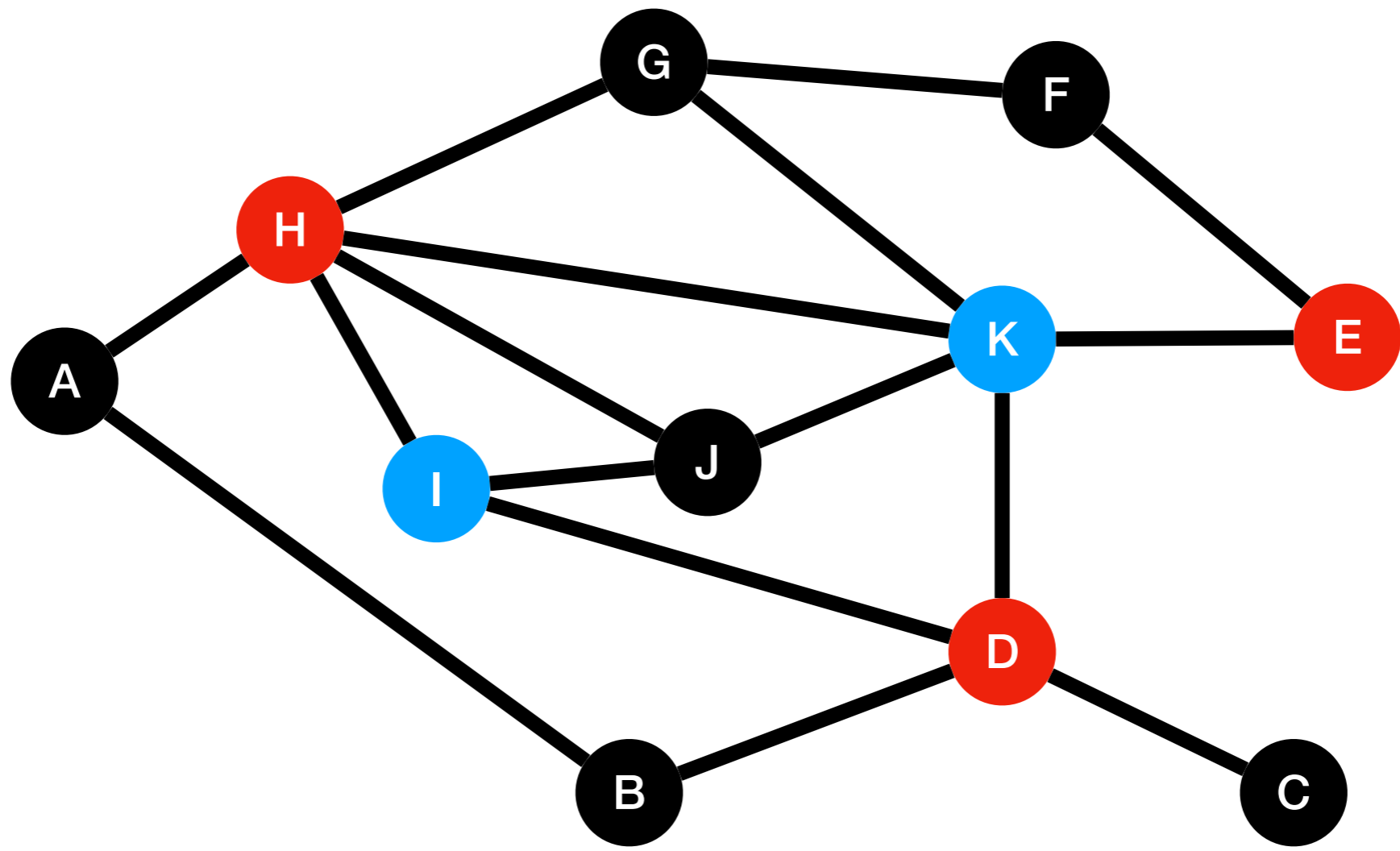


Algorithme de Welsh-Powell



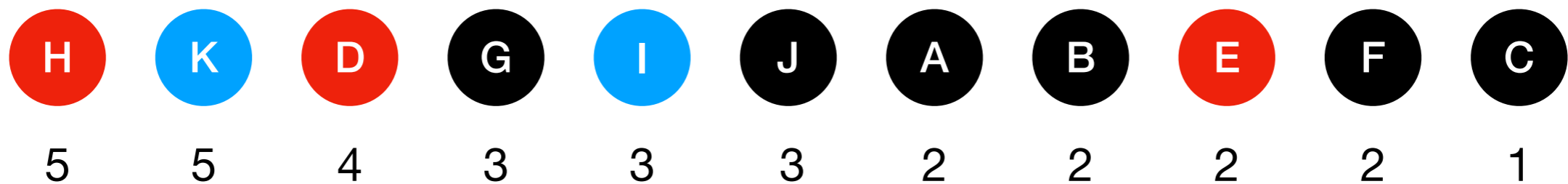
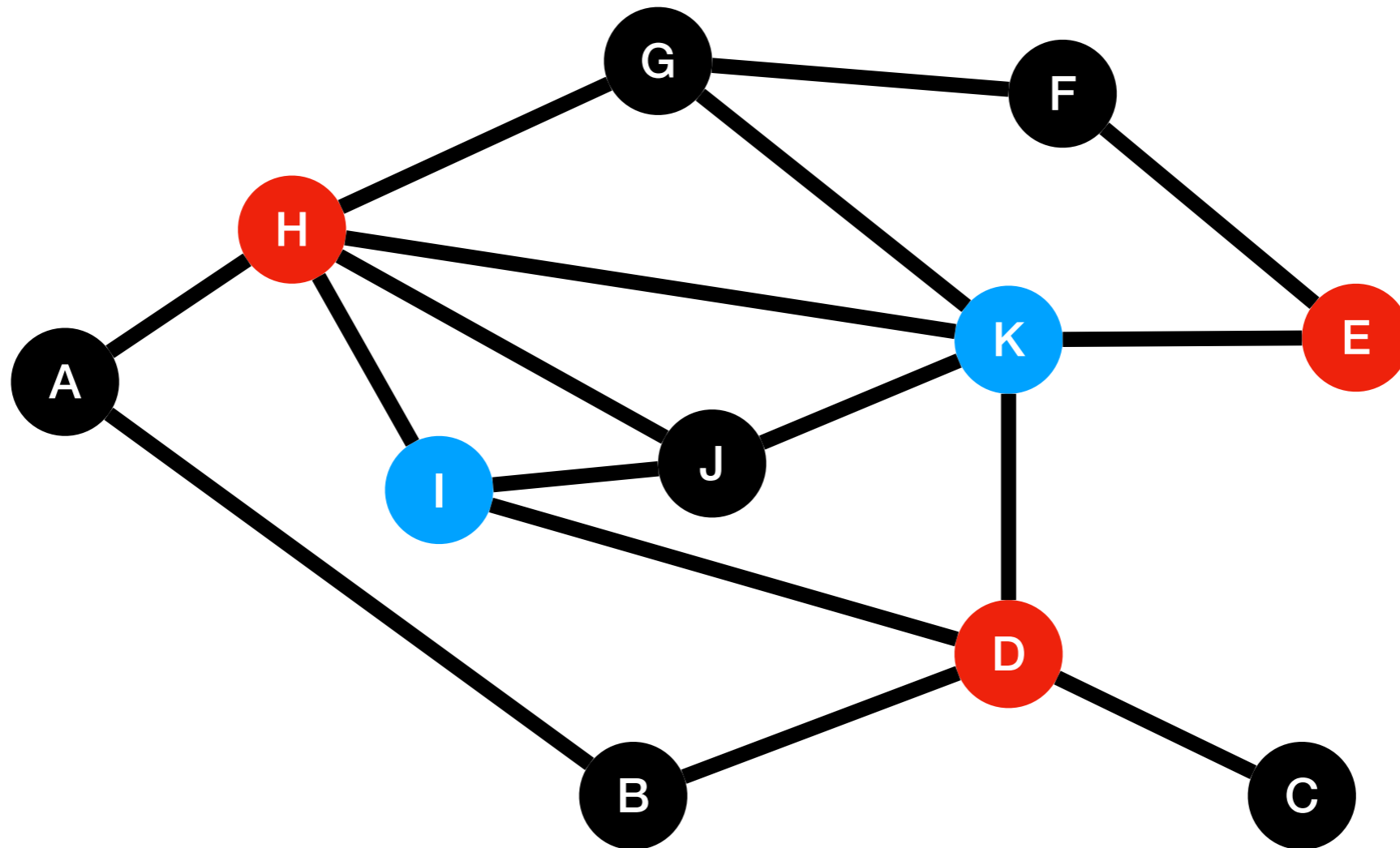
- H (red) 5
- K (blue) 5
- D (red) 4
- G (black) 3
- I (blue) 3
- J (black) 3
- A (black) 2
- B (black) 2
- E (red) 2
- F (black) 2
- C (black) 1

Algorithme de Welsh-Powell

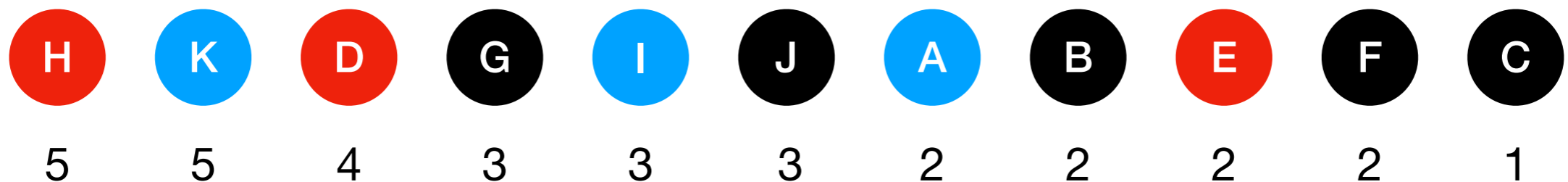
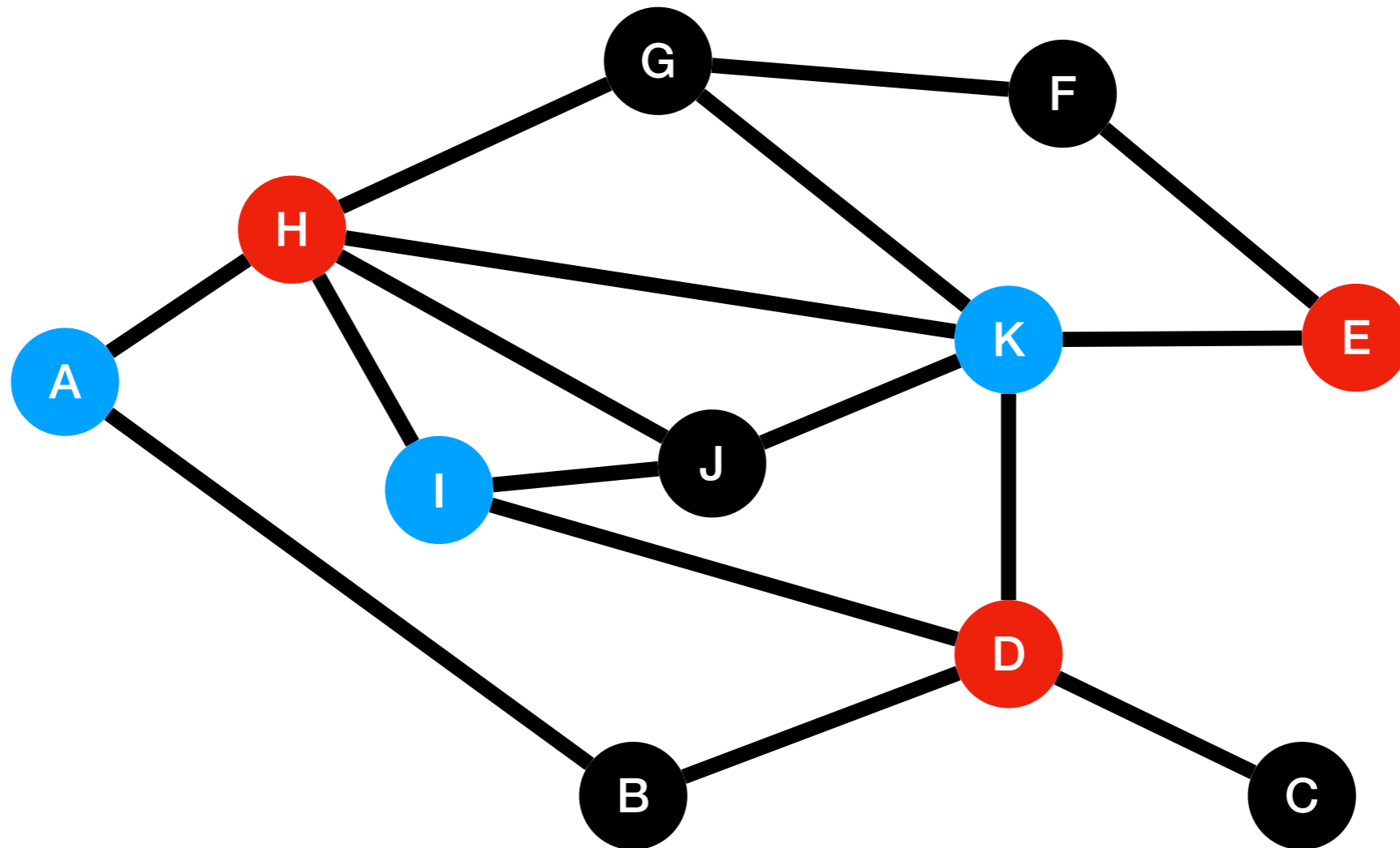


- ↓
- H (5)
 - K (5)
 - D (4)
 - G (3)
 - I (3)
 - J (3)
 - A (2)
 - B (2)
 - E (2)
 - F (2)
 - C (1)

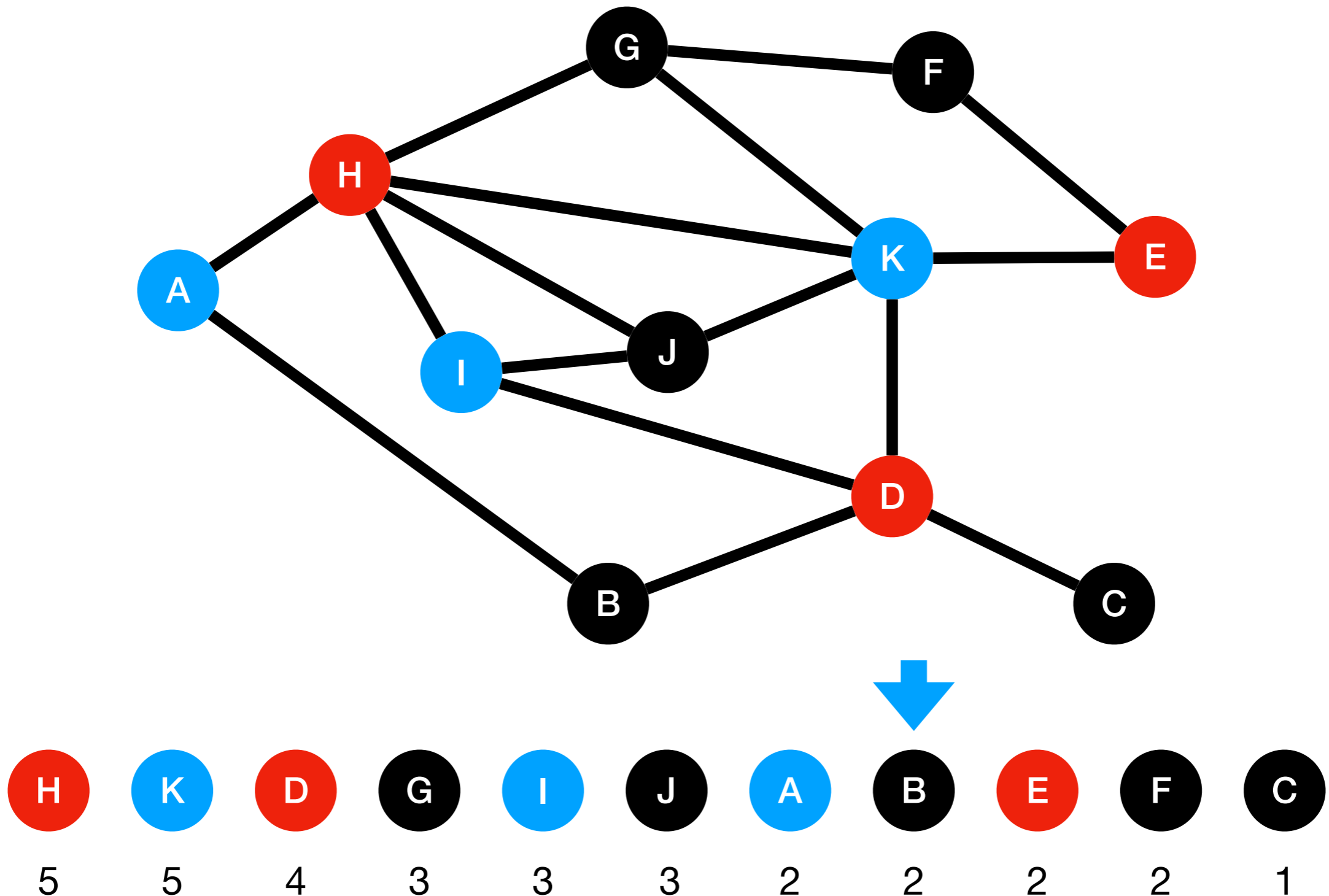
Algorithme de Welsh-Powell



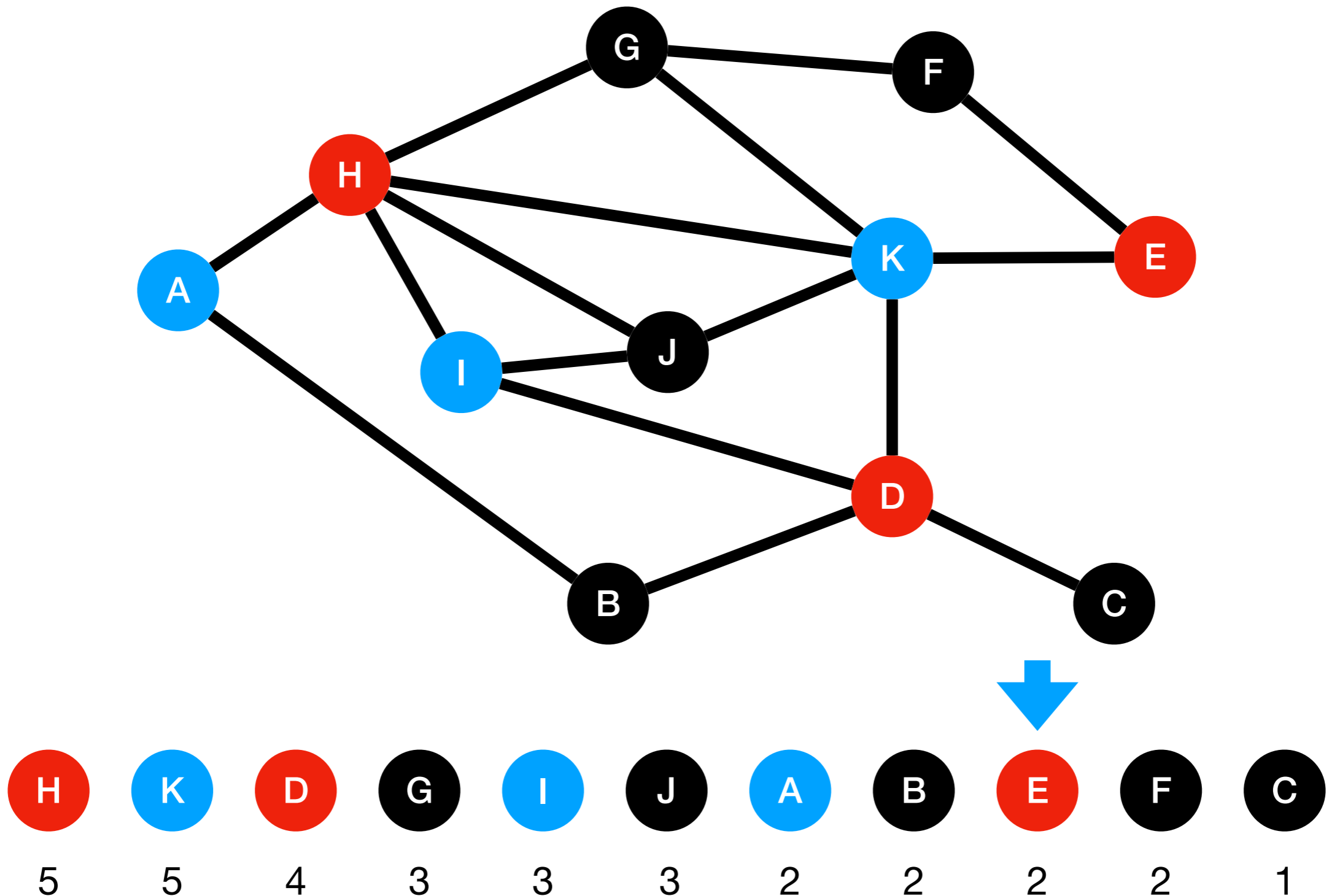
Algorithme de Welsh-Powell



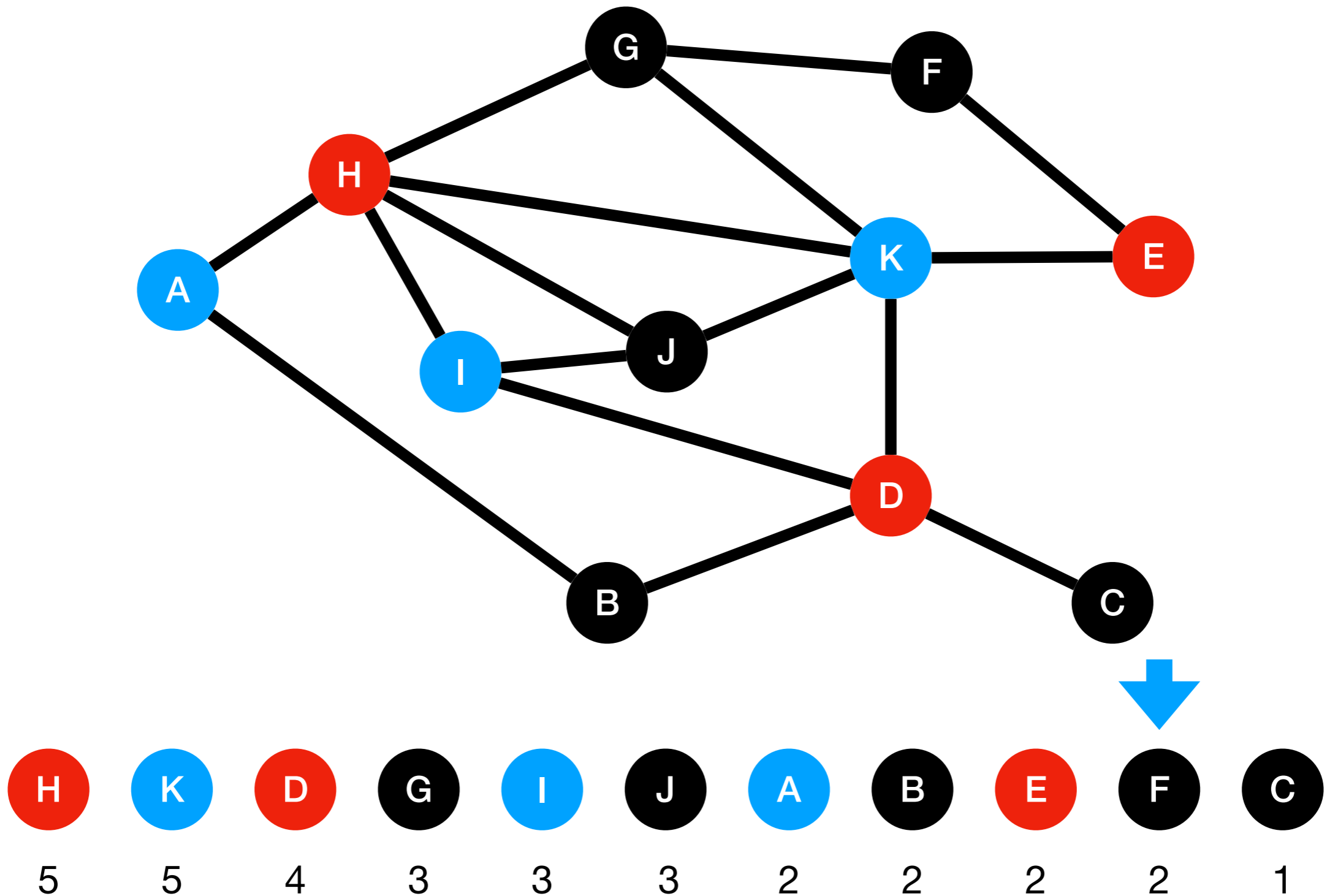
Algorithme de Welsh-Powell



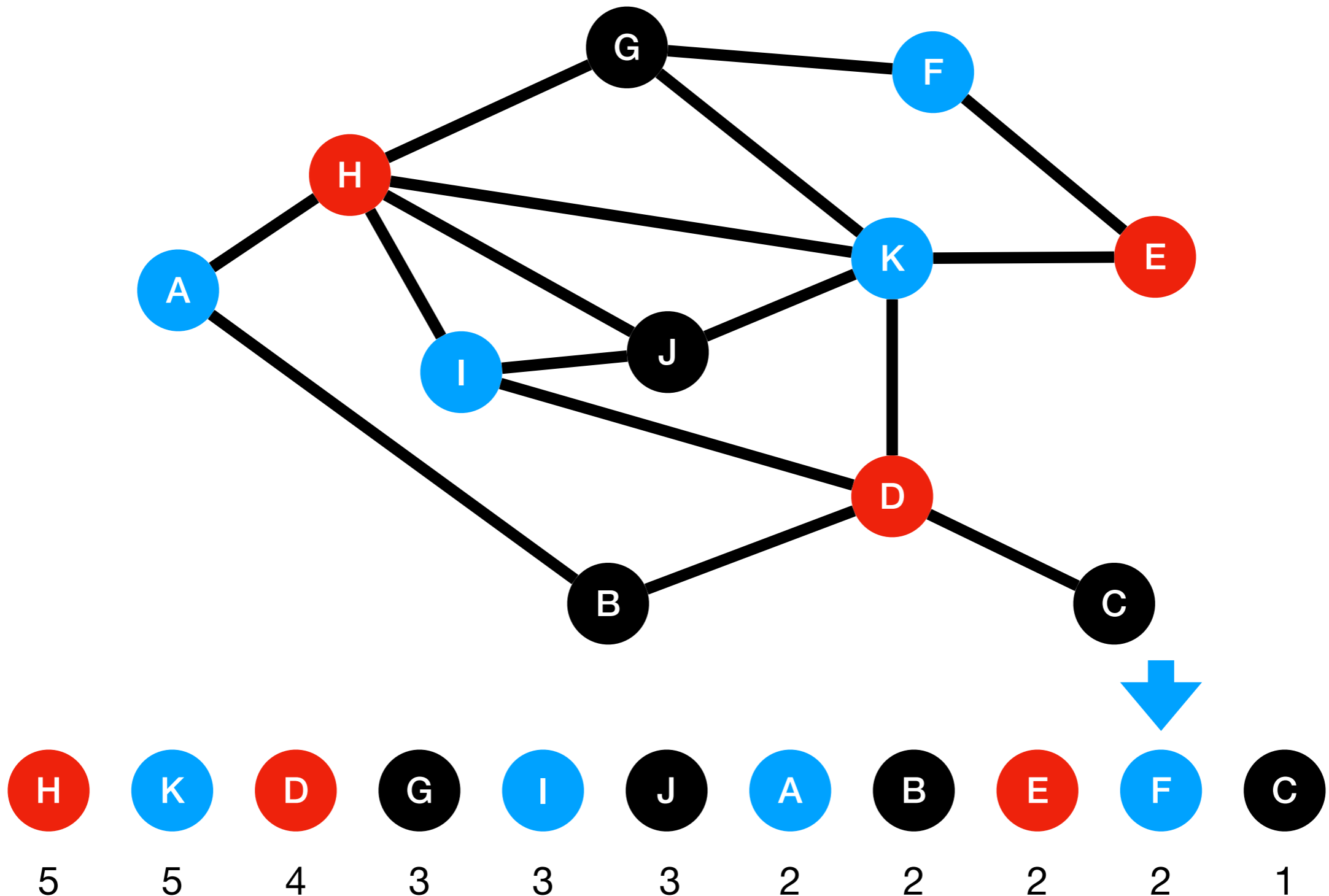
Algorithme de Welsh-Powell



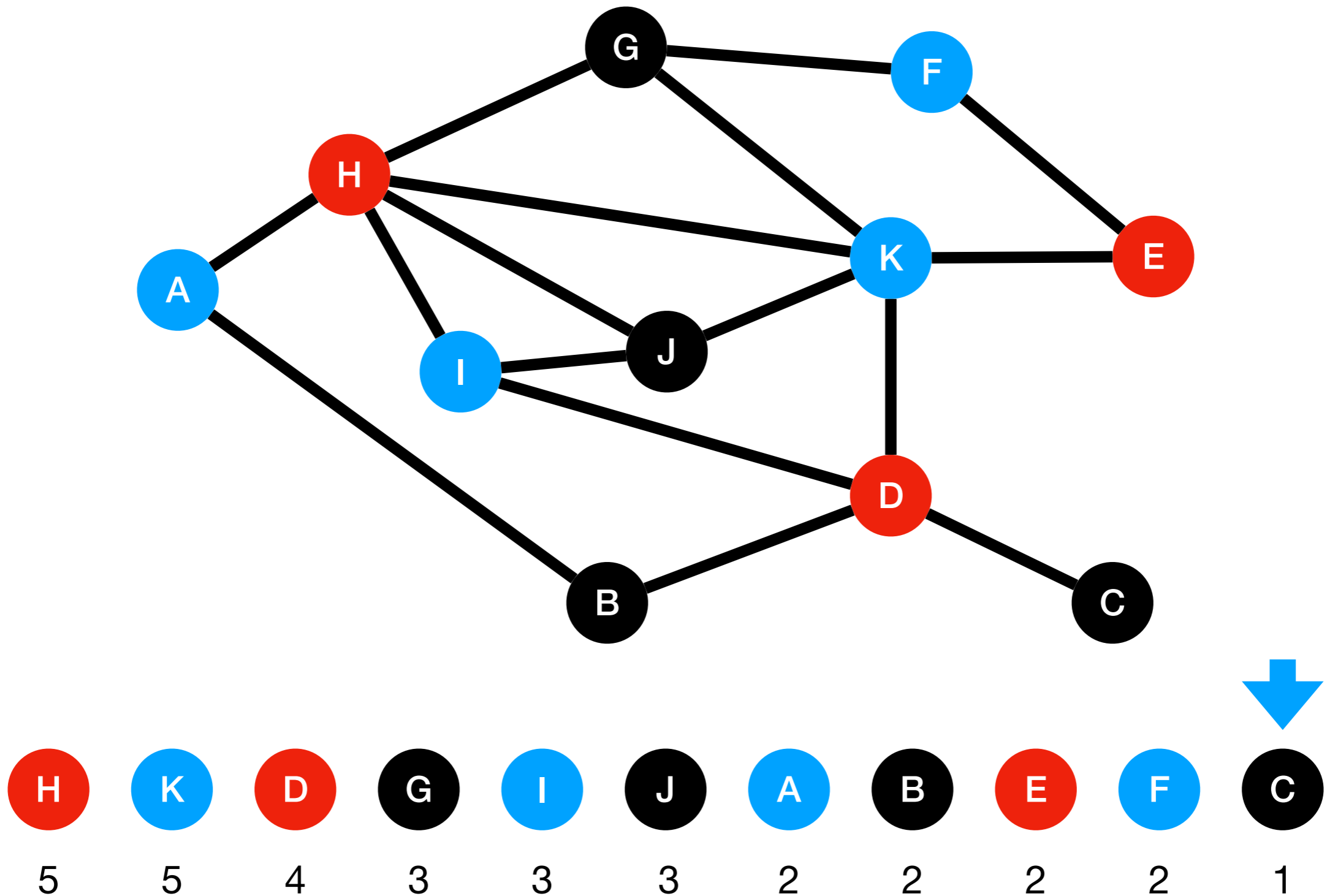
Algorithme de Welsh-Powell



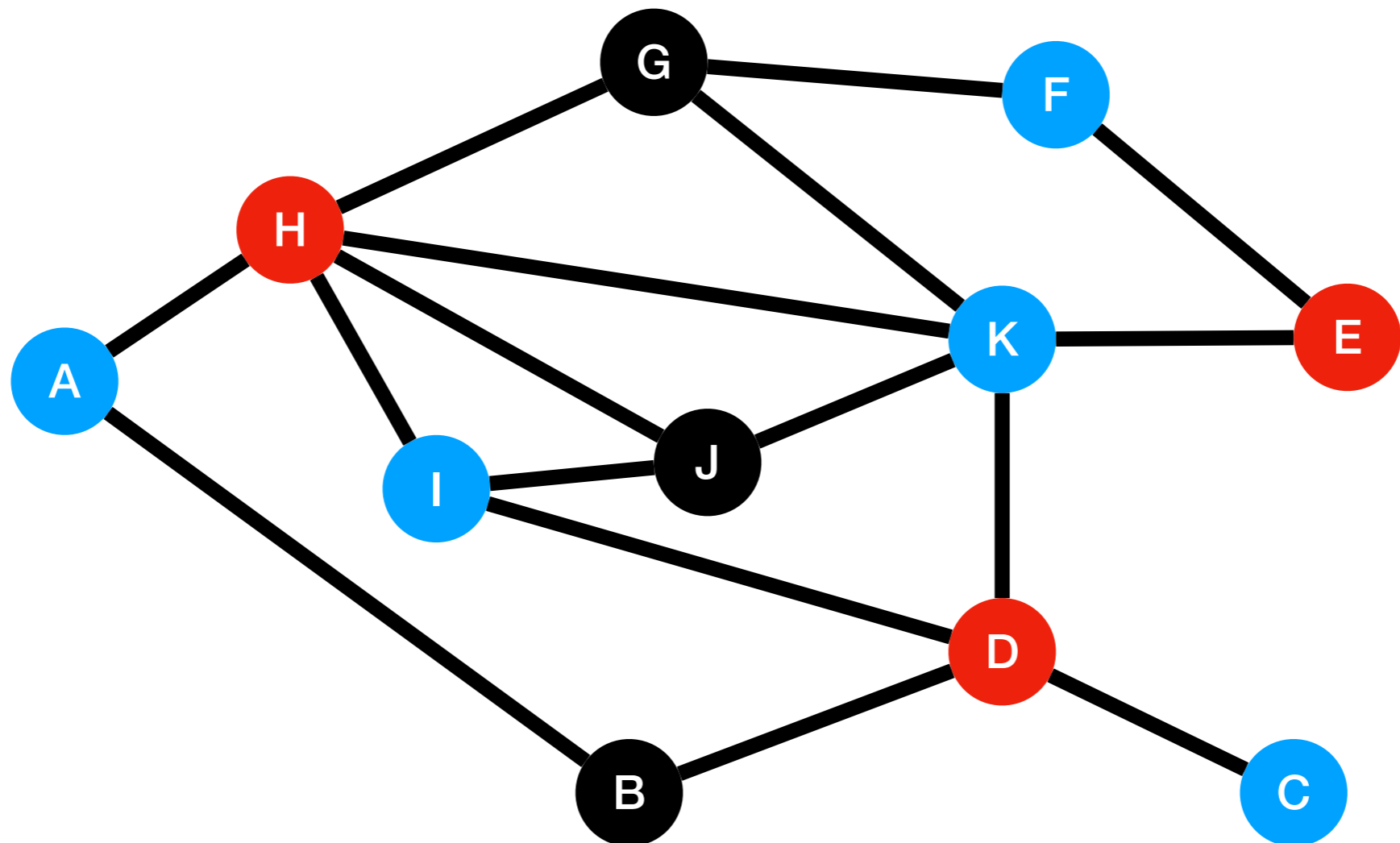
Algorithme de Welsh-Powell



Algorithme de Welsh-Powell



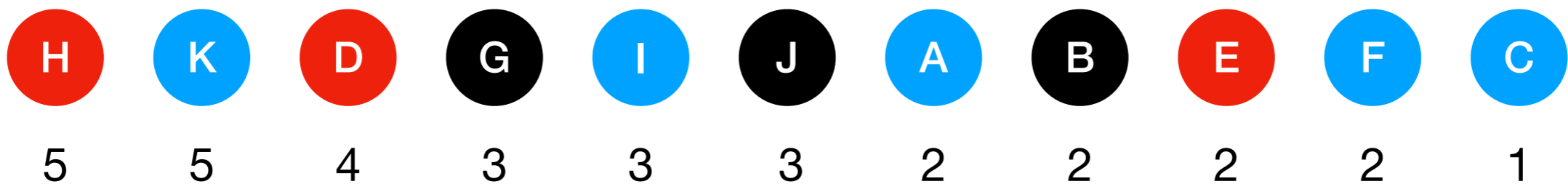
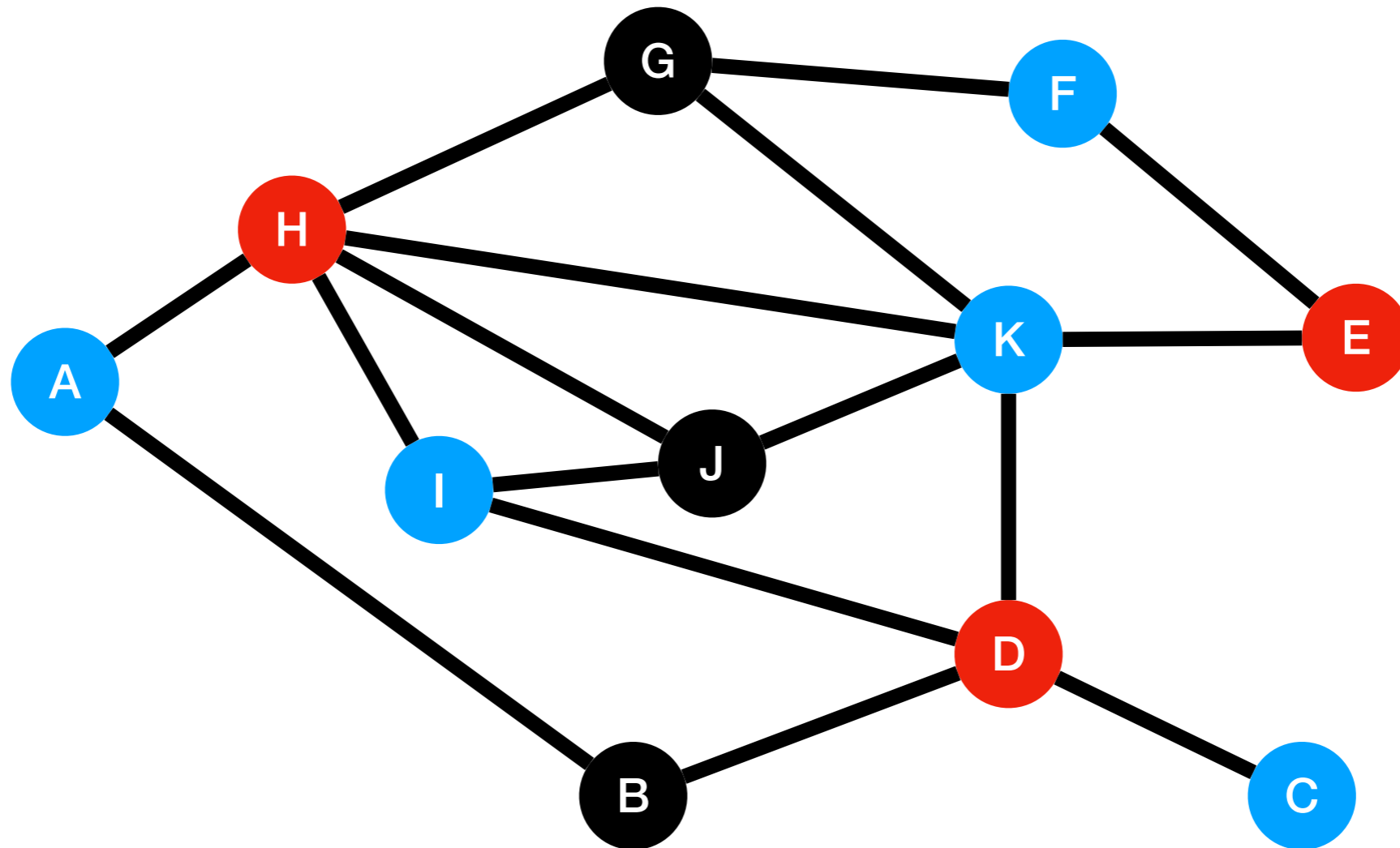
Algorithme de Welsh-Powell



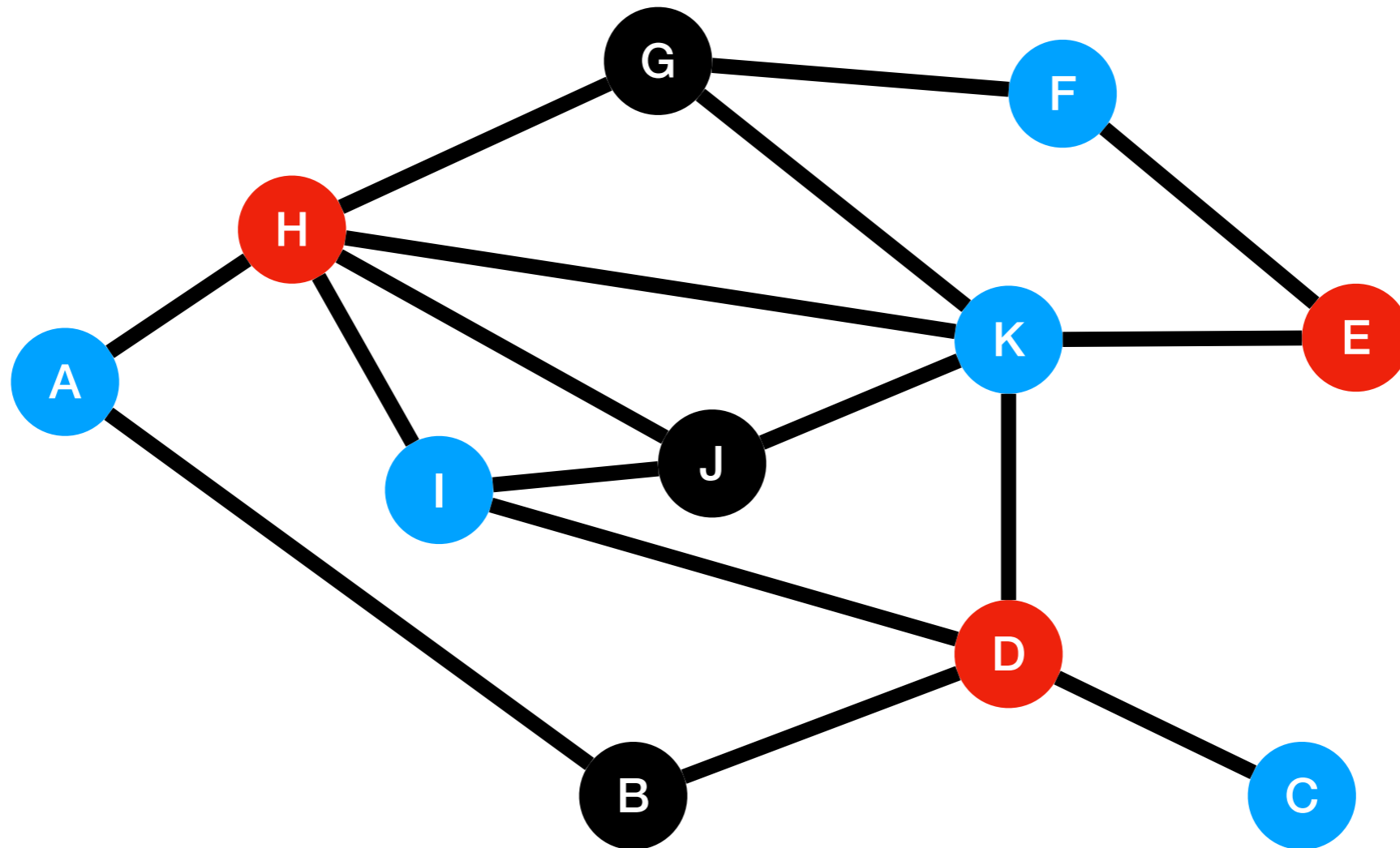
- H (red) 5
- K (blue) 5
- D (red) 4
- G (black) 3
- I (blue) 3
- J (black) 3
- A (blue) 2
- B (black) 2
- E (red) 2
- F (blue) 2
- C (blue) 1



Algorithme de Welsh-Powell



Algorithme de Welsh-Powell



5



5



4



3



3



3



2



2



2

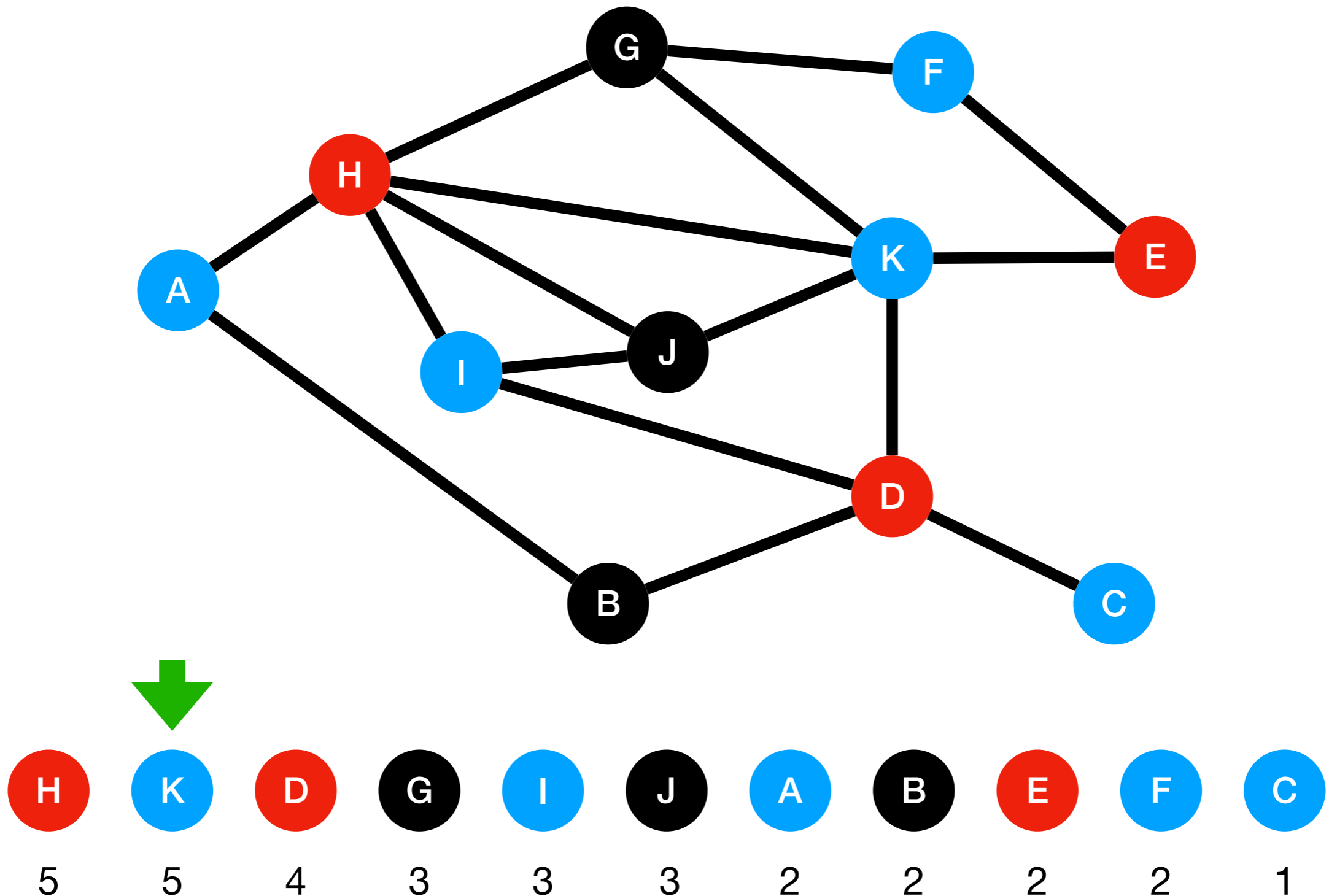


2

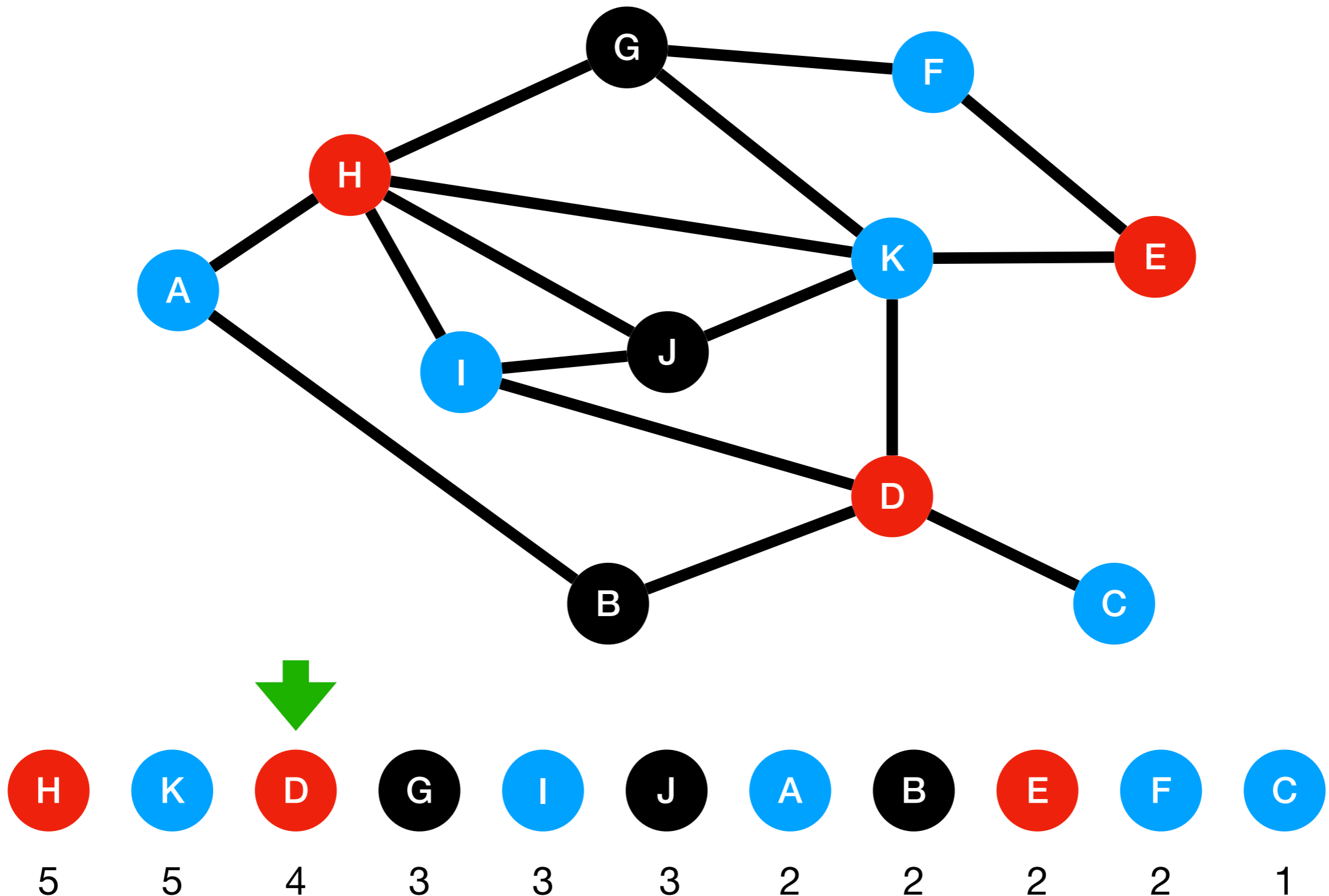


1

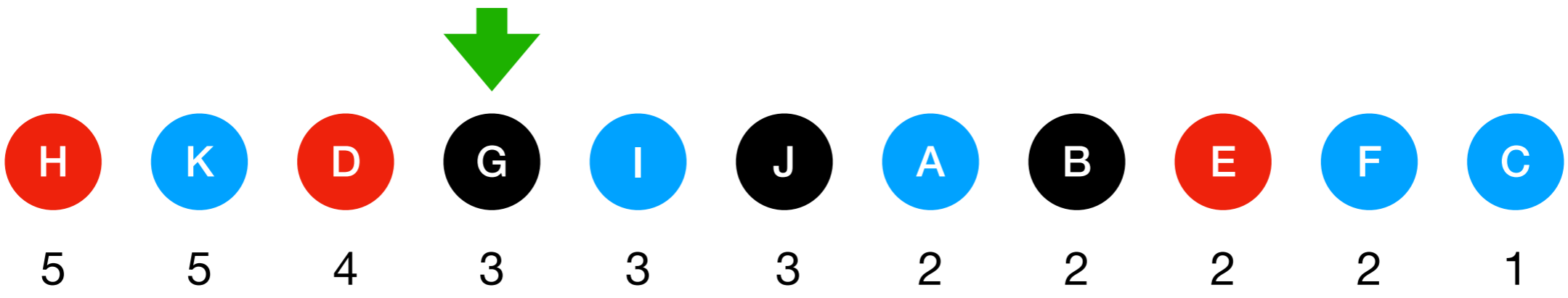
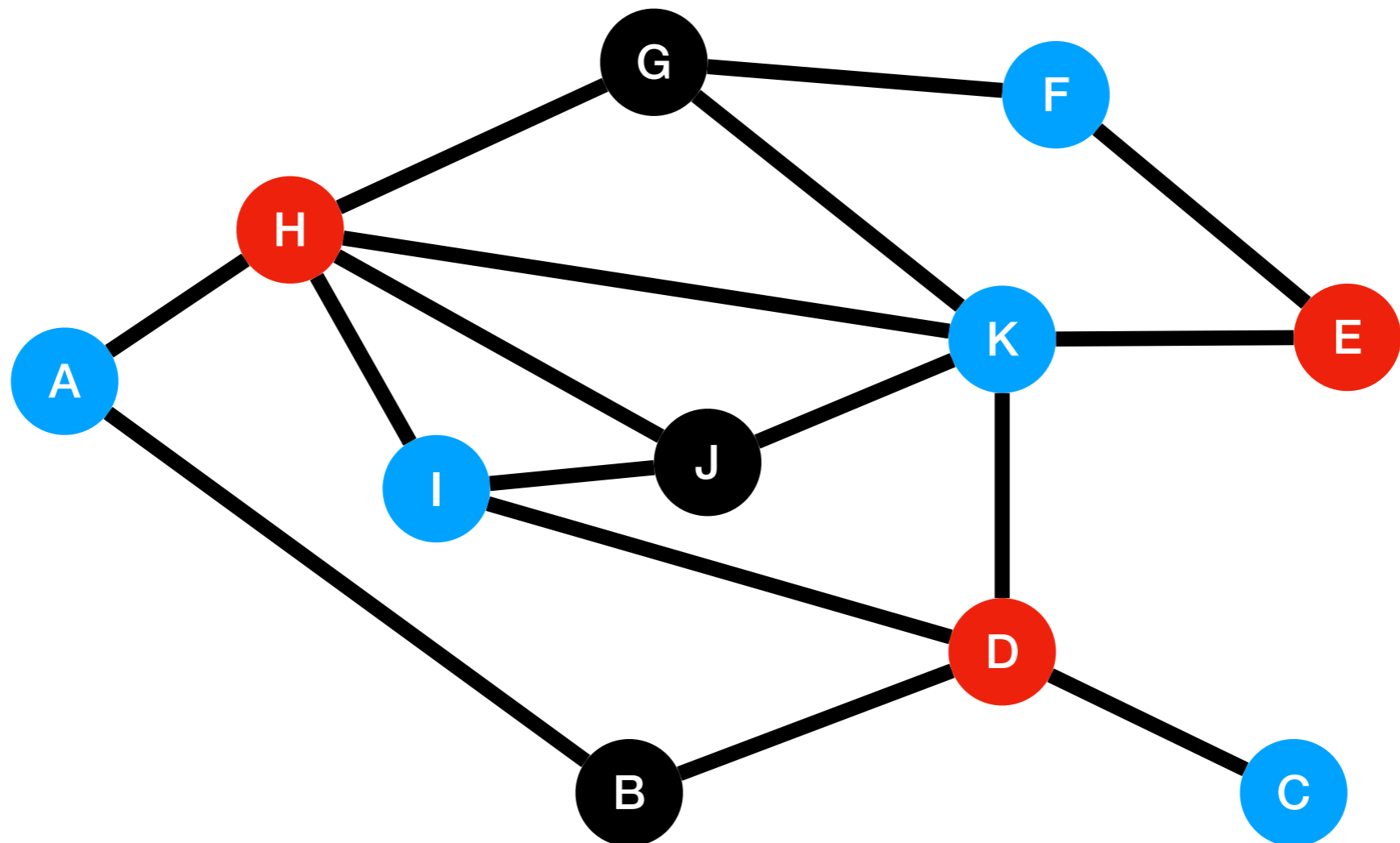
Algorithme de Welsh-Powell



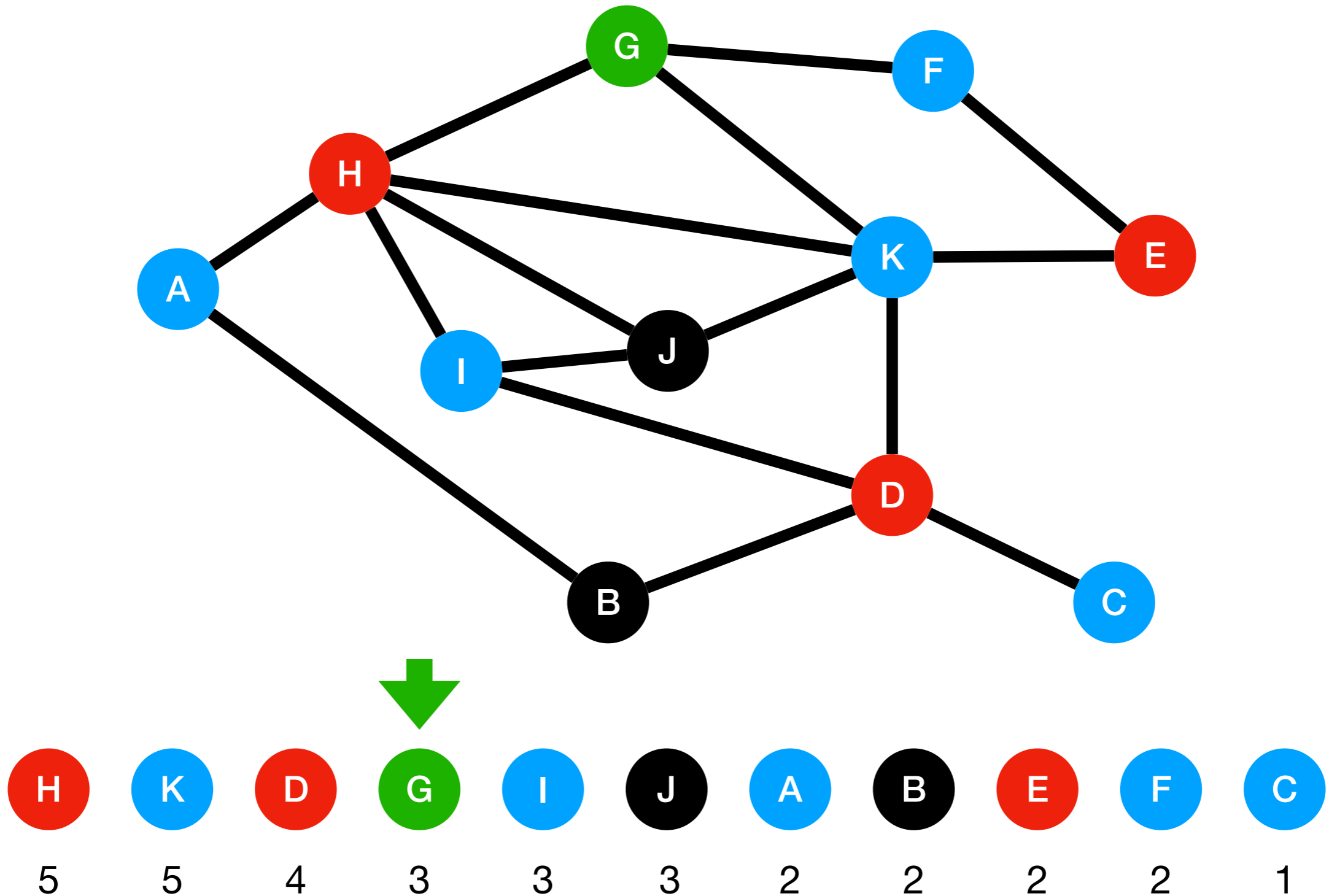
Algorithme de Welsh-Powell



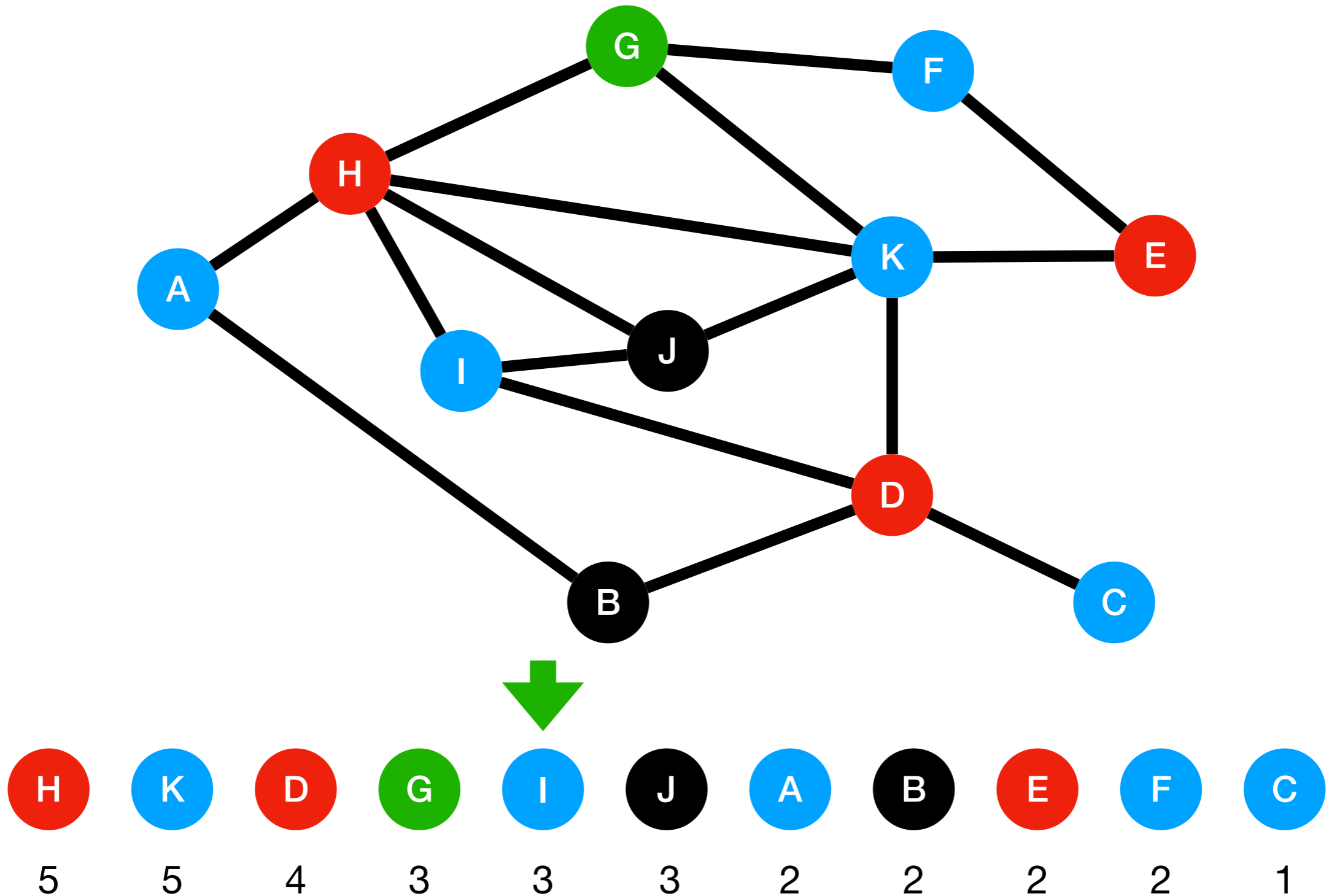
Algorithme de Welsh-Powell



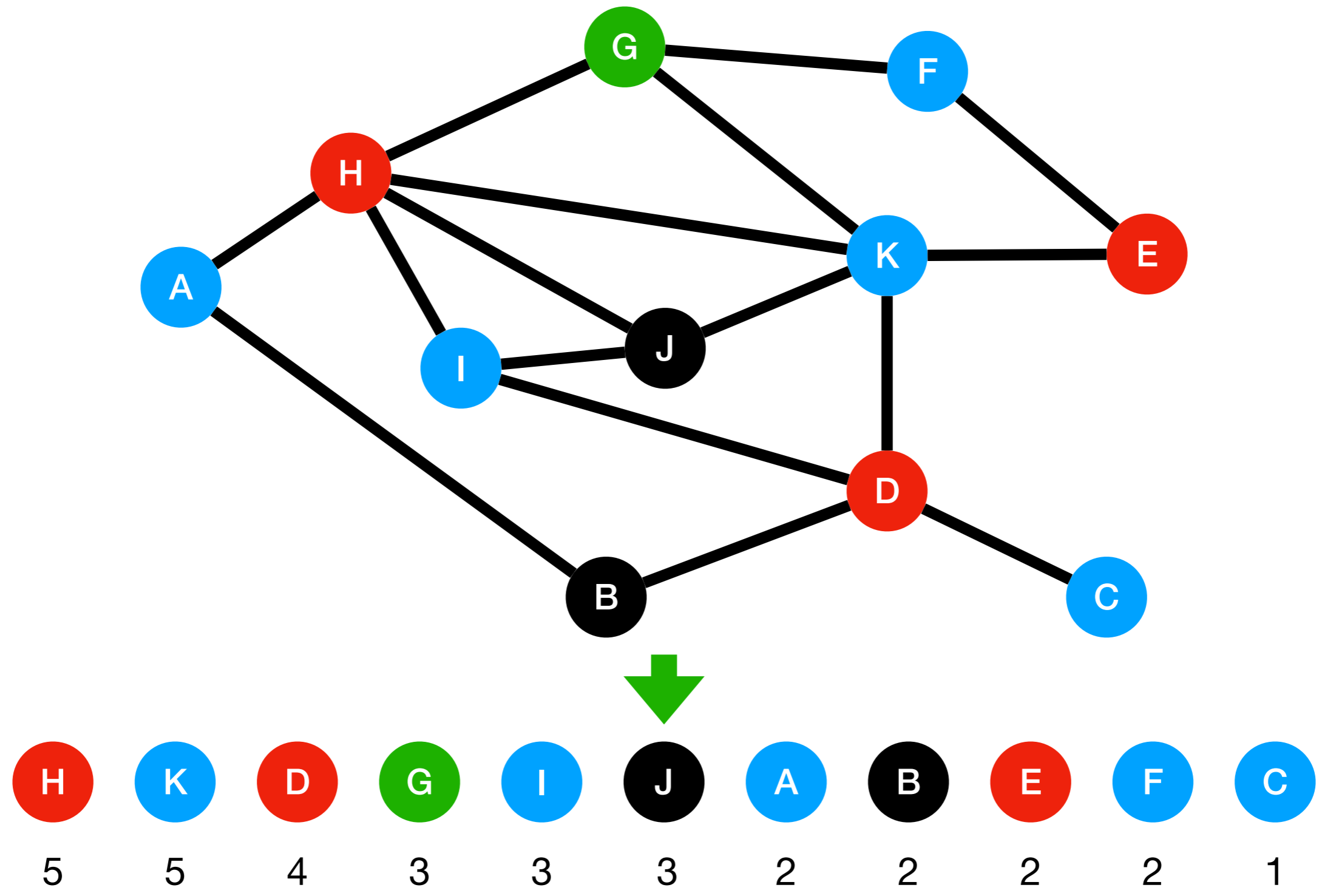
Algorithme de Welsh-Powell



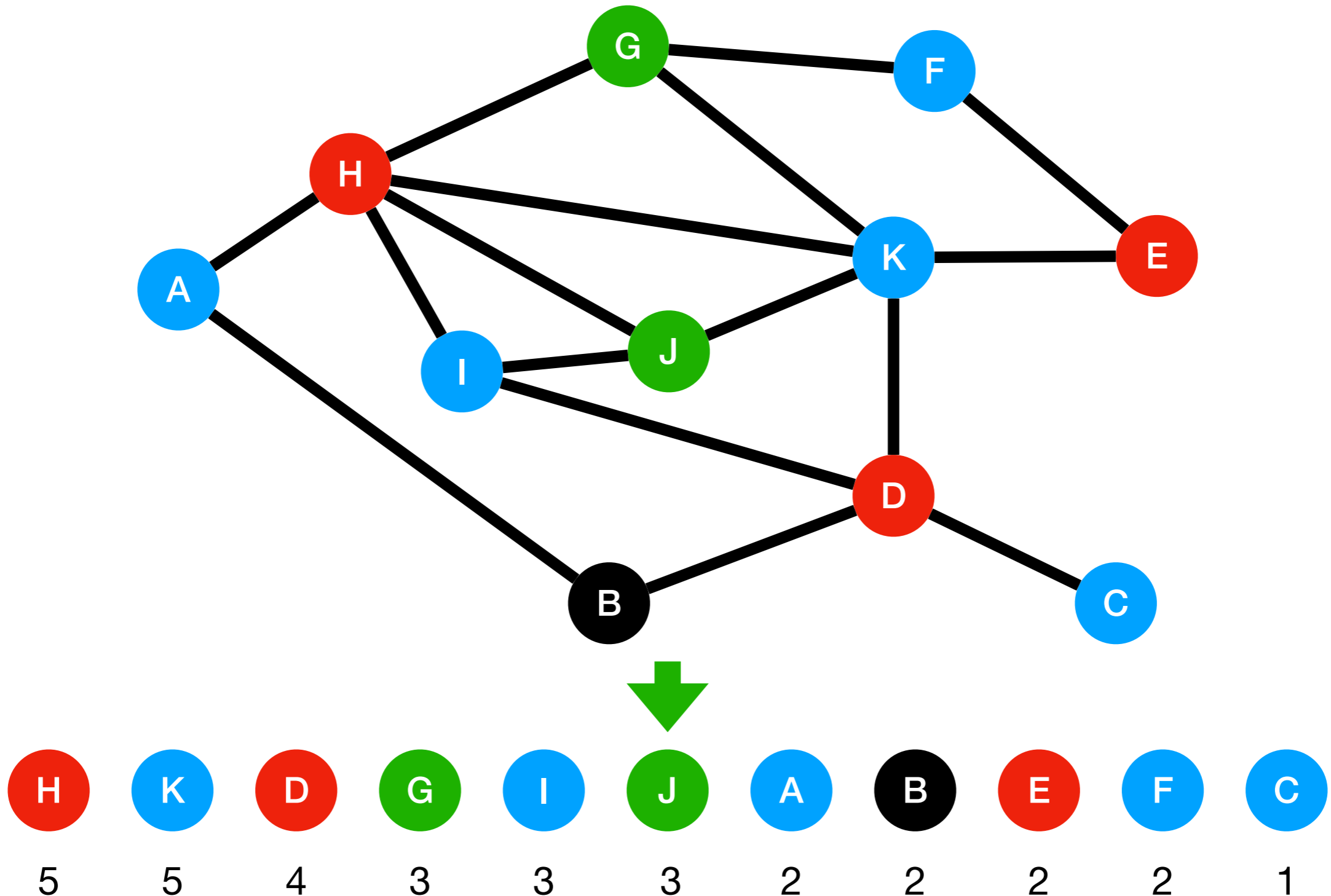
Algorithme de Welsh-Powell



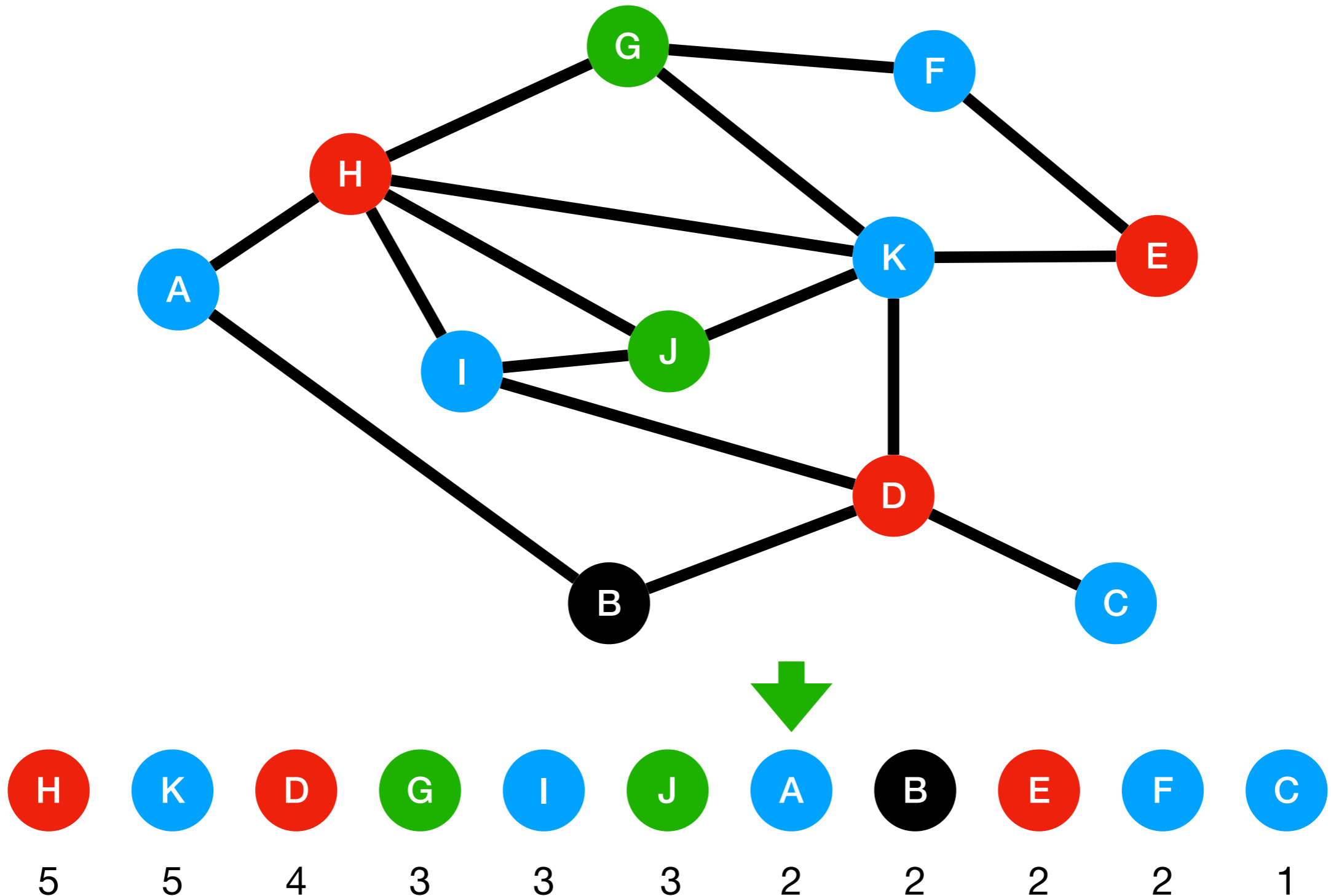
Algorithme de Welsh-Powell



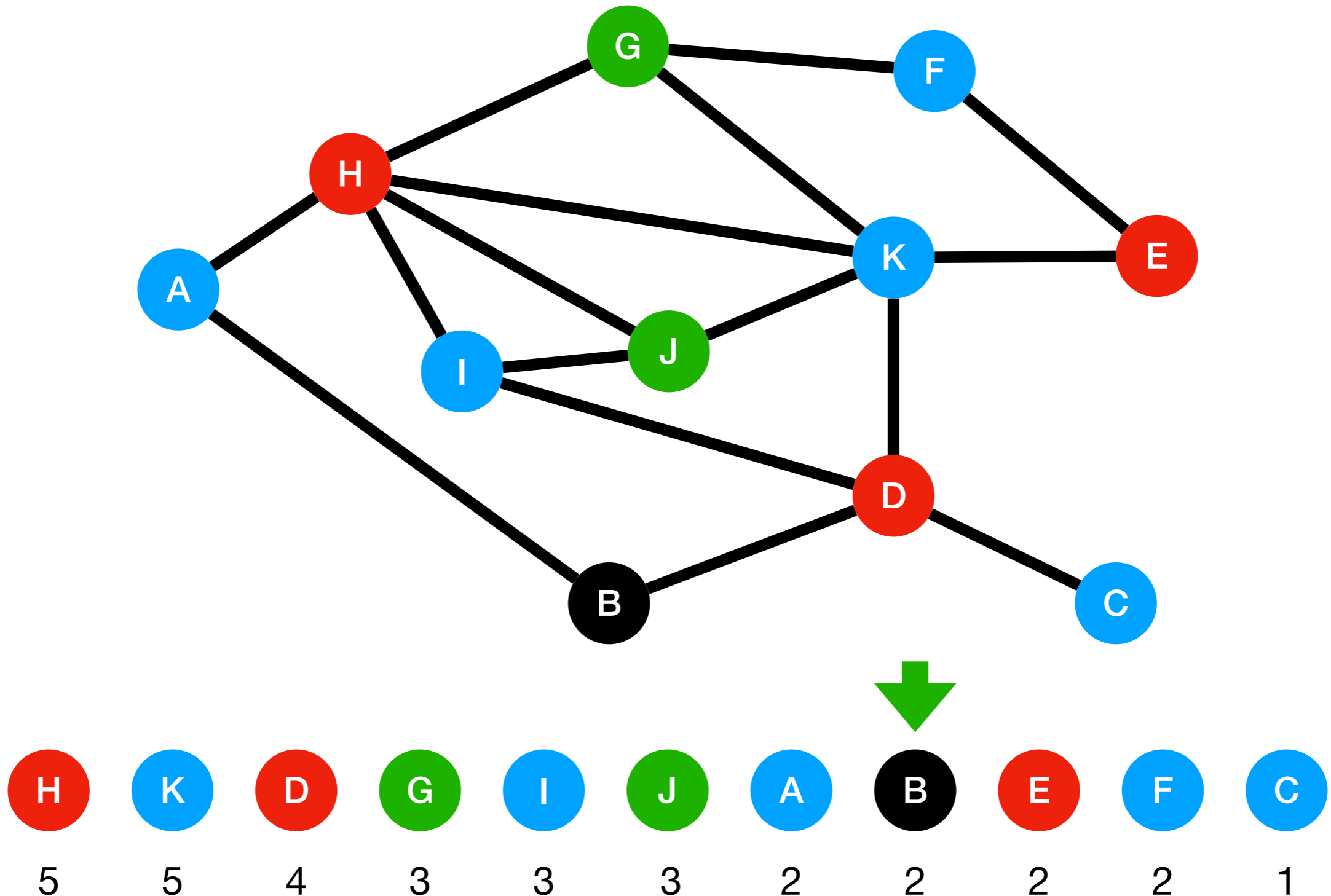
Algorithme de Welsh-Powell



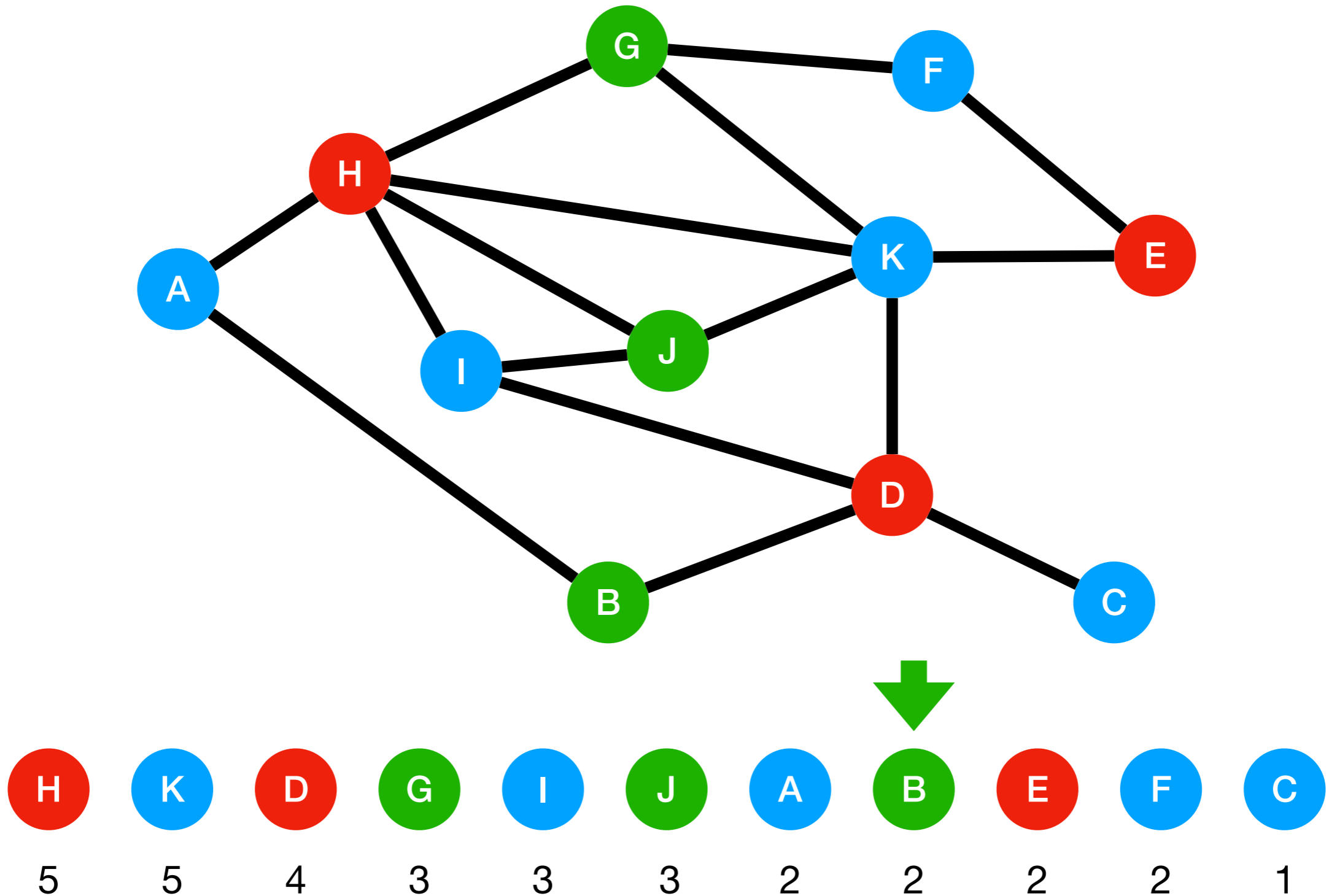
Algorithme de Welsh-Powell



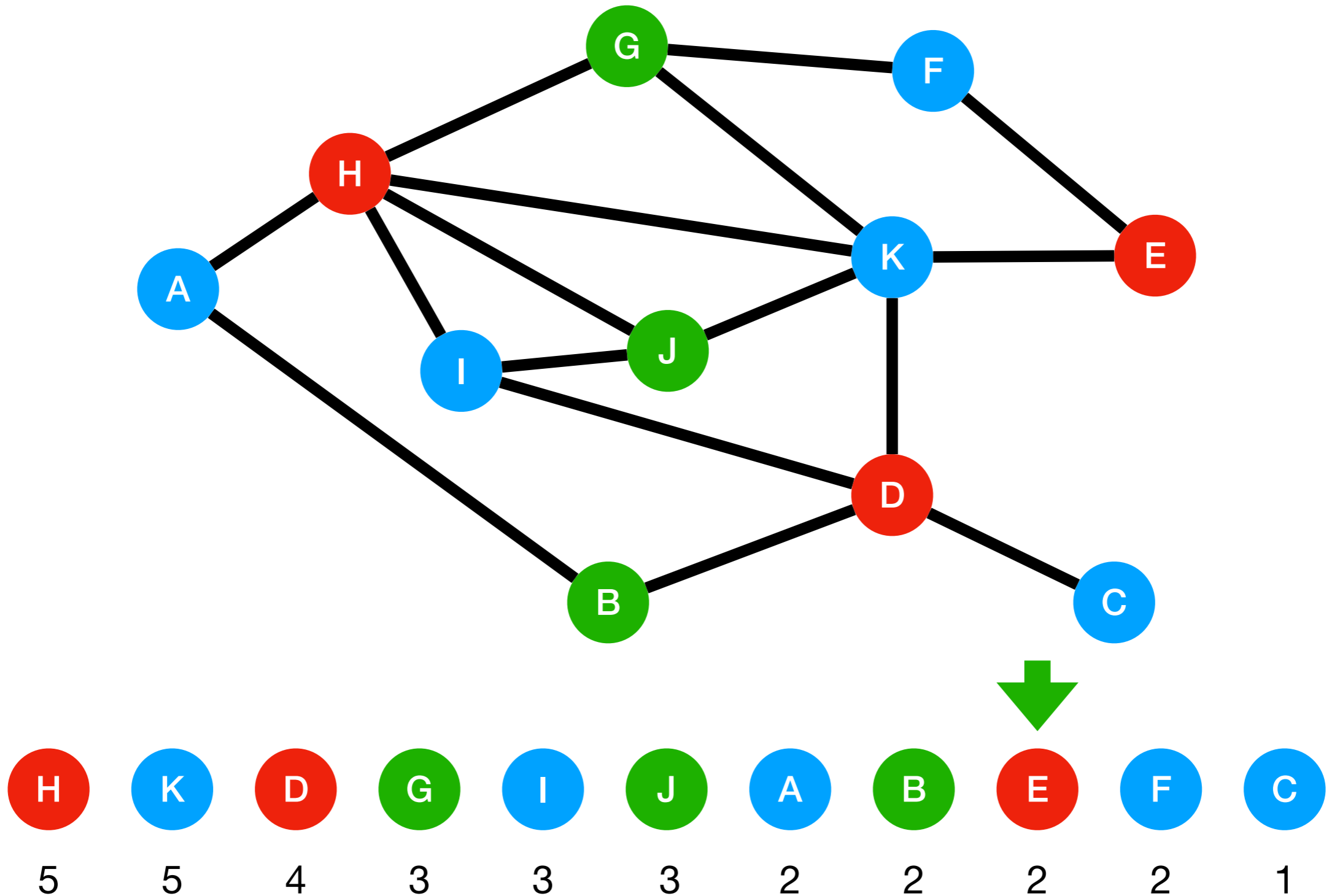
Algorithme de Welsh-Powell



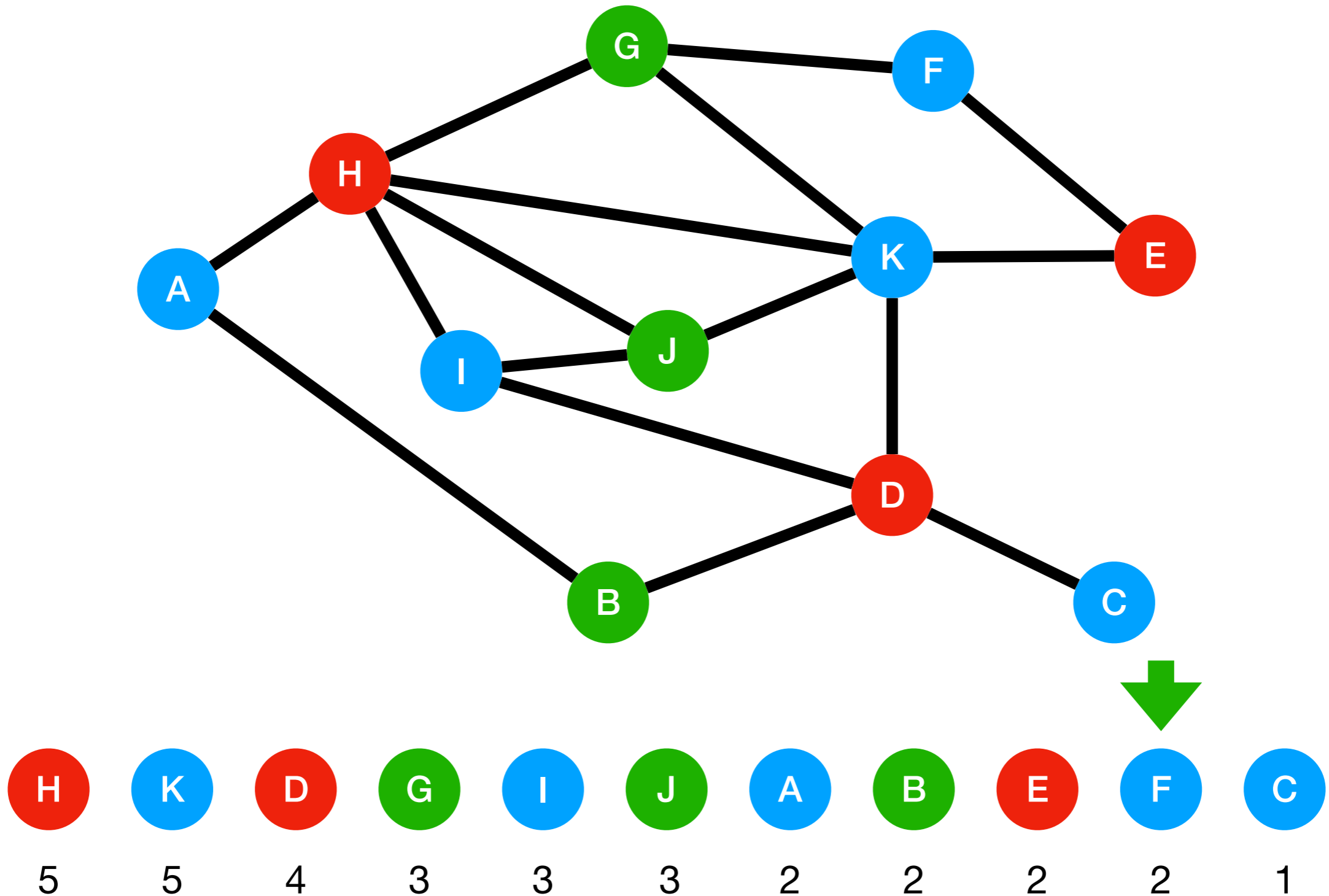
Algorithme de Welsh-Powell



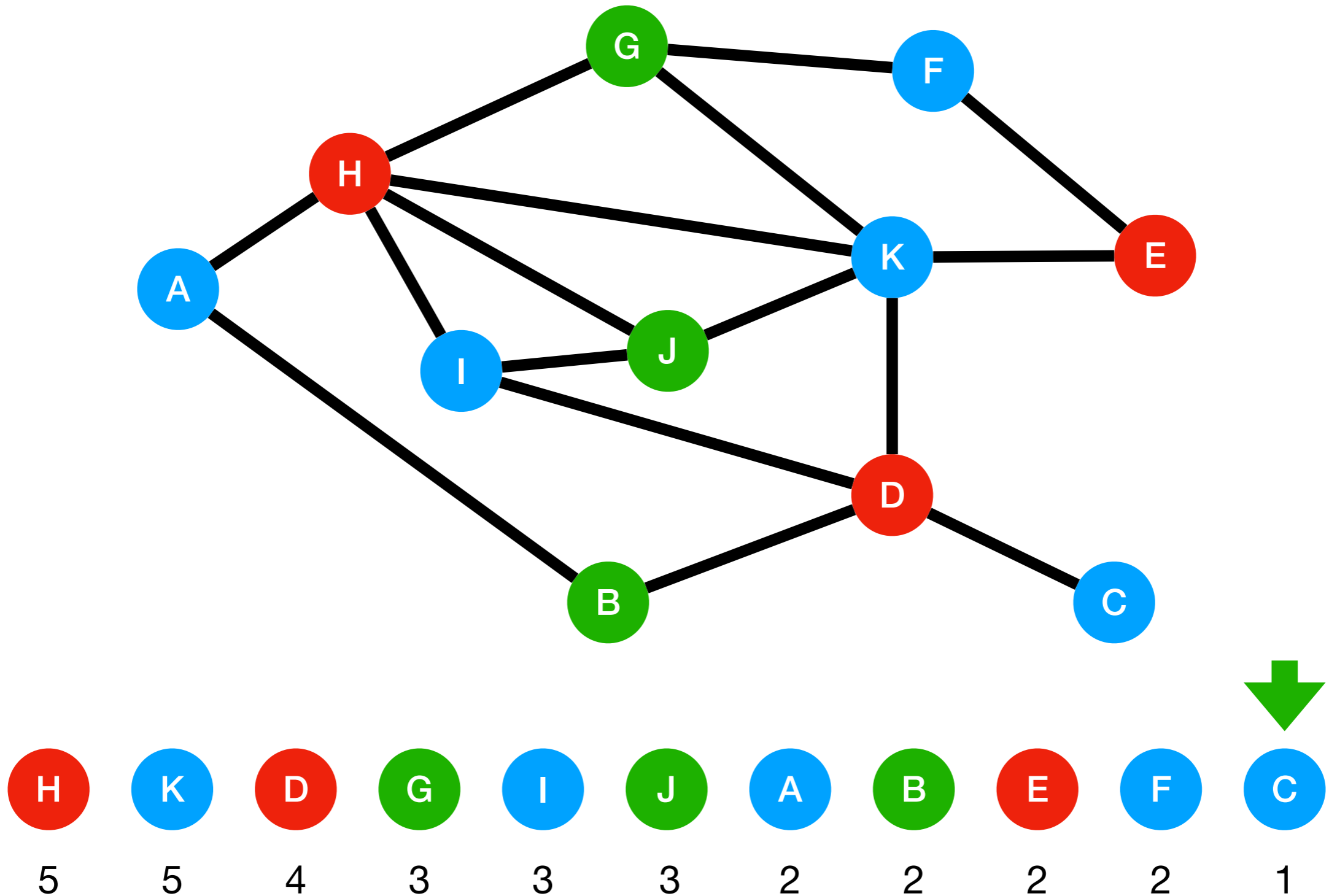
Algorithme de Welsh-Powell



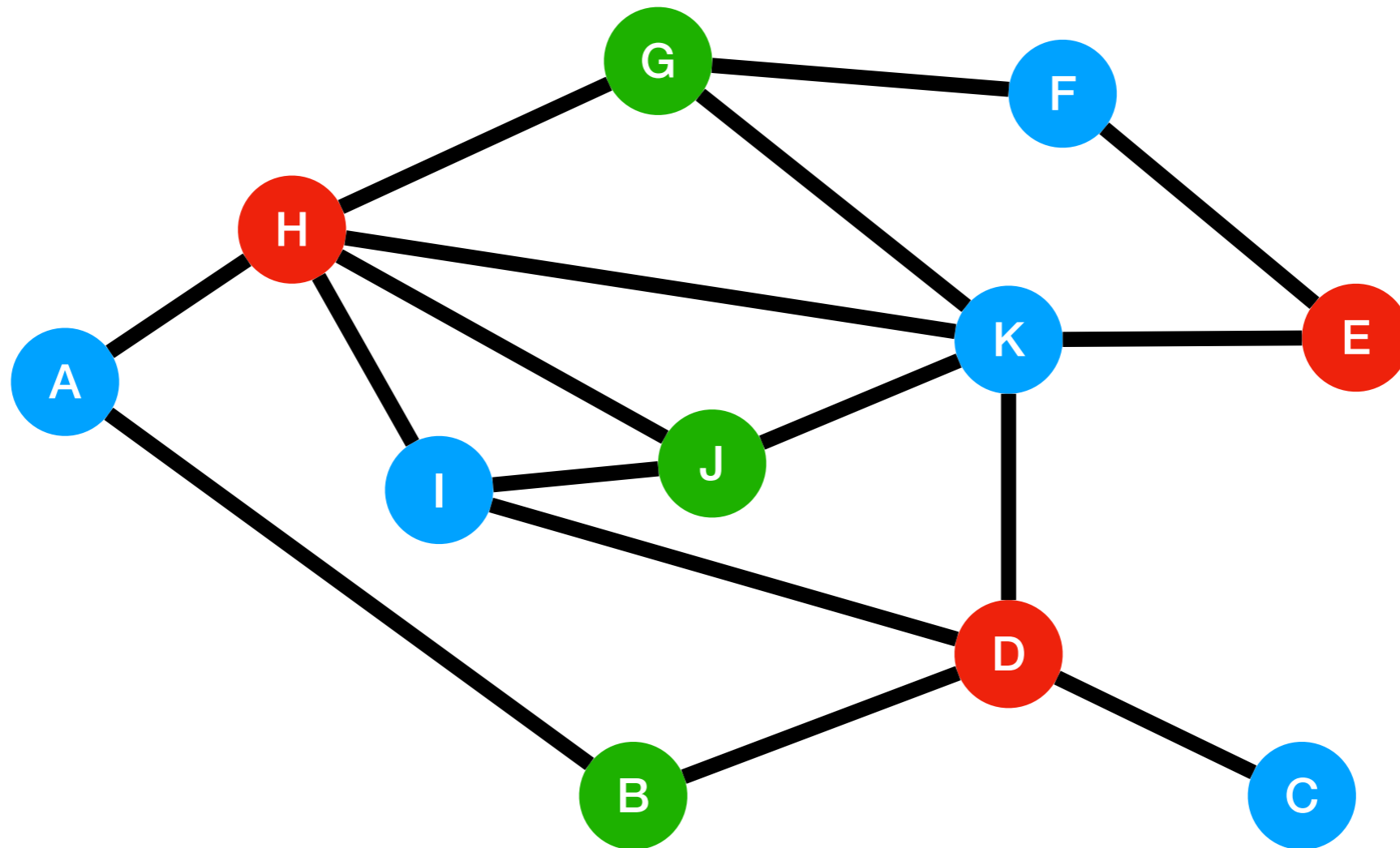
Algorithme de Welsh-Powell



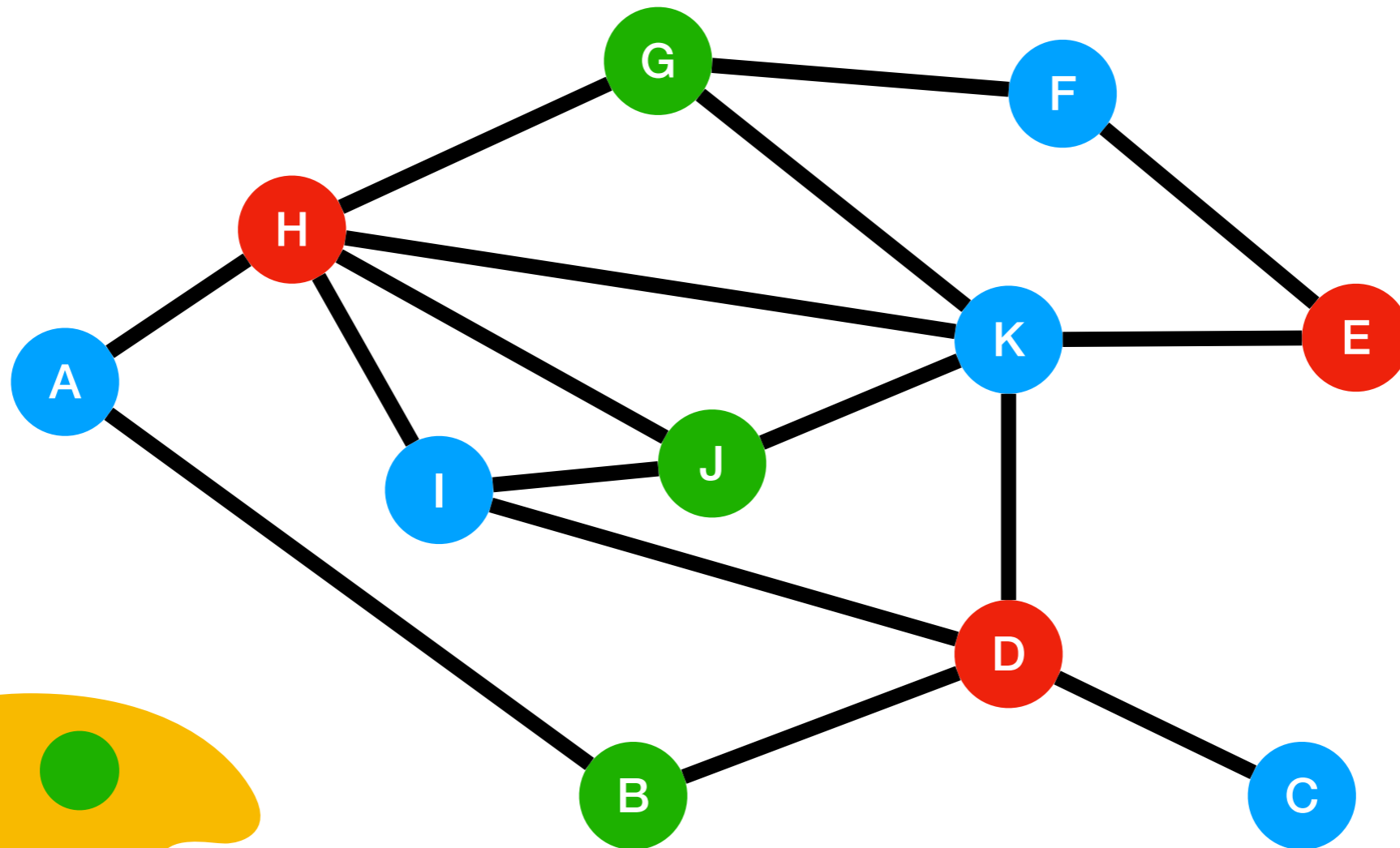
Algorithme de Welsh-Powell



Algorithme de Welsh-Powell

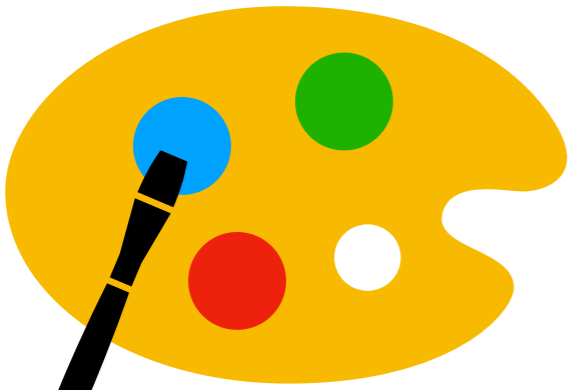
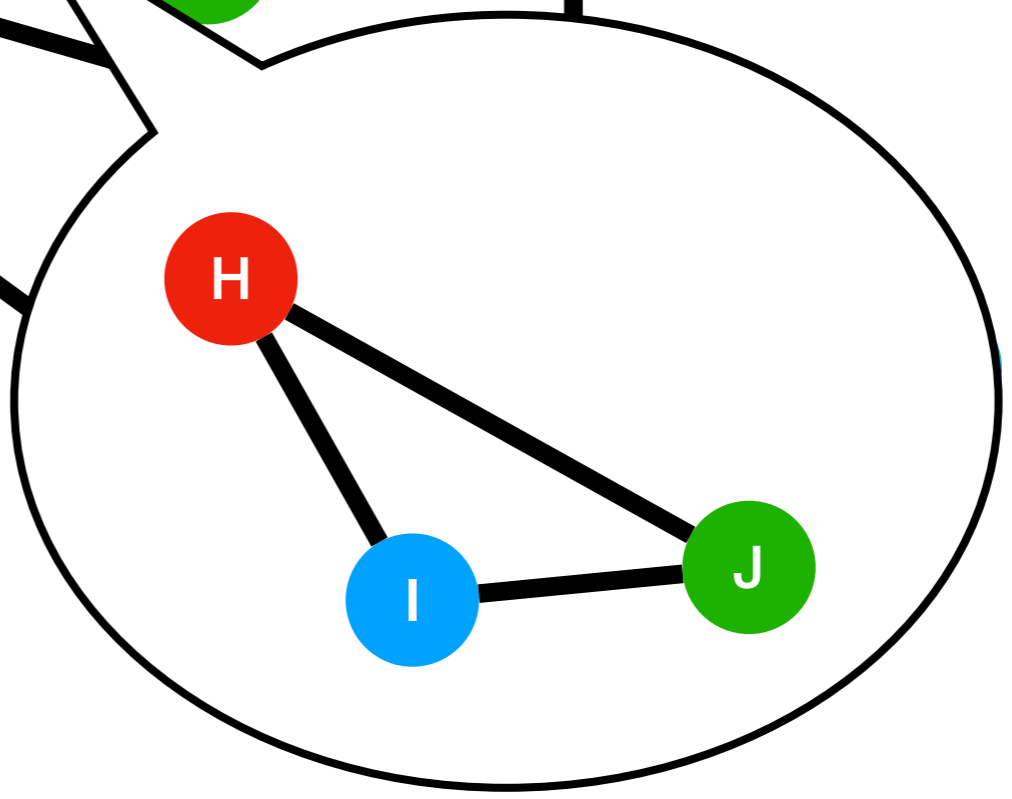
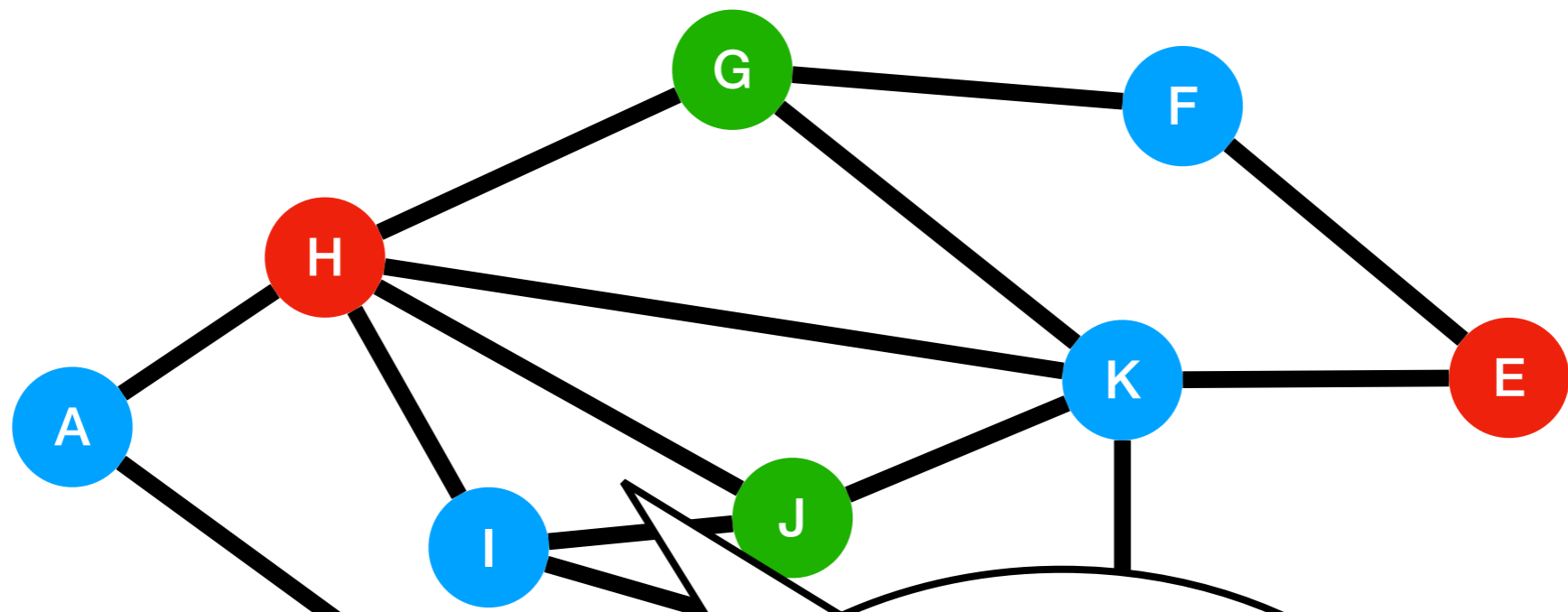


Algorithme de Welsh-Powell



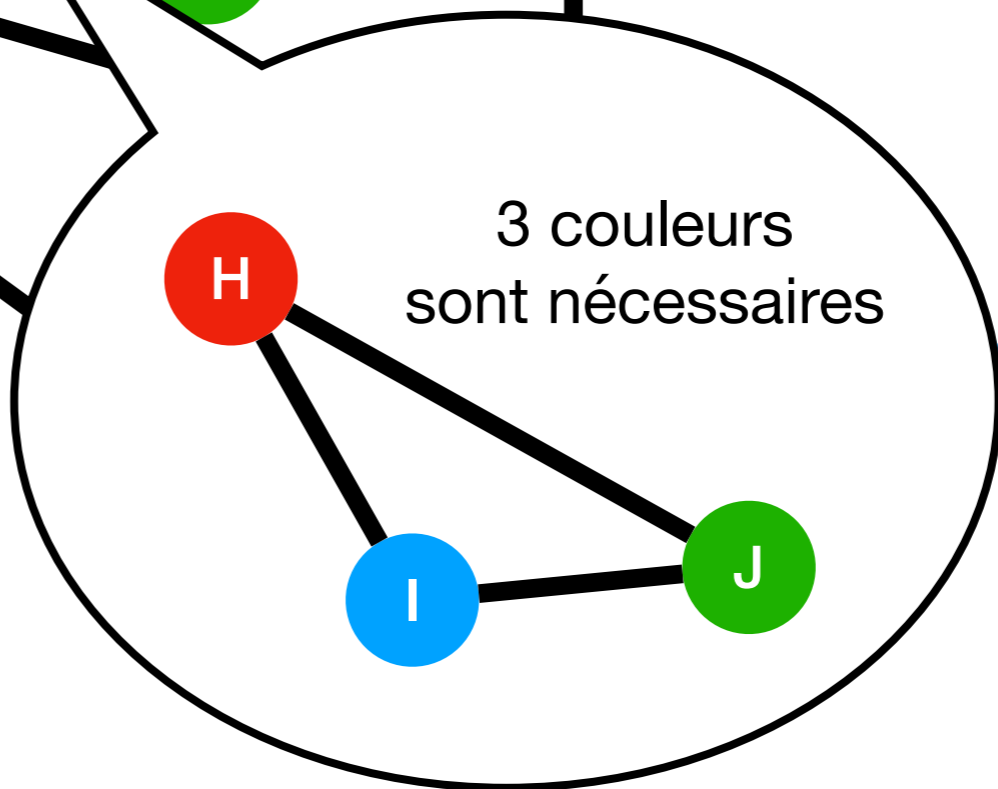
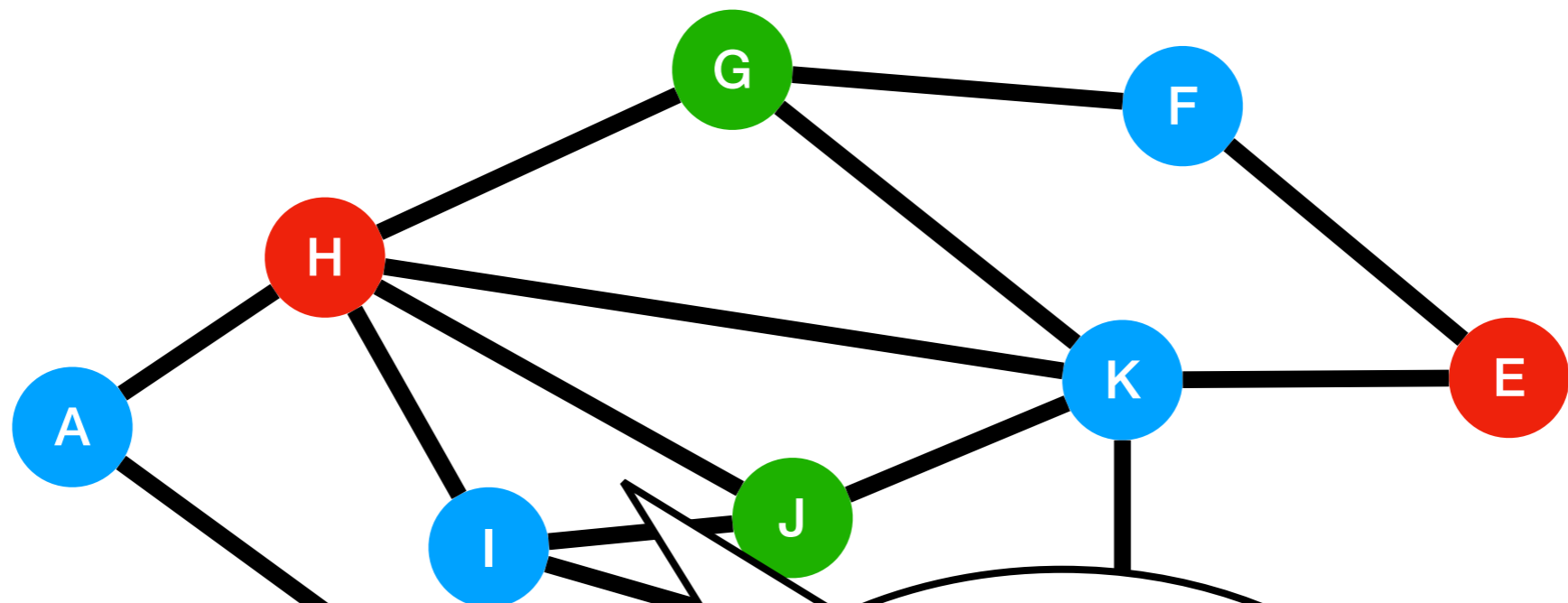
3 couleurs
suffisent

Algorithme de Welsh-Powell



3 couleurs
suffisent

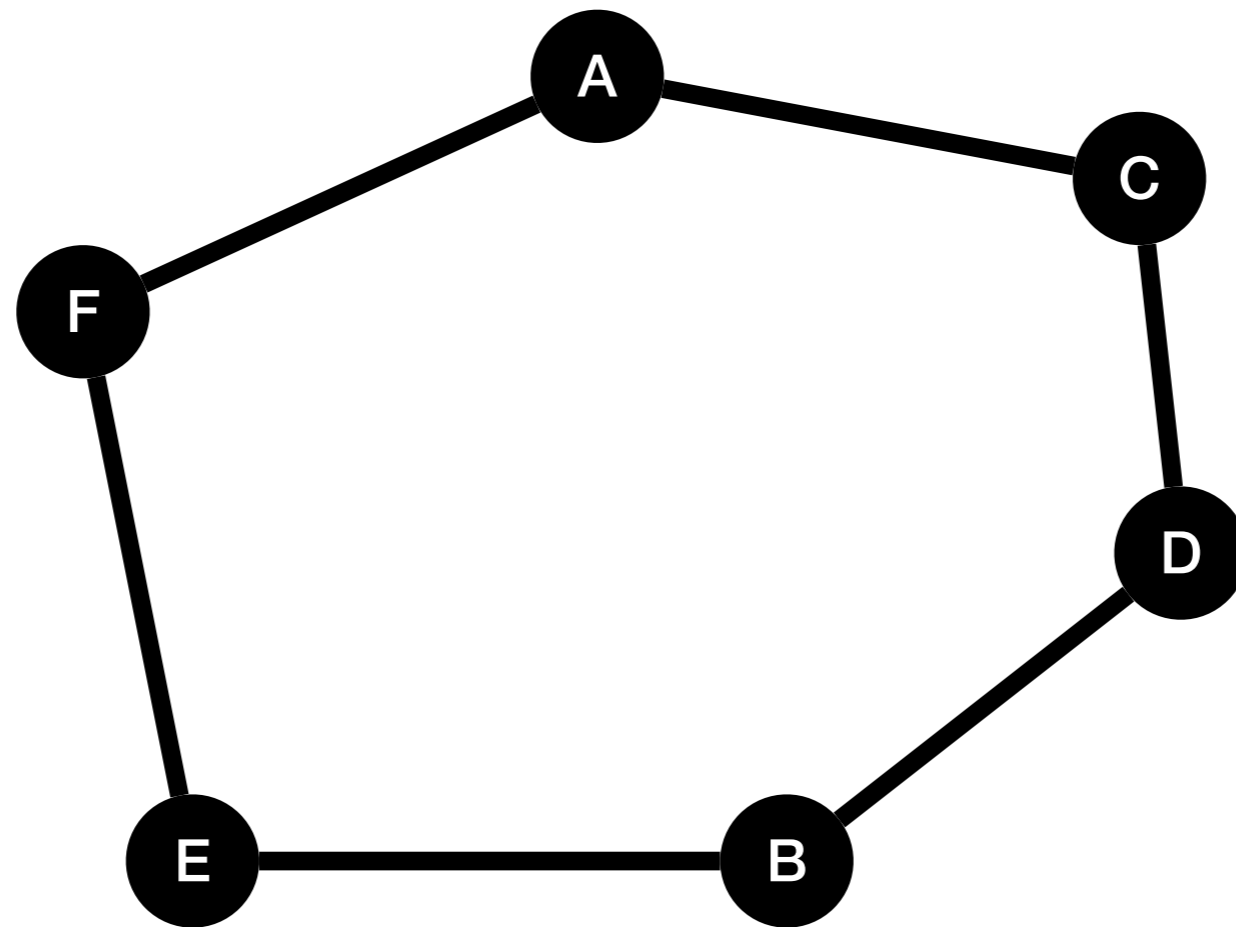
Algorithme de Welsh-Powell



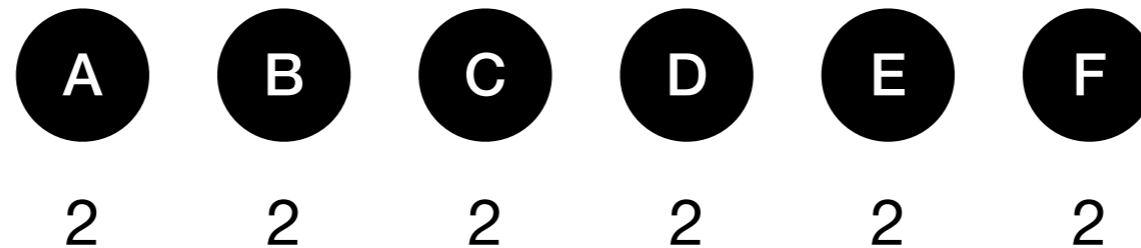
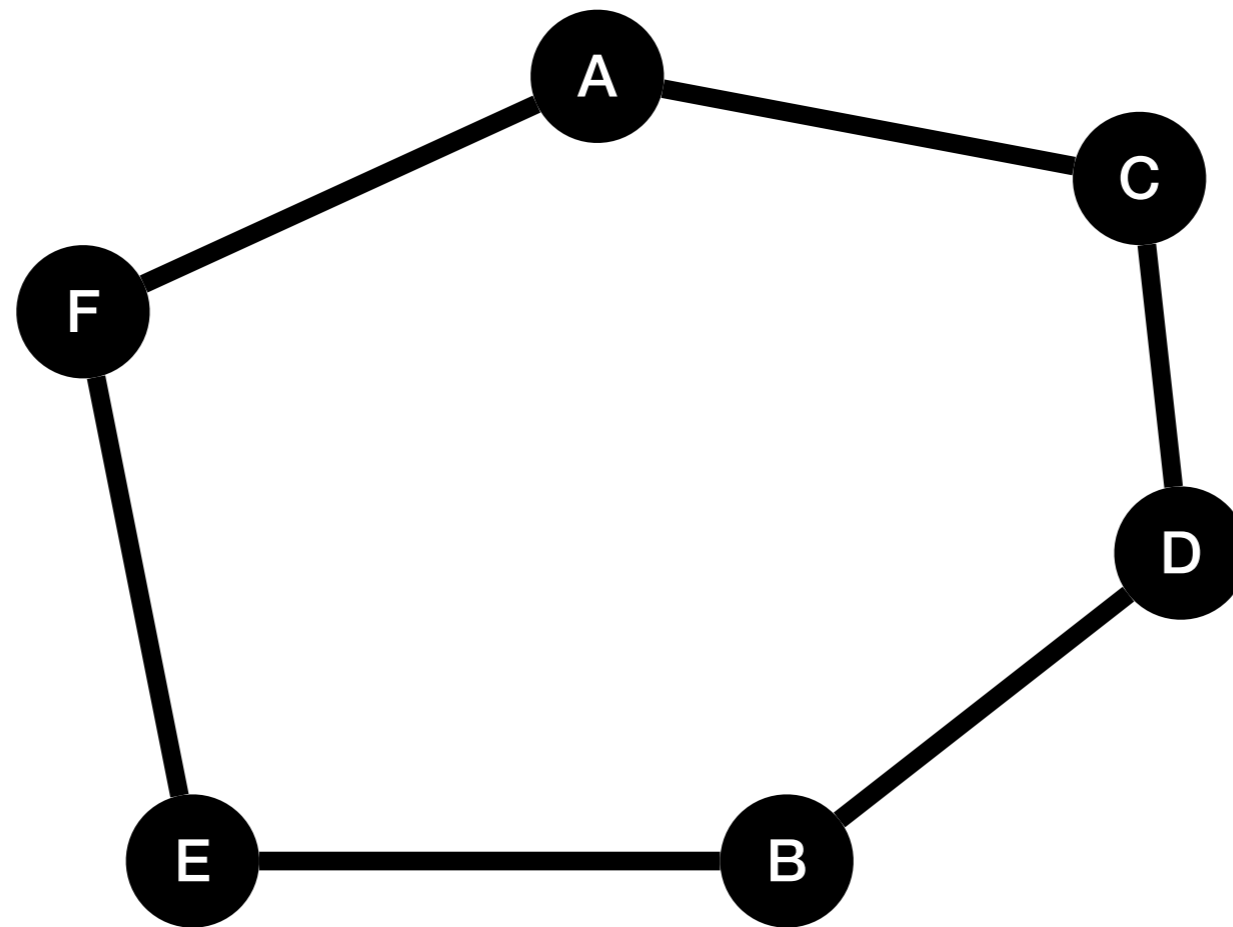
Algorithme de Welsh-Powell

- Trier les sommets du graphe par ordre de degré (nombre de voisins) décroissant
- **couleur** := rouge
- **Tant qu'il y a encore des sommets en noir faire**
 - Parcourir la liste triée des sommets et colorier en **couleur** les sommets en noir qui ne sont pas connectés à d'autres sommets de la même couleur
 - Choisir une nouvelle **couleur**

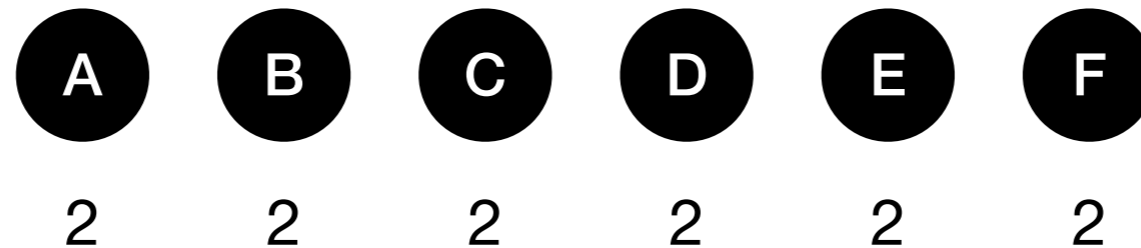
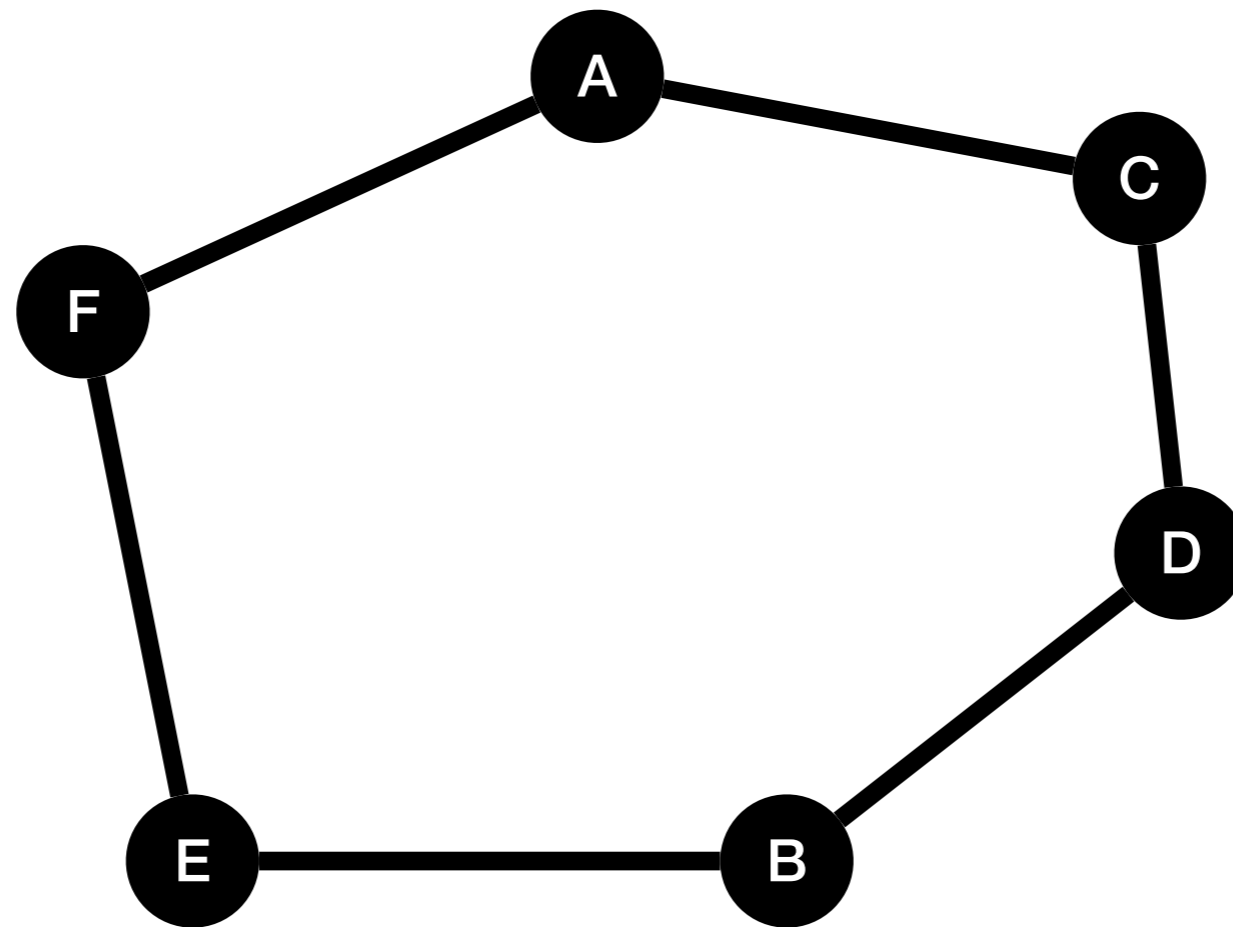
Colorations non optimales



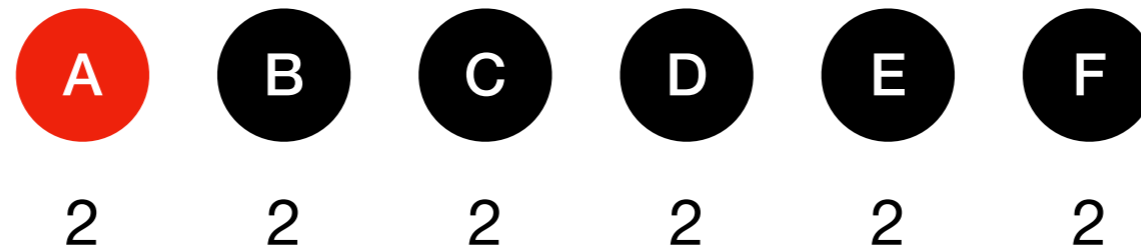
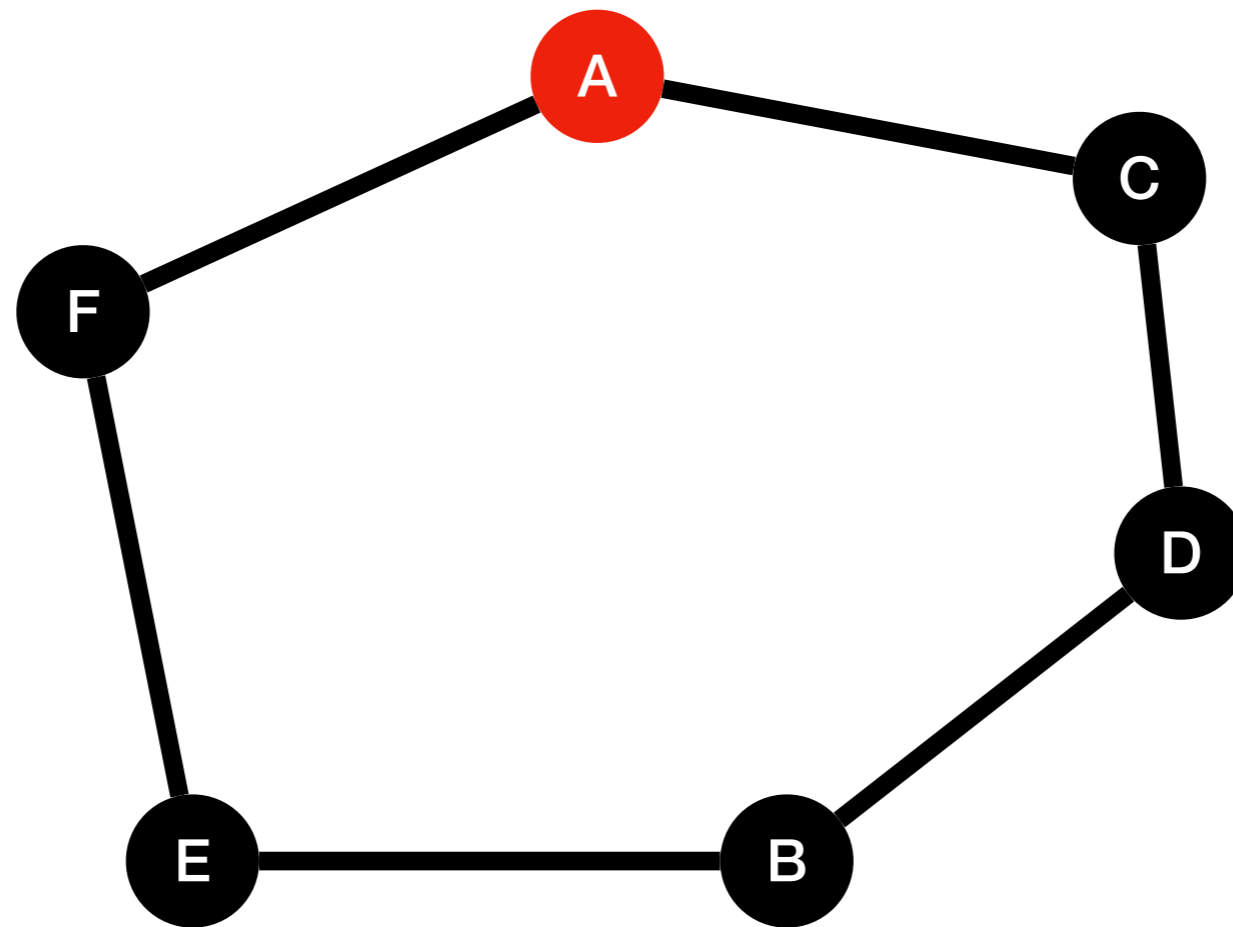
Colorations non optimales



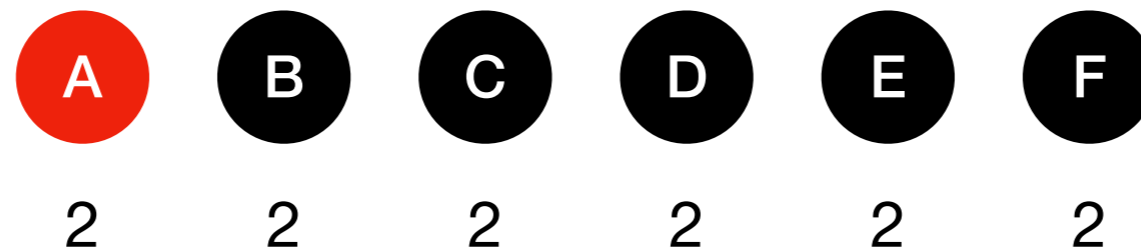
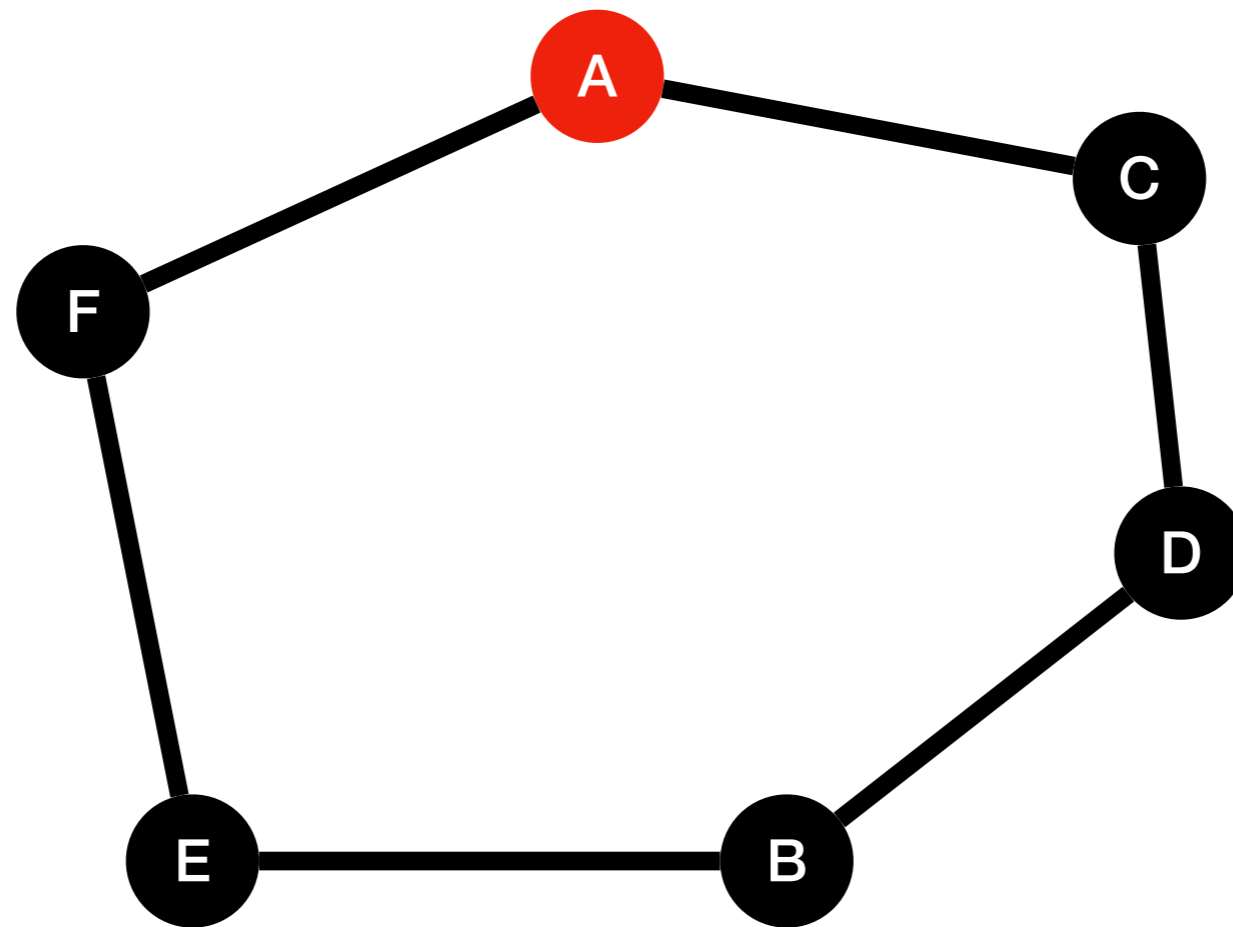
Colorations non optimales



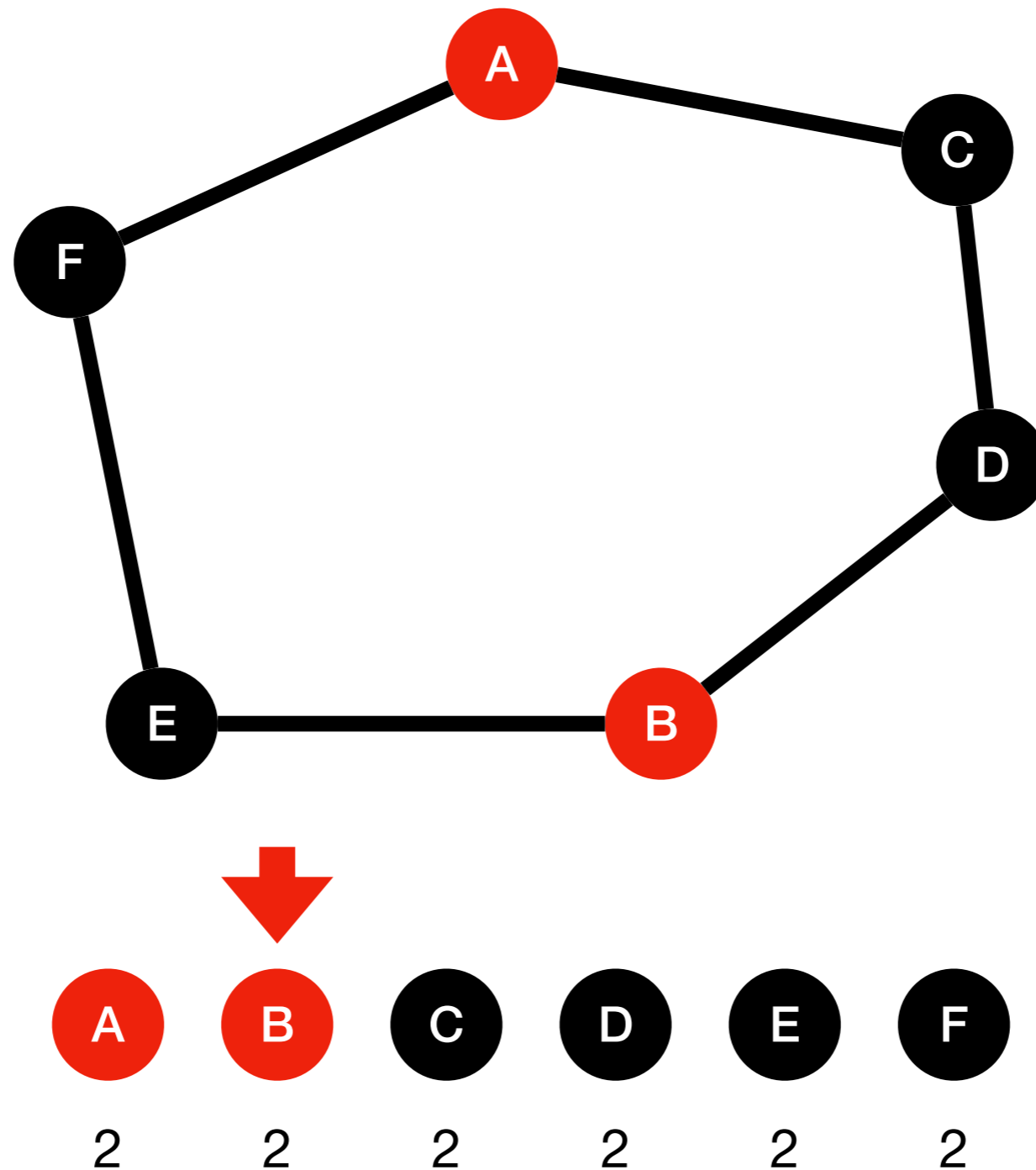
Colorations non optimales



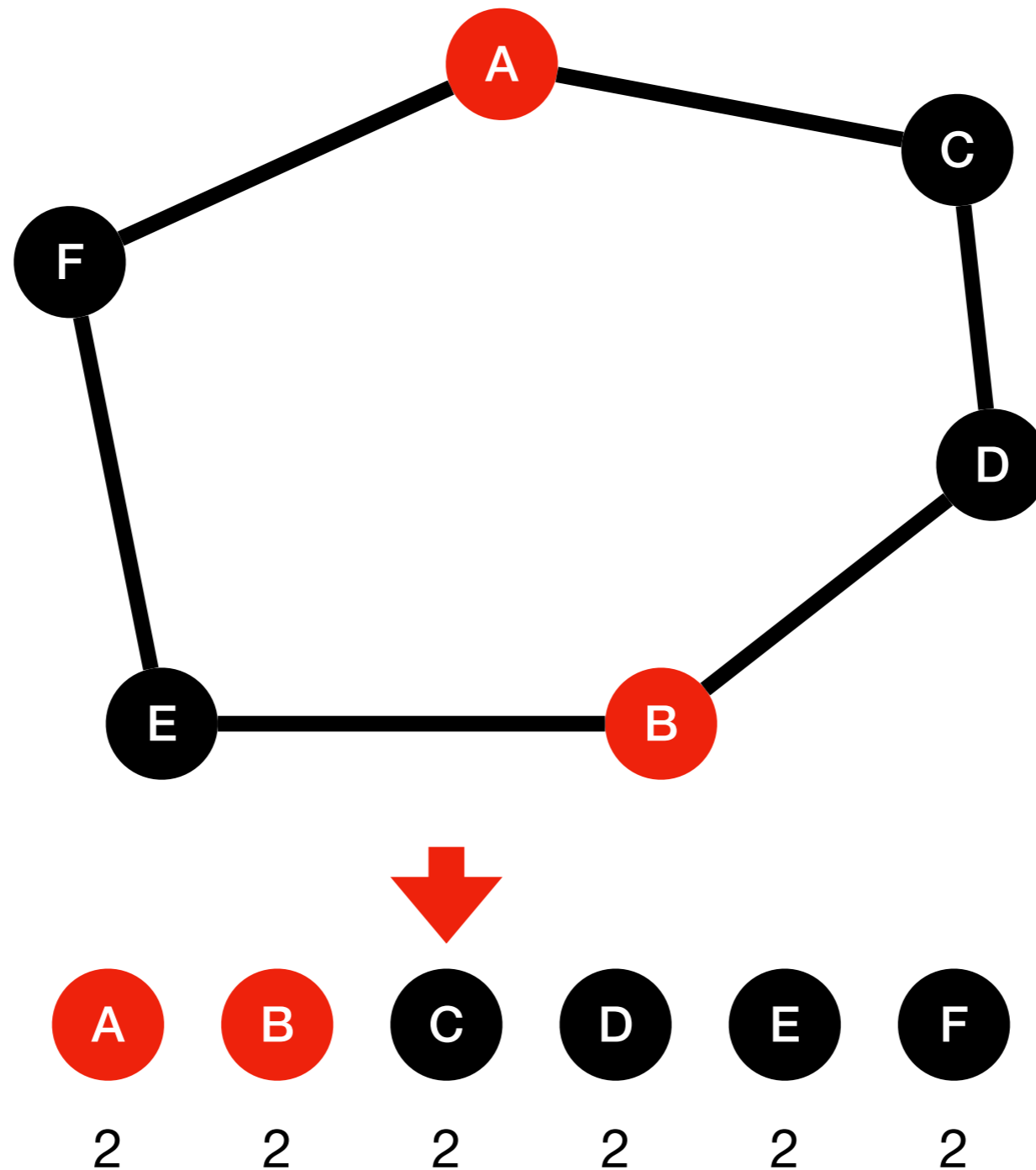
Colorations non optimales



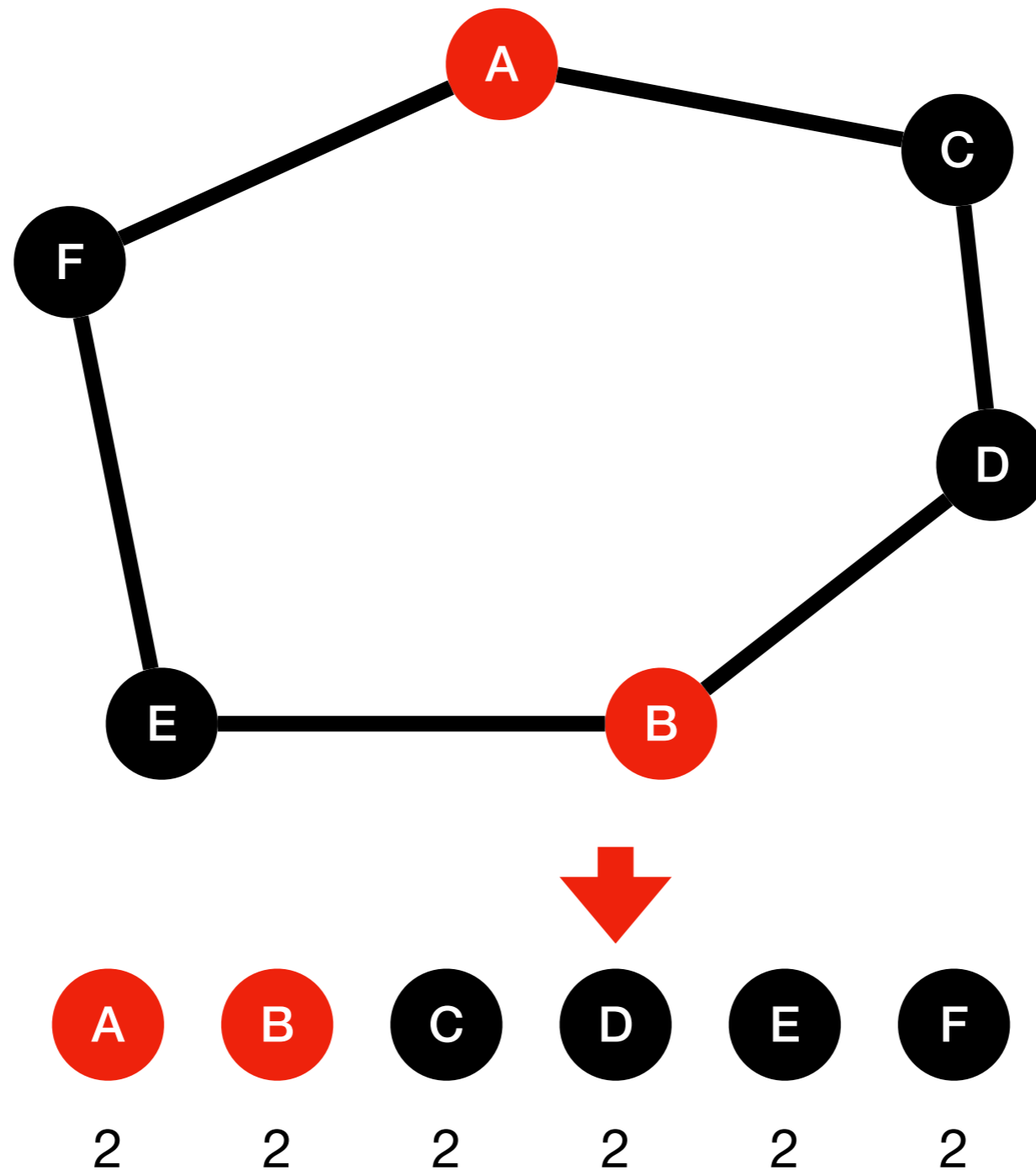
Colorations non optimales



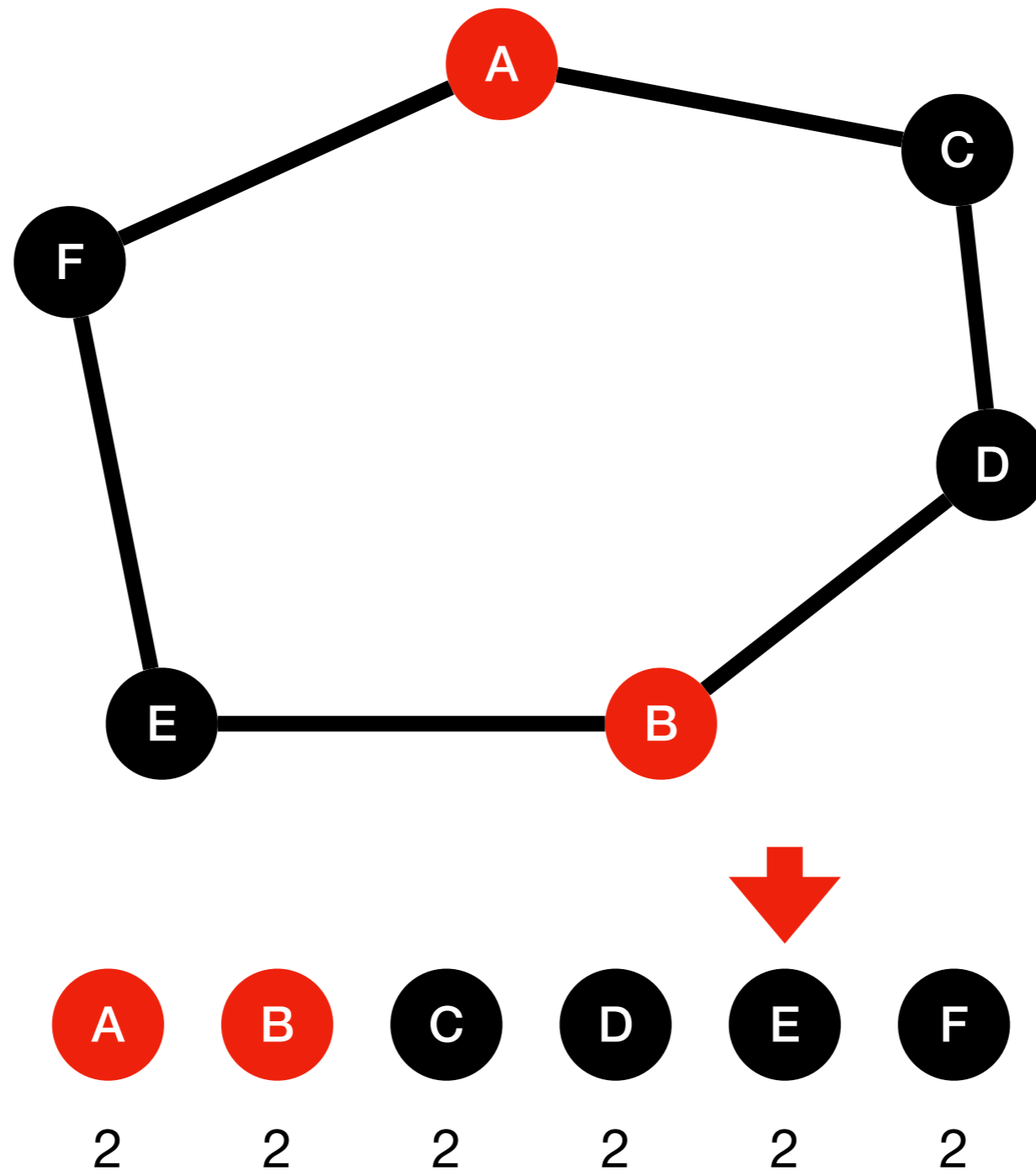
Colorations non optimales



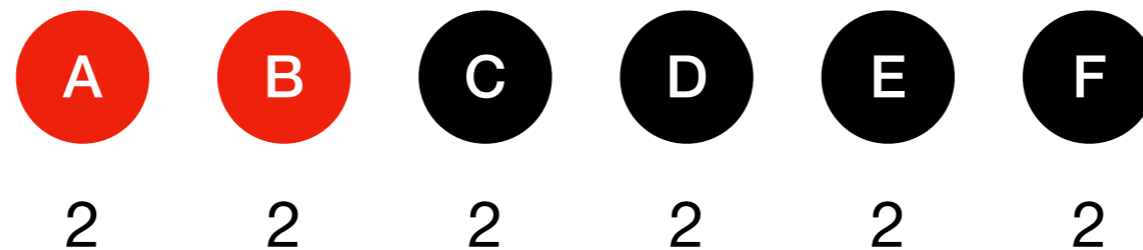
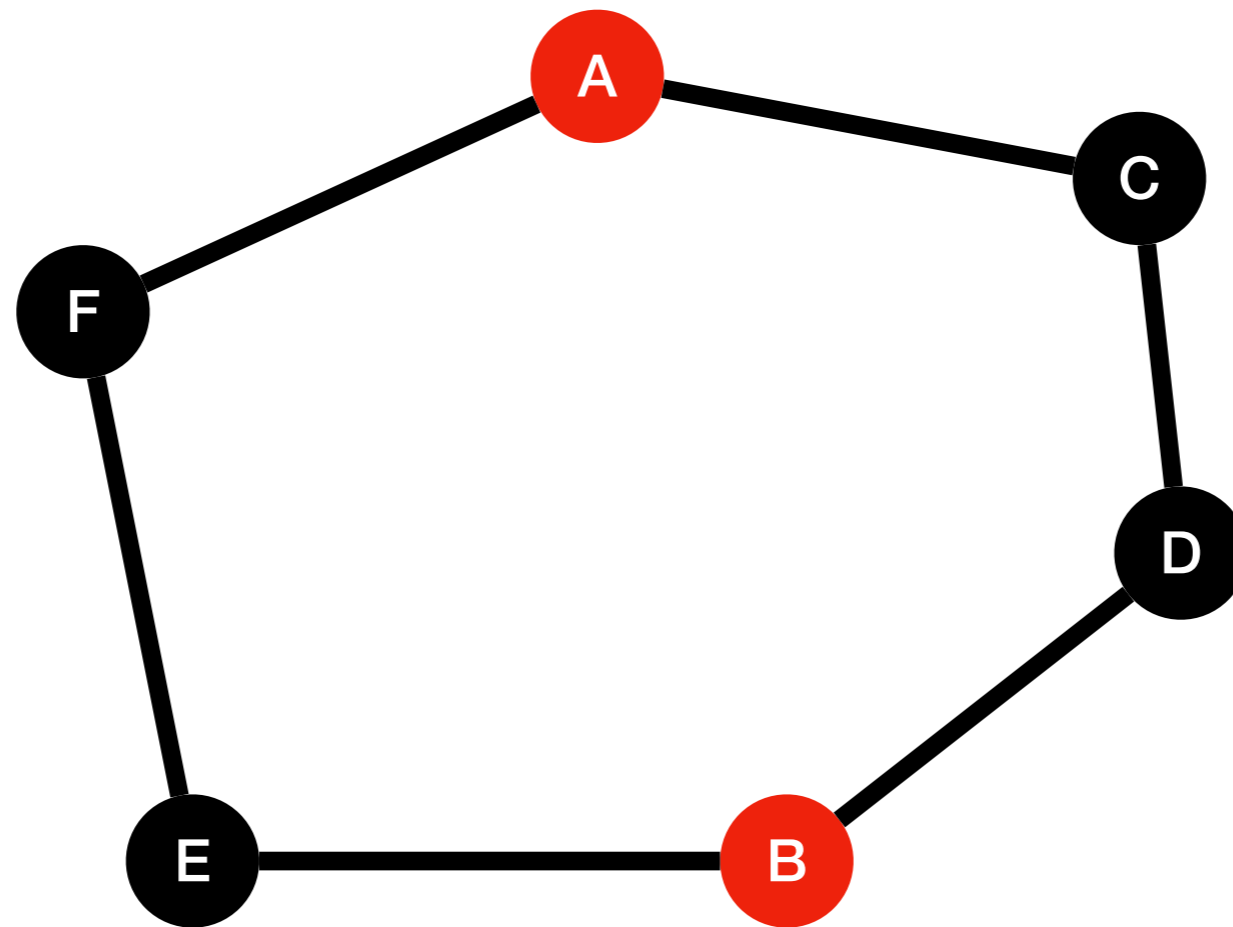
Colorations non optimales



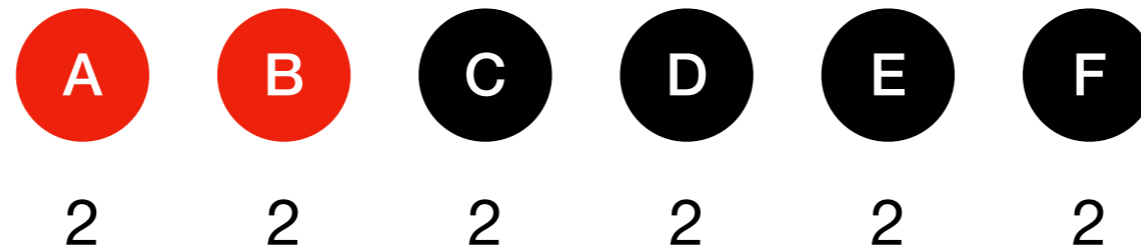
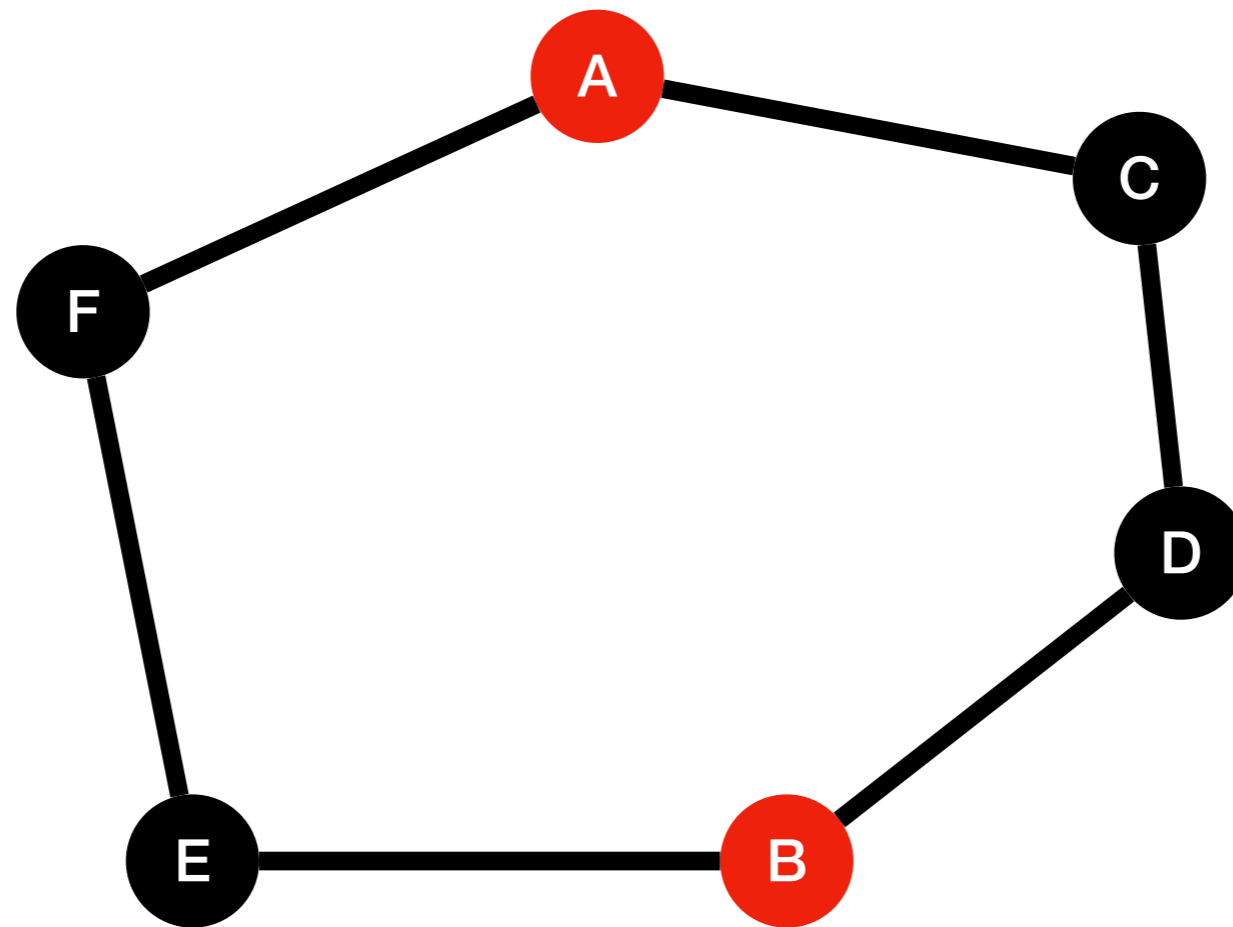
Colorations non optimales



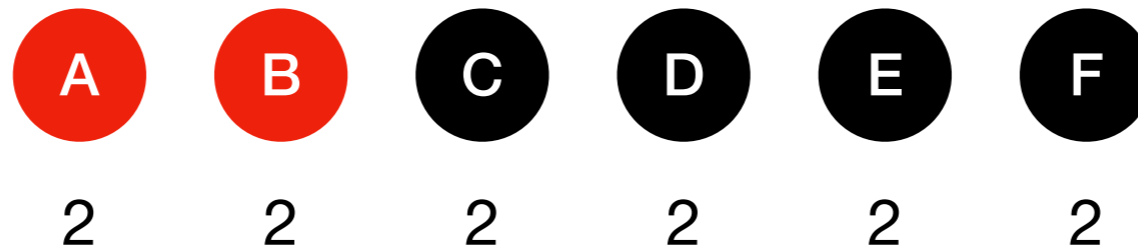
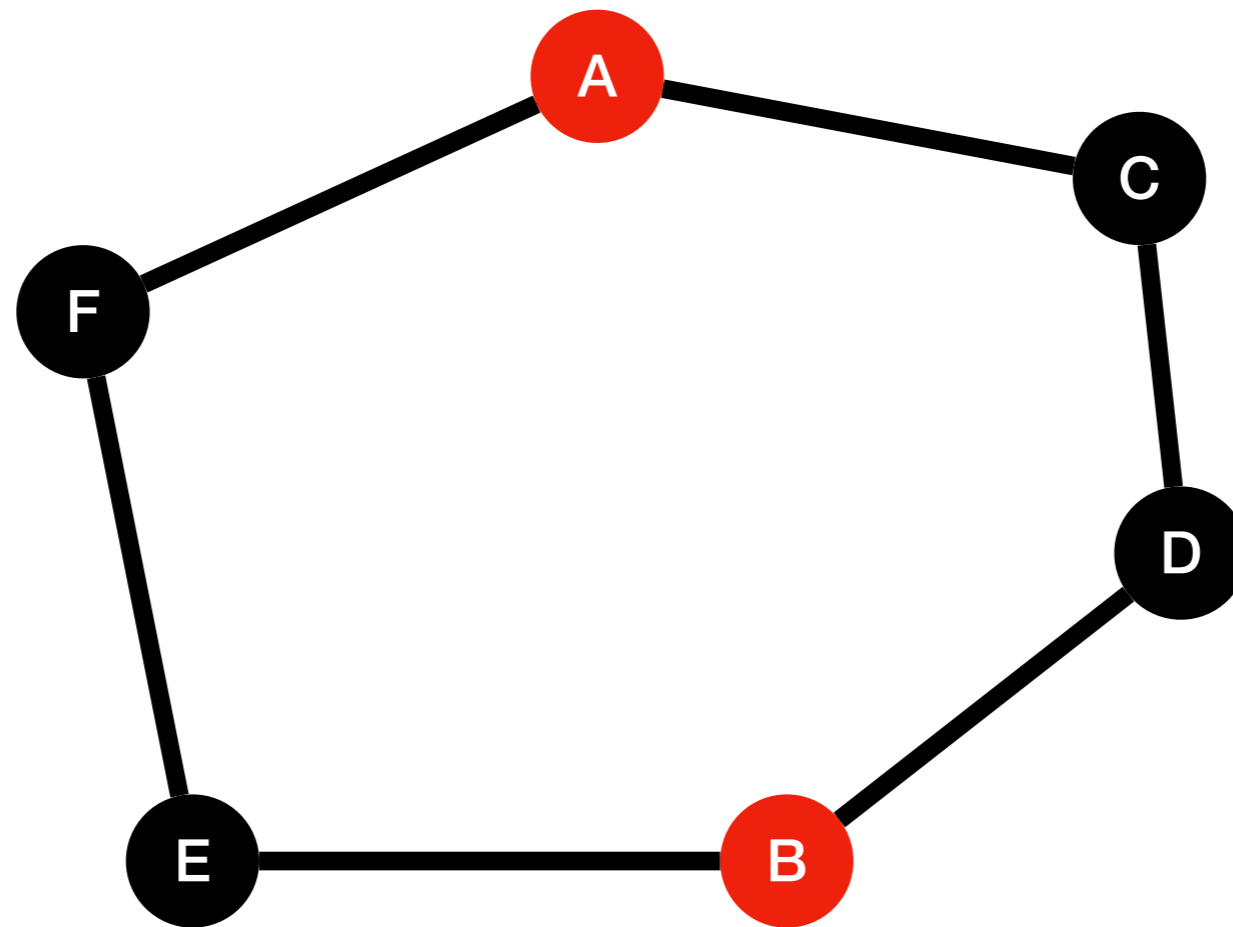
Colorations non optimales



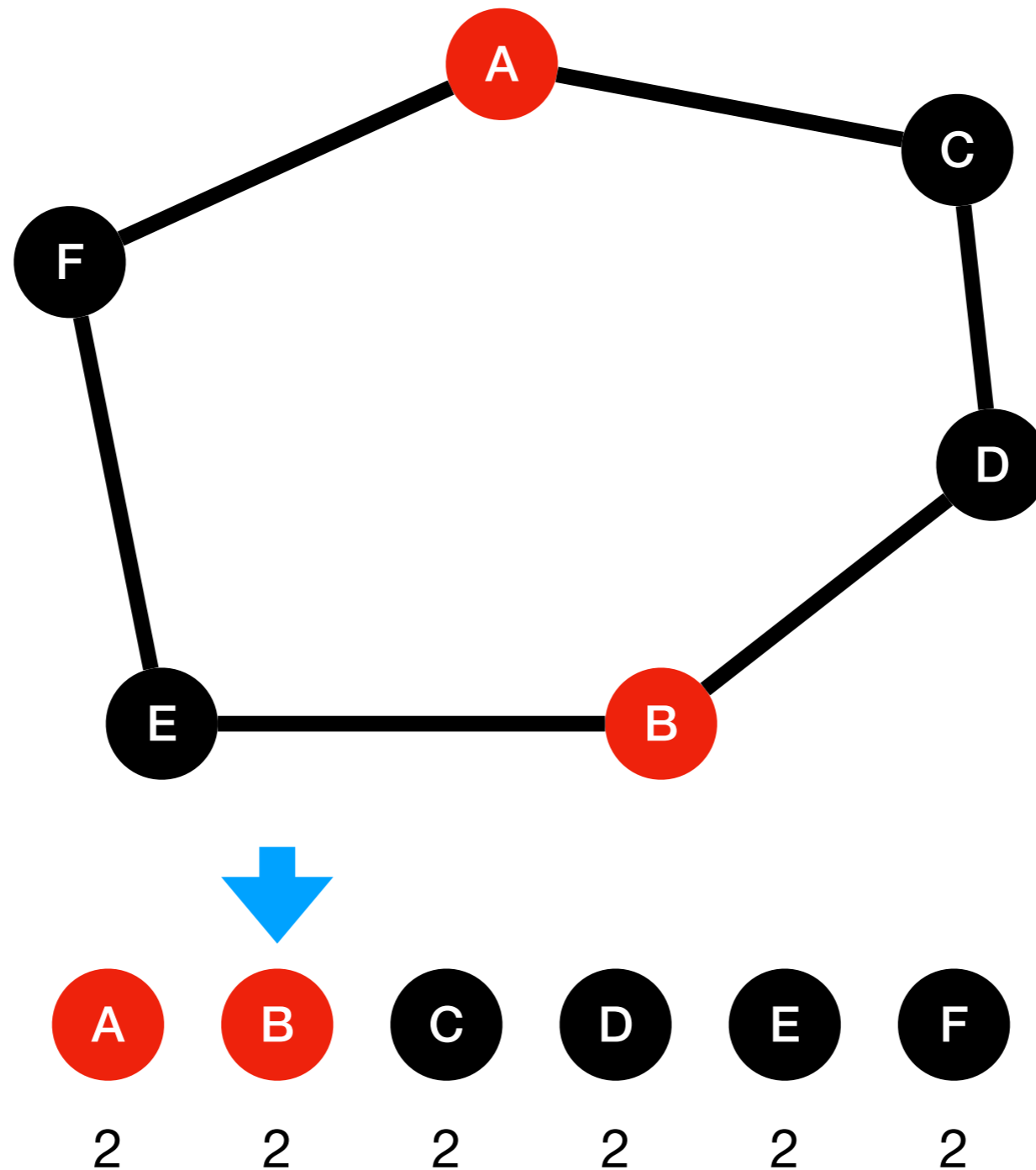
Colorations non optimales



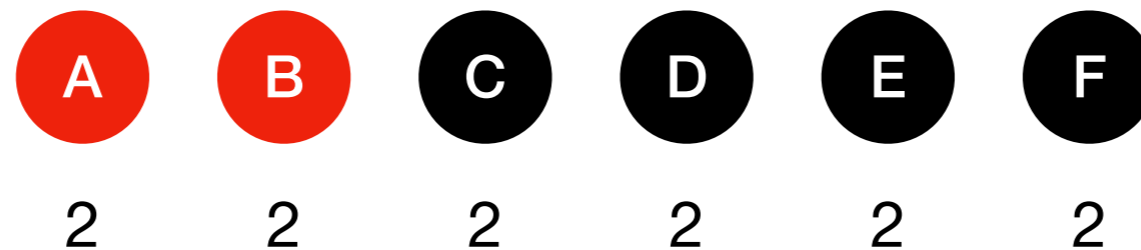
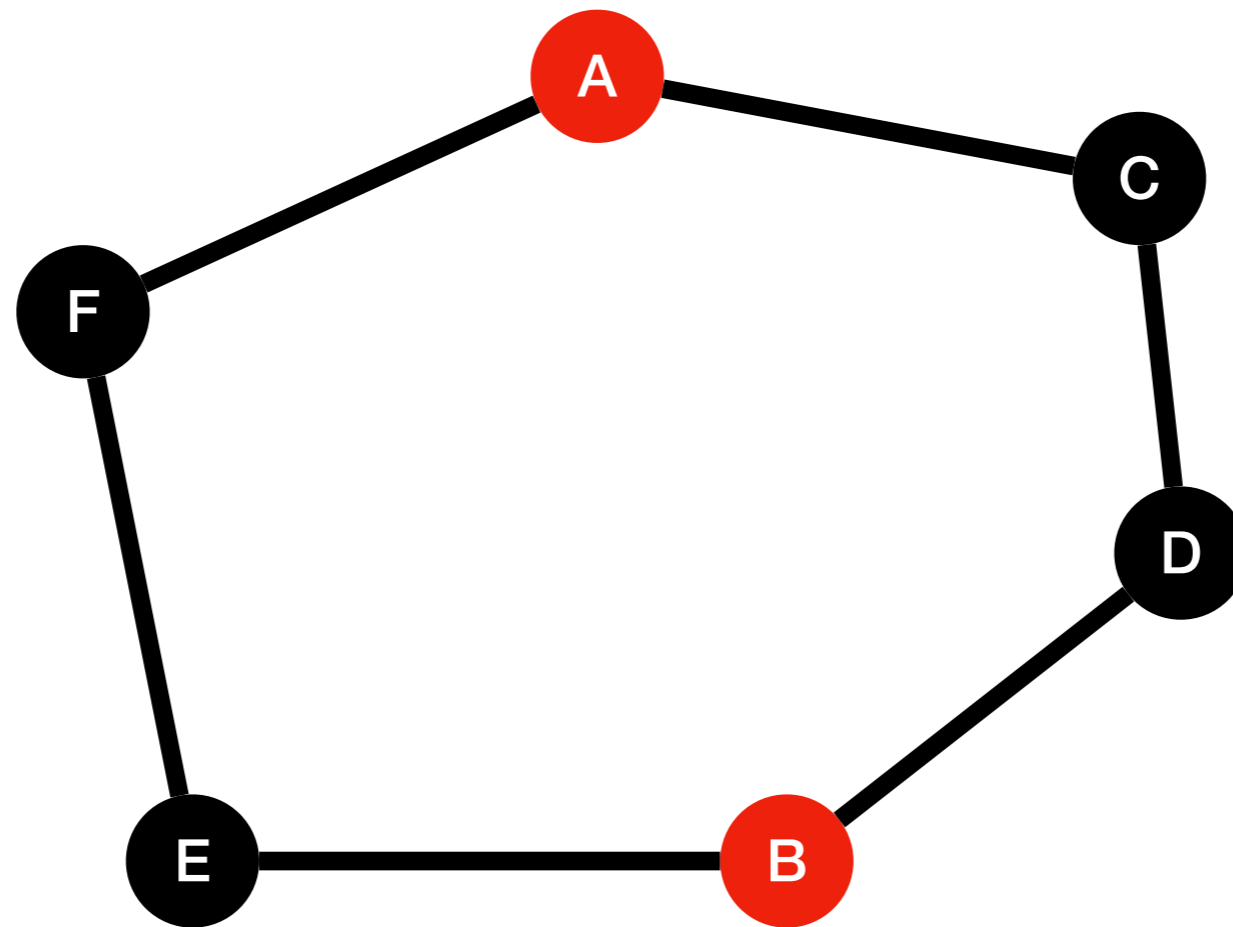
Colorations non optimales



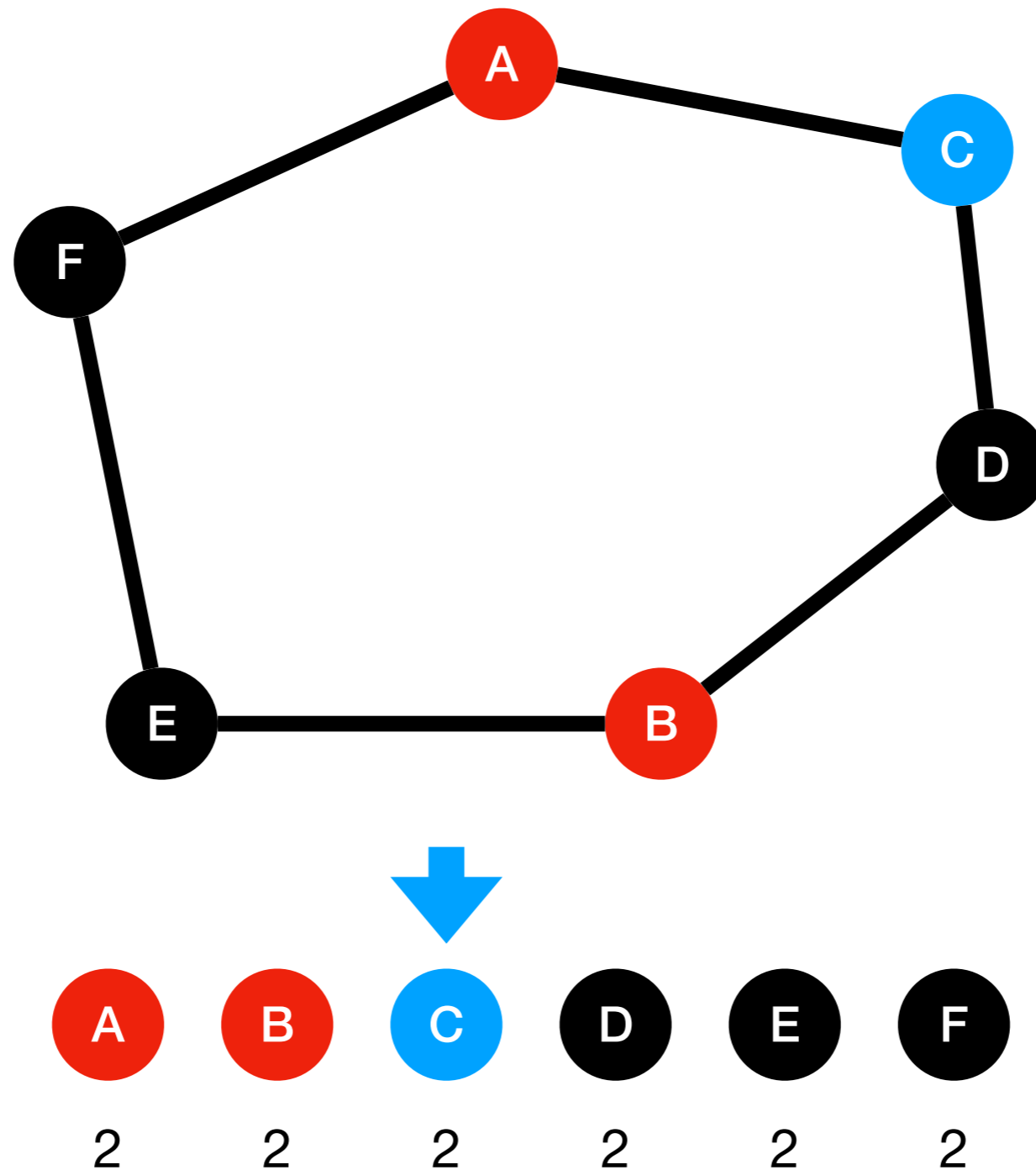
Colorations non optimales



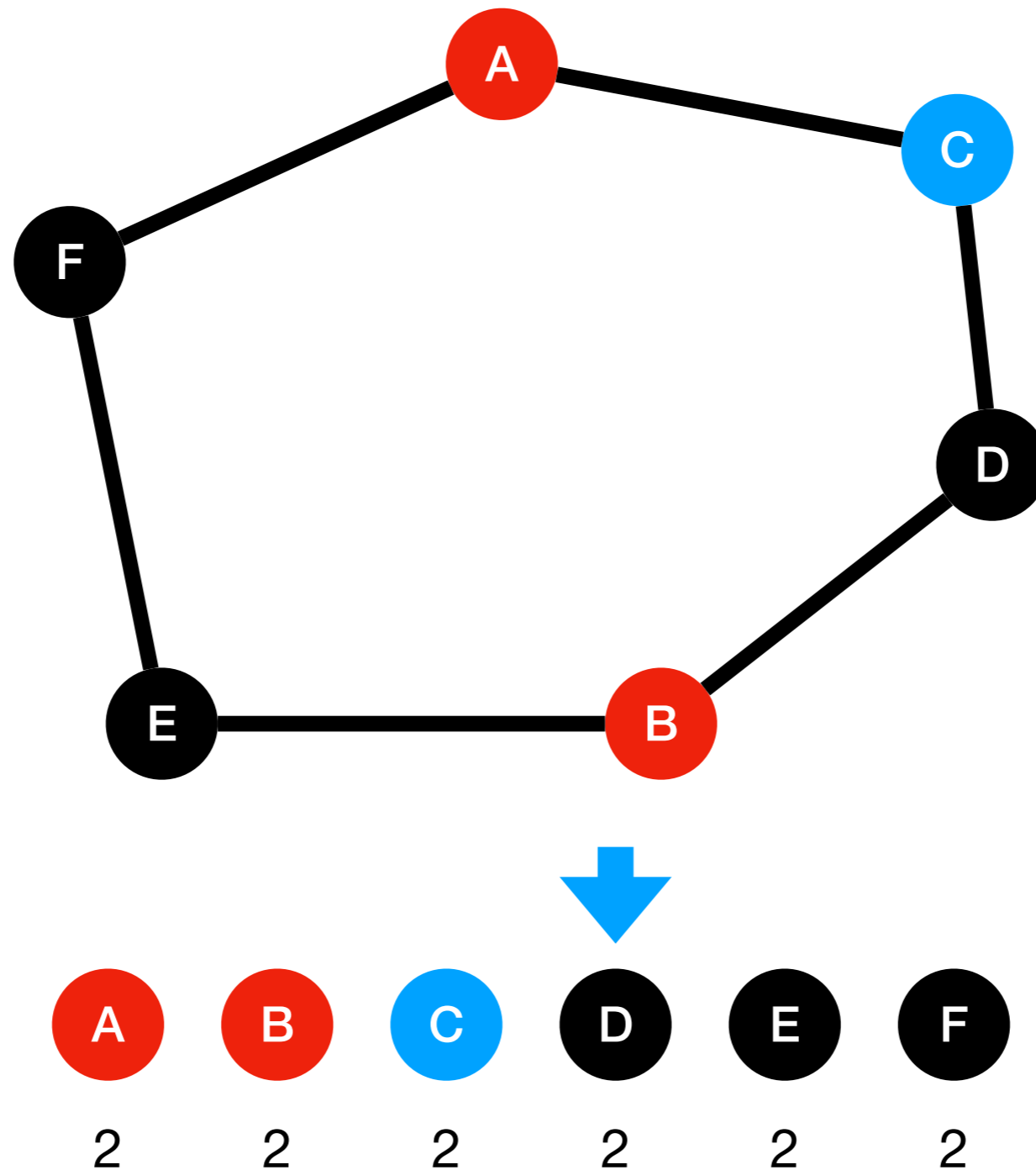
Colorations non optimales



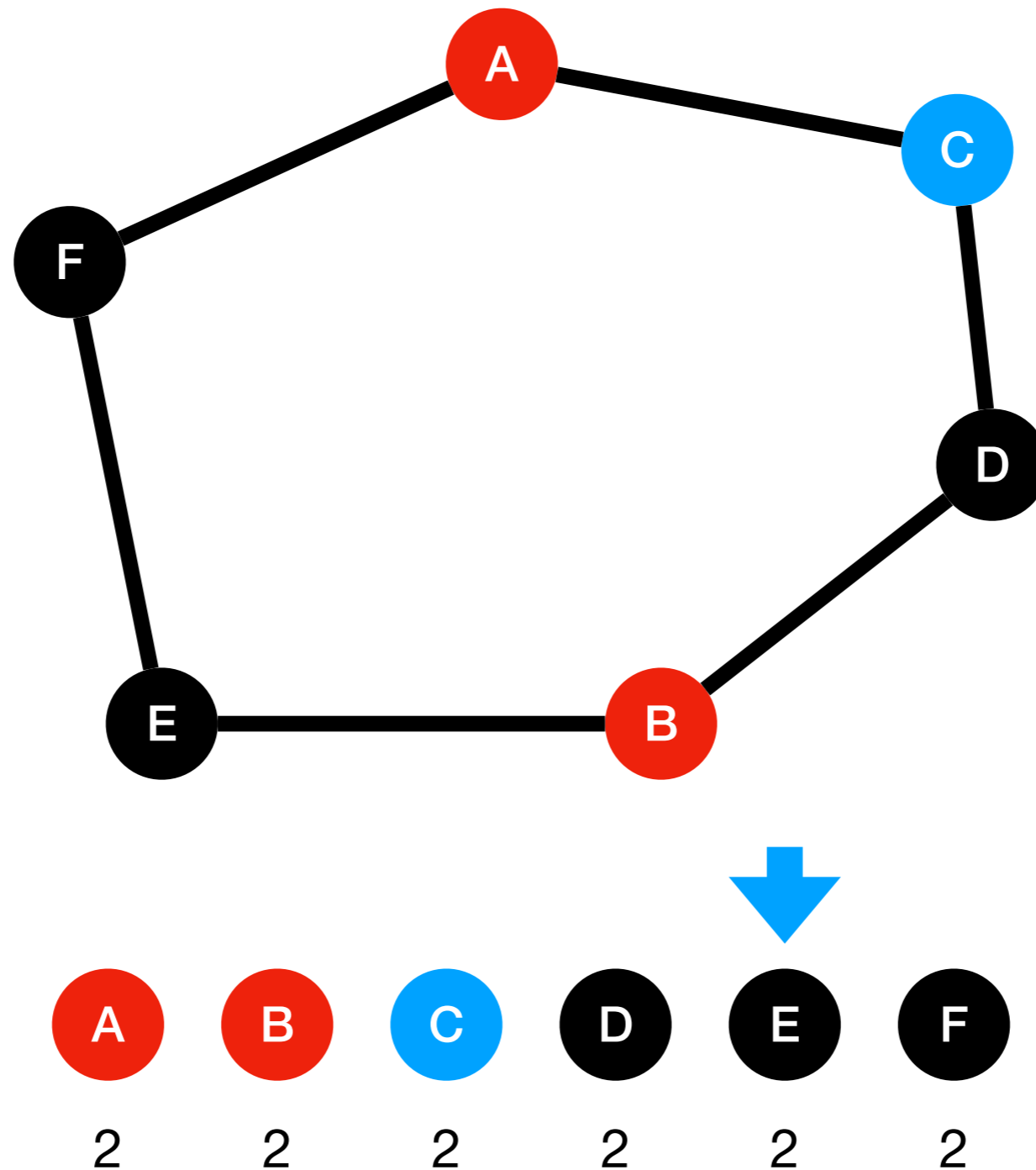
Colorations non optimales



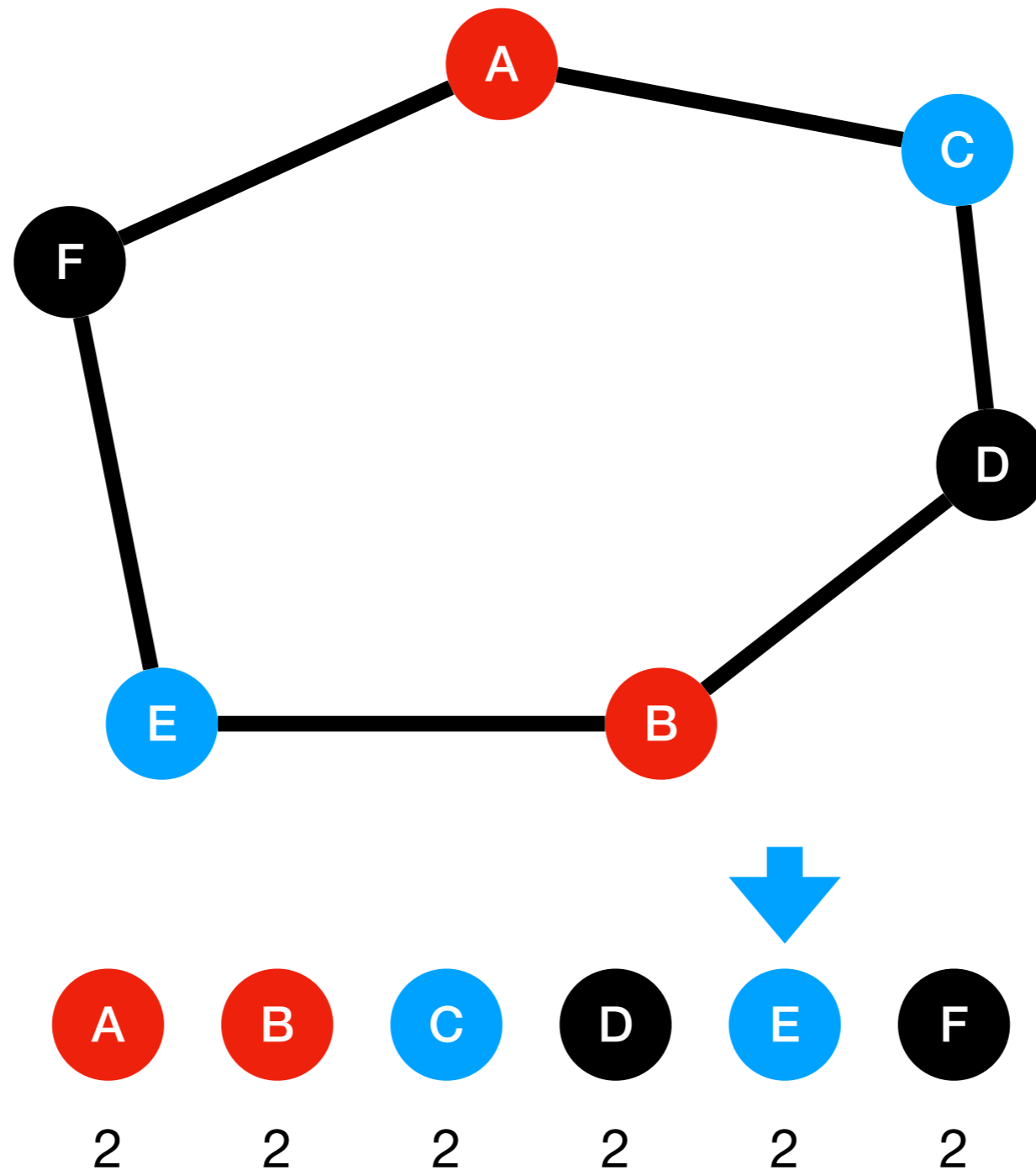
Colorations non optimales



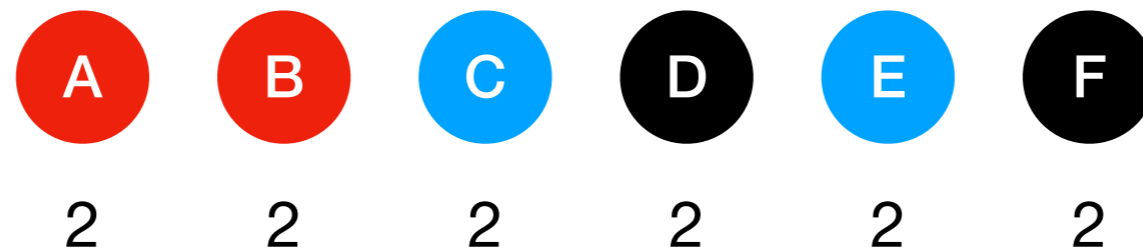
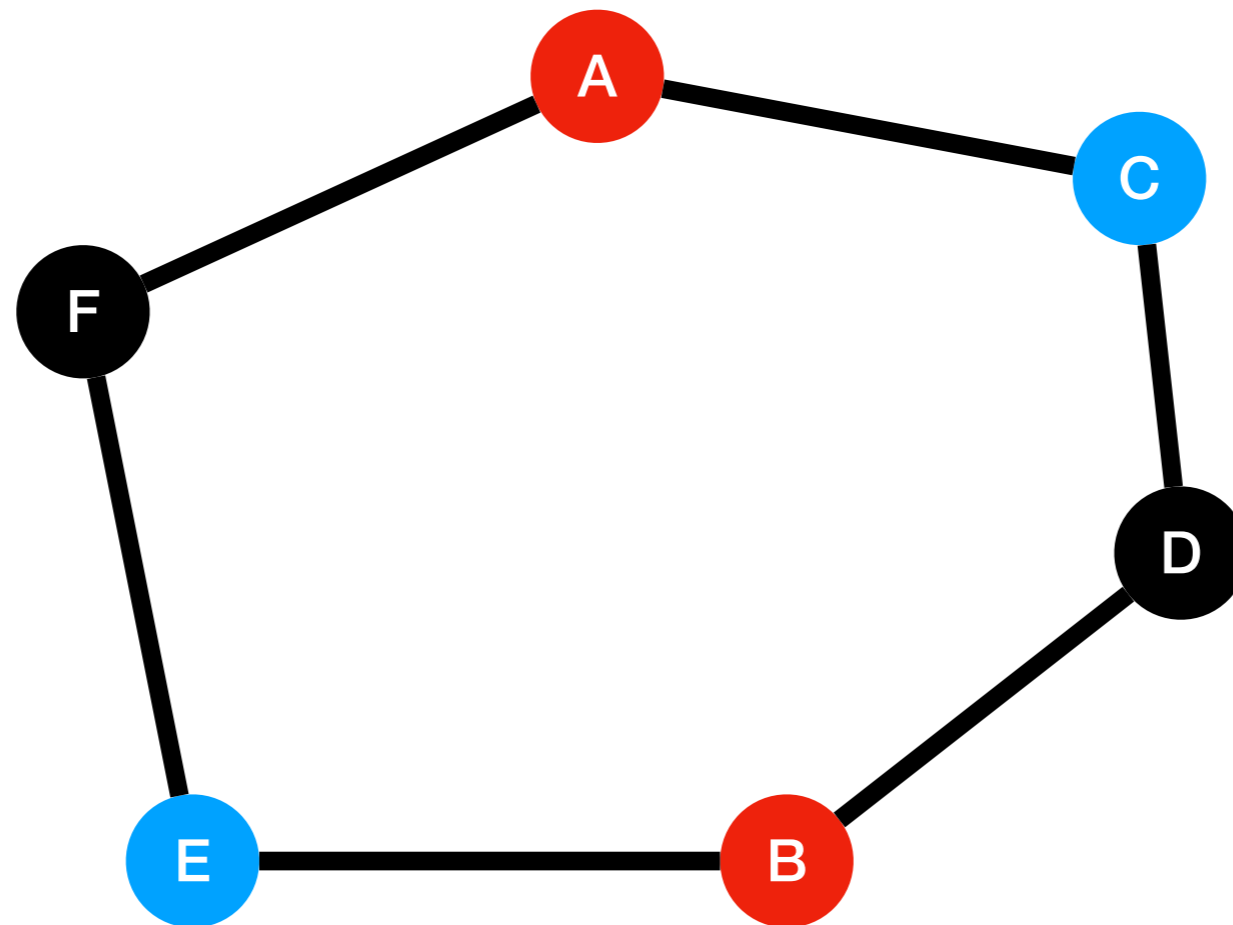
Colorations non optimales



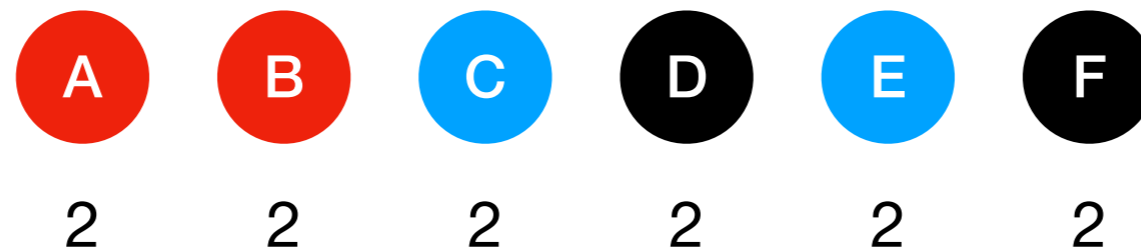
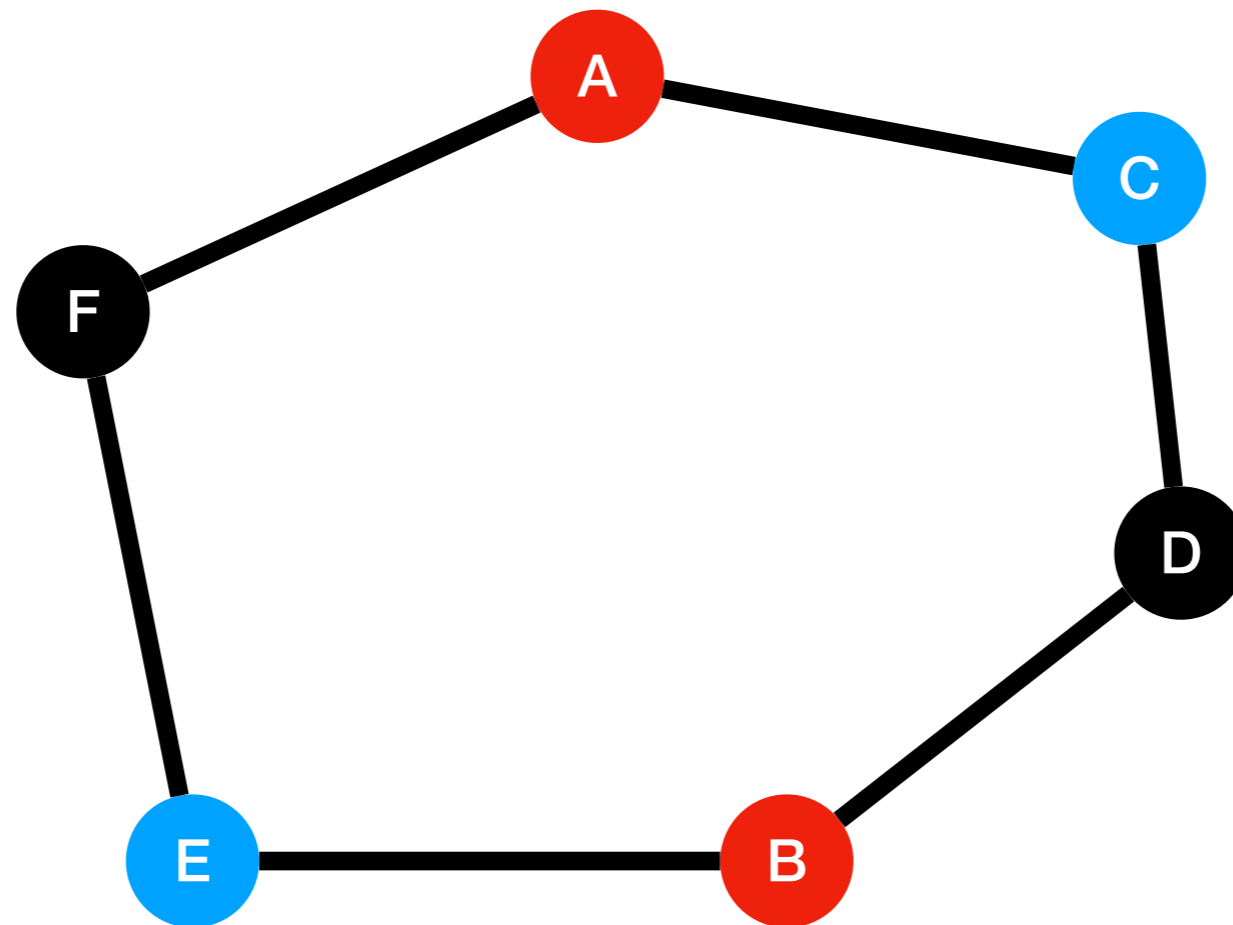
Colorations non optimales



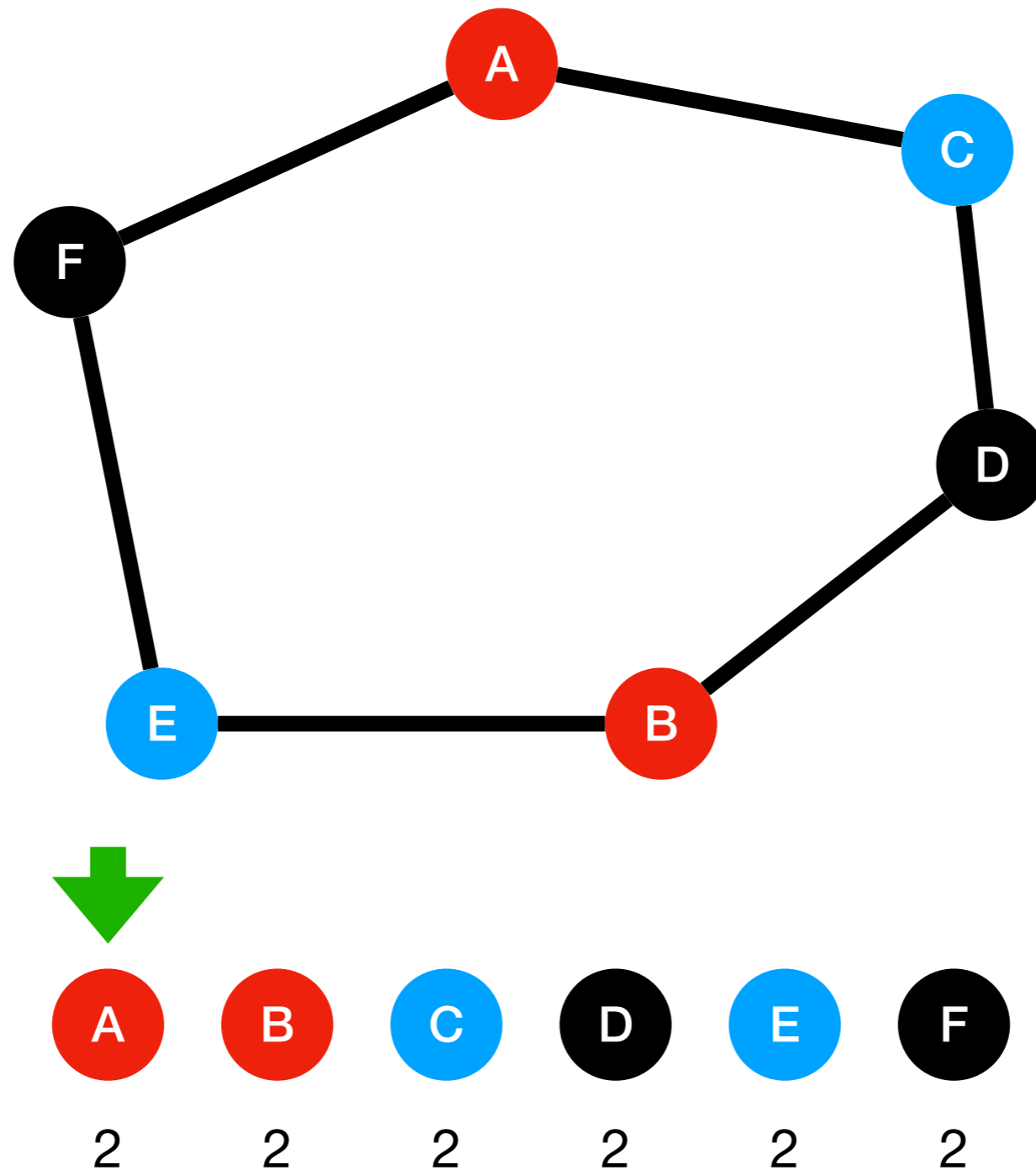
Colorations non optimales



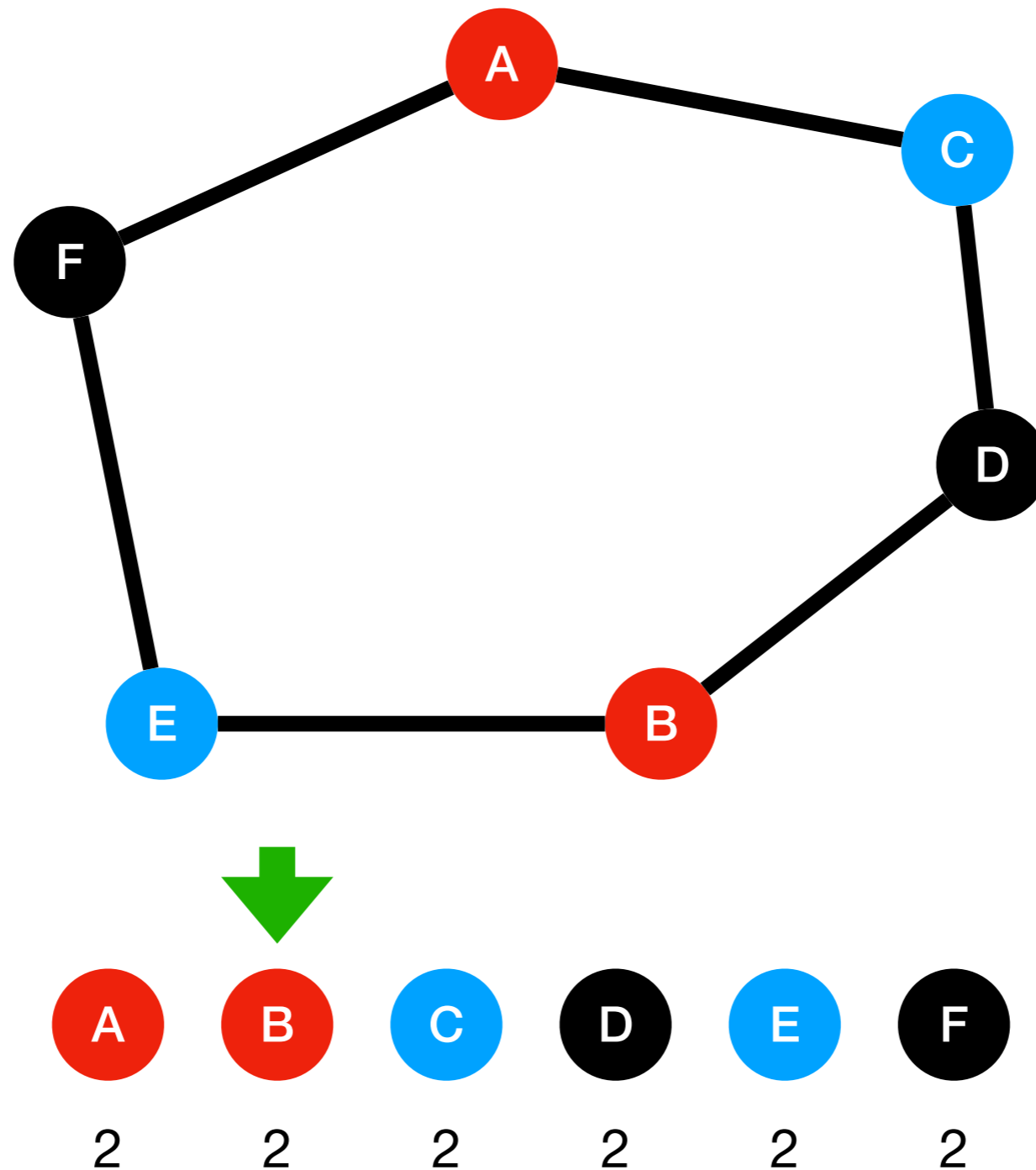
Colorations non optimales



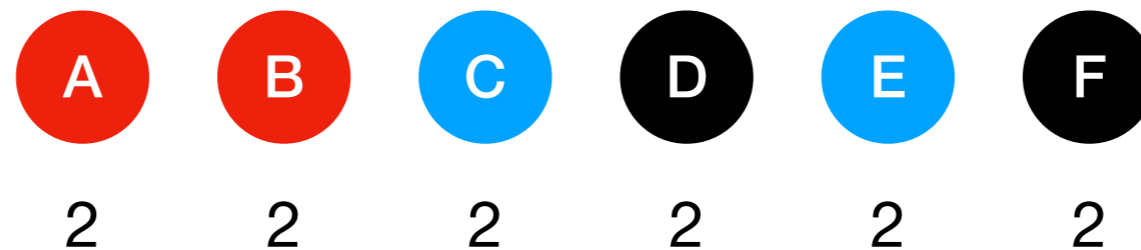
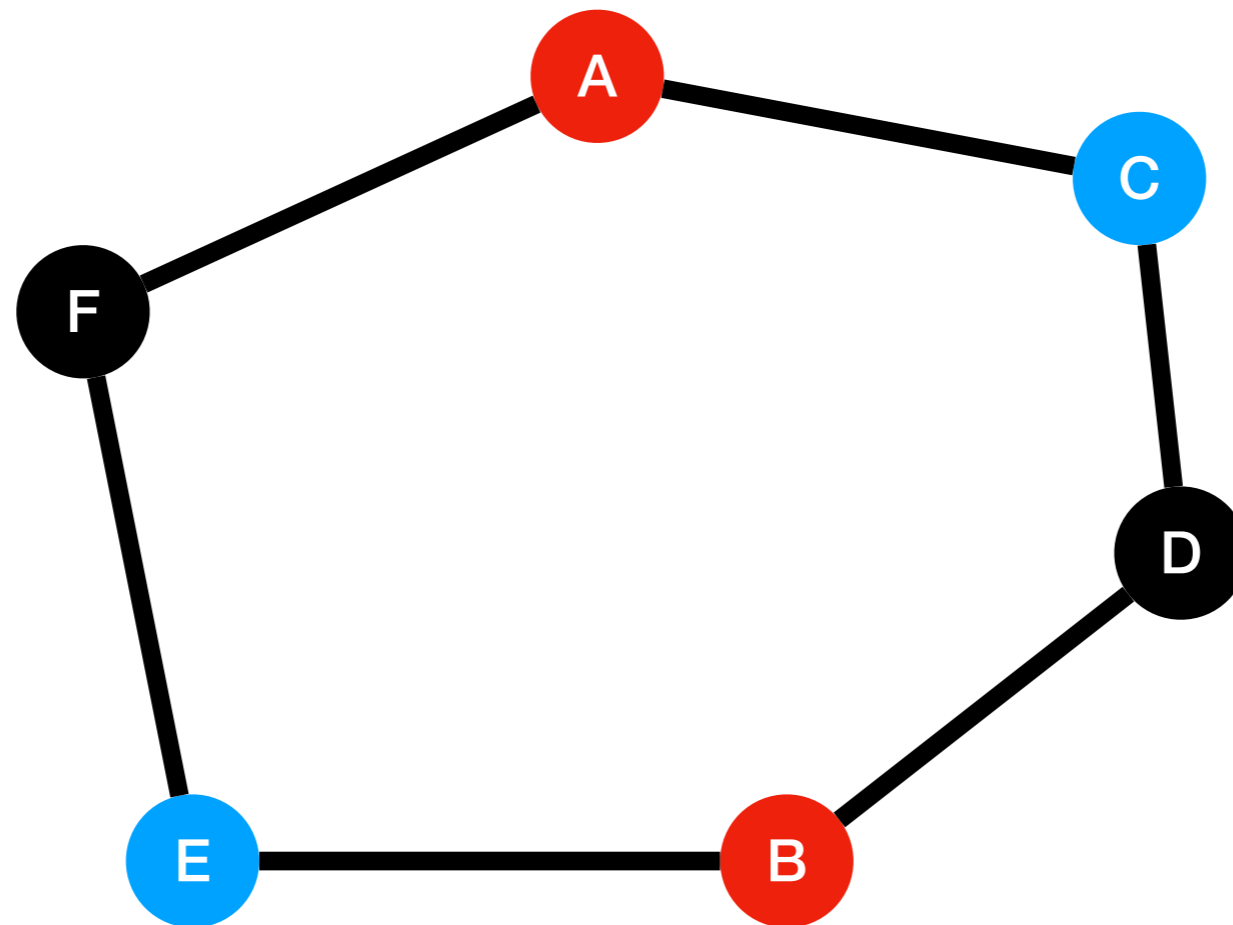
Colorations non optimales



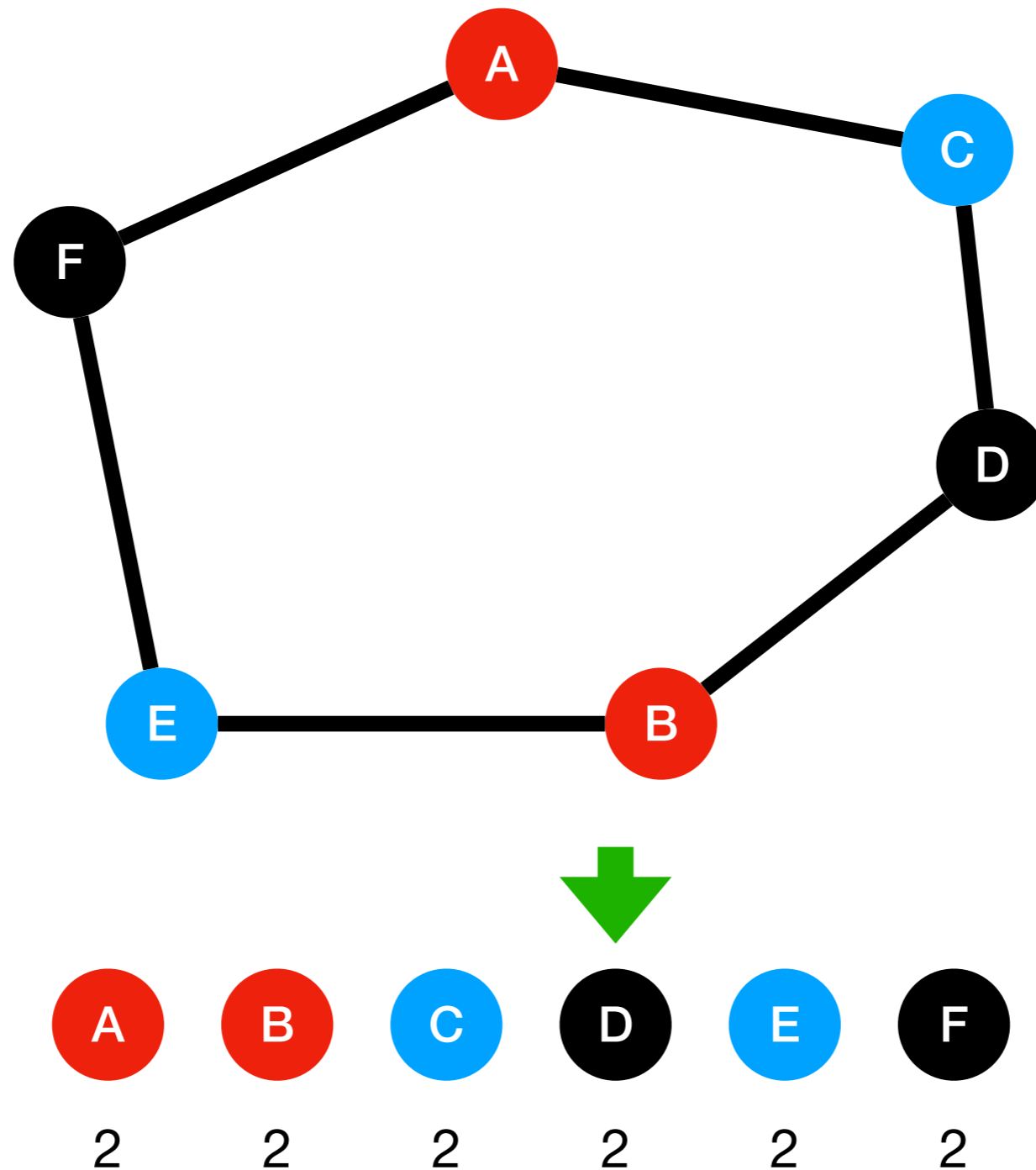
Colorations non optimales



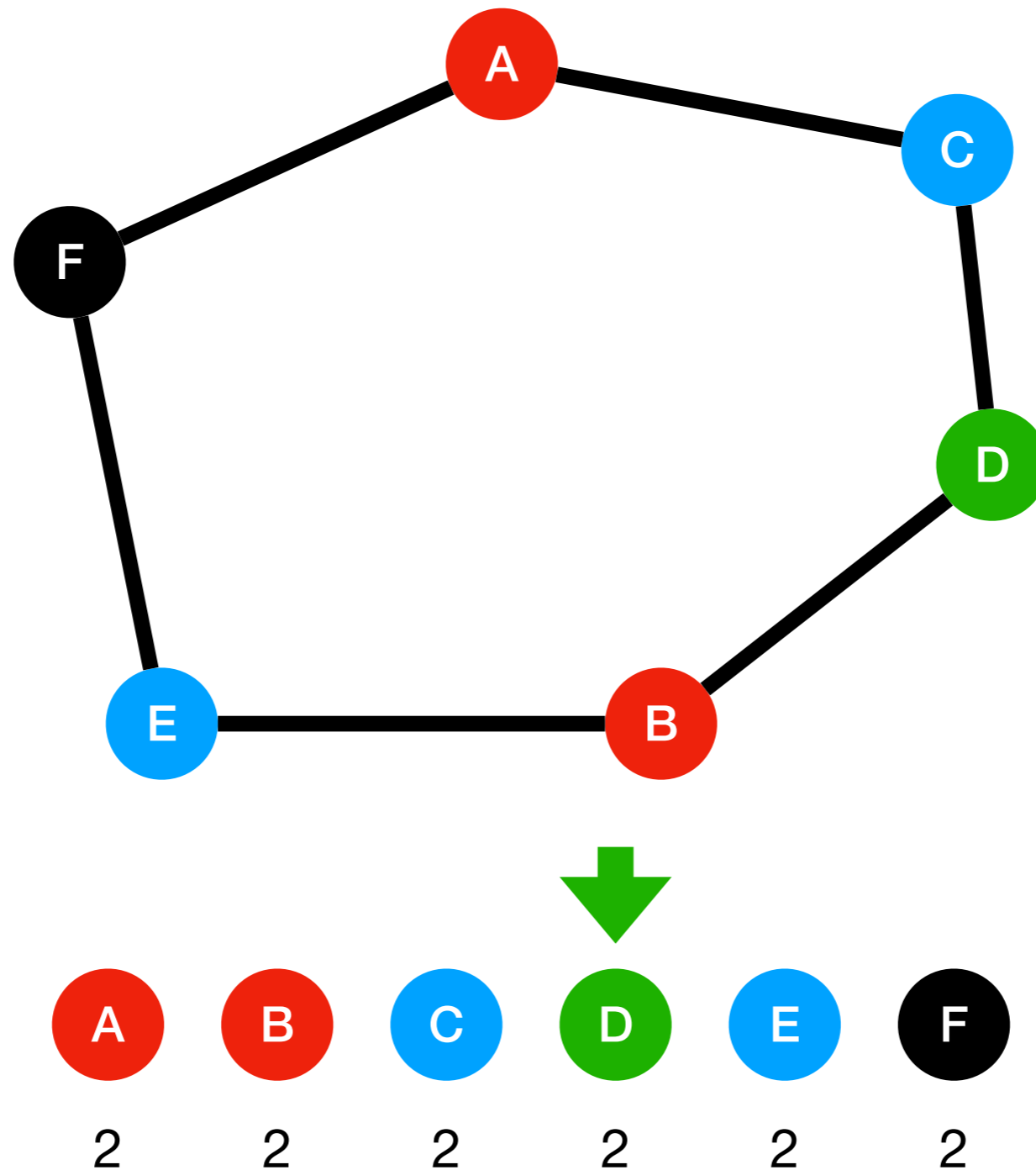
Colorations non optimales



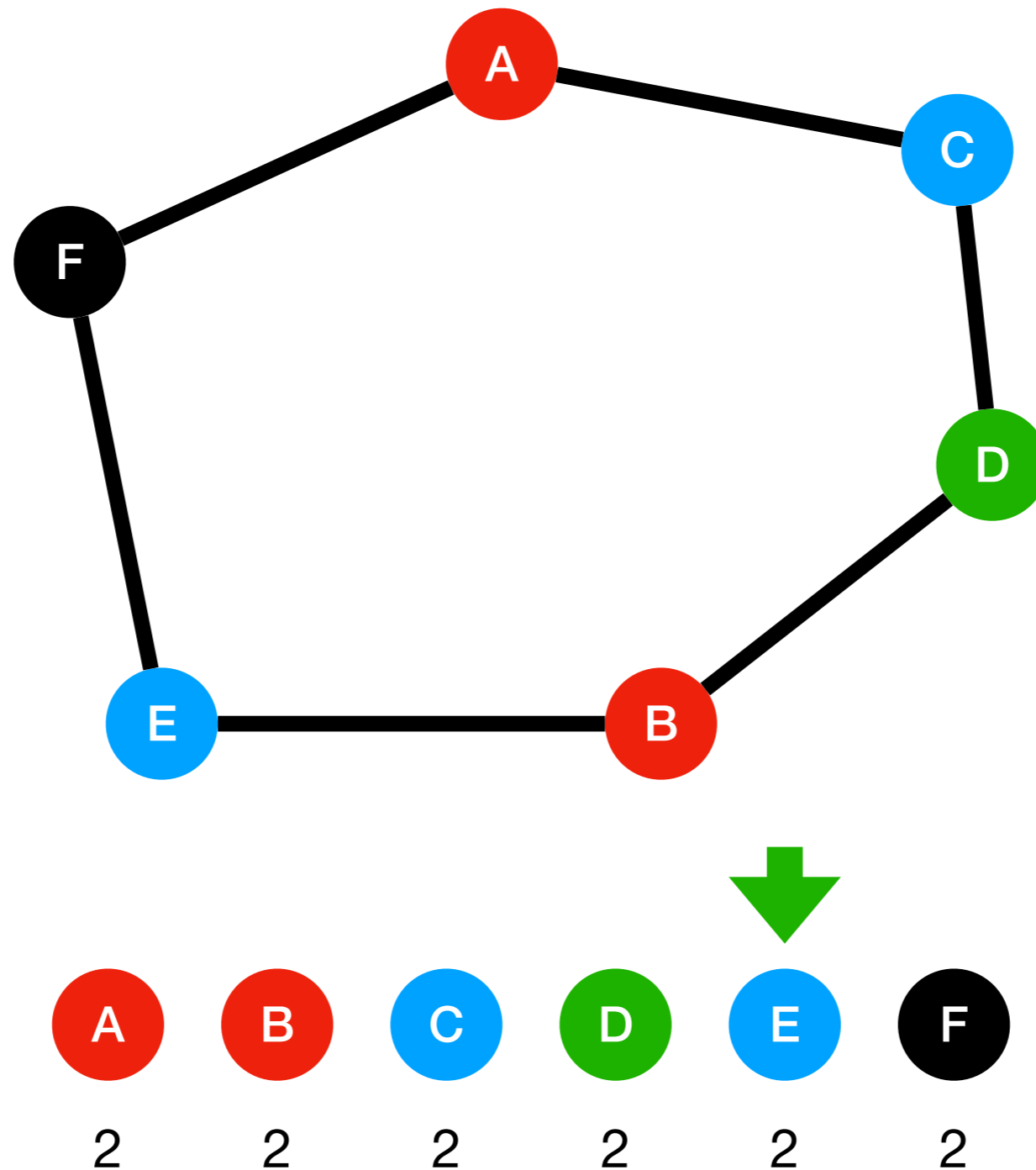
Colorations non optimales



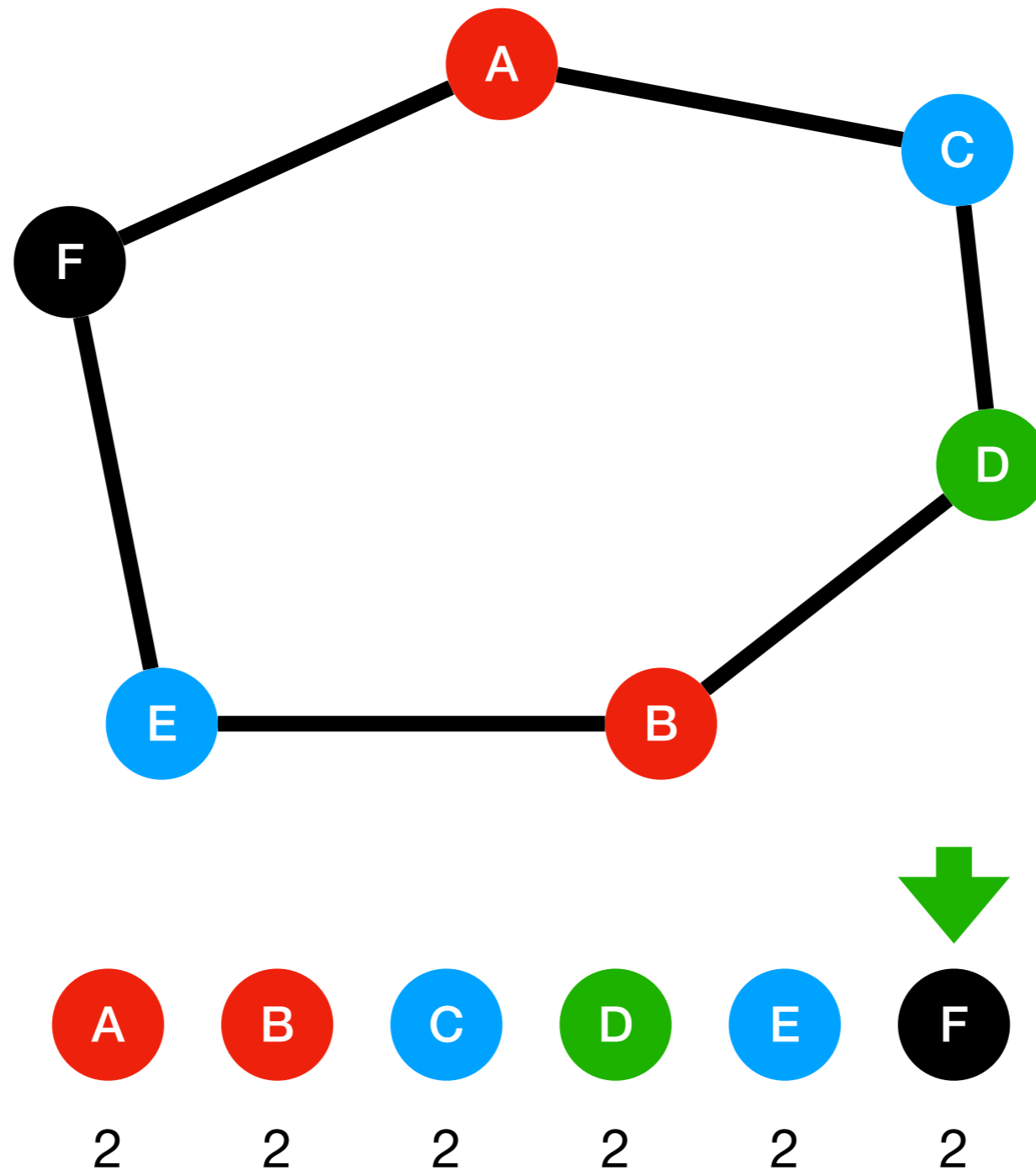
Colorations non optimales



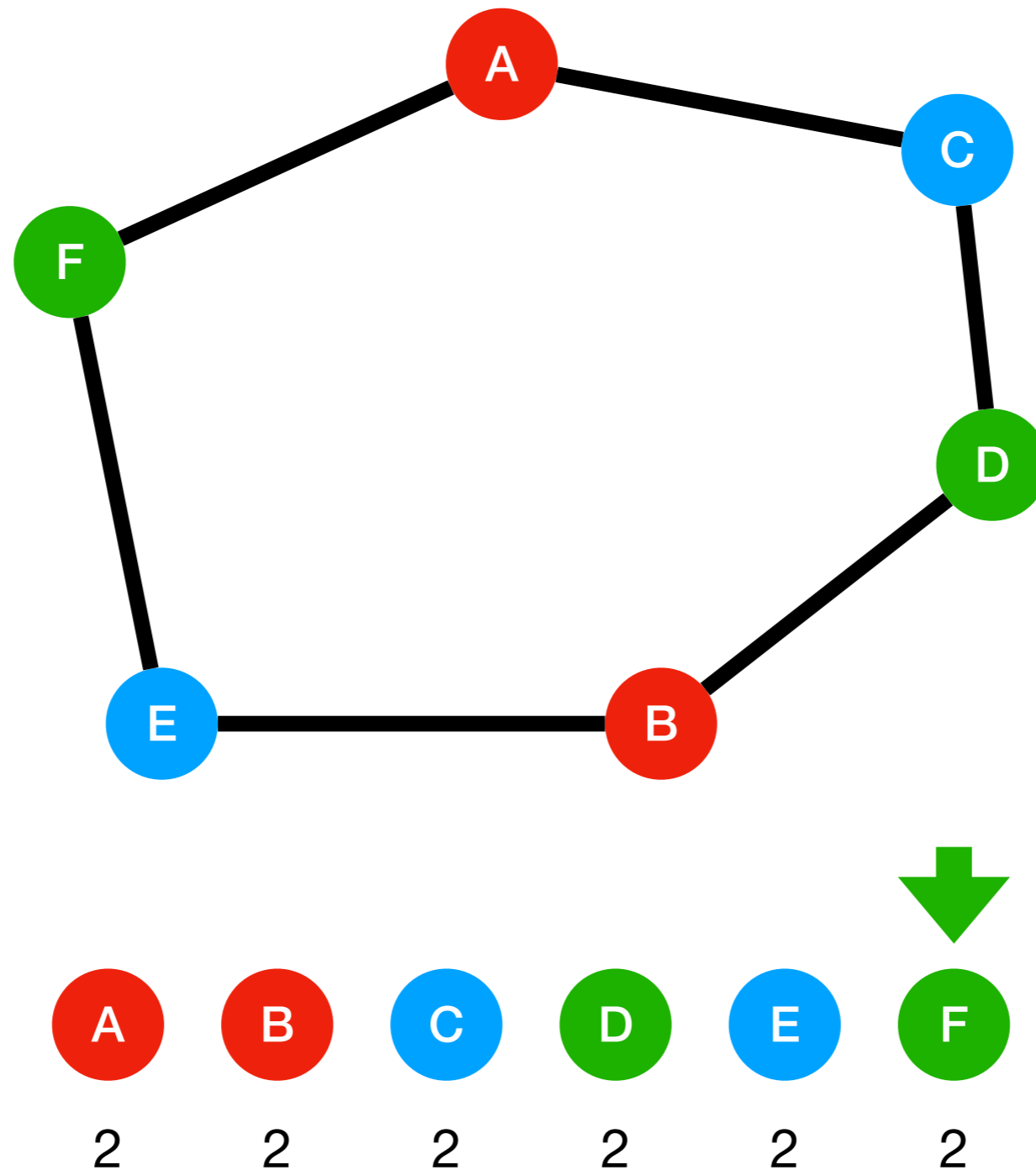
Colorations non optimales



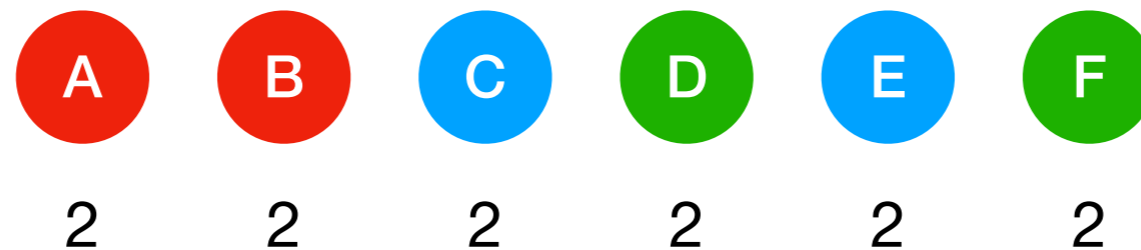
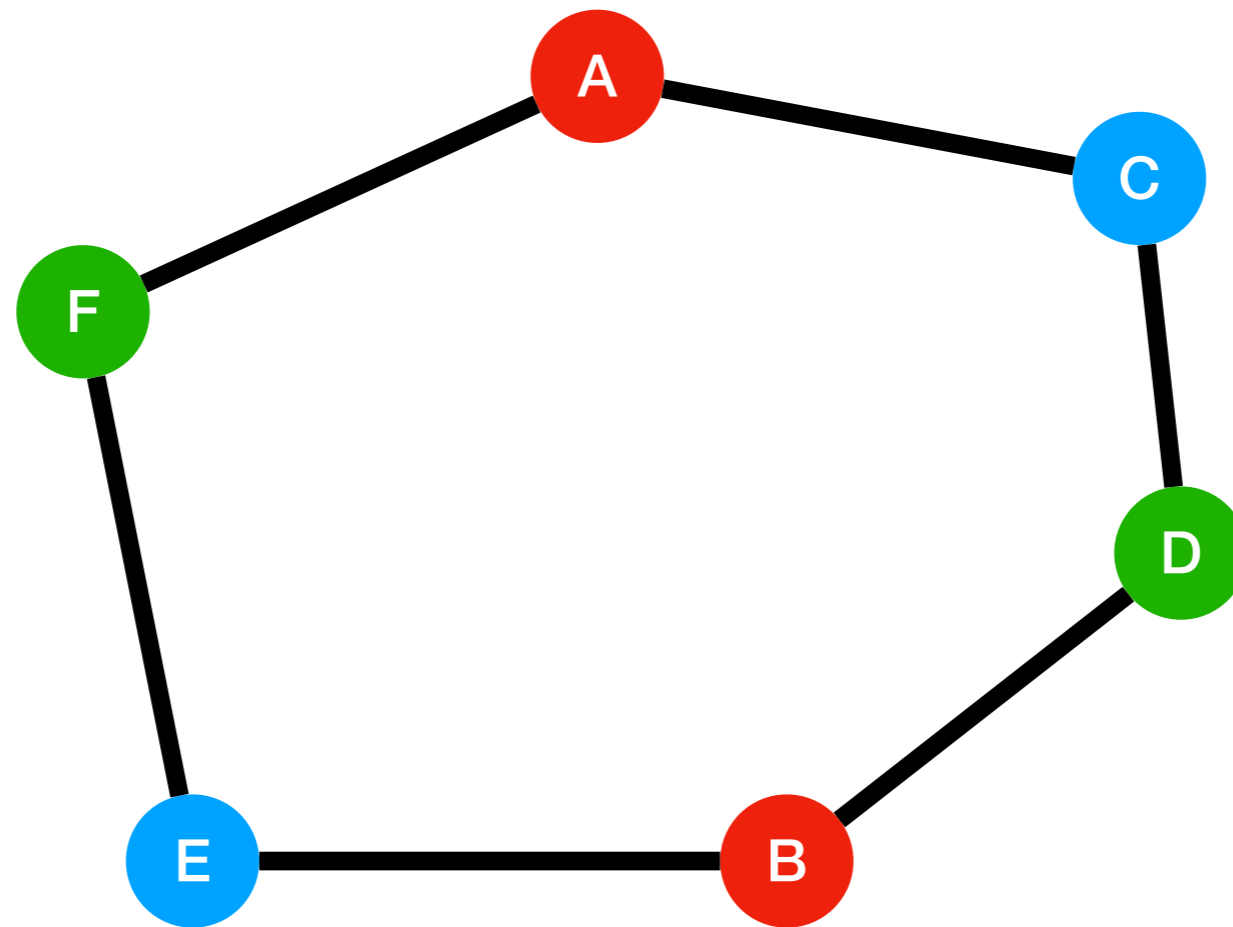
Colorations non optimales



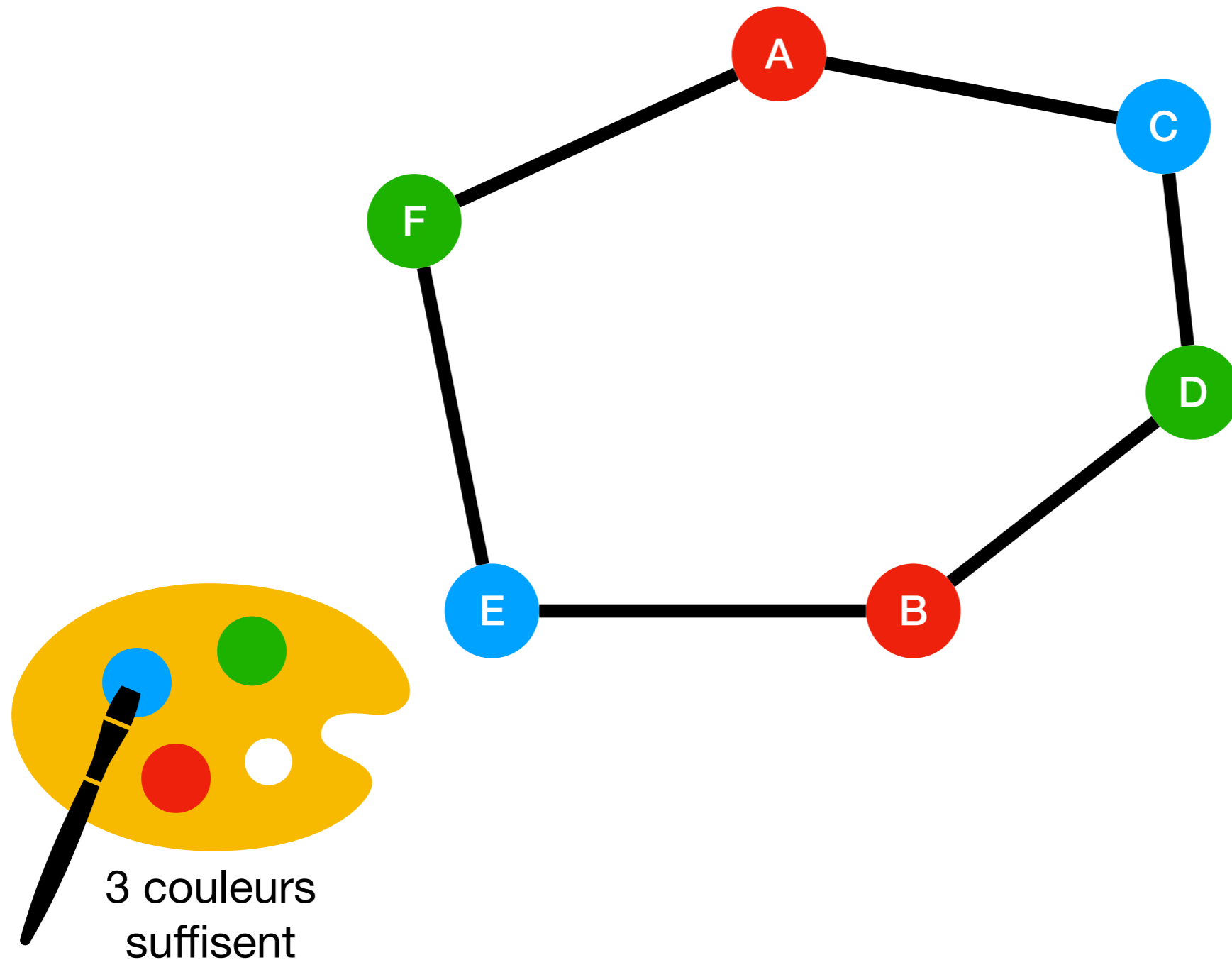
Colorations non optimales



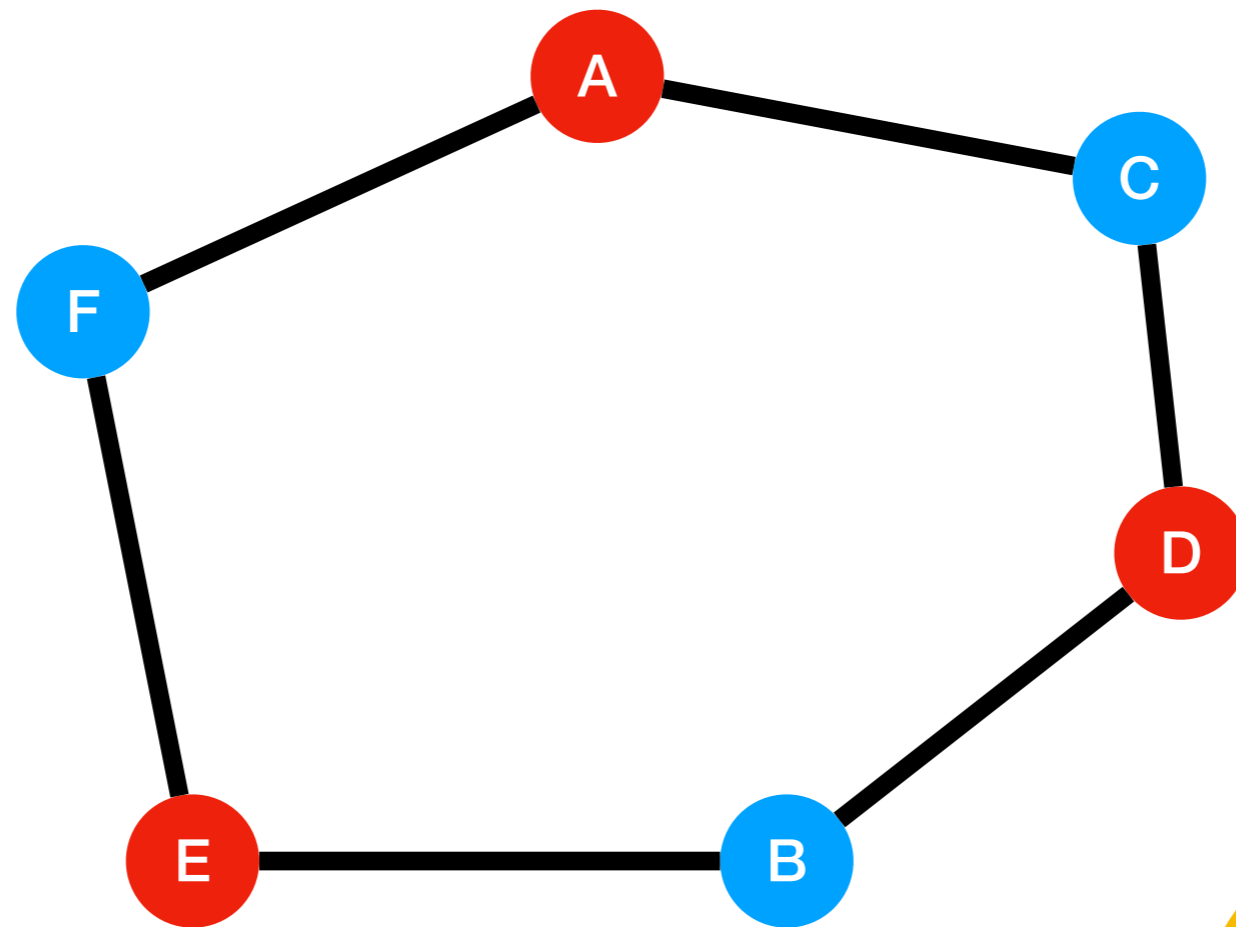
Colorations non optimales



Colorations non optimales



Colorations non optimales



mais 2 couleurs
suffisent aussi !

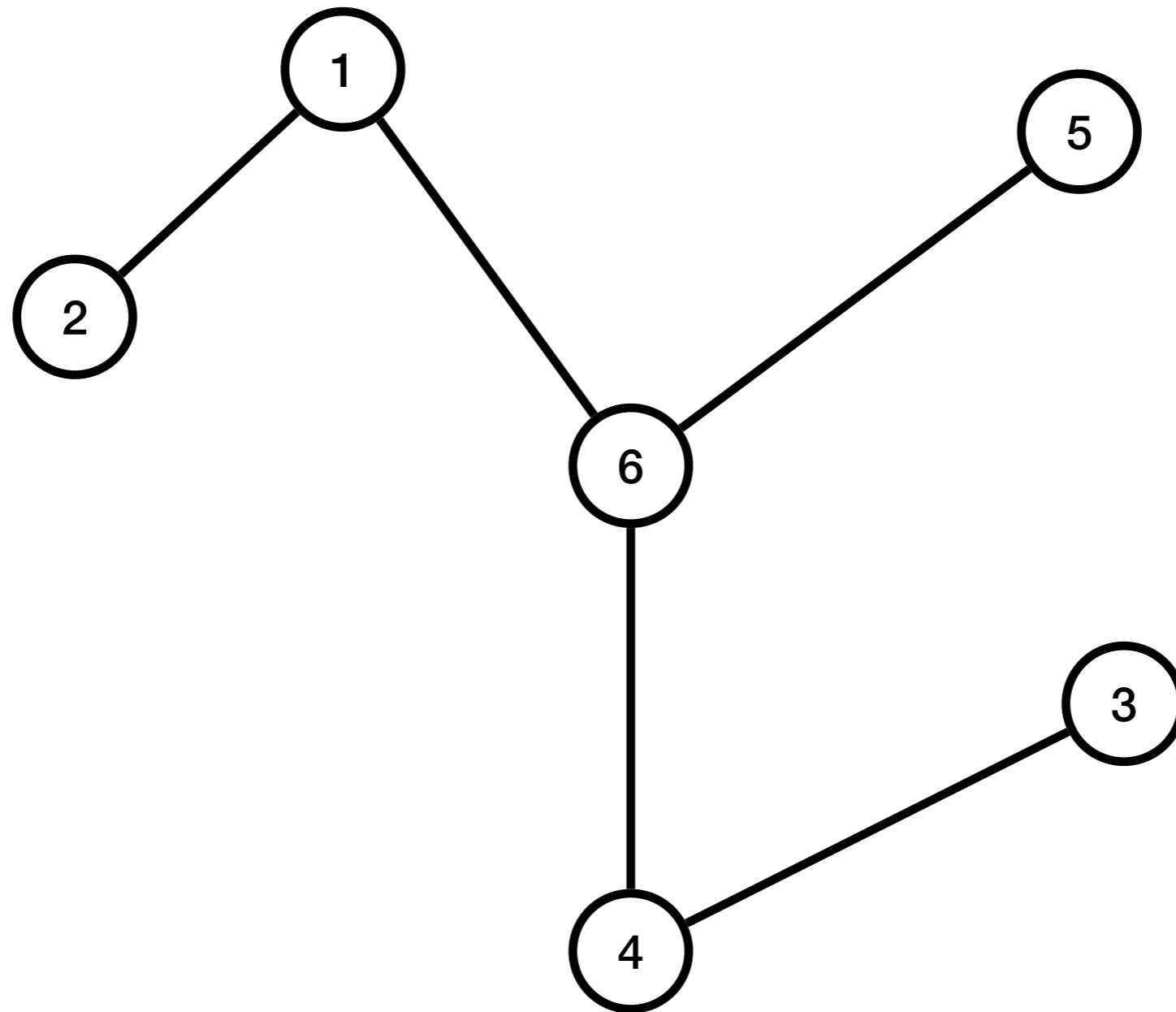
Exercices 3 et 4 du TD4

Arbres

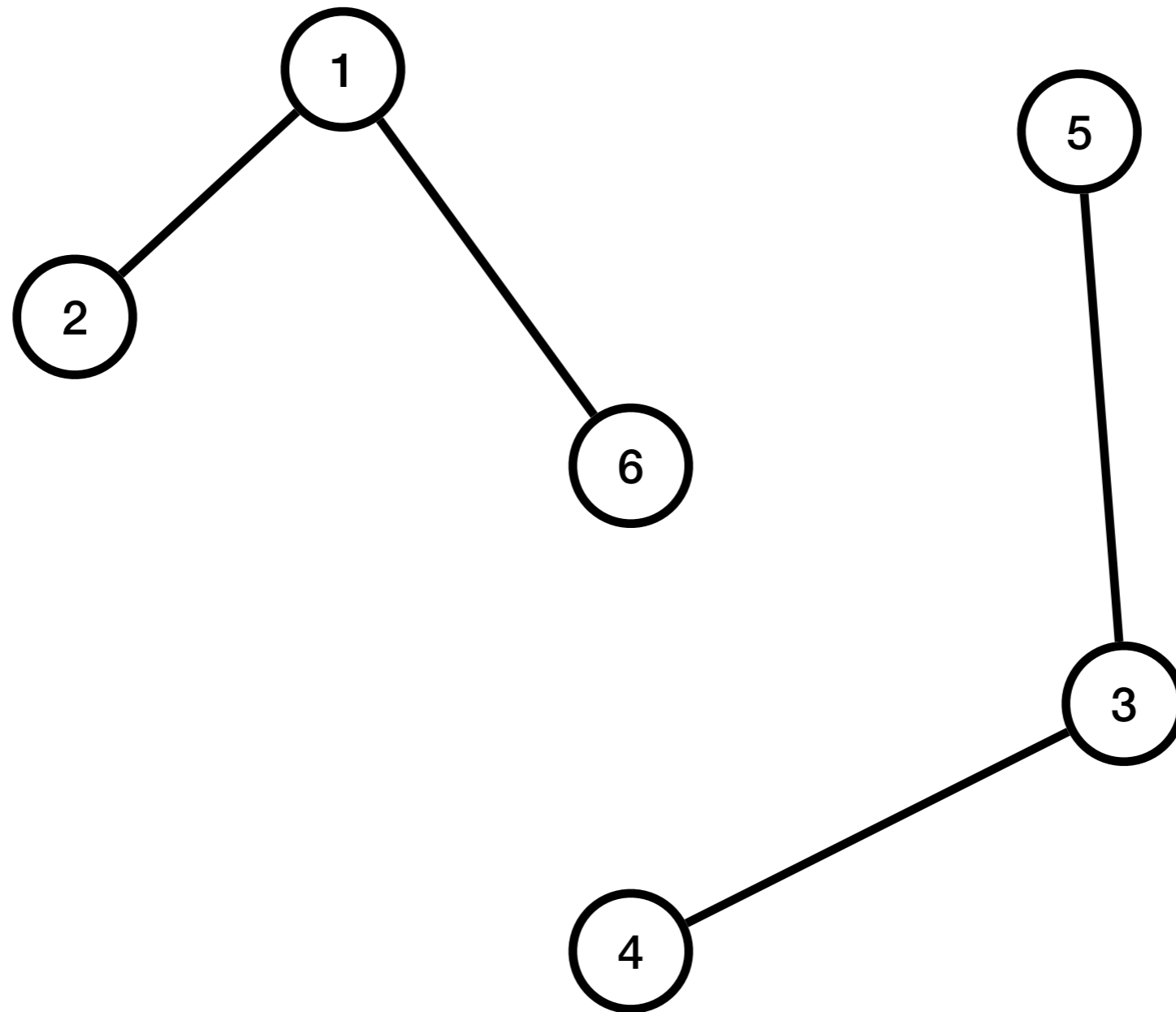


**Un arbre est un graphe
non orienté, connexe
et acyclique**

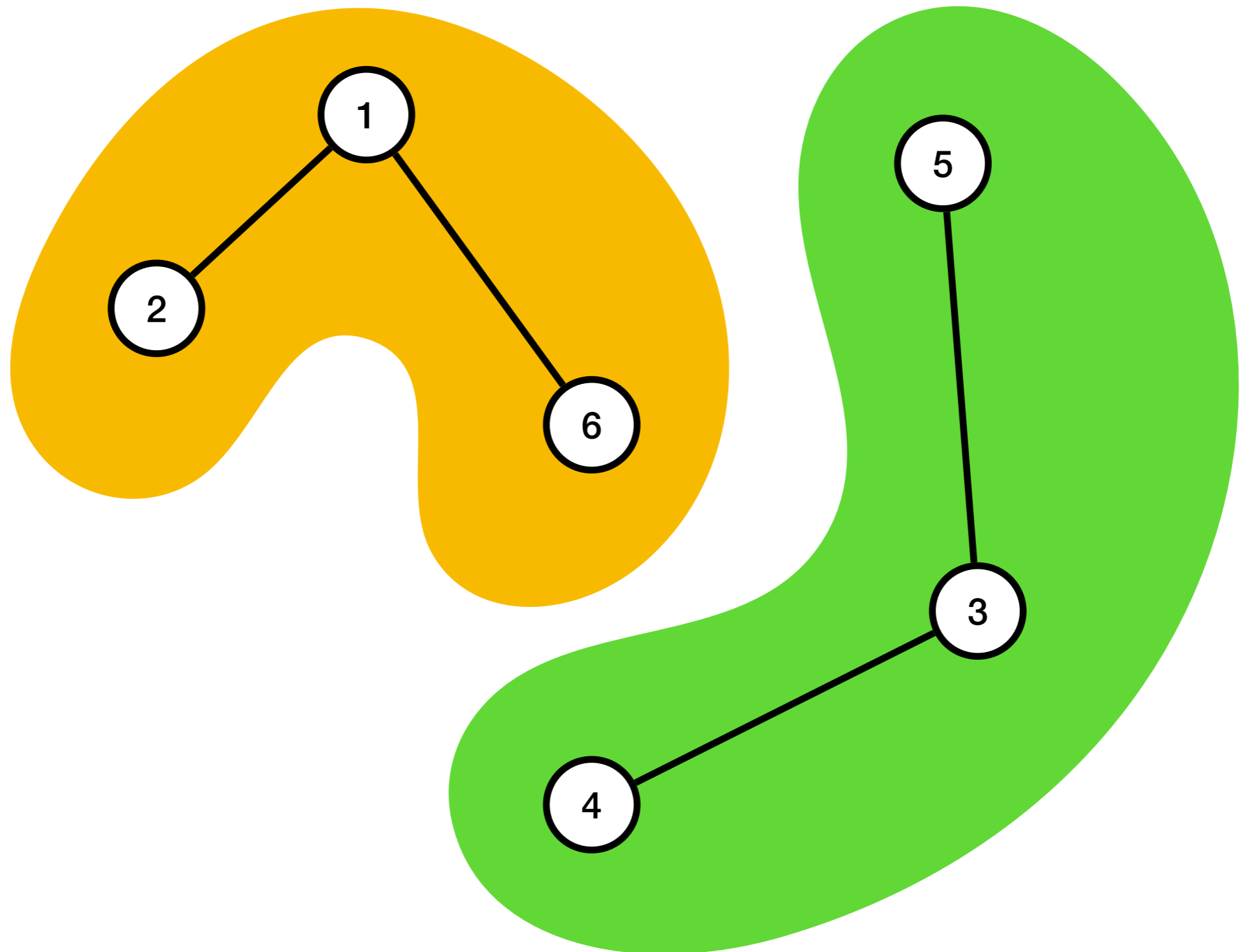
Un arbre



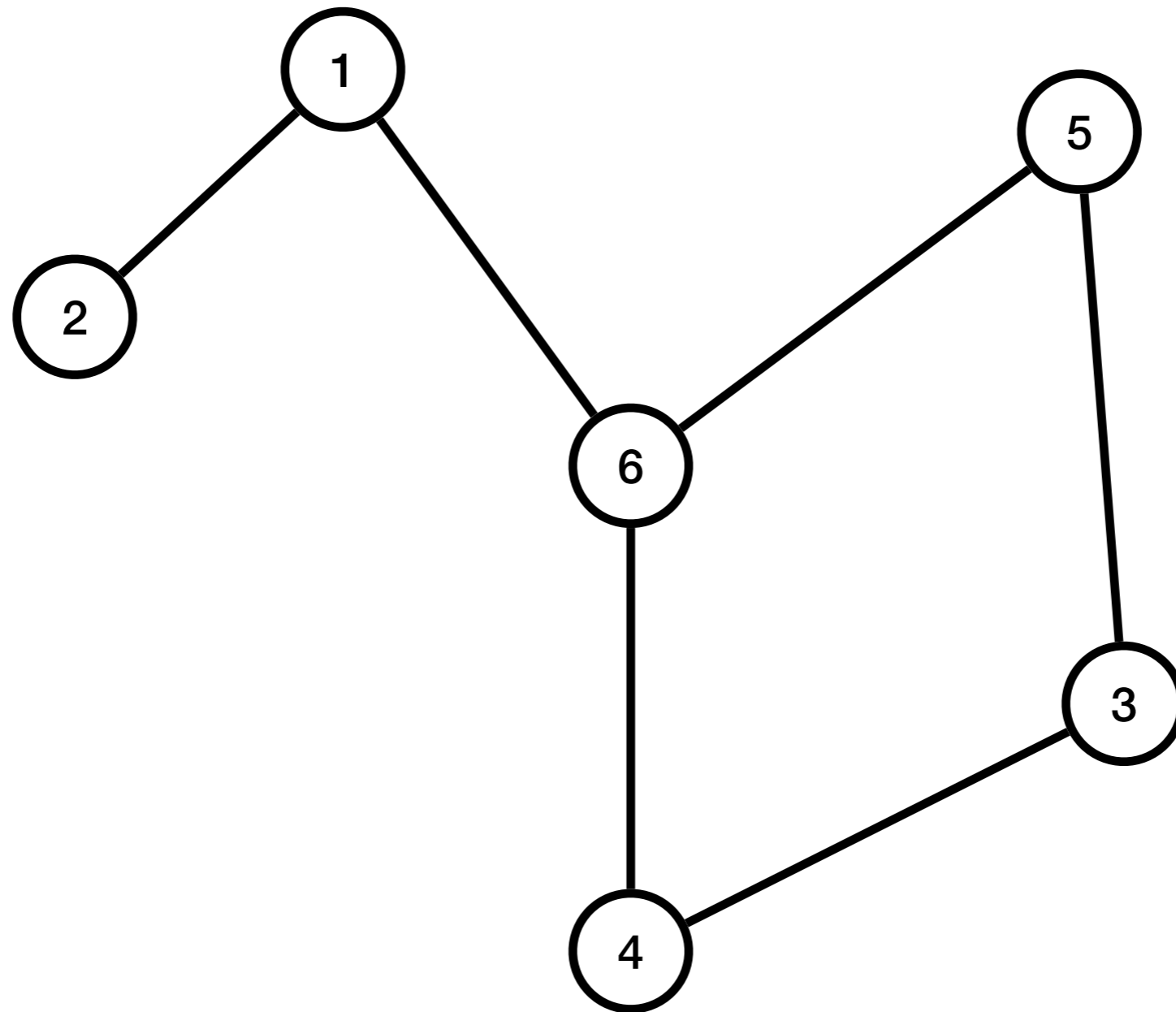
Pas un arbre



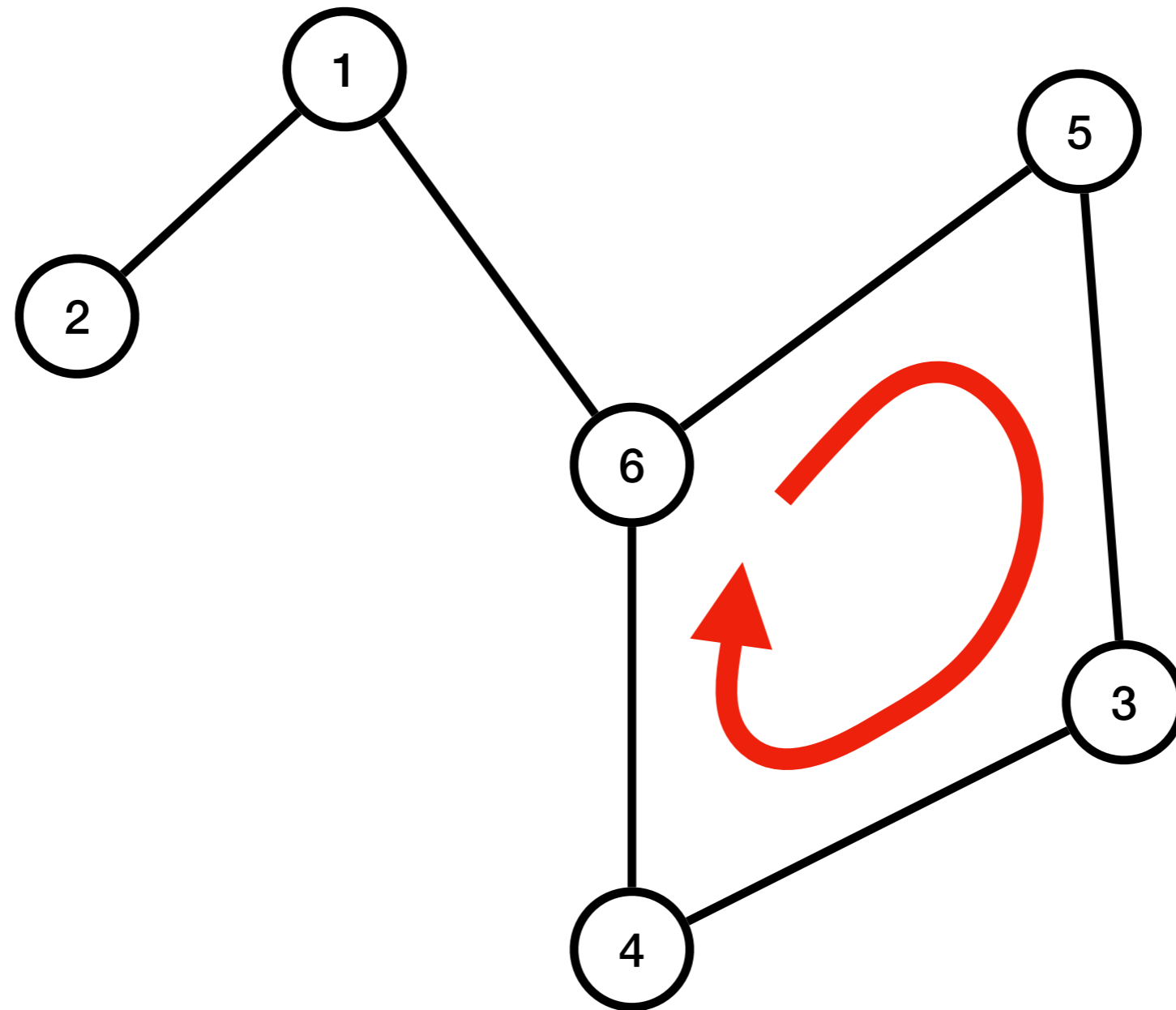
Pas un arbre



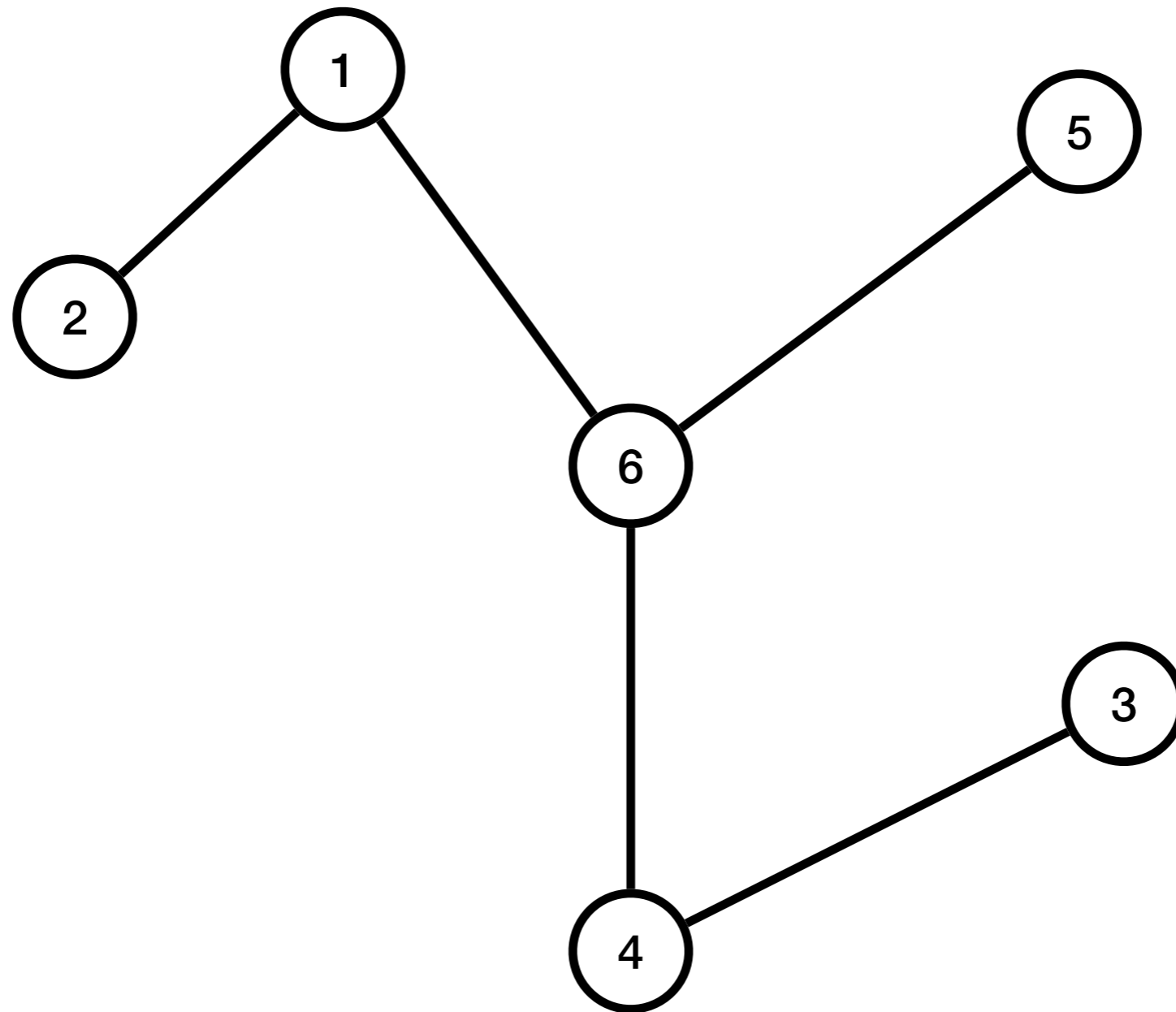
Pas un arbre



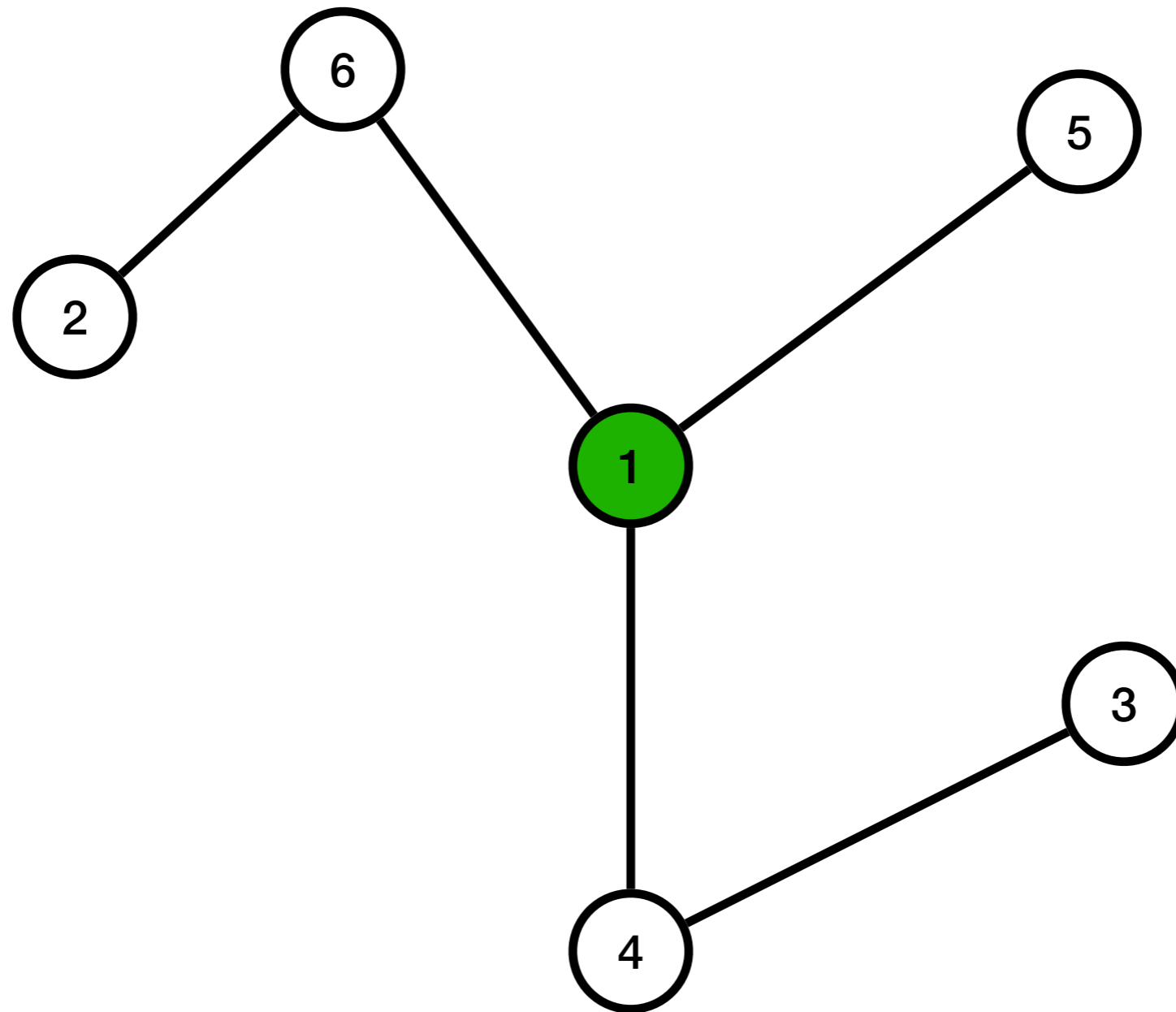
Pas un arbre



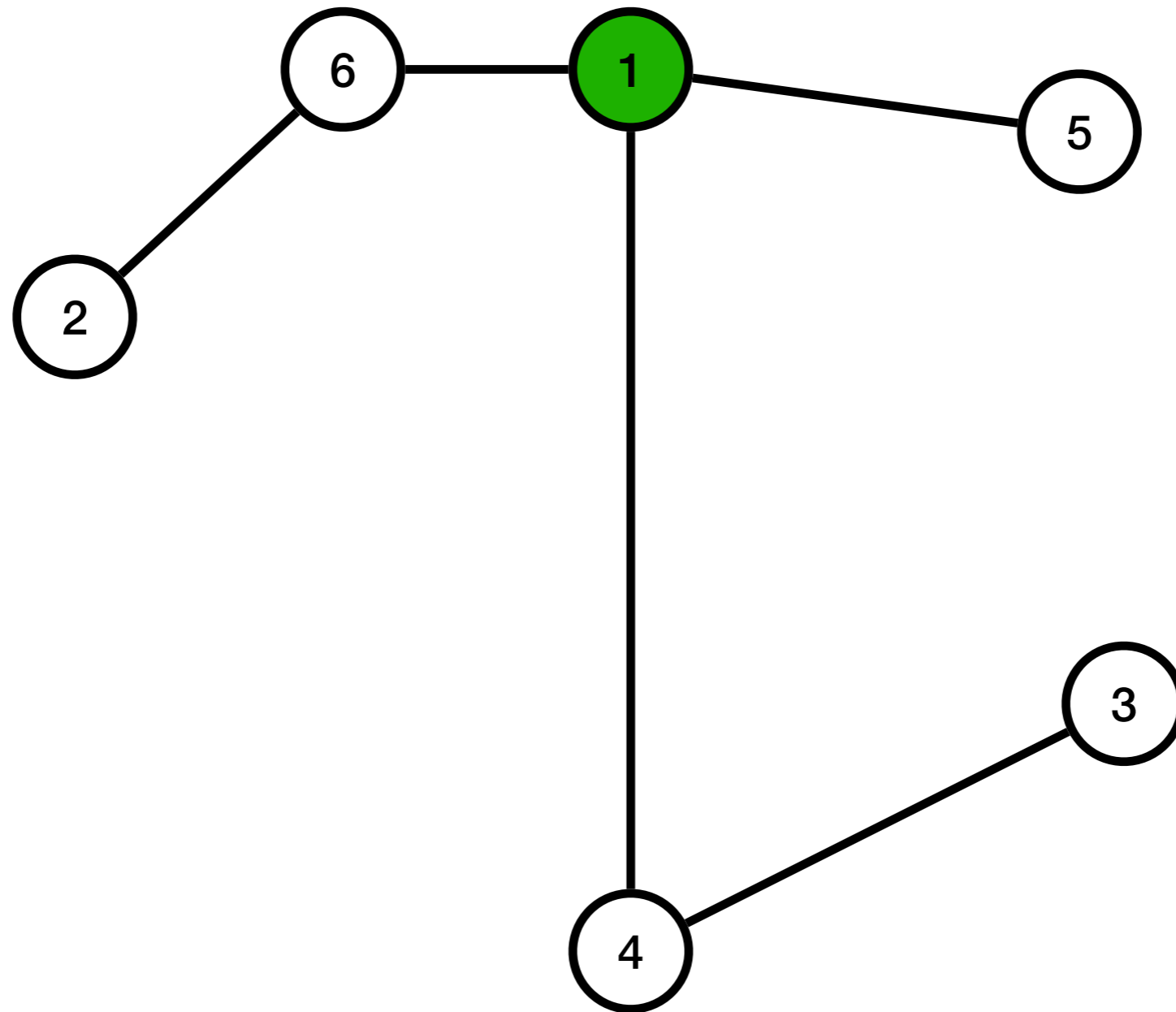
Arbres enracinés



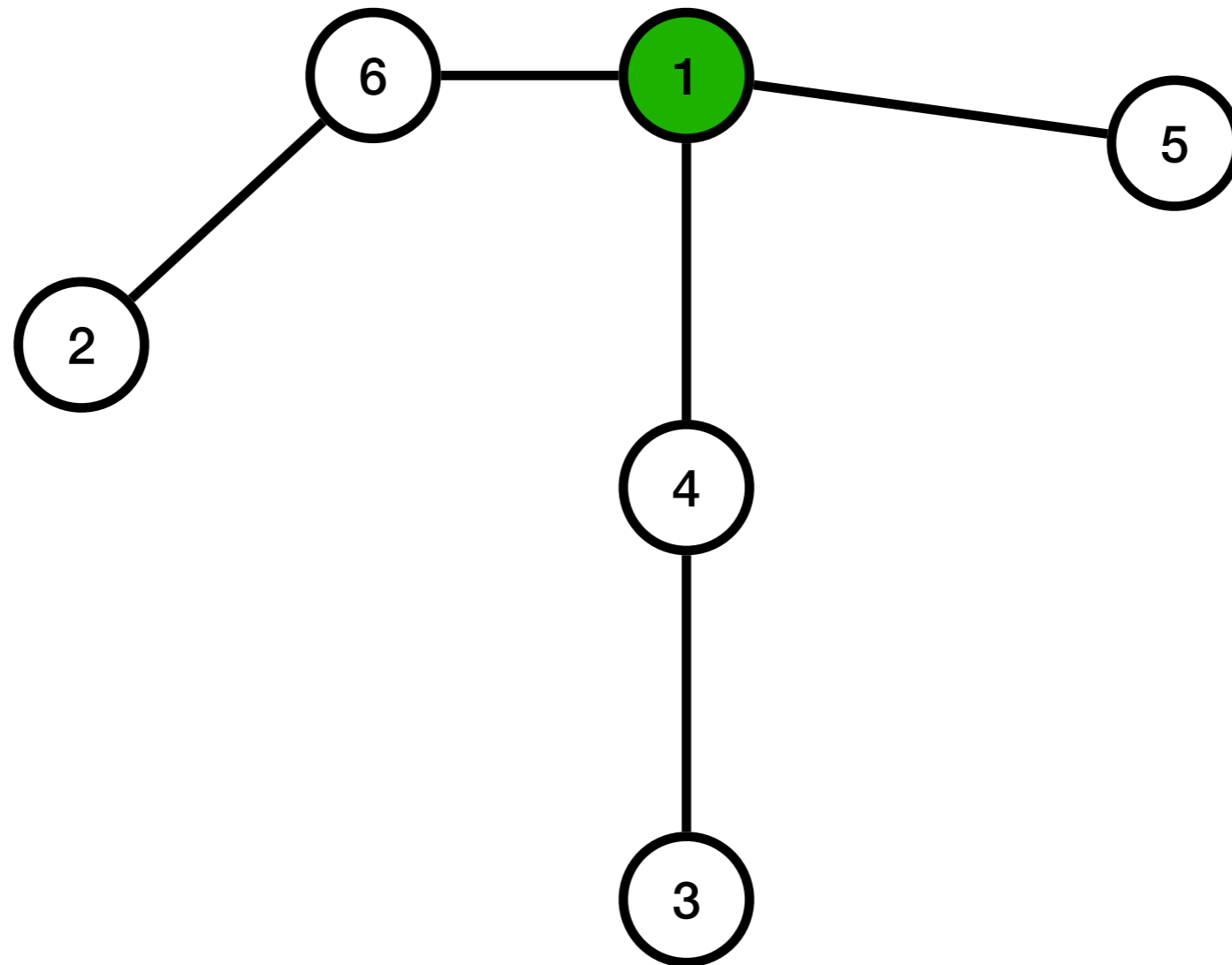
Arbres enracinés



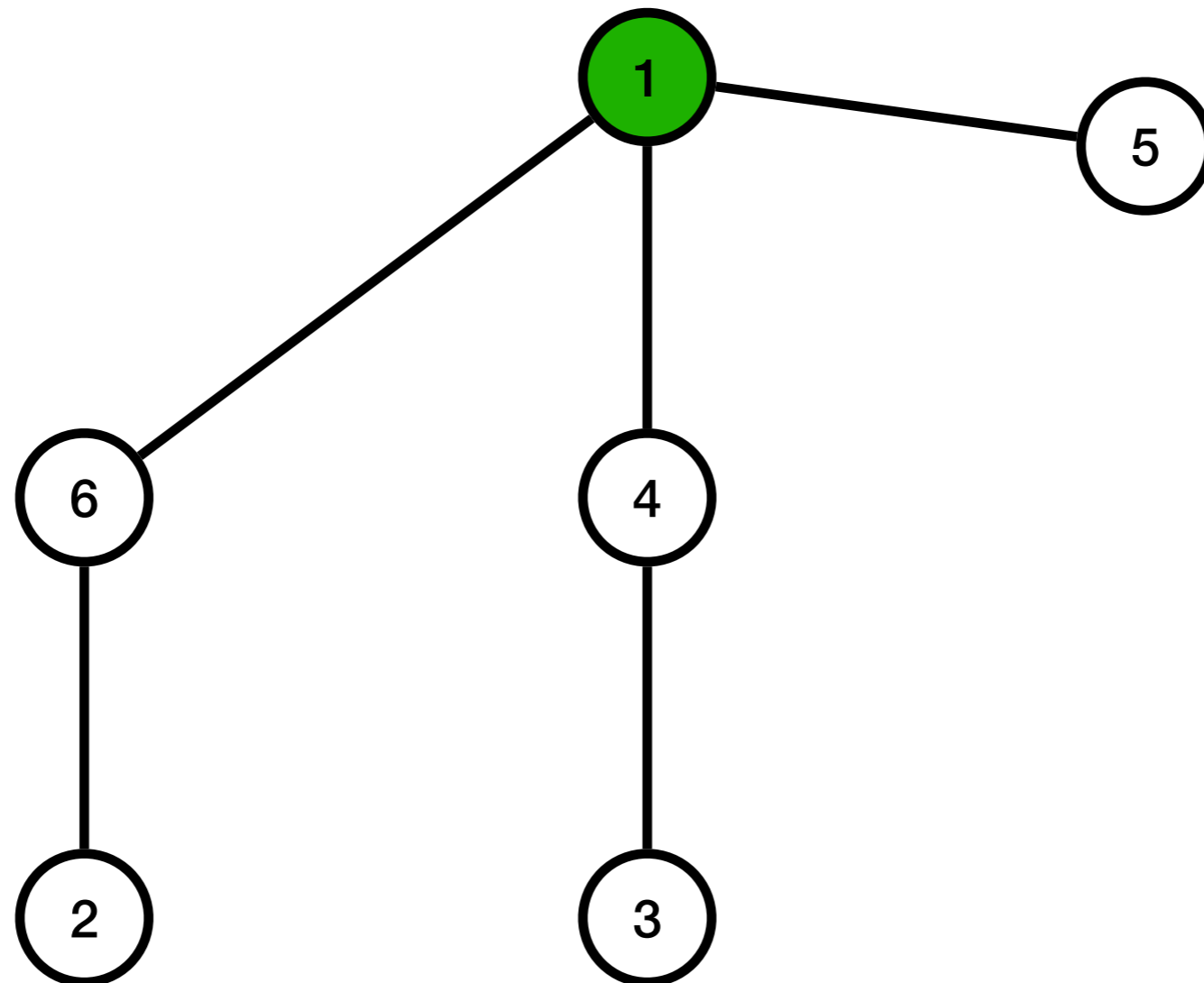
Arbres enracinés



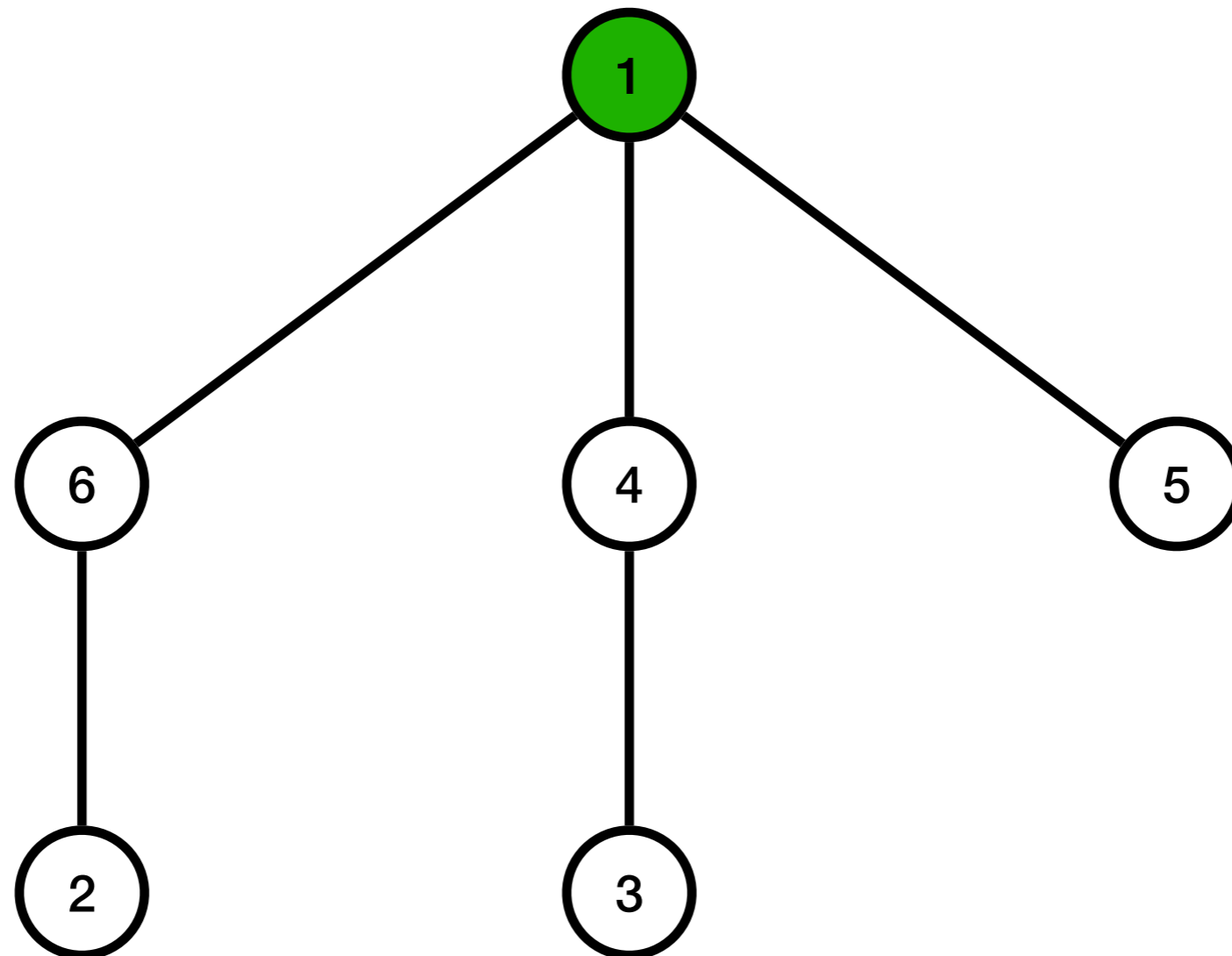
Arbres enracinés



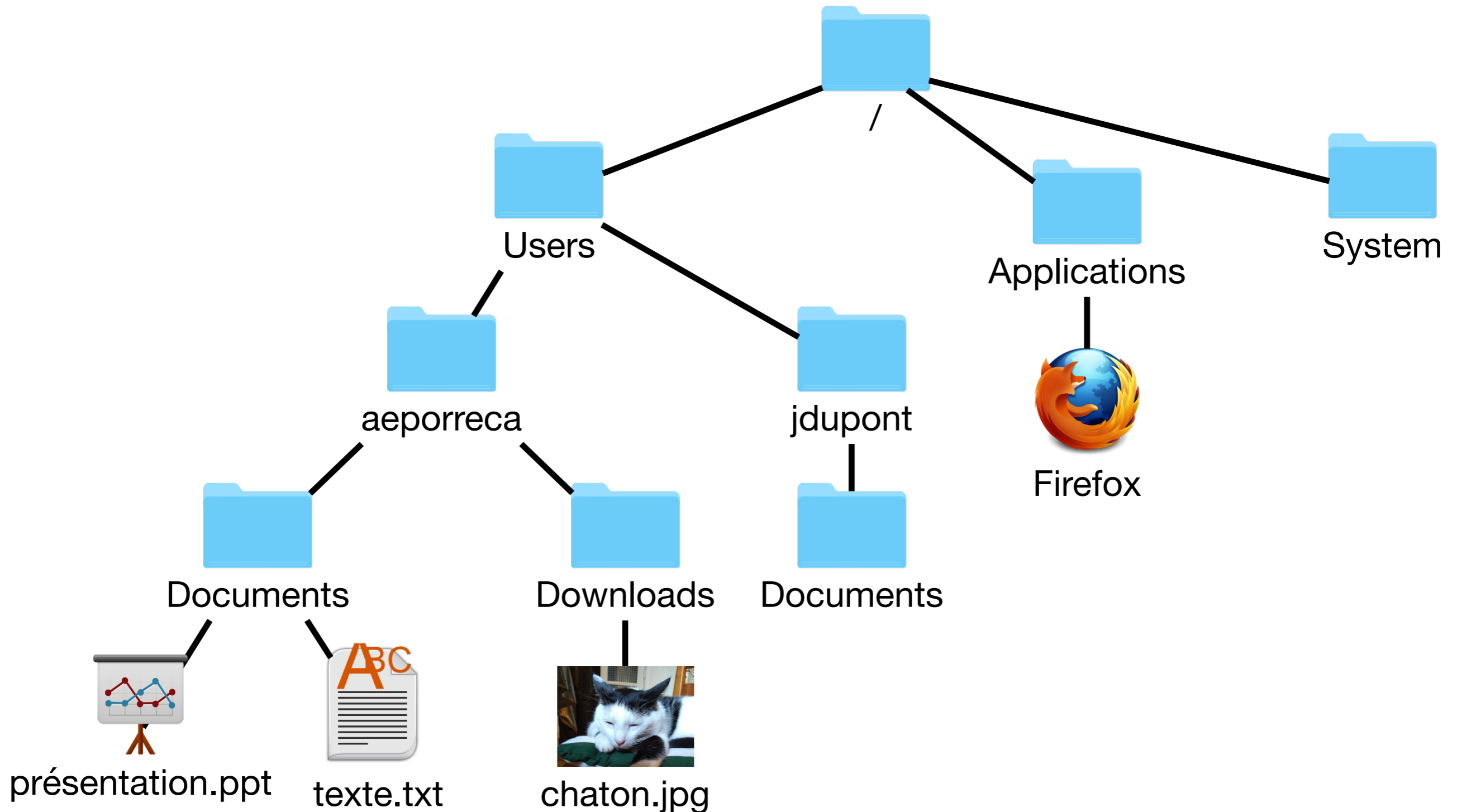
Arbres enracinés



Arbres enracinés



Arborescence des répertoires (dossiers) et fichiers



Deux codages pour le texte

lettre	codage fixe
<i>a</i>	00000
<i>b</i>	00001
<i>c</i>	00010
<i>d</i>	00011
<i>e</i>	00100
<i>f</i>	00101
<i>g</i>	00110
<i>h</i>	00111
<i>i</i>	01000
<i>j</i>	01001
<i>k</i>	01010
<i>l</i>	01011
<i>m</i>	01100
<i>n</i>	01101
<i>o</i>	01110
<i>p</i>	01111
<i>q</i>	10000
<i>r</i>	10001
<i>s</i>	10010
<i>t</i>	10011
<i>u</i>	10100
<i>v</i>	10101
<i>w</i>	10110
<i>x</i>	10111
<i>y</i>	11000
<i>z</i>	11001
<i>espace</i>	11010

lettre	codage variable
<i>a</i>	1010
<i>b</i>	0010011
<i>c</i>	01001
<i>d</i>	01110
<i>e</i>	110
<i>f</i>	0111100
<i>g</i>	0111110
<i>h</i>	0010010
<i>i</i>	1000
<i>j</i>	011111110
<i>k</i>	011111111001
<i>l</i>	0001
<i>m</i>	00101
<i>n</i>	1001
<i>o</i>	0000
<i>p</i>	01000
<i>q</i>	0111101
<i>r</i>	0101
<i>s</i>	1011
<i>t</i>	0110
<i>u</i>	0011
<i>v</i>	001000
<i>w</i>	011111111000
<i>x</i>	01111110
<i>y</i>	0111111111
<i>z</i>	01111111101
<i>espace</i>	111

Deux codages pour le texte

encore

lettre	codage fixe
<i>a</i>	00000
<i>b</i>	00001
<i>c</i>	00010
<i>d</i>	00011
<i>e</i>	00100
<i>f</i>	00101
<i>g</i>	00110
<i>h</i>	00111
<i>i</i>	01000
<i>j</i>	01001
<i>k</i>	01010
<i>l</i>	01011
<i>m</i>	01100
<i>n</i>	01101
<i>o</i>	01110
<i>p</i>	01111
<i>q</i>	10000
<i>r</i>	10001
<i>s</i>	10010
<i>t</i>	10011
<i>u</i>	10100
<i>v</i>	10101
<i>w</i>	10110
<i>x</i>	10111
<i>y</i>	11000
<i>z</i>	11001
<i>espace</i>	11010

lettre	codage variable
<i>a</i>	1010
<i>b</i>	0010011
<i>c</i>	01001
<i>d</i>	01110
<i>e</i>	110
<i>f</i>	0111100
<i>g</i>	0111110
<i>h</i>	0010010
<i>i</i>	1000
<i>j</i>	011111110
<i>k</i>	011111111001
<i>l</i>	0001
<i>m</i>	00101
<i>n</i>	1001
<i>o</i>	0000
<i>p</i>	01000
<i>q</i>	0111101
<i>r</i>	0101
<i>s</i>	1011
<i>t</i>	0110
<i>u</i>	0011
<i>v</i>	001000
<i>w</i>	011111111000
<i>x</i>	01111110
<i>y</i>	0111111111
<i>z</i>	01111111101
<i>espace</i>	111

Deux codages pour le texte

lettre	codage fixe
<i>a</i>	00000
<i>b</i>	00001
<i>c</i>	00010
<i>d</i>	00011
<i>e</i>	00100
<i>f</i>	00101
<i>g</i>	00110
<i>h</i>	00111
<i>i</i>	01000
<i>j</i>	01001
<i>k</i>	01010
<i>l</i>	01011
<i>m</i>	01100
<i>n</i>	01101
<i>o</i>	01110
<i>p</i>	01111
<i>q</i>	10000
<i>r</i>	10001
<i>s</i>	10010
<i>t</i>	10011
<i>u</i>	10100
<i>v</i>	10101
<i>w</i>	10110
<i>x</i>	10111
<i>y</i>	11000
<i>z</i>	11001
<i>espace</i>	11010

encore



lettre	codage variable
<i>a</i>	1010
<i>b</i>	0010011
<i>c</i>	01001
<i>d</i>	01110
<i>e</i>	110
<i>f</i>	0111100
<i>g</i>	0111110
<i>h</i>	0010010
<i>i</i>	1000
<i>j</i>	011111110
<i>k</i>	011111111001
<i>l</i>	0001
<i>m</i>	00101
<i>n</i>	1001
<i>o</i>	0000
<i>p</i>	01000
<i>q</i>	0111101
<i>r</i>	0101
<i>s</i>	1011
<i>t</i>	0110
<i>u</i>	0011
<i>v</i>	001000
<i>w</i>	011111111000
<i>x</i>	01111110
<i>y</i>	0111111111
<i>z</i>	01111111101
<i>espace</i>	111

Deux codages pour le texte

lettre	codage fixe
<i>a</i>	00000
<i>b</i>	00001
<i>c</i>	00010
<i>d</i>	00011
<i>e</i>	00100
<i>f</i>	00101
<i>g</i>	00110
<i>h</i>	00111
<i>i</i>	01000
<i>j</i>	01001
<i>k</i>	01010
<i>l</i>	01011
<i>m</i>	01100
<i>n</i>	01101
<i>o</i>	01110
<i>p</i>	01111
<i>q</i>	10000
<i>r</i>	10001
<i>s</i>	10010
<i>t</i>	10011
<i>u</i>	10100
<i>v</i>	10101
<i>w</i>	10110
<i>x</i>	10111
<i>y</i>	11000
<i>z</i>	11001
<i>espace</i>	11010

encore



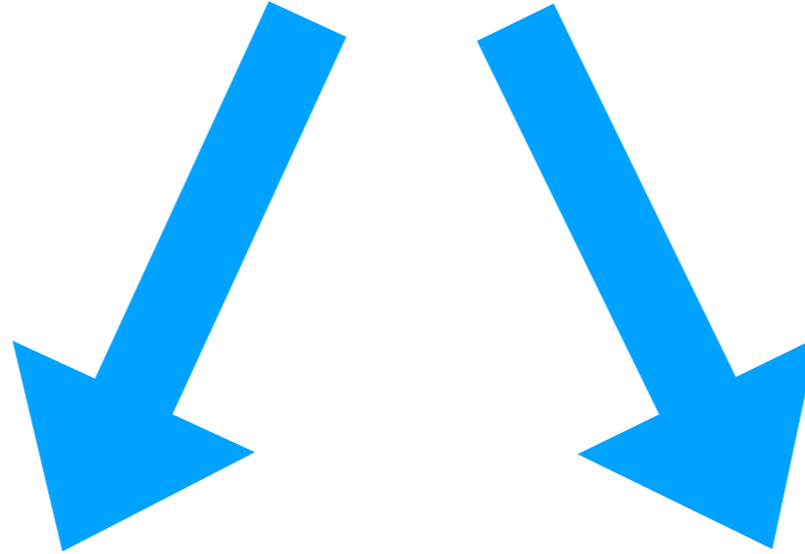
00100
01101
00010
01110
10001
00100

lettre	codage variable
<i>a</i>	1010
<i>b</i>	0010011
<i>c</i>	01001
<i>d</i>	01110
<i>e</i>	110
<i>f</i>	0111100
<i>g</i>	0111110
<i>h</i>	0010010
<i>i</i>	1000
<i>j</i>	011111110
<i>k</i>	011111111001
<i>l</i>	0001
<i>m</i>	00101
<i>n</i>	1001
<i>o</i>	0000
<i>p</i>	01000
<i>q</i>	0111101
<i>r</i>	0101
<i>s</i>	1011
<i>t</i>	0110
<i>u</i>	0011
<i>v</i>	001000
<i>w</i>	011111111000
<i>x</i>	01111110
<i>y</i>	0111111111
<i>z</i>	01111111101
<i>espace</i>	111

Deux codages pour le texte

lettre	codage fixe
<i>a</i>	00000
<i>b</i>	00001
<i>c</i>	00010
<i>d</i>	00011
<i>e</i>	00100
<i>f</i>	00101
<i>g</i>	00110
<i>h</i>	00111
<i>i</i>	01000
<i>j</i>	01001
<i>k</i>	01010
<i>l</i>	01011
<i>m</i>	01100
<i>n</i>	01101
<i>o</i>	01110
<i>p</i>	01111
<i>q</i>	10000
<i>r</i>	10001
<i>s</i>	10010
<i>t</i>	10011
<i>u</i>	10100
<i>v</i>	10101
<i>w</i>	10110
<i>x</i>	10111
<i>y</i>	11000
<i>z</i>	11001
<i>espace</i>	11010

encore



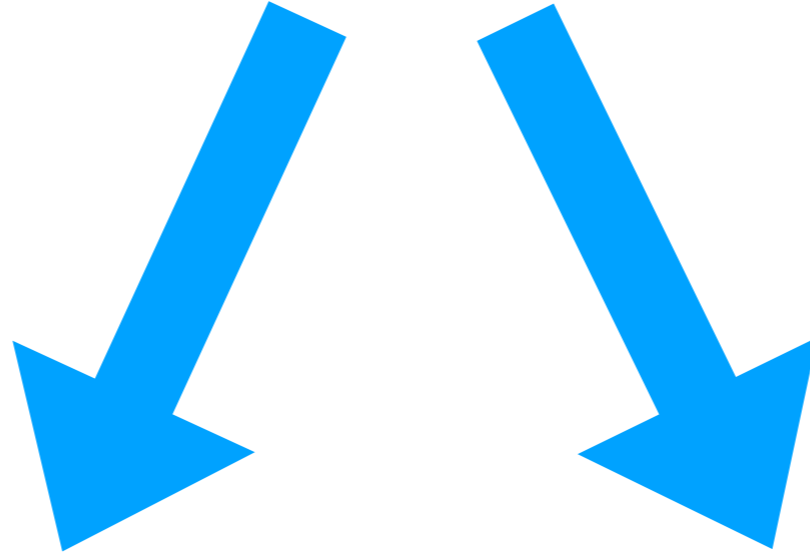
00100
01101
00010
01110
10001
00100

lettre	codage variable
<i>a</i>	1010
<i>b</i>	0010011
<i>c</i>	01001
<i>d</i>	01110
<i>e</i>	110
<i>f</i>	0111100
<i>g</i>	0111110
<i>h</i>	0010010
<i>i</i>	1000
<i>j</i>	011111110
<i>k</i>	011111111001
<i>l</i>	0001
<i>m</i>	00101
<i>n</i>	1001
<i>o</i>	0000
<i>p</i>	01000
<i>q</i>	0111101
<i>r</i>	0101
<i>s</i>	1011
<i>t</i>	0110
<i>u</i>	0011
<i>v</i>	001000
<i>w</i>	011111111000
<i>x</i>	01111110
<i>y</i>	0111111111
<i>z</i>	01111111101
<i>espace</i>	111

Deux codages pour le texte

lettre	codage fixe
<i>a</i>	00000
<i>b</i>	00001
<i>c</i>	00010
<i>d</i>	00011
<i>e</i>	00100
<i>f</i>	00101
<i>g</i>	00110
<i>h</i>	00111
<i>i</i>	01000
<i>j</i>	01001
<i>k</i>	01010
<i>l</i>	01011
<i>m</i>	01100
<i>n</i>	01101
<i>o</i>	01110
<i>p</i>	01111
<i>q</i>	10000
<i>r</i>	10001
<i>s</i>	10010
<i>t</i>	10011
<i>u</i>	10100
<i>v</i>	10101
<i>w</i>	10110
<i>x</i>	10111
<i>y</i>	11000
<i>z</i>	11001
<i>espace</i>	11010

encore



00100
01101
00010
01110
10001
00100

110
1001
01001
0000
0101
110

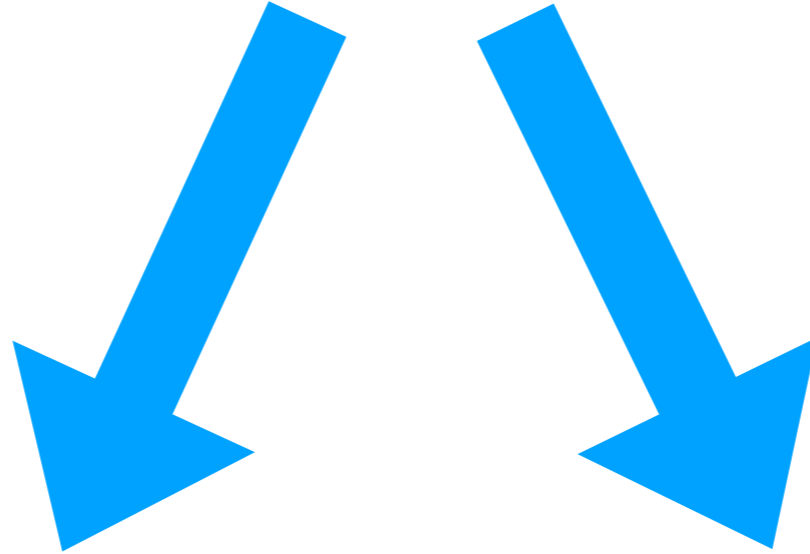
lettre	codage variable
<i>a</i>	1010
<i>b</i>	0010011
<i>c</i>	01001
<i>d</i>	01110
<i>e</i>	110
<i>f</i>	0111100
<i>g</i>	0111110
<i>h</i>	0010010
<i>i</i>	1000
<i>j</i>	011111110
<i>k</i>	011111111001
<i>l</i>	0001
<i>m</i>	00101
<i>n</i>	1001
<i>o</i>	0000
<i>p</i>	01000
<i>q</i>	0111101
<i>r</i>	0101
<i>s</i>	1011
<i>t</i>	0110
<i>u</i>	0011
<i>v</i>	001000
<i>w</i>	011111111000
<i>x</i>	01111110
<i>y</i>	0111111111
<i>z</i>	01111111101
<i>espace</i>	111

Deux codages pour le texte

lettre	codage fixe
<i>a</i>	00000
<i>b</i>	00001
<i>c</i>	00010
<i>d</i>	00011
<i>e</i>	00100
<i>f</i>	00101
<i>g</i>	00110
<i>h</i>	00111
<i>i</i>	01000
<i>j</i>	01001
<i>k</i>	01010
<i>l</i>	01011
<i>m</i>	01100
<i>n</i>	01101
<i>o</i>	01110
<i>p</i>	01111
<i>q</i>	10000
<i>r</i>	10001
<i>s</i>	10010
<i>t</i>	10011
<i>u</i>	10100
<i>v</i>	10101
<i>w</i>	10110
<i>x</i>	10111
<i>y</i>	11000
<i>z</i>	11001
<i>espace</i>	11010

lettre	codage variable
<i>a</i>	1010
<i>b</i>	0010011
<i>c</i>	01001
<i>d</i>	01110
<i>e</i>	110
<i>f</i>	0111100
<i>g</i>	0111110
<i>h</i>	0010010
<i>i</i>	1000
<i>j</i>	011111110
<i>k</i>	011111111001
<i>l</i>	0001
<i>m</i>	00101
<i>n</i>	1001
<i>o</i>	0000
<i>p</i>	01000
<i>q</i>	0111101
<i>r</i>	0101
<i>s</i>	1011
<i>t</i>	0110
<i>u</i>	0011
<i>v</i>	001000
<i>w</i>	011111111000
<i>x</i>	01111110
<i>y</i>	0111111111
<i>z</i>	01111111101
<i>espace</i>	111

encore



00100
01101
00010
01110
10001
00100

30 bits

110
1001
01001
0000
0101
110

23 bits

Décoder avec le codage variable

100010010111100000001010010110100110100000001001

lettre	codage variable
<i>a</i>	1010
<i>b</i>	0010011
<i>c</i>	01001
<i>d</i>	01110
<i>e</i>	110
<i>f</i>	0111100
<i>g</i>	0111110
<i>h</i>	0010010
<i>i</i>	1000
<i>j</i>	011111110
<i>k</i>	011111111001
<i>l</i>	0001
<i>m</i>	00101
<i>n</i>	1001
<i>o</i>	0000
<i>p</i>	01000
<i>q</i>	0111101
<i>r</i>	0101
<i>s</i>	1011
<i>t</i>	0110
<i>u</i>	0011
<i>v</i>	001000
<i>w</i>	011111111000
<i>x</i>	01111110
<i>y</i>	0111111111
<i>z</i>	01111111101
<i>espace</i>	111

Décoder avec le codage variable

100010010111100000001010010110100110100000001001

lettre	codage variable
<i>a</i>	1010
<i>b</i>	0010011
<i>c</i>	01001
<i>d</i>	01110
<i>e</i>	110
<i>f</i>	0111100
<i>g</i>	0111110
<i>h</i>	0010010
<i>i</i>	1000
<i>j</i>	011111110
<i>k</i>	011111111001
<i>l</i>	0001
<i>m</i>	00101
<i>n</i>	1001
<i>o</i>	0000
<i>p</i>	01000
<i>q</i>	0111101
<i>r</i>	0101
<i>s</i>	1011
<i>t</i>	0110
<i>u</i>	0011
<i>v</i>	001000
<i>w</i>	011111111000
<i>x</i>	01111110
<i>y</i>	0111111111
<i>z</i>	01111111101
<i>espace</i>	111

Décoder avec le codage variable

100010010111100000001010010110100110100000001001

i

lettre	codage variable
<i>a</i>	1010
<i>b</i>	0010011
<i>c</i>	01001
<i>d</i>	01110
<i>e</i>	110
<i>f</i>	0111100
<i>g</i>	0111110
<i>h</i>	0010010
<i>i</i>	1000
<i>j</i>	011111110
<i>k</i>	011111111001
<i>l</i>	0001
<i>m</i>	00101
<i>n</i>	1001
<i>o</i>	0000
<i>p</i>	01000
<i>q</i>	0111101
<i>r</i>	0101
<i>s</i>	1011
<i>t</i>	0110
<i>u</i>	0011
<i>v</i>	001000
<i>w</i>	011111111000
<i>x</i>	01111110
<i>y</i>	0111111111
<i>z</i>	01111111101
<i>espace</i>	111

Décoder avec le codage variable

100010010111100000001010010110100110100000001001

i

lettre	codage variable
<i>a</i>	1010
<i>b</i>	0010011
<i>c</i>	01001
<i>d</i>	01110
<i>e</i>	110
<i>f</i>	0111100
<i>g</i>	0111110
<i>h</i>	0010010
<i>i</i>	1000
<i>j</i>	011111110
<i>k</i>	011111111001
<i>l</i>	0001
<i>m</i>	00101
<i>n</i>	1001
<i>o</i>	0000
<i>p</i>	01000
<i>q</i>	0111101
<i>r</i>	0101
<i>s</i>	1011
<i>t</i>	0110
<i>u</i>	0011
<i>v</i>	001000
<i>w</i>	011111111000
<i>x</i>	01111110
<i>y</i>	0111111111
<i>z</i>	01111111101
<i>espace</i>	111

Décoder avec le codage variable

100010010111100000001010010110100110100000001001
i n

lettre	codage variable
<i>a</i>	1010
<i>b</i>	0010011
<i>c</i>	01001
<i>d</i>	01110
<i>e</i>	110
<i>f</i>	0111100
<i>g</i>	0111110
<i>h</i>	0010010
<i>i</i>	1000
<i>j</i>	011111110
<i>k</i>	011111111001
<i>l</i>	0001
<i>m</i>	00101
<i>n</i>	1001
<i>o</i>	0000
<i>p</i>	01000
<i>q</i>	0111101
<i>r</i>	0101
<i>s</i>	1011
<i>t</i>	0110
<i>u</i>	0011
<i>v</i>	001000
<i>w</i>	011111111000
<i>x</i>	01111110
<i>y</i>	0111111111
<i>z</i>	01111111101
<i>espace</i>	111

Décoder avec le codage variable

100010010111100000001010010110100110100000001001

i n

lettre	codage variable
<i>a</i>	1010
<i>b</i>	0010011
<i>c</i>	01001
<i>d</i>	01110
<i>e</i>	110
<i>f</i>	0111100
<i>g</i>	0111110
<i>h</i>	0010010
<i>i</i>	1000
<i>j</i>	011111110
<i>k</i>	011111111001
<i>l</i>	0001
<i>m</i>	00101
<i>n</i>	1001
<i>o</i>	0000
<i>p</i>	01000
<i>q</i>	0111101
<i>r</i>	0101
<i>s</i>	1011
<i>t</i>	0110
<i>u</i>	0011
<i>v</i>	001000
<i>w</i>	011111111000
<i>x</i>	01111110
<i>y</i>	0111111111
<i>z</i>	01111111101
<i>espace</i>	111

Décoder avec le codage variable

100010010111100000001010010110100110100000001001
i n f

lettre	codage variable
<i>a</i>	1010
<i>b</i>	0010011
<i>c</i>	01001
<i>d</i>	01110
<i>e</i>	110
<i>f</i>	0111100
<i>g</i>	0111110
<i>h</i>	0010010
<i>i</i>	1000
<i>j</i>	011111110
<i>k</i>	011111111001
<i>l</i>	0001
<i>m</i>	00101
<i>n</i>	1001
<i>o</i>	0000
<i>p</i>	01000
<i>q</i>	0111101
<i>r</i>	0101
<i>s</i>	1011
<i>t</i>	0110
<i>u</i>	0011
<i>v</i>	001000
<i>w</i>	011111111000
<i>x</i>	01111110
<i>y</i>	0111111111
<i>z</i>	01111111101
<i>espace</i>	111

Décoder avec le codage variable

100010010111100000001010010110100110100000001001
i n f

lettre	codage variable
<i>a</i>	1010
<i>b</i>	0010011
<i>c</i>	01001
<i>d</i>	01110
<i>e</i>	110
<i>f</i>	0111100
<i>g</i>	0111110
<i>h</i>	0010010
<i>i</i>	1000
<i>j</i>	011111110
<i>k</i>	011111111001
<i>l</i>	0001
<i>m</i>	00101
<i>n</i>	1001
<i>o</i>	0000
<i>p</i>	01000
<i>q</i>	0111101
<i>r</i>	0101
<i>s</i>	1011
<i>t</i>	0110
<i>u</i>	0011
<i>v</i>	001000
<i>w</i>	011111111000
<i>x</i>	01111110
<i>y</i>	0111111111
<i>z</i>	01111111101
<i>espace</i>	111

Décoder avec le codage variable

10001001011110000000101010010110100110100000001001
i n f o

lettre	codage variable
<i>a</i>	1010
<i>b</i>	0010011
<i>c</i>	01001
<i>d</i>	01110
<i>e</i>	110
<i>f</i>	0111100
<i>g</i>	0111110
<i>h</i>	0010010
<i>i</i>	1000
<i>j</i>	011111110
<i>k</i>	011111111001
<i>l</i>	0001
<i>m</i>	00101
<i>n</i>	1001
<i>o</i>	0000
<i>p</i>	01000
<i>q</i>	0111101
<i>r</i>	0101
<i>s</i>	1011
<i>t</i>	0110
<i>u</i>	0011
<i>v</i>	001000
<i>w</i>	011111111000
<i>x</i>	01111110
<i>y</i>	0111111111
<i>z</i>	01111111101
<i>espace</i>	111

Décoder avec le codage variable

100010010111100000001010010110100110100000001001
i n f o

lettre	codage variable
<i>a</i>	1010
<i>b</i>	0010011
<i>c</i>	01001
<i>d</i>	01110
<i>e</i>	110
<i>f</i>	0111100
<i>g</i>	0111110
<i>h</i>	0010010
<i>i</i>	1000
<i>j</i>	011111110
<i>k</i>	011111111001
<i>l</i>	0001
<i>m</i>	00101
<i>n</i>	1001
<i>o</i>	0000
<i>p</i>	01000
<i>q</i>	0111101
<i>r</i>	0101
<i>s</i>	1011
<i>t</i>	0110
<i>u</i>	0011
<i>v</i>	001000
<i>w</i>	011111111000
<i>x</i>	01111110
<i>y</i>	0111111111
<i>z</i>	01111111101
<i>espace</i>	111

Décoder avec le codage variable

100010010111100000001010010110100110100000001001
i n f o r

lettre	codage variable
<i>a</i>	1010
<i>b</i>	0010011
<i>c</i>	01001
<i>d</i>	01110
<i>e</i>	110
<i>f</i>	0111100
<i>g</i>	0111110
<i>h</i>	0010010
<i>i</i>	1000
<i>j</i>	011111110
<i>k</i>	011111111001
<i>l</i>	0001
<i>m</i>	00101
<i>n</i>	1001
<i>o</i>	0000
<i>p</i>	01000
<i>q</i>	0111101
<i>r</i>	0101
<i>s</i>	1011
<i>t</i>	0110
<i>u</i>	0011
<i>v</i>	001000
<i>w</i>	011111111000
<i>x</i>	01111110
<i>y</i>	0111111111
<i>z</i>	01111111101
<i>espace</i>	111

Décoder avec le codage variable

100010010111100000001010010110100110100000001001
i n f o r

lettre	codage variable
<i>a</i>	1010
<i>b</i>	0010011
<i>c</i>	01001
<i>d</i>	01110
<i>e</i>	110
<i>f</i>	0111100
<i>g</i>	0111110
<i>h</i>	0010010
<i>i</i>	1000
<i>j</i>	011111110
<i>k</i>	011111111001
<i>l</i>	0001
<i>m</i>	00101
<i>n</i>	1001
<i>o</i>	0000
<i>p</i>	01000
<i>q</i>	0111101
<i>r</i>	0101
<i>s</i>	1011
<i>t</i>	0110
<i>u</i>	0011
<i>v</i>	001000
<i>w</i>	011111111000
<i>x</i>	01111110
<i>y</i>	0111111111
<i>z</i>	01111111101
<i>espace</i>	111

Décoder avec le codage variable

100010010111100000001010010110100110100000001001
i n f o r m

lettre	codage variable
<i>a</i>	1010
<i>b</i>	0010011
<i>c</i>	01001
<i>d</i>	01110
<i>e</i>	110
<i>f</i>	0111100
<i>g</i>	0111110
<i>h</i>	0010010
<i>i</i>	1000
<i>j</i>	011111110
<i>k</i>	011111111001
<i>l</i>	0001
<i>m</i>	00101
<i>n</i>	1001
<i>o</i>	0000
<i>p</i>	01000
<i>q</i>	0111101
<i>r</i>	0101
<i>s</i>	1011
<i>t</i>	0110
<i>u</i>	0011
<i>v</i>	001000
<i>w</i>	011111111000
<i>x</i>	01111110
<i>y</i>	0111111111
<i>z</i>	01111111101
<i>espace</i>	111

Décoder avec le codage variable

100010010111000000010100101101001101001000000001001
i n f o r m

lettre	codage variable
<i>a</i>	1010
<i>b</i>	0010011
<i>c</i>	01001
<i>d</i>	01110
<i>e</i>	110
<i>f</i>	0111100
<i>g</i>	0111110
<i>h</i>	0010010
<i>i</i>	1000
<i>j</i>	011111110
<i>k</i>	011111111001
<i>l</i>	0001
<i>m</i>	00101
<i>n</i>	1001
<i>o</i>	0000
<i>p</i>	01000
<i>q</i>	0111101
<i>r</i>	0101
<i>s</i>	1011
<i>t</i>	0110
<i>u</i>	0011
<i>v</i>	001000
<i>w</i>	011111111000
<i>x</i>	01111110
<i>y</i>	0111111111
<i>z</i>	01111111101
<i>espace</i>	111

Décoder avec le codage variable

1000100101111000000010100101101001101000000001001
i n f o r m a

lettre	codage variable
<i>a</i>	1010
<i>b</i>	0010011
<i>c</i>	01001
<i>d</i>	01110
<i>e</i>	110
<i>f</i>	0111100
<i>g</i>	0111110
<i>h</i>	0010010
<i>i</i>	1000
<i>j</i>	011111110
<i>k</i>	011111111001
<i>l</i>	0001
<i>m</i>	00101
<i>n</i>	1001
<i>o</i>	0000
<i>p</i>	01000
<i>q</i>	0111101
<i>r</i>	0101
<i>s</i>	1011
<i>t</i>	0110
<i>u</i>	0011
<i>v</i>	001000
<i>w</i>	011111111000
<i>x</i>	01111110
<i>y</i>	0111111111
<i>z</i>	01111111101
<i>espace</i>	111

Décoder avec le codage variable

100010010111100000001010010110100110100000001001
i n f o r m a

lettre	codage variable
<i>a</i>	1010
<i>b</i>	0010011
<i>c</i>	01001
<i>d</i>	01110
<i>e</i>	110
<i>f</i>	0111100
<i>g</i>	0111110
<i>h</i>	0010010
<i>i</i>	1000
<i>j</i>	011111110
<i>k</i>	011111111001
<i>l</i>	0001
<i>m</i>	00101
<i>n</i>	1001
<i>o</i>	0000
<i>p</i>	01000
<i>q</i>	0111101
<i>r</i>	0101
<i>s</i>	1011
<i>t</i>	0110
<i>u</i>	0011
<i>v</i>	001000
<i>w</i>	011111111000
<i>x</i>	01111110
<i>y</i>	0111111111
<i>z</i>	01111111101
<i>espace</i>	111

Décoder avec le codage variable

100010010111100000001010010110100110100000001001
i n f o r m a t

lettre	codage variable
<i>a</i>	1010
<i>b</i>	0010011
<i>c</i>	01001
<i>d</i>	01110
<i>e</i>	110
<i>f</i>	0111100
<i>g</i>	0111110
<i>h</i>	0010010
<i>i</i>	1000
<i>j</i>	011111110
<i>k</i>	011111111001
<i>l</i>	0001
<i>m</i>	00101
<i>n</i>	1001
<i>o</i>	0000
<i>p</i>	01000
<i>q</i>	0111101
<i>r</i>	0101
<i>s</i>	1011
<i>t</i>	0110
<i>u</i>	0011
<i>v</i>	001000
<i>w</i>	011111111000
<i>x</i>	01111110
<i>y</i>	0111111111
<i>z</i>	01111111101
<i>espace</i>	111

Décoder avec le codage variable

100010010111100000001010010110100110100000001001
i n f o r m a t

lettre	codage variable
<i>a</i>	1010
<i>b</i>	0010011
<i>c</i>	01001
<i>d</i>	01110
<i>e</i>	110
<i>f</i>	0111100
<i>g</i>	0111110
<i>h</i>	0010010
<i>i</i>	1000
<i>j</i>	011111110
<i>k</i>	011111111001
<i>l</i>	0001
<i>m</i>	00101
<i>n</i>	1001
<i>o</i>	0000
<i>p</i>	01000
<i>q</i>	0111101
<i>r</i>	0101
<i>s</i>	1011
<i>t</i>	0110
<i>u</i>	0011
<i>v</i>	001000
<i>w</i>	011111111000
<i>x</i>	01111110
<i>y</i>	0111111111
<i>z</i>	01111111101
<i>espace</i>	111

Décoder avec le codage variable

100010010111100000001010010110100110100000001001
i n f o r m a t i

lettre	codage variable
<i>a</i>	1010
<i>b</i>	0010011
<i>c</i>	01001
<i>d</i>	01110
<i>e</i>	110
<i>f</i>	0111100
<i>g</i>	0111110
<i>h</i>	0010010
<i>i</i>	1000
<i>j</i>	011111110
<i>k</i>	011111111001
<i>l</i>	0001
<i>m</i>	00101
<i>n</i>	1001
<i>o</i>	0000
<i>p</i>	01000
<i>q</i>	0111101
<i>r</i>	0101
<i>s</i>	1011
<i>t</i>	0110
<i>u</i>	0011
<i>v</i>	001000
<i>w</i>	011111111000
<i>x</i>	01111110
<i>y</i>	0111111111
<i>z</i>	01111111101
<i>espace</i>	111

Décoder avec le codage variable

100010010111100000001010010110100110100000001001
i n f o r m a t i

lettre	codage variable
<i>a</i>	1010
<i>b</i>	0010011
<i>c</i>	01001
<i>d</i>	01110
<i>e</i>	110
<i>f</i>	0111100
<i>g</i>	0111110
<i>h</i>	0010010
<i>i</i>	1000
<i>j</i>	011111110
<i>k</i>	011111111001
<i>l</i>	0001
<i>m</i>	00101
<i>n</i>	1001
<i>o</i>	0000
<i>p</i>	01000
<i>q</i>	0111101
<i>r</i>	0101
<i>s</i>	1011
<i>t</i>	0110
<i>u</i>	0011
<i>v</i>	001000
<i>w</i>	011111111000
<i>x</i>	01111110
<i>y</i>	0111111111
<i>z</i>	01111111101
<i>espace</i>	111

Décoder avec le codage variable

100010010111100000001010010110100110100000001001
i n f o r m a t i o

lettre	codage variable
<i>a</i>	1010
<i>b</i>	0010011
<i>c</i>	01001
<i>d</i>	01110
<i>e</i>	110
<i>f</i>	0111100
<i>g</i>	0111110
<i>h</i>	0010010
<i>i</i>	1000
<i>j</i>	011111110
<i>k</i>	011111111001
<i>l</i>	0001
<i>m</i>	00101
<i>n</i>	1001
<i>o</i>	0000
<i>p</i>	01000
<i>q</i>	0111101
<i>r</i>	0101
<i>s</i>	1011
<i>t</i>	0110
<i>u</i>	0011
<i>v</i>	001000
<i>w</i>	011111111000
<i>x</i>	01111110
<i>y</i>	0111111111
<i>z</i>	01111111101
<i>espace</i>	111

Décoder avec le codage variable

100010010111100000001010010110100110100000001001

i n f o r m a t i o

lettre	codage variable
<i>a</i>	1010
<i>b</i>	0010011
<i>c</i>	01001
<i>d</i>	01110
<i>e</i>	110
<i>f</i>	0111100
<i>g</i>	0111110
<i>h</i>	0010010
<i>i</i>	1000
<i>j</i>	011111110
<i>k</i>	011111111001
<i>l</i>	0001
<i>m</i>	00101
<i>n</i>	1001
<i>o</i>	0000
<i>p</i>	01000
<i>q</i>	0111101
<i>r</i>	0101
<i>s</i>	1011
<i>t</i>	0110
<i>u</i>	0011
<i>v</i>	001000
<i>w</i>	011111111000
<i>x</i>	01111110
<i>y</i>	0111111111
<i>z</i>	01111111101
<i>espace</i>	111

Décoder avec le codage variable

100010010111100000001010010110100110100000001001
i n f o r m a t i o n

lettre	codage variable
<i>a</i>	1010
<i>b</i>	0010011
<i>c</i>	01001
<i>d</i>	01110
<i>e</i>	110
<i>f</i>	0111100
<i>g</i>	0111110
<i>h</i>	0010010
<i>i</i>	1000
<i>j</i>	011111110
<i>k</i>	011111111001
<i>l</i>	0001
<i>m</i>	00101
<i>n</i>	1001
<i>o</i>	0000
<i>p</i>	01000
<i>q</i>	0111101
<i>r</i>	0101
<i>s</i>	1011
<i>t</i>	0110
<i>u</i>	0011
<i>v</i>	001000
<i>w</i>	011111111000
<i>x</i>	01111110
<i>y</i>	0111111111
<i>z</i>	01111111101
<i>espace</i>	111

Décoder avec le codage variable

100010010111100000001010010110100110100000001001
i n f o r m a t i o n

Pourquoi n'a-t-on jamais le choix du décodage ?

lettre	codage variable
<i>a</i>	1010
<i>b</i>	0010011
<i>c</i>	01001
<i>d</i>	01110
<i>e</i>	110
<i>f</i>	0111100
<i>g</i>	0111110
<i>h</i>	0010010
<i>i</i>	1000
<i>j</i>	011111110
<i>k</i>	011111111001
<i>l</i>	0001
<i>m</i>	00101
<i>n</i>	1001
<i>o</i>	0000
<i>p</i>	01000
<i>q</i>	0111101
<i>r</i>	0101
<i>s</i>	1011
<i>t</i>	0110
<i>u</i>	0011
<i>v</i>	001000
<i>w</i>	011111111000
<i>x</i>	01111110
<i>y</i>	0111111111
<i>z</i>	01111111101
<i>espace</i>	111

Décoder avec le codage variable

100010010111100000001010010110100110100000001001
i n f o r m a t i o n

Pourquoi n'a-t-on jamais le choix du décodage ?

Aucun code n'est le **préfixe** d'un autre...

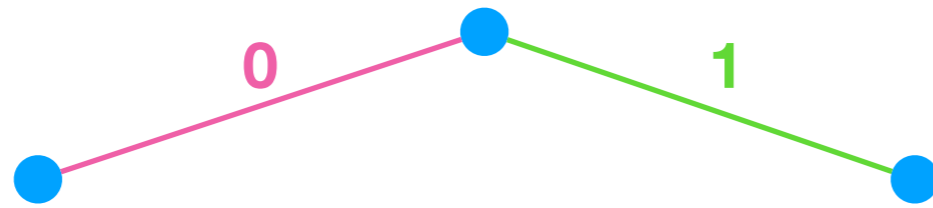
lettre	codage variable
<i>a</i>	1010
<i>b</i>	0010011
<i>c</i>	01001
<i>d</i>	01110
<i>e</i>	110
<i>f</i>	0111100
<i>g</i>	0111110
<i>h</i>	0010010
<i>i</i>	1000
<i>j</i>	011111110
<i>k</i>	011111111001
<i>l</i>	0001
<i>m</i>	00101
<i>n</i>	1001
<i>o</i>	0000
<i>p</i>	01000
<i>q</i>	0111101
<i>r</i>	0101
<i>s</i>	1011
<i>t</i>	0110
<i>u</i>	0011
<i>v</i>	001000
<i>w</i>	011111111000
<i>x</i>	01111110
<i>y</i>	0111111111
<i>z</i>	01111111101
<i>espace</i>	111

Arbre de codage variable



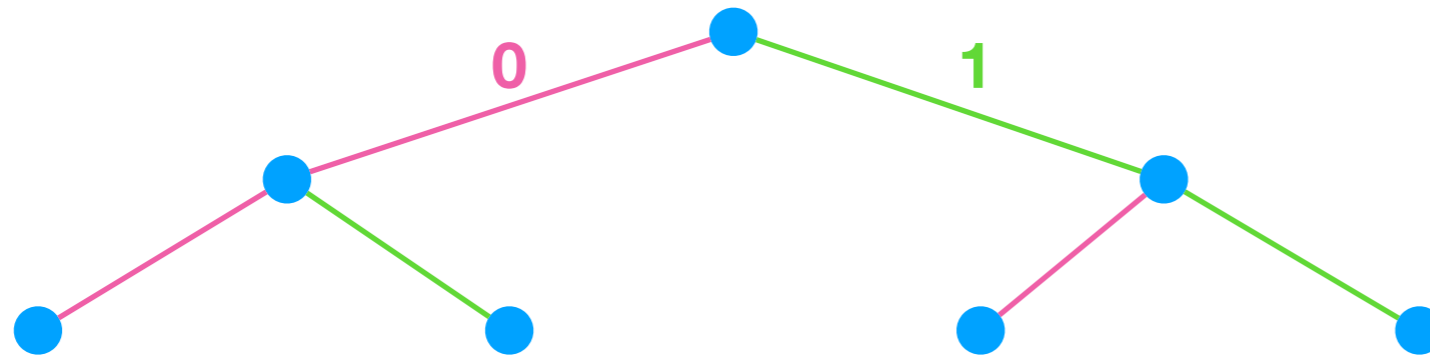
lettre	codage variable
<i>a</i>	1010
<i>b</i>	0010011
<i>c</i>	01001
<i>d</i>	01110
<i>e</i>	110
<i>f</i>	0111100
<i>g</i>	0111110
<i>h</i>	0010010
<i>i</i>	1000
<i>j</i>	011111110
<i>k</i>	011111111001
<i>l</i>	0001
<i>m</i>	00101
<i>n</i>	1001
<i>o</i>	0000
<i>p</i>	01000
<i>q</i>	0111101
<i>r</i>	0101
<i>s</i>	1011
<i>t</i>	0110
<i>u</i>	0011
<i>v</i>	001000
<i>w</i>	011111111000
<i>x</i>	01111110
<i>y</i>	0111111111
<i>z</i>	01111111101
<i>espace</i>	111

Arbre de codage variable



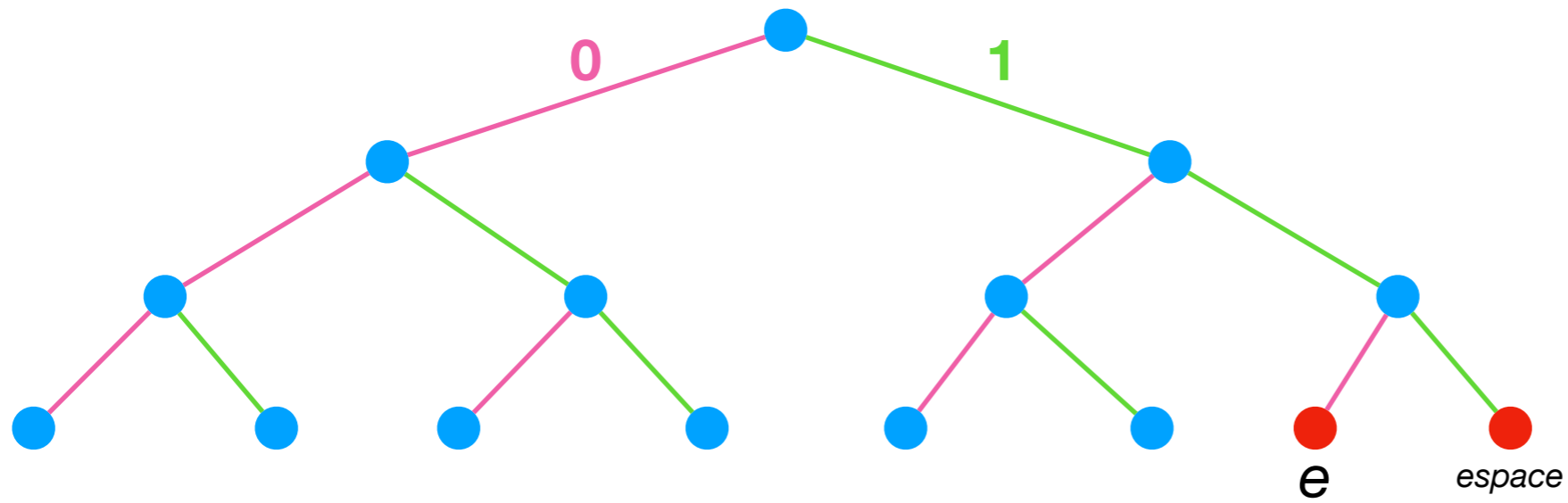
lettre	codage variable
<i>a</i>	1010
<i>b</i>	0010011
<i>c</i>	01001
<i>d</i>	01110
<i>e</i>	110
<i>f</i>	0111100
<i>g</i>	0111110
<i>h</i>	0010010
<i>i</i>	1000
<i>j</i>	011111110
<i>k</i>	011111111001
<i>l</i>	0001
<i>m</i>	00101
<i>n</i>	1001
<i>o</i>	0000
<i>p</i>	01000
<i>q</i>	0111101
<i>r</i>	0101
<i>s</i>	1011
<i>t</i>	0110
<i>u</i>	0011
<i>v</i>	001000
<i>w</i>	011111111000
<i>x</i>	01111110
<i>y</i>	0111111111
<i>z</i>	01111111101
<i>espace</i>	111

Arbre de codage variable



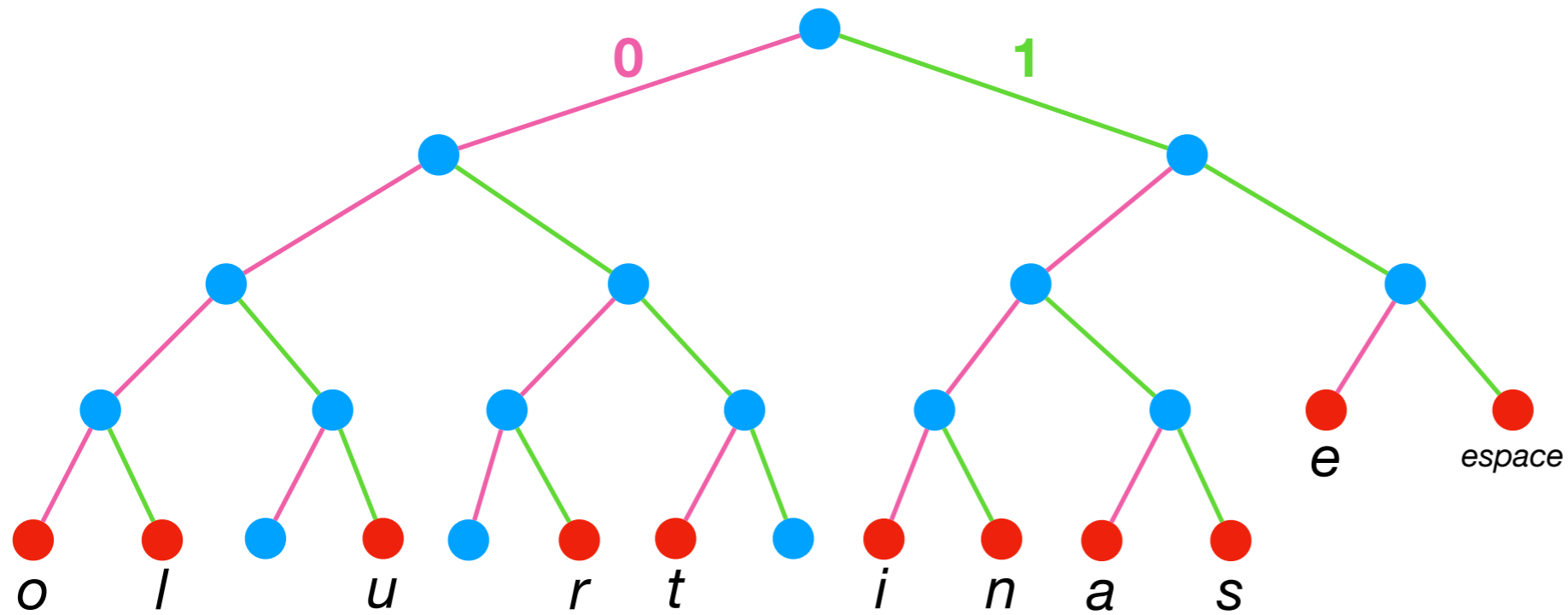
lettre	codage variable
<i>a</i>	1010
<i>b</i>	0010011
<i>c</i>	01001
<i>d</i>	01110
<i>e</i>	110
<i>f</i>	0111100
<i>g</i>	0111110
<i>h</i>	0010010
<i>i</i>	1000
<i>j</i>	011111110
<i>k</i>	011111111001
<i>l</i>	0001
<i>m</i>	00101
<i>n</i>	1001
<i>o</i>	0000
<i>p</i>	01000
<i>q</i>	0111101
<i>r</i>	0101
<i>s</i>	1011
<i>t</i>	0110
<i>u</i>	0011
<i>v</i>	001000
<i>w</i>	011111111000
<i>x</i>	01111110
<i>y</i>	0111111111
<i>z</i>	01111111101
<i>espace</i>	111

Arbre de codage variable



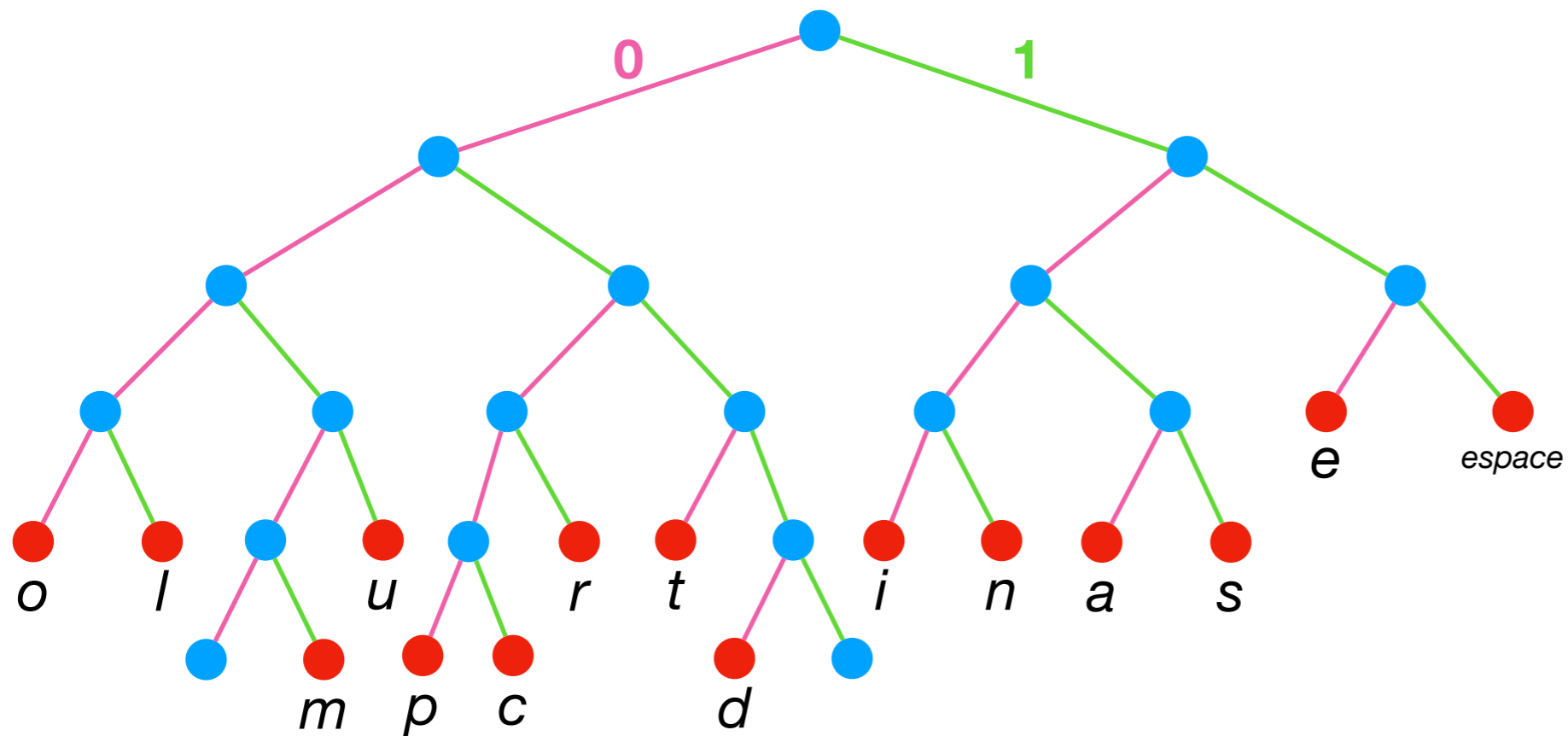
lettre	codage variable
<i>a</i>	1010
<i>b</i>	0010011
<i>c</i>	01001
<i>d</i>	01110
<i>e</i>	110
<i>f</i>	0111100
<i>g</i>	0111110
<i>h</i>	0010010
<i>i</i>	1000
<i>j</i>	011111110
<i>k</i>	011111111001
<i>l</i>	0001
<i>m</i>	00101
<i>n</i>	1001
<i>o</i>	0000
<i>p</i>	01000
<i>q</i>	0111101
<i>r</i>	0101
<i>s</i>	1011
<i>t</i>	0110
<i>u</i>	0011
<i>v</i>	001000
<i>w</i>	011111111000
<i>x</i>	01111110
<i>y</i>	0111111111
<i>z</i>	01111111101
<i>espace</i>	111

Arbre de codage variable



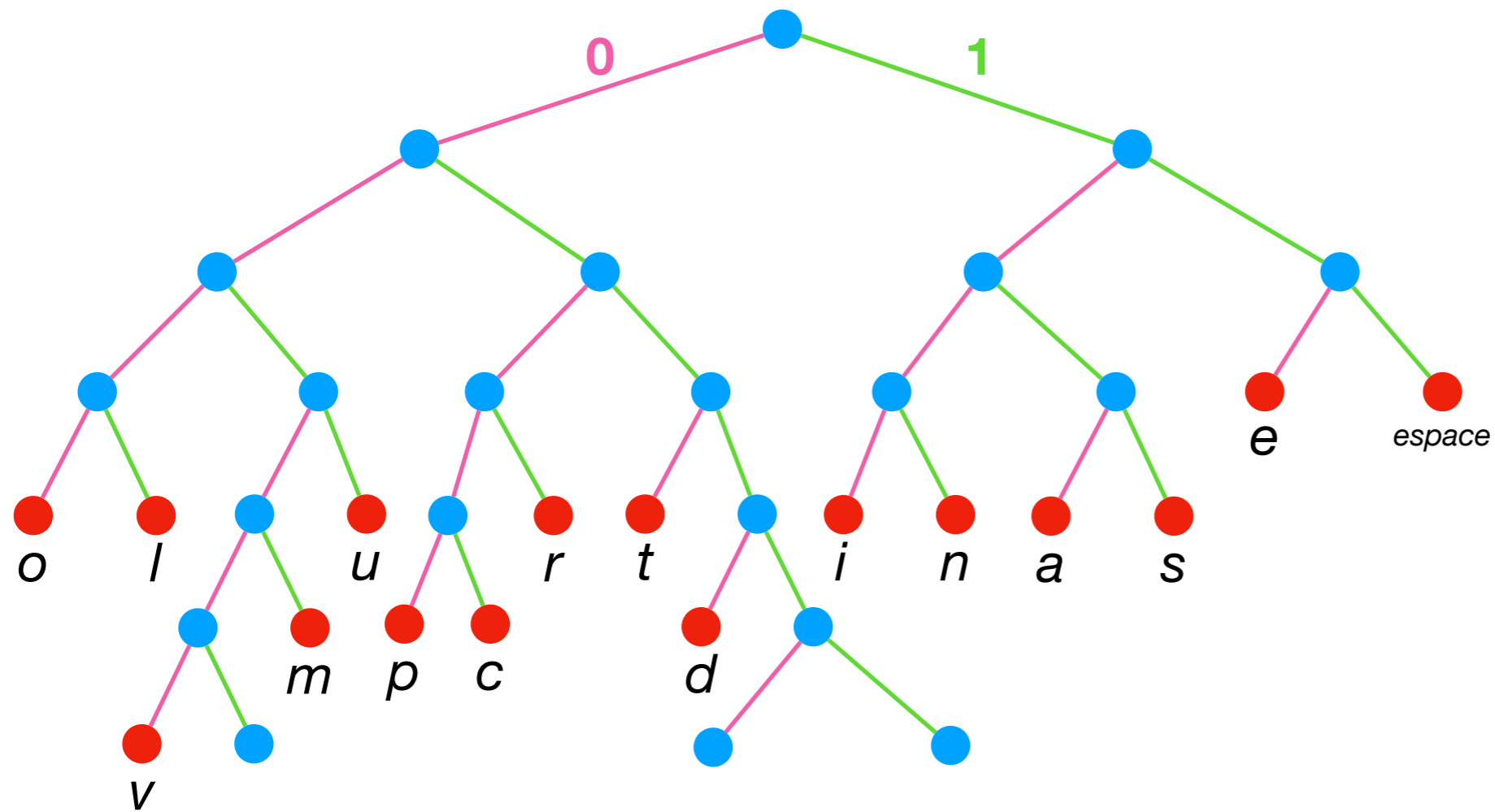
lettre	codage variable
<i>a</i>	1010
<i>b</i>	0010011
<i>c</i>	01001
<i>d</i>	01110
<i>e</i>	110
<i>f</i>	0111100
<i>g</i>	0111110
<i>h</i>	0010010
<i>i</i>	1000
<i>j</i>	011111110
<i>k</i>	011111111001
<i>l</i>	0001
<i>m</i>	00101
<i>n</i>	1001
<i>o</i>	0000
<i>p</i>	01000
<i>q</i>	0111101
<i>r</i>	0101
<i>s</i>	1011
<i>t</i>	0110
<i>u</i>	0011
<i>v</i>	001000
<i>w</i>	011111111000
<i>x</i>	01111110
<i>y</i>	0111111111
<i>z</i>	01111111101
<i>espace</i>	111

Arbre de codage variable



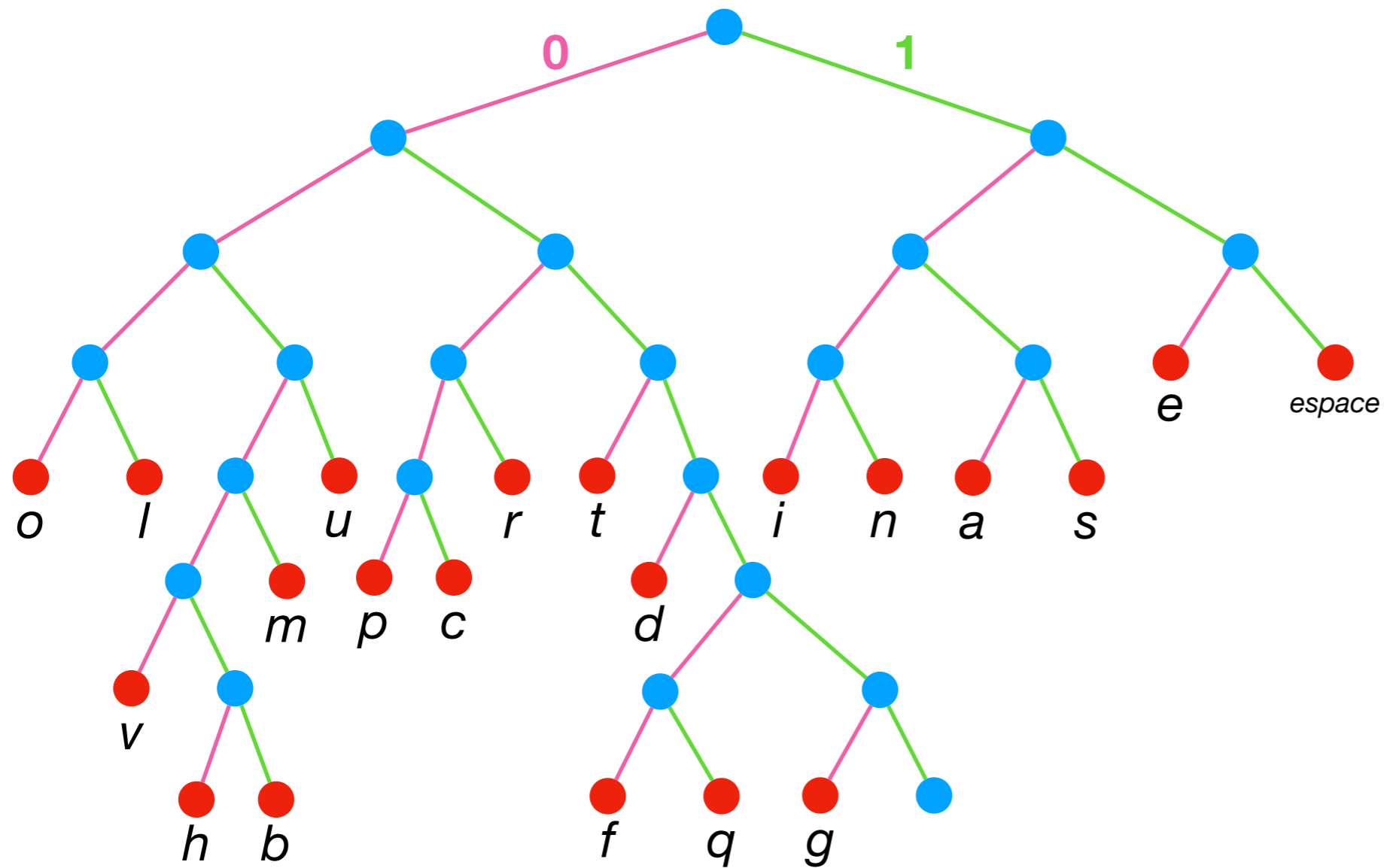
lettre	codage variable
<i>a</i>	1010
<i>b</i>	0010011
<i>c</i>	01001
<i>d</i>	01110
<i>e</i>	110
<i>f</i>	0111100
<i>g</i>	0111110
<i>h</i>	0010010
<i>i</i>	1000
<i>j</i>	011111110
<i>k</i>	011111111001
<i>l</i>	0001
<i>m</i>	00101
<i>n</i>	1001
<i>o</i>	0000
<i>p</i>	01000
<i>q</i>	0111101
<i>r</i>	0101
<i>s</i>	1011
<i>t</i>	0110
<i>u</i>	0011
<i>v</i>	001000
<i>w</i>	011111111000
<i>x</i>	01111110
<i>y</i>	0111111111
<i>z</i>	01111111101
<i>espace</i>	111

Arbre de codage variable



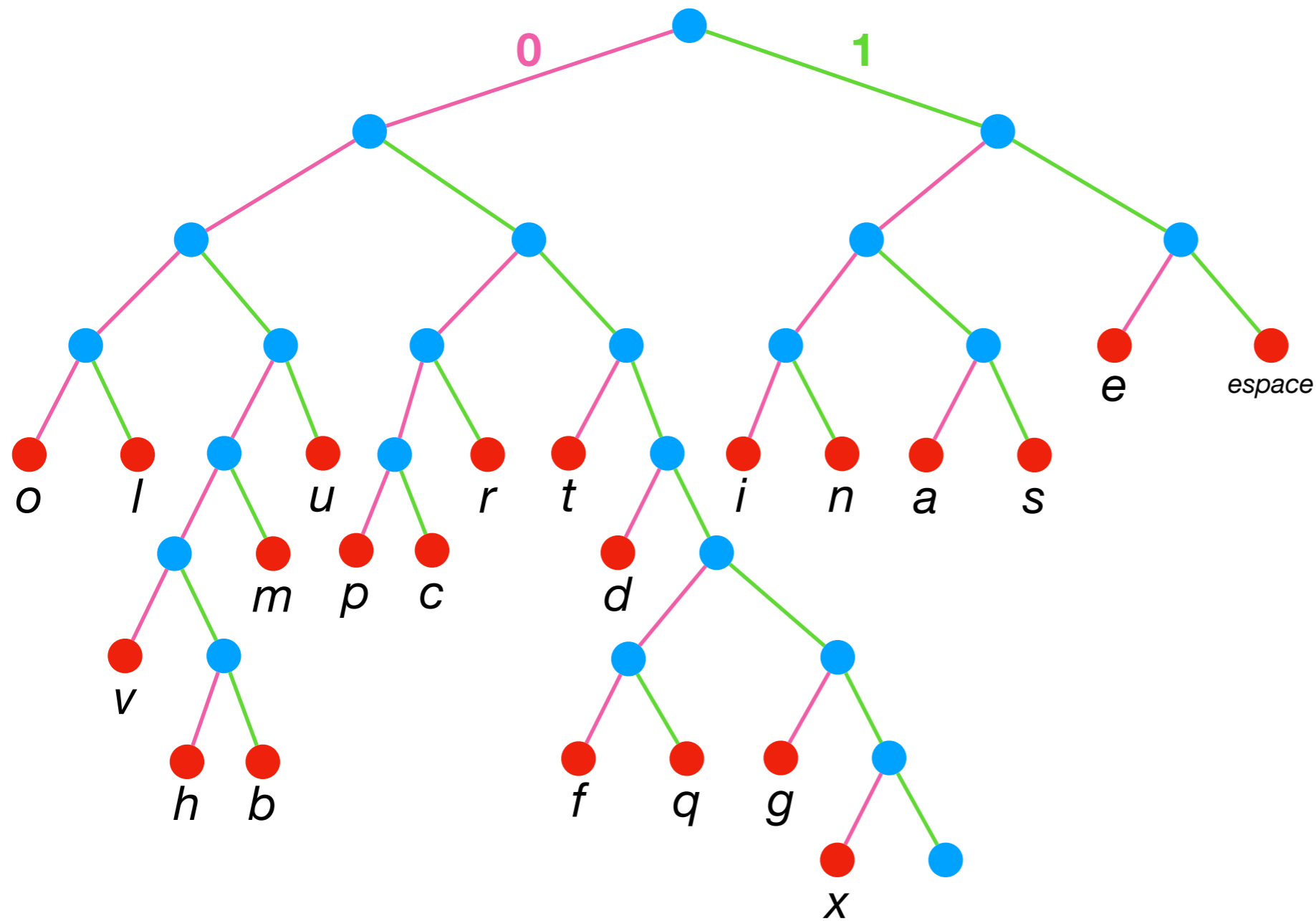
lettre	codage variable
<i>a</i>	1010
<i>b</i>	0010011
<i>c</i>	01001
<i>d</i>	01110
<i>e</i>	110
<i>f</i>	0111100
<i>g</i>	0111110
<i>h</i>	0010010
<i>i</i>	1000
<i>j</i>	011111110
<i>k</i>	011111111001
<i>l</i>	0001
<i>m</i>	00101
<i>n</i>	1001
<i>o</i>	0000
<i>p</i>	01000
<i>q</i>	0111101
<i>r</i>	0101
<i>s</i>	1011
<i>t</i>	0110
<i>u</i>	0011
<i>v</i>	001000
<i>w</i>	011111111000
<i>x</i>	01111110
<i>y</i>	0111111111
<i>z</i>	01111111101
<i>espace</i>	111

Arbre de codage variable



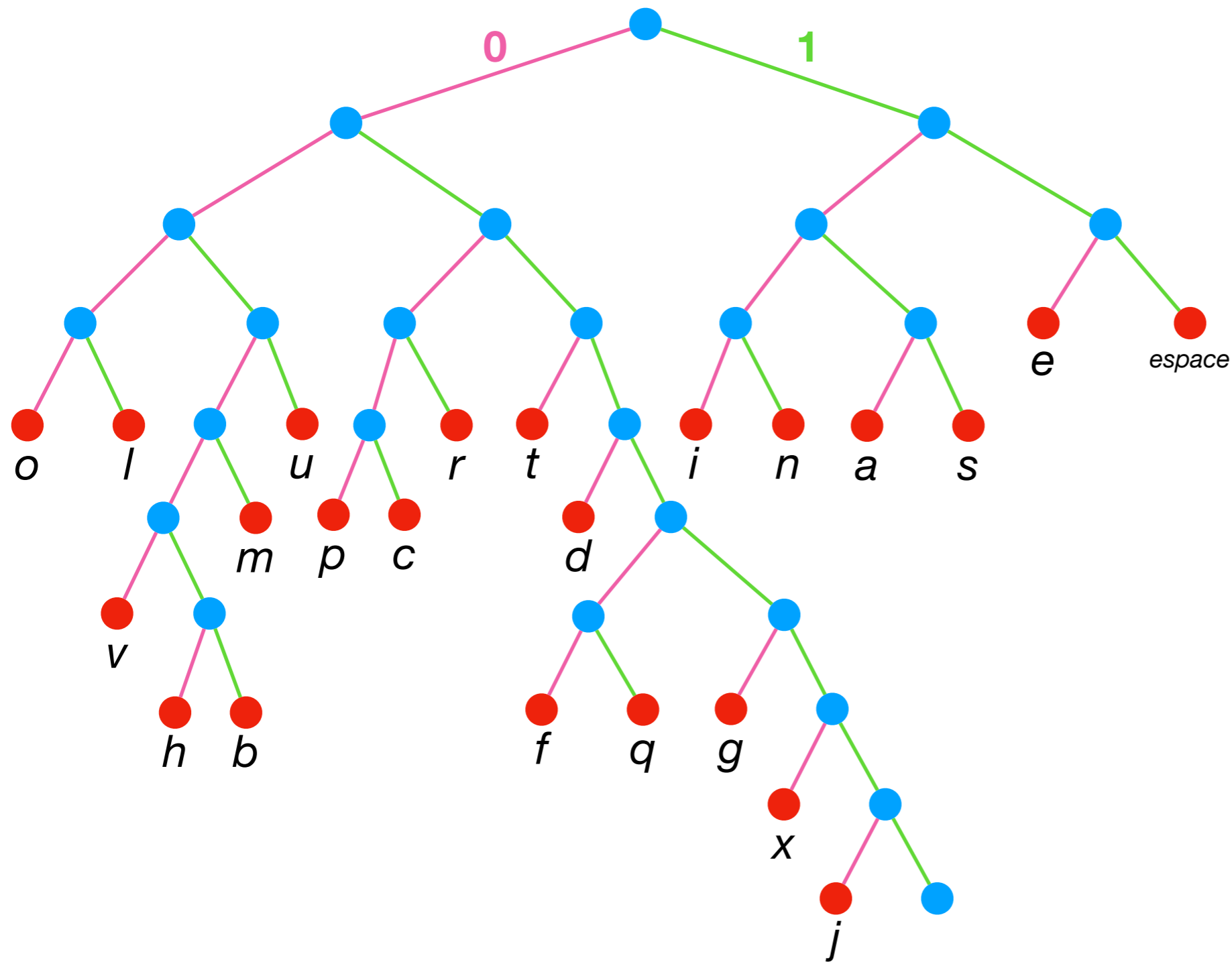
lettre	codage variable
<i>a</i>	1010
<i>b</i>	0010011
<i>c</i>	01001
<i>d</i>	01110
<i>e</i>	110
<i>f</i>	0111100
<i>g</i>	0111110
<i>h</i>	0010010
<i>i</i>	1000
<i>j</i>	011111110
<i>k</i>	011111111001
<i>l</i>	0001
<i>m</i>	00101
<i>n</i>	1001
<i>o</i>	0000
<i>p</i>	01000
<i>q</i>	0111101
<i>r</i>	0101
<i>s</i>	1011
<i>t</i>	0110
<i>u</i>	0011
<i>v</i>	001000
<i>w</i>	011111111000
<i>x</i>	01111110
<i>y</i>	0111111111
<i>z</i>	01111111101
<i>espace</i>	111

Arbre de codage variable



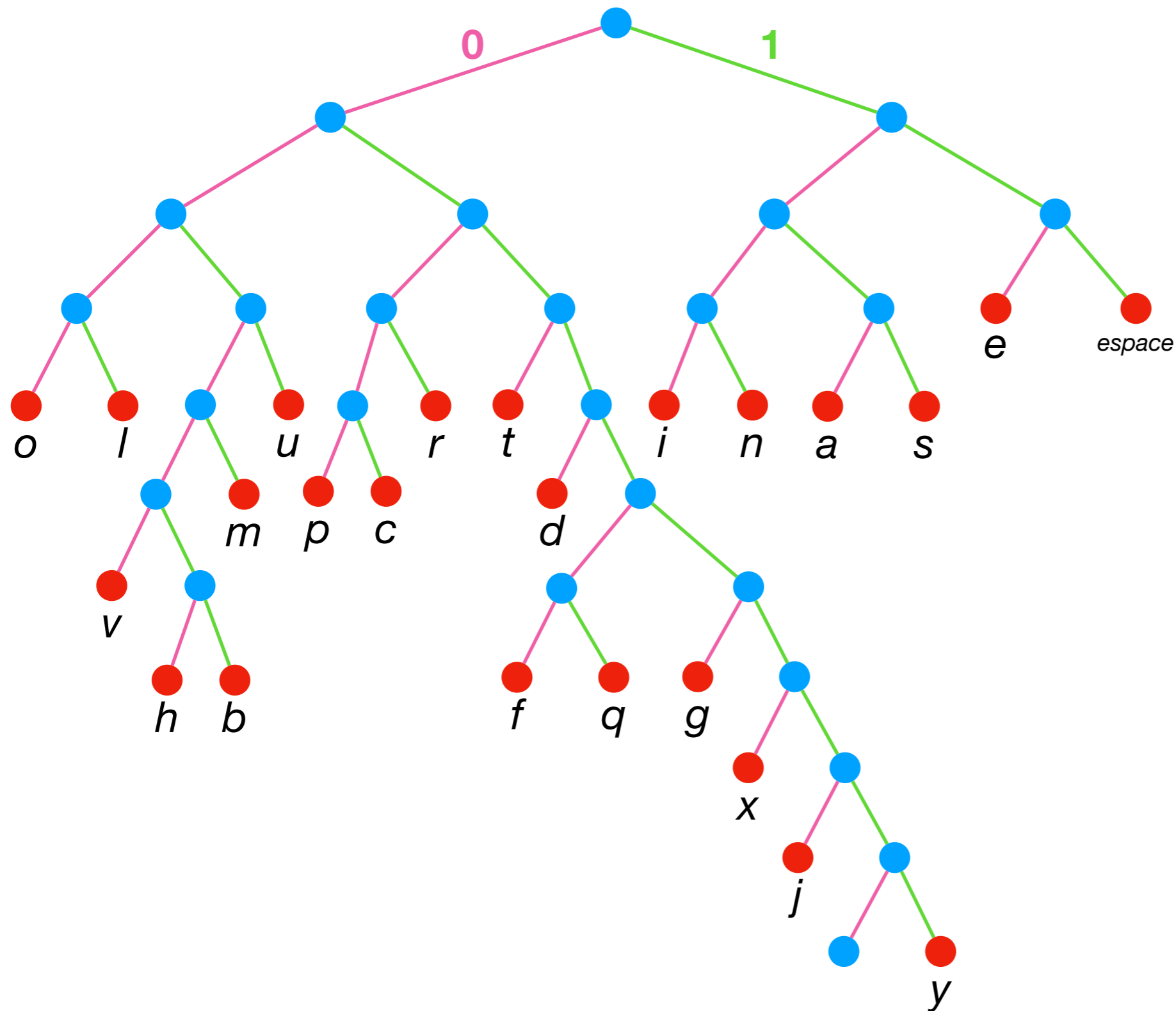
lettre	codage variable
<i>a</i>	1010
<i>b</i>	0010011
<i>c</i>	01001
<i>d</i>	01110
<i>e</i>	110
<i>f</i>	0111100
<i>g</i>	0111110
<i>h</i>	0010010
<i>i</i>	1000
<i>j</i>	011111110
<i>k</i>	011111111001
<i>l</i>	0001
<i>m</i>	00101
<i>n</i>	1001
<i>o</i>	0000
<i>p</i>	01000
<i>q</i>	0111101
<i>r</i>	0101
<i>s</i>	1011
<i>t</i>	0110
<i>u</i>	0011
<i>v</i>	001000
<i>w</i>	011111111000
<i>x</i>	01111110
<i>y</i>	0111111111
<i>z</i>	01111111101
<i>espace</i>	111

Arbre de codage variable



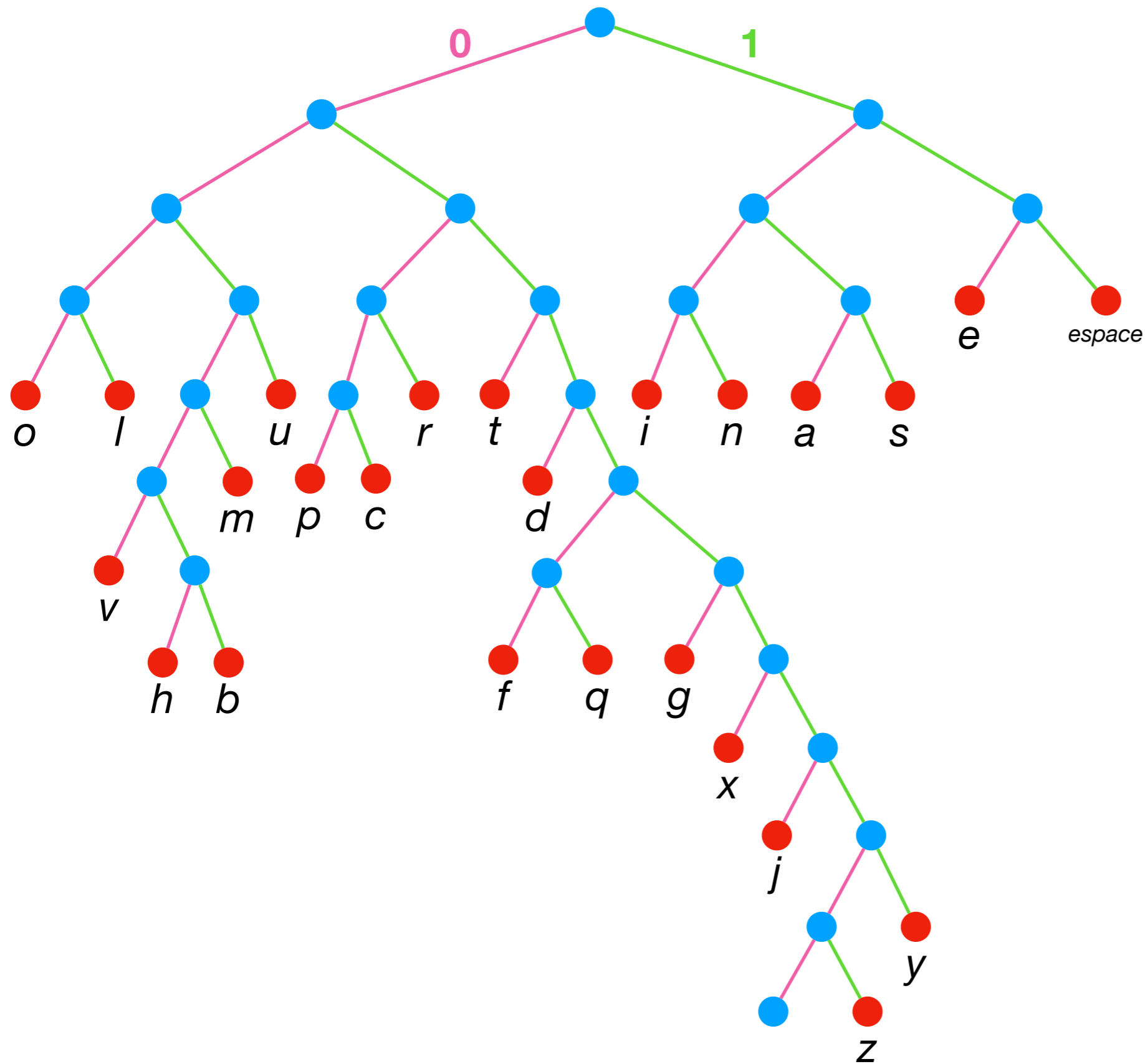
lettre	codage variable
<i>a</i>	1010
<i>b</i>	0010011
<i>c</i>	01001
<i>d</i>	01110
<i>e</i>	110
<i>f</i>	0111100
<i>g</i>	0111110
<i>h</i>	0010010
<i>i</i>	1000
<i>j</i>	011111110
<i>k</i>	011111111001
<i>l</i>	0001
<i>m</i>	00101
<i>n</i>	1001
<i>o</i>	0000
<i>p</i>	01000
<i>q</i>	0111101
<i>r</i>	0101
<i>s</i>	1011
<i>t</i>	0110
<i>u</i>	0011
<i>v</i>	001000
<i>w</i>	011111111000
<i>x</i>	01111110
<i>y</i>	0111111111
<i>z</i>	01111111101
<i>espace</i>	111

Arbre de codage variable



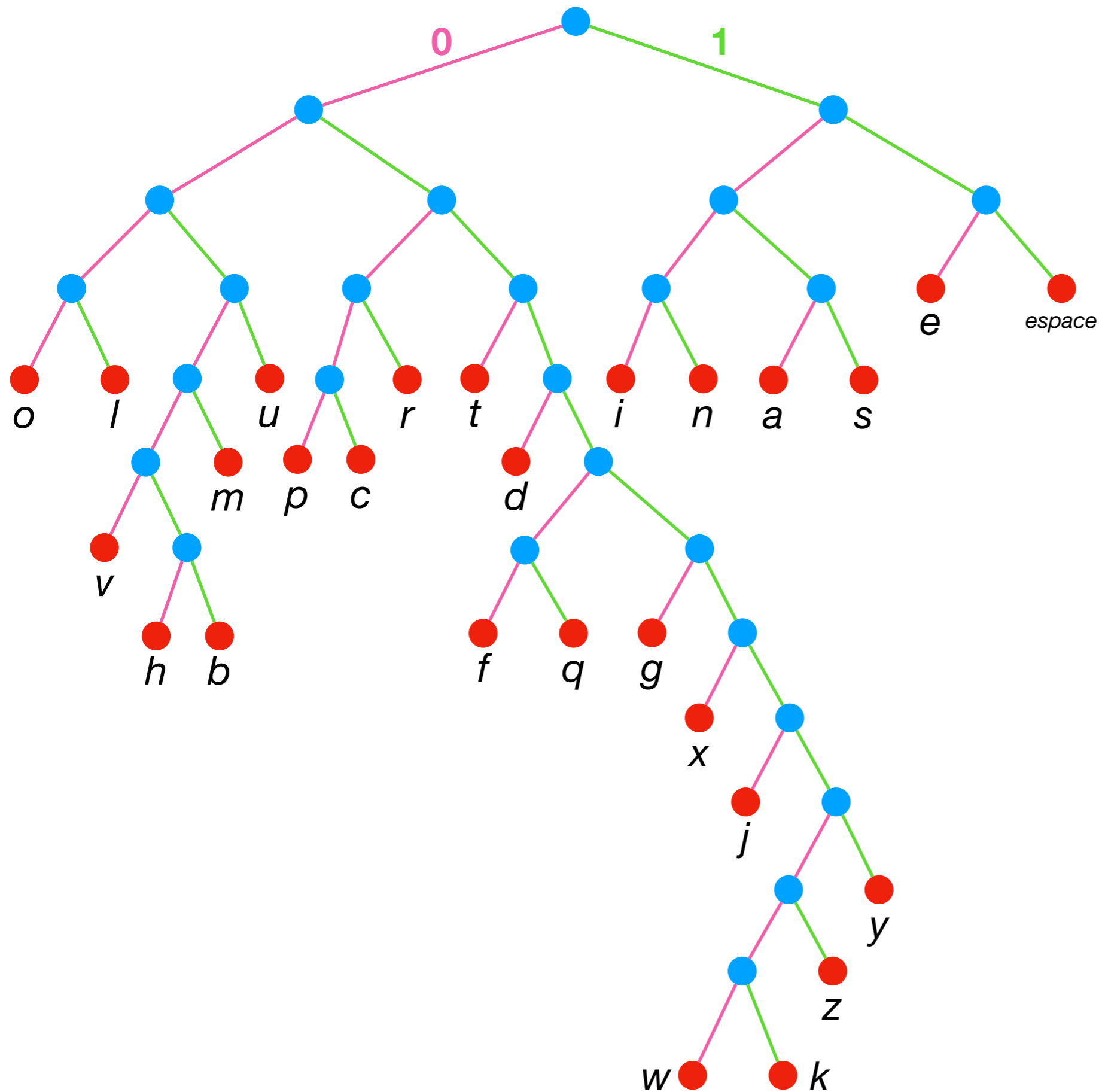
lettre	codage variable
<i>a</i>	1010
<i>b</i>	0010011
<i>c</i>	01001
<i>d</i>	01110
<i>e</i>	110
<i>f</i>	0111100
<i>g</i>	0111110
<i>h</i>	0010010
<i>i</i>	1000
<i>j</i>	011111110
<i>k</i>	011111111001
<i>l</i>	0001
<i>m</i>	00101
<i>n</i>	1001
<i>o</i>	0000
<i>p</i>	01000
<i>q</i>	0111101
<i>r</i>	0101
<i>s</i>	1011
<i>t</i>	0110
<i>u</i>	0011
<i>v</i>	001000
<i>w</i>	011111111000
<i>x</i>	01111110
<i>y</i>	0111111111
<i>z</i>	01111111101
<i>espace</i>	111

Arbre de codage variable



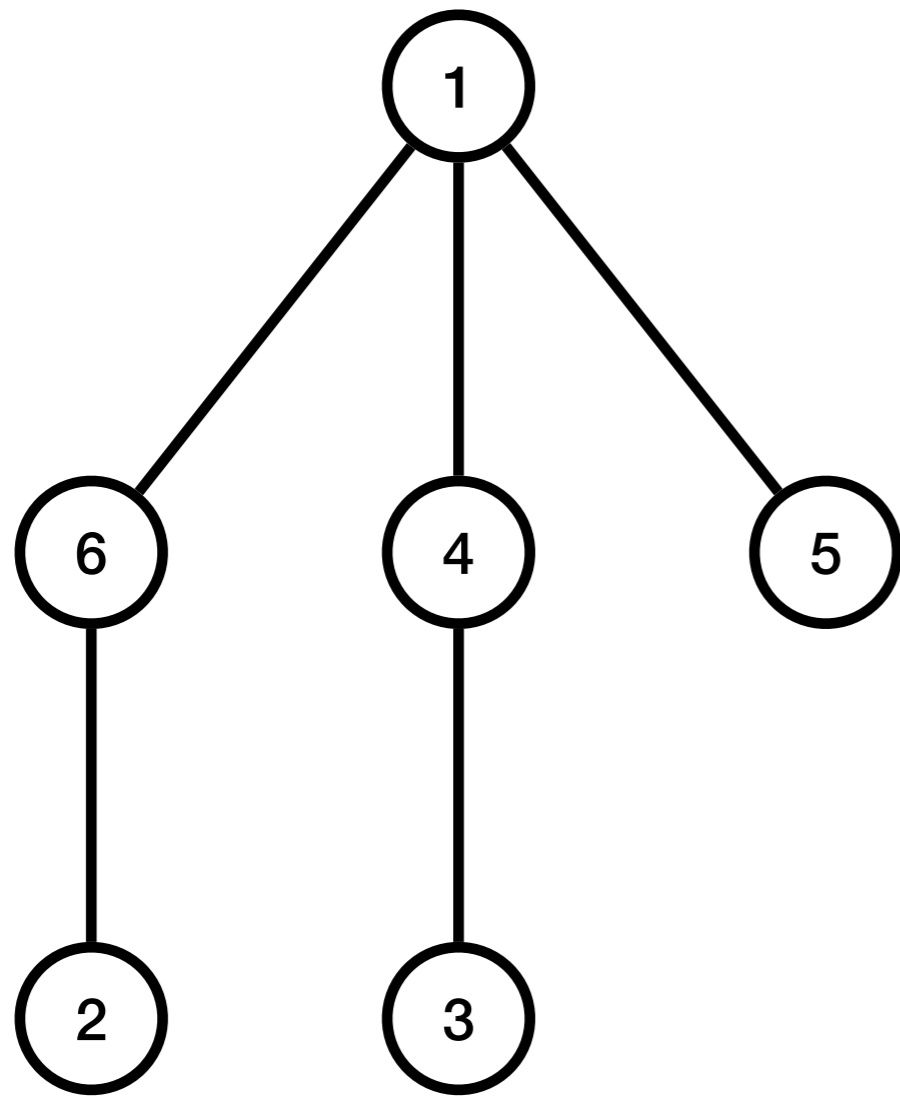
lettre	codage variable
<i>a</i>	1010
<i>b</i>	0010011
<i>c</i>	01001
<i>d</i>	01110
<i>e</i>	110
<i>f</i>	0111100
<i>g</i>	0111110
<i>h</i>	0010010
<i>i</i>	1000
<i>j</i>	011111110
<i>k</i>	011111111001
<i>l</i>	0001
<i>m</i>	00101
<i>n</i>	1001
<i>o</i>	0000
<i>p</i>	01000
<i>q</i>	0111101
<i>r</i>	0101
<i>s</i>	1011
<i>t</i>	0110
<i>u</i>	0011
<i>v</i>	001000
<i>w</i>	011111111000
<i>x</i>	01111110
<i>y</i>	0111111111
<i>z</i>	01111111101
<i>espace</i>	111

Arbre de codage variable



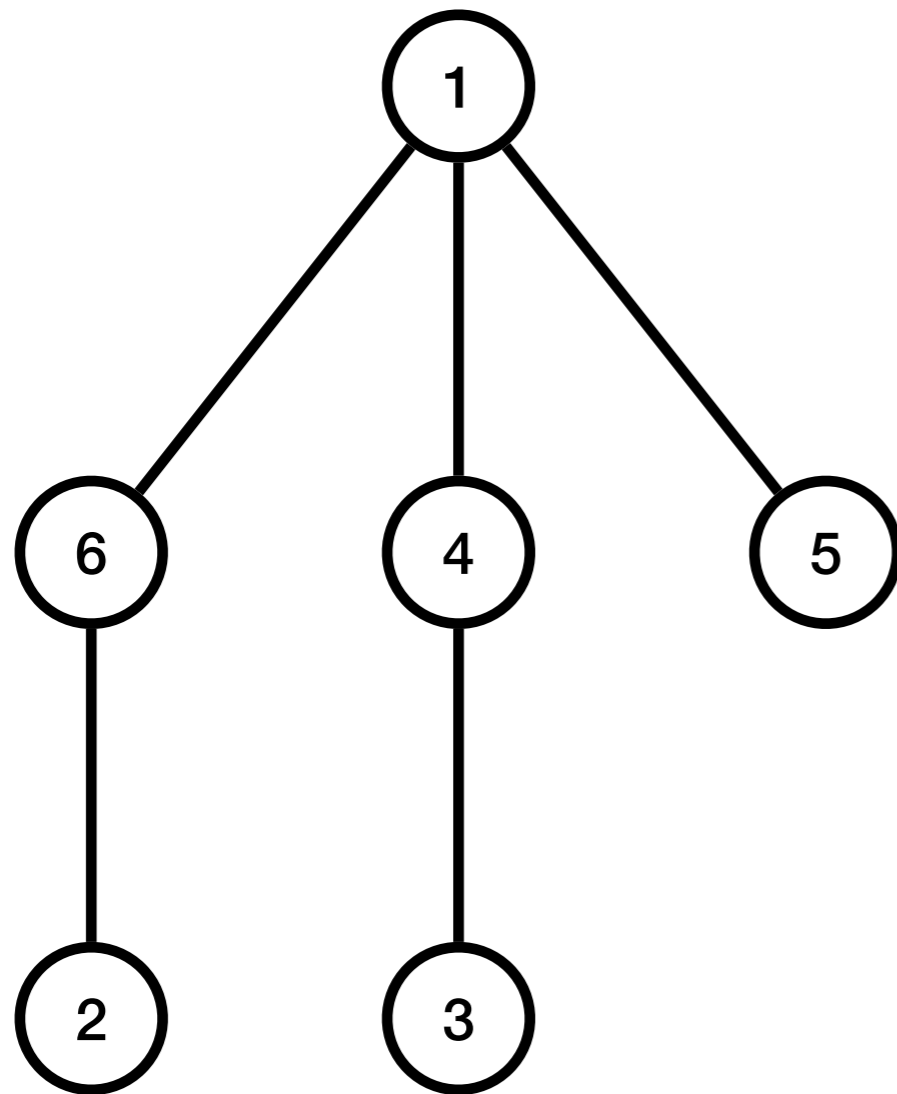
lettre	codage variable
<i>a</i>	1010
<i>b</i>	0010011
<i>c</i>	01001
<i>d</i>	01110
<i>e</i>	110
<i>f</i>	0111100
<i>g</i>	0111110
<i>h</i>	0010010
<i>i</i>	1000
<i>j</i>	011111110
<i>k</i>	011111111001
<i>l</i>	0001
<i>m</i>	00101
<i>n</i>	1001
<i>o</i>	0000
<i>p</i>	01000
<i>q</i>	0111101
<i>r</i>	0101
<i>s</i>	1011
<i>t</i>	0110
<i>u</i>	0011
<i>v</i>	001000
<i>w</i>	011111111000
<i>x</i>	01111110
<i>y</i>	0111111111
<i>z</i>	01111111101
<i>espace</i>	111

Représentation d'arbres en tant que graphes



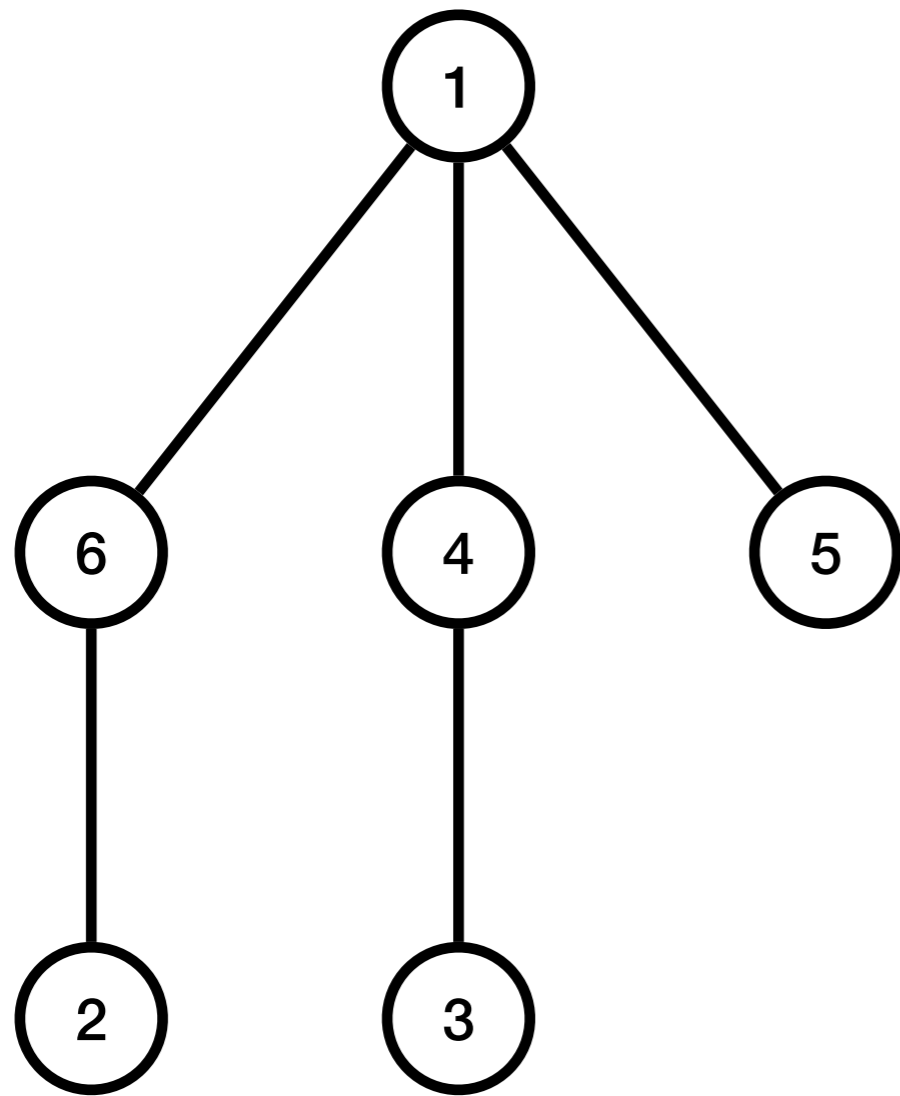
	1	2	3	4	5	6
1						
2						
3						
4						
5						
6						

Représentation d'arbres en tant que graphes



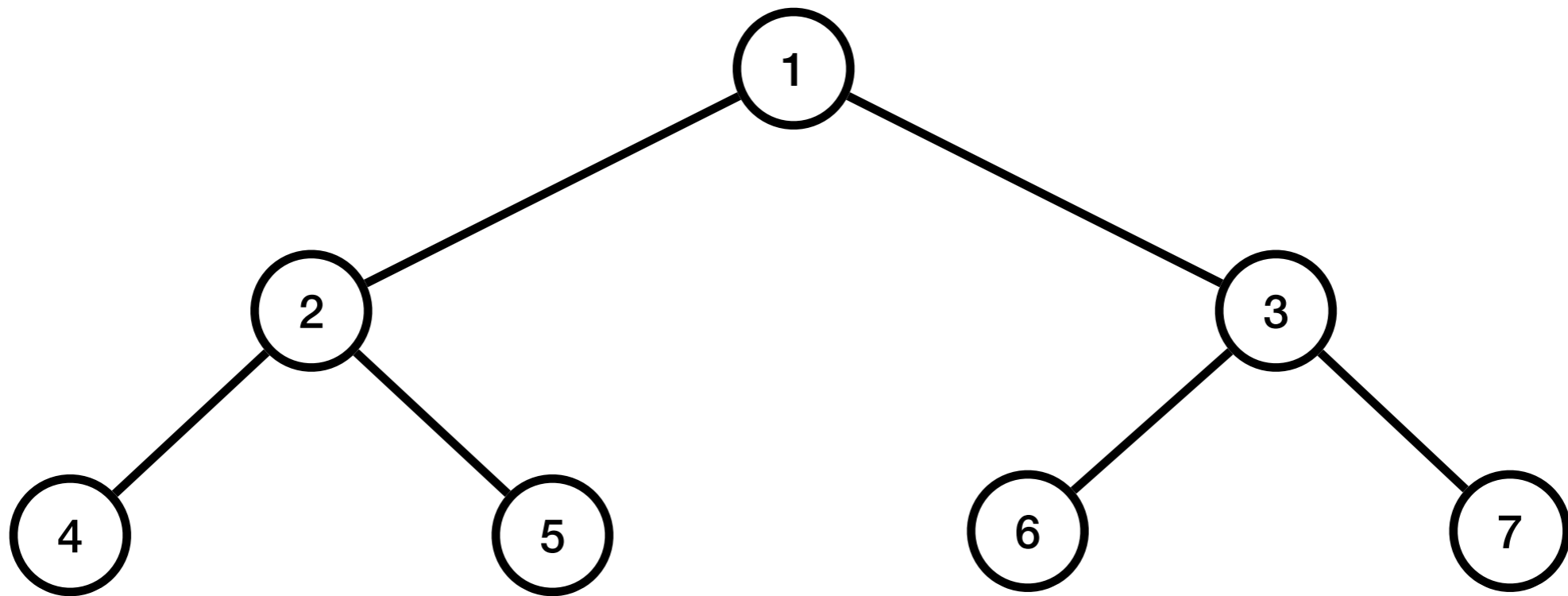
	1	2	3	4	5	6
1				1	1	1
2						1
3				1		
4	1		1			
5	1					
6	1	1				

Représentation d'arbres en tant que graphes

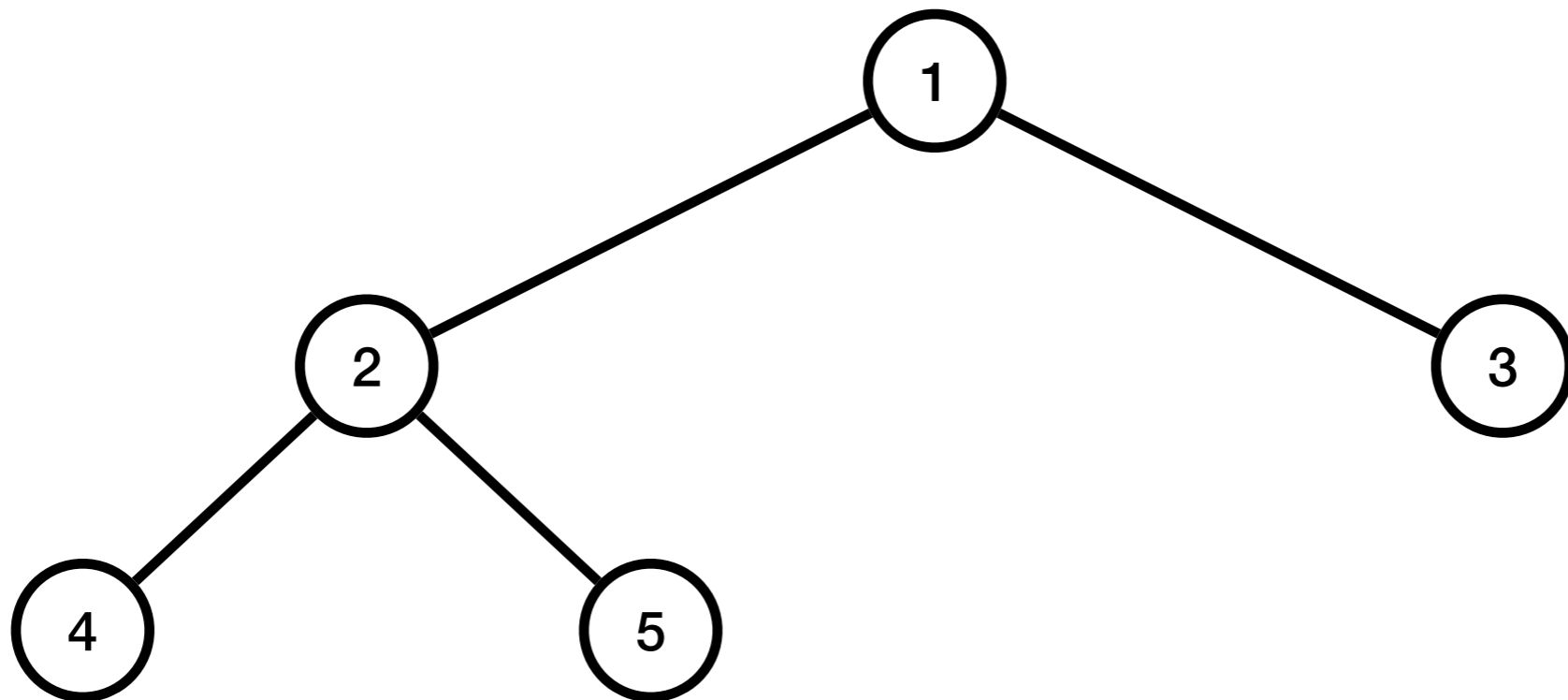


	1	2	3	4	5	6
1	0	0	0	1	1	1
2	0	0	0	0	0	1
3	0	0	0	1	0	0
4	1	0	1	0	0	0
5	1	0	0	0	0	0
6	1	1	0	0	0	0

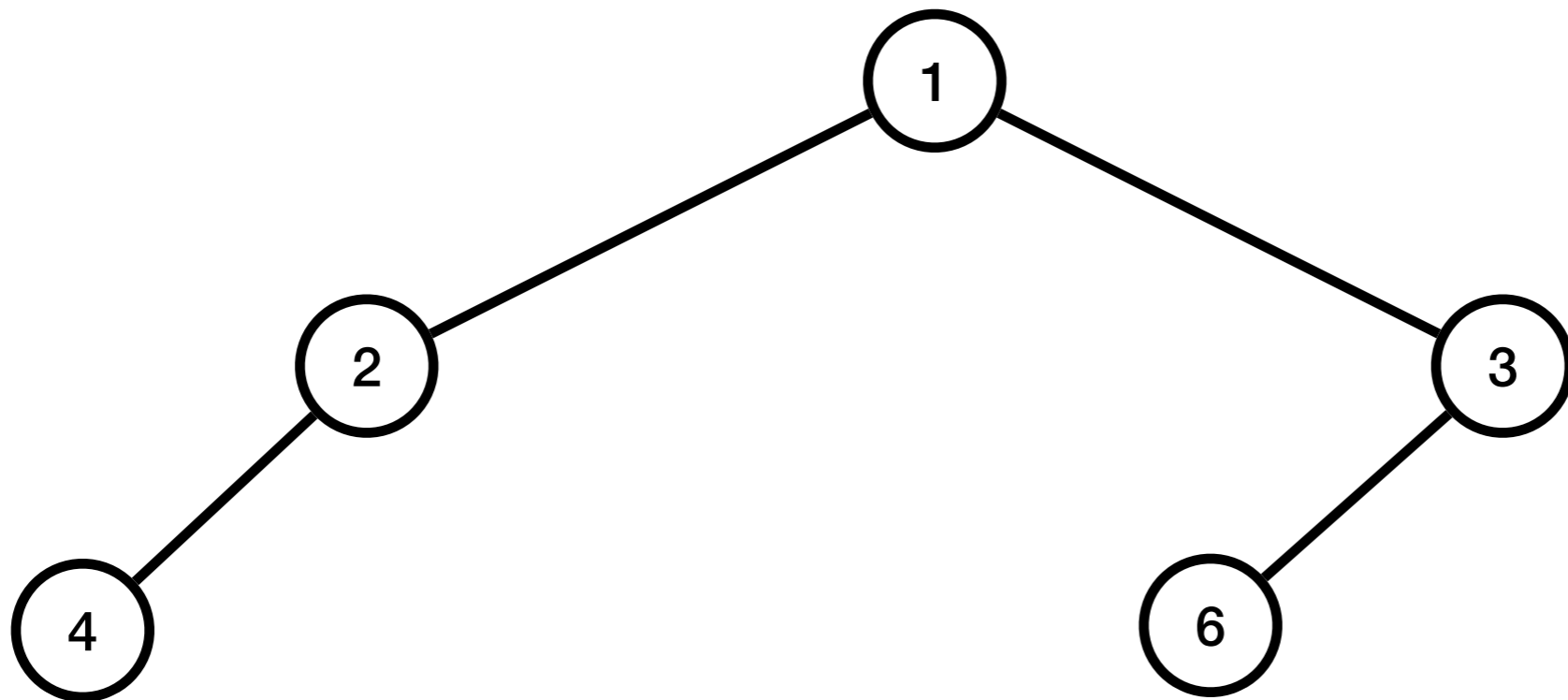
Arbres binaires



Arbres binaires



Arbres binaires



Arbres binaires : définition récursive

Un arbre binaire est :

- soit l'arbre vide, qu'on dénote par **nil** (None en Python)
- soit un nœud avec une certaine valeur (par exemple, un nombre entier), un sous-arbre gauche et un sous-arbre droit ; les sous-arbres sont, à leur tour, des arbres binaires

Exemples d'arbres binaires

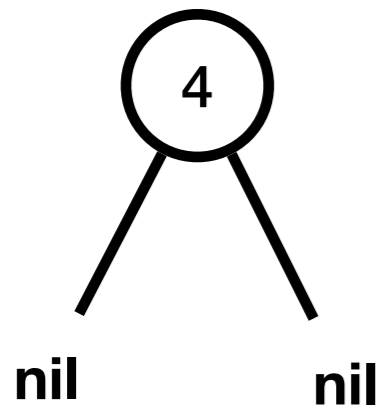
nil

Exemples d'arbres binaires

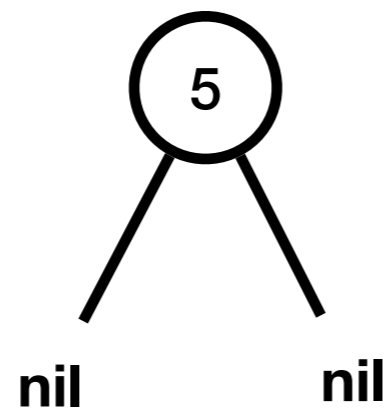
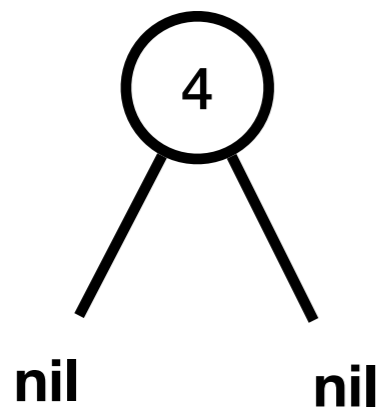
nil

nil

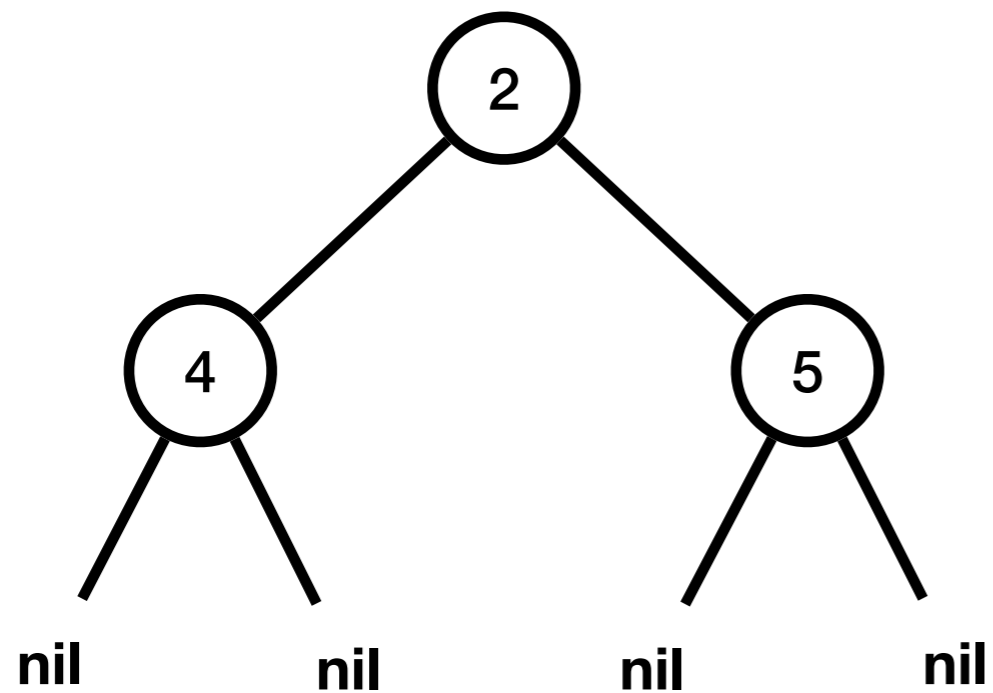
Exemples d'arbres binaires



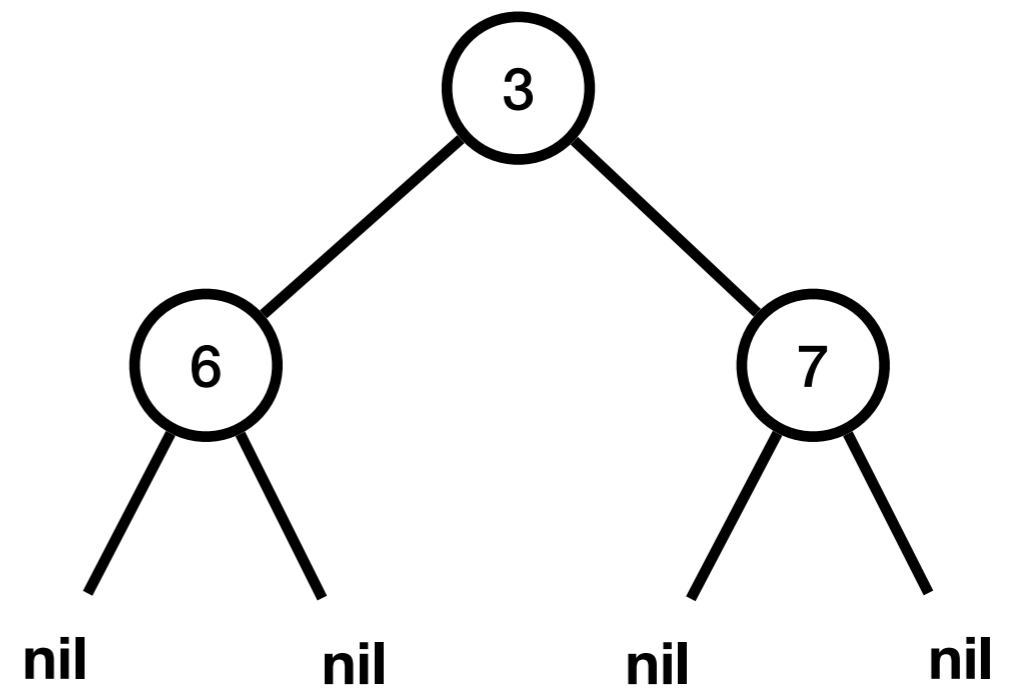
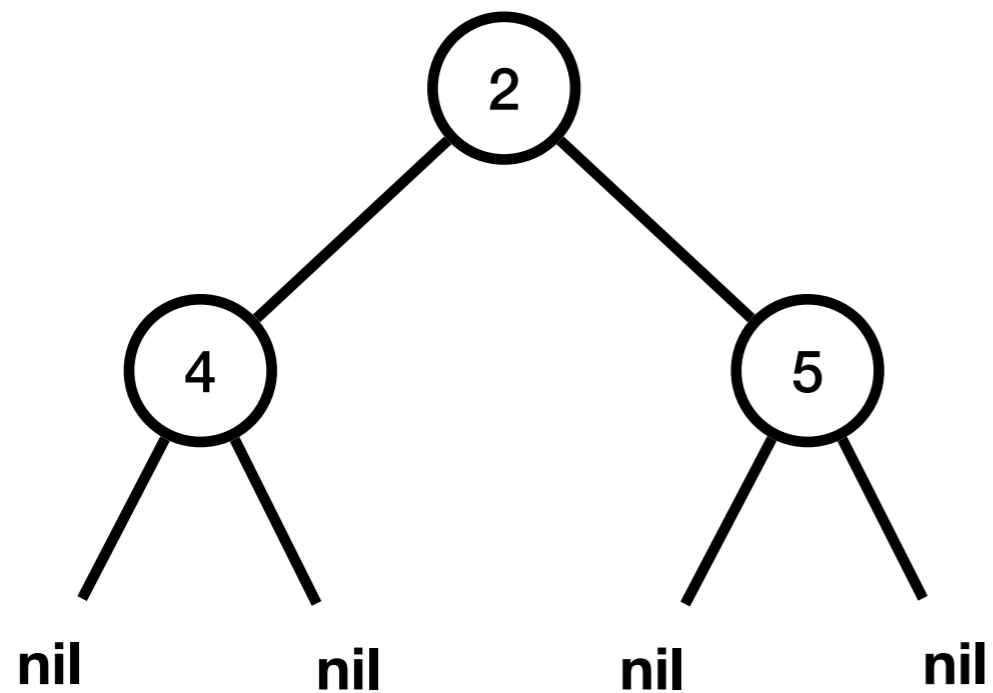
Exemples d'arbres binaires



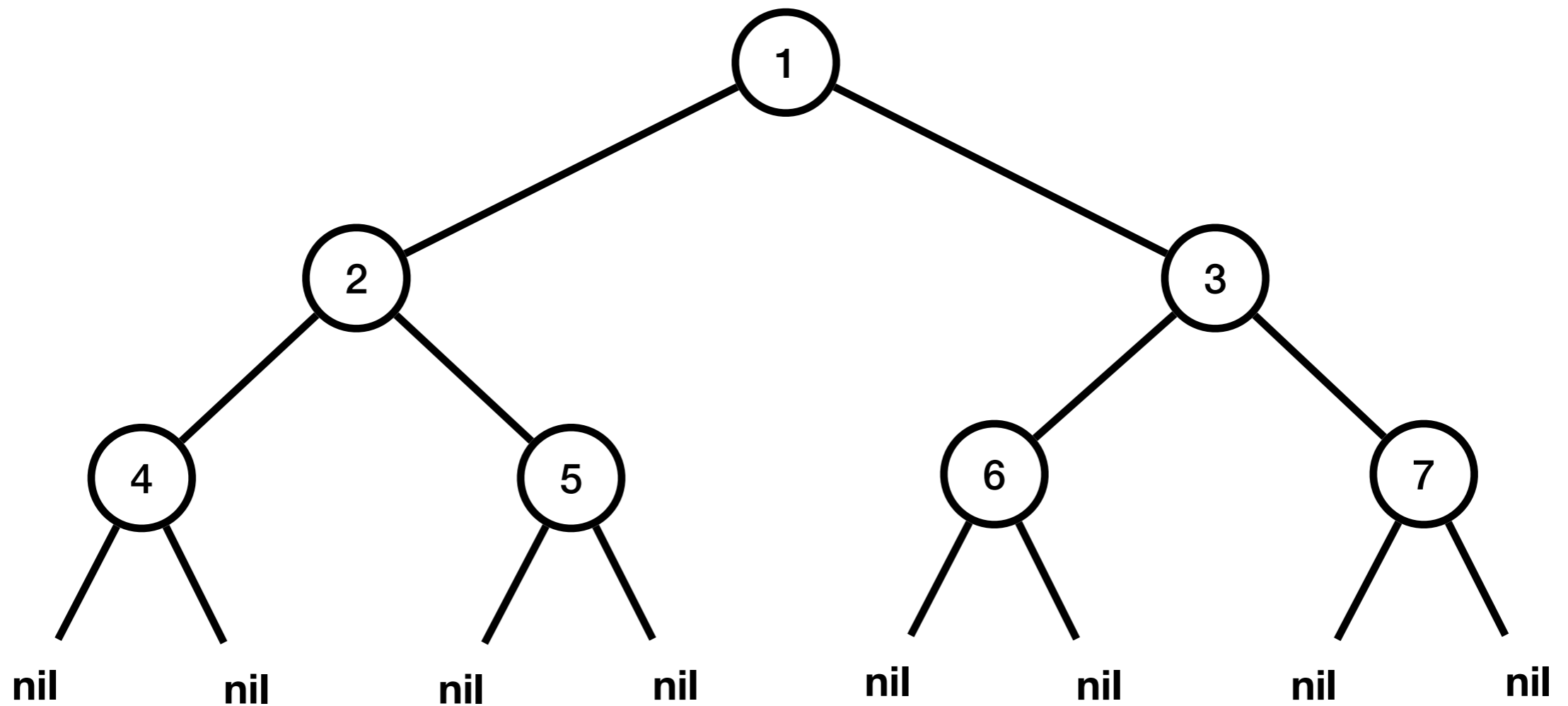
Exemples d'arbres binaires



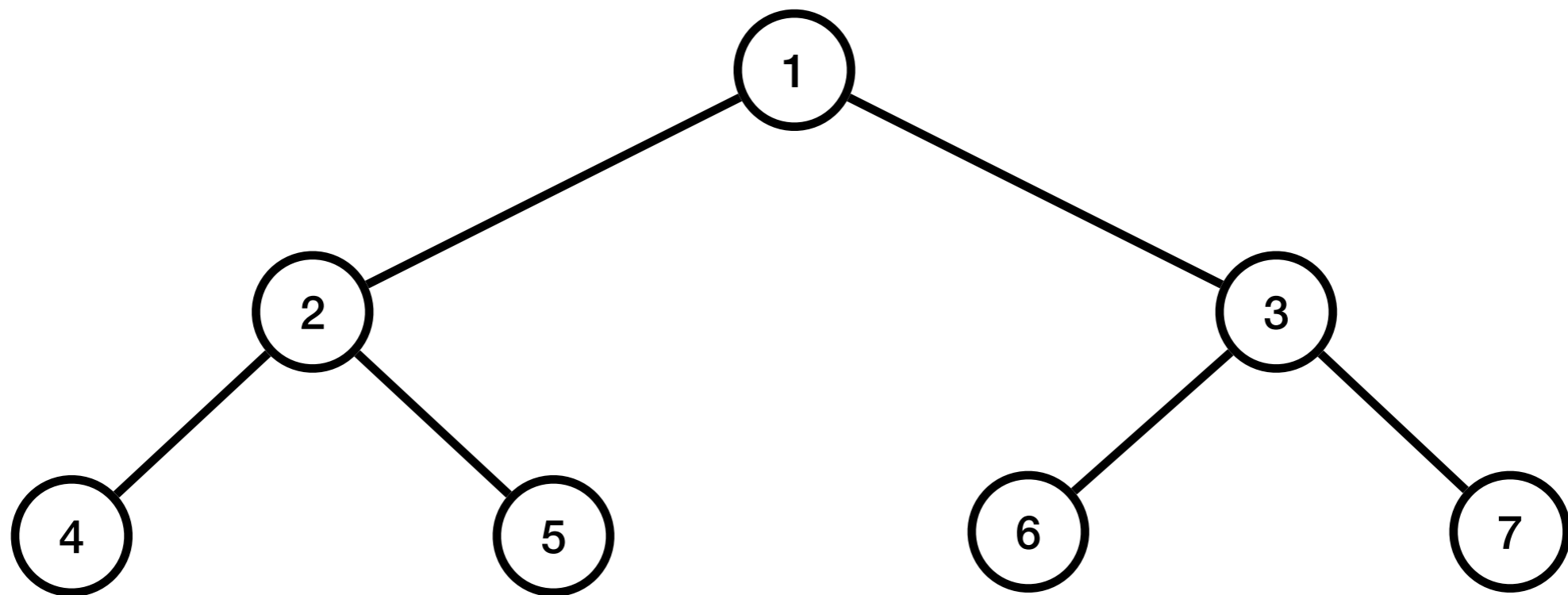
Exemples d'arbres binaires



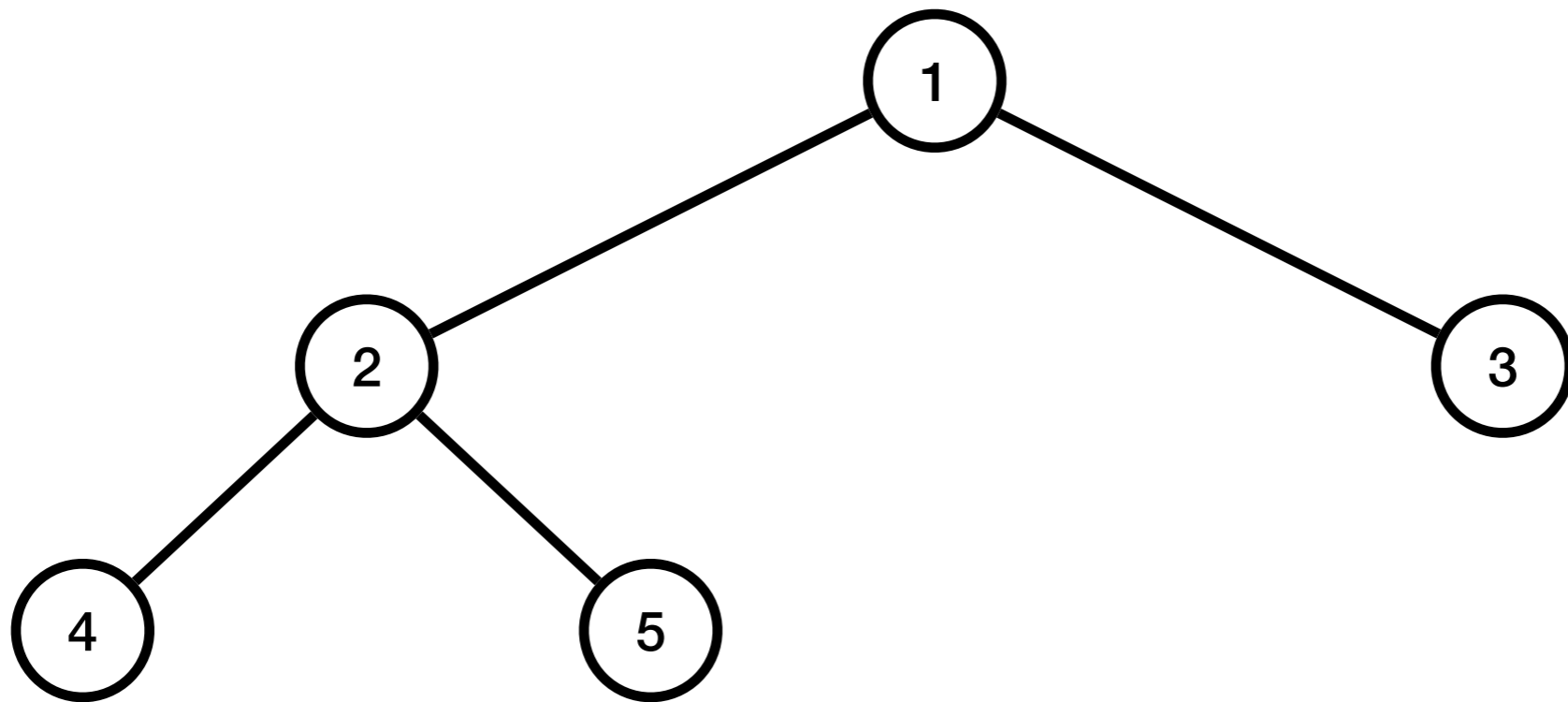
Exemples d'arbres binaires



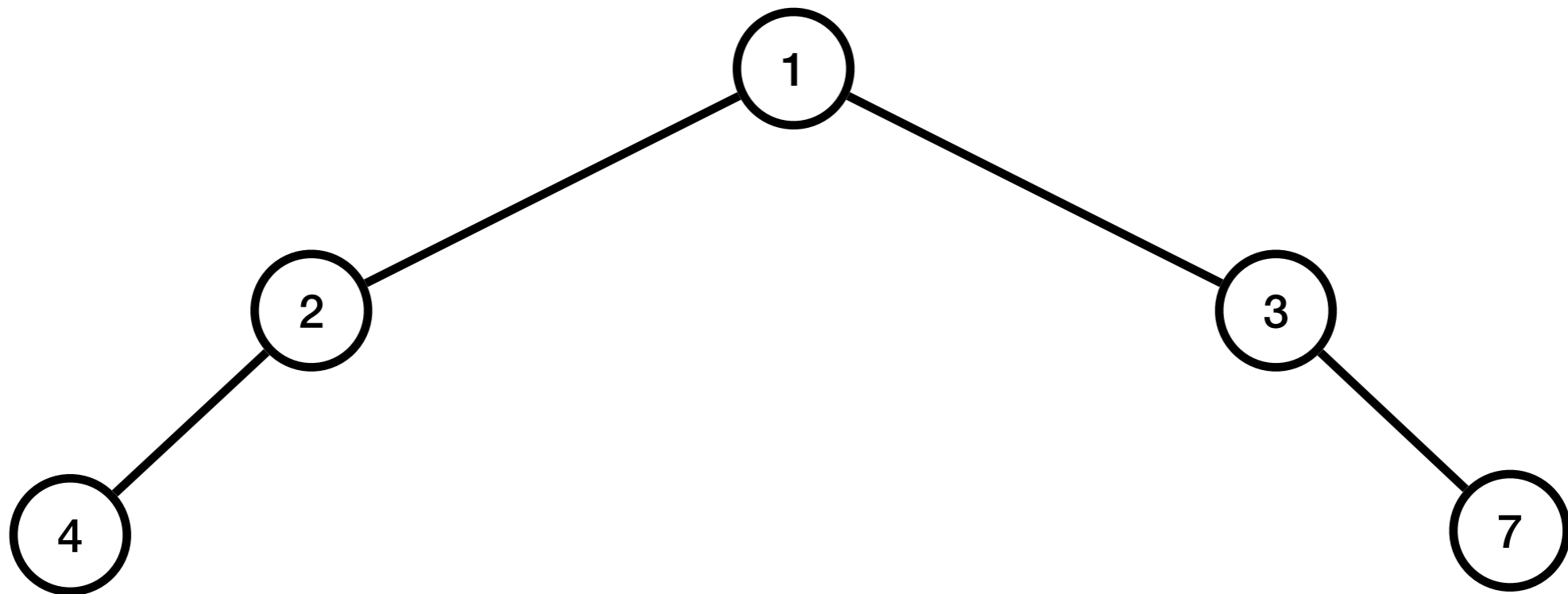
Exemples d'arbres binaires



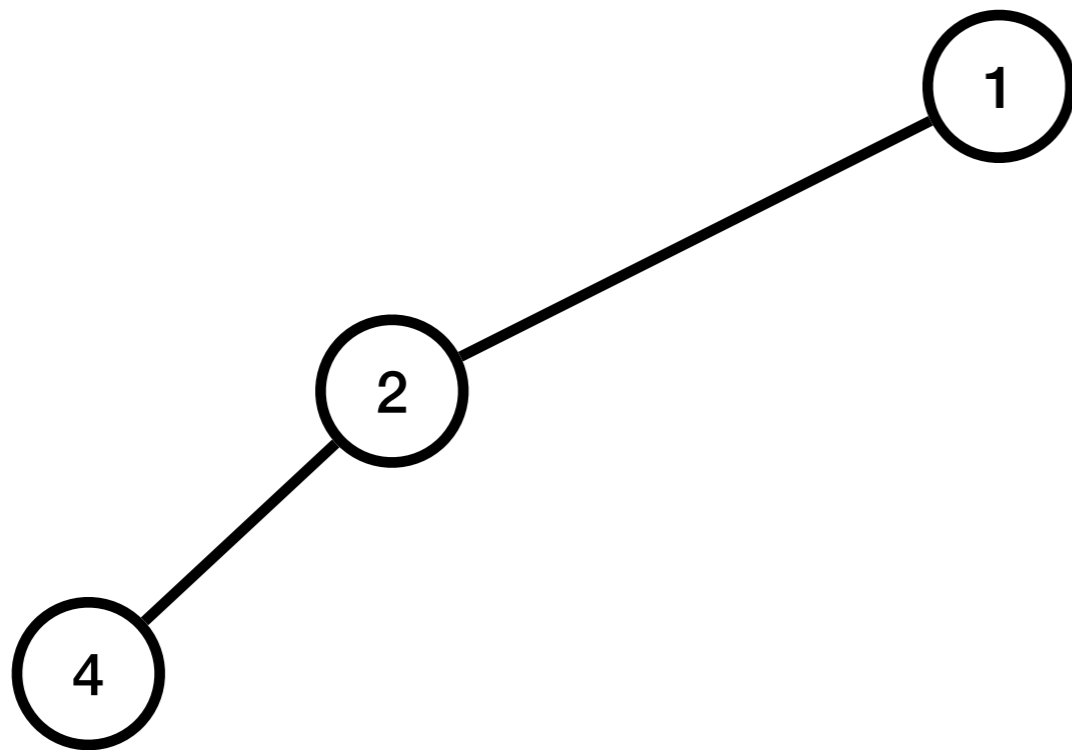
Exemples d'arbres binaires



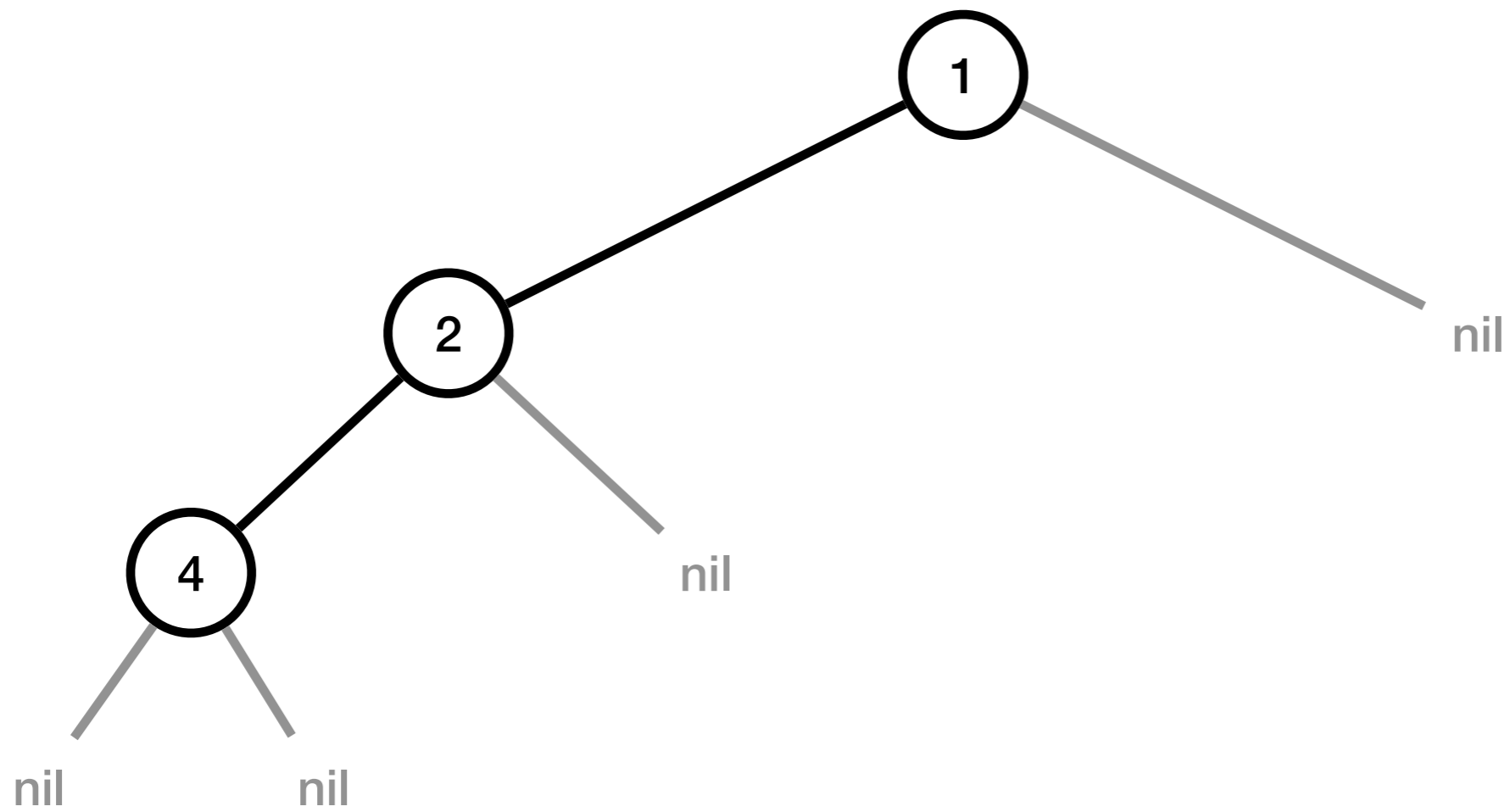
Exemples d'arbres binaires



Exemples d'arbres binaires



Exemples d'arbres binaires



Interface pour les arbres binaire

```
def nouveau_nœud(valeur, enfant_gauche, enfant_droit):  
    ...
```

```
def est_vide(arbre):  
    ...
```

```
def valeur(arbre):  
    ...
```

```
def enfant_gauche(arbre):  
    ...
```

```
def enfant_droit(arbre):  
    ...
```


Implémentation avec des dictionnaires

```
def nouveau_nœud(valeur, enfant_gauche, enfant_droit):  
    return {'valeur': valeur,  
            'enfant_gauche': enfant_gauche,  
            'enfant_droit': enfant_droit}
```

```
def est_vide(arbre):  
    return arbre is None
```

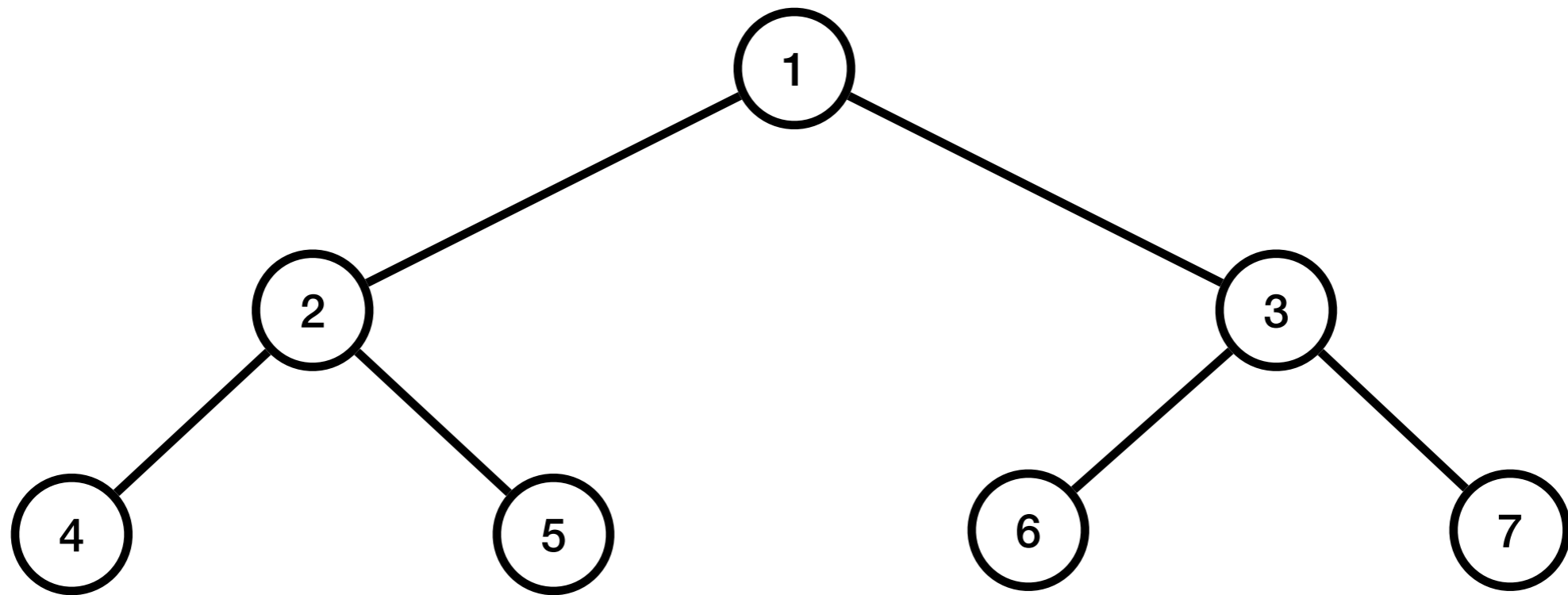
```
def valeur(nœud):  
    return nœud['valeur']
```

```
def enfant_gauche(nœud):  
    return nœud['enfant_gauche']
```

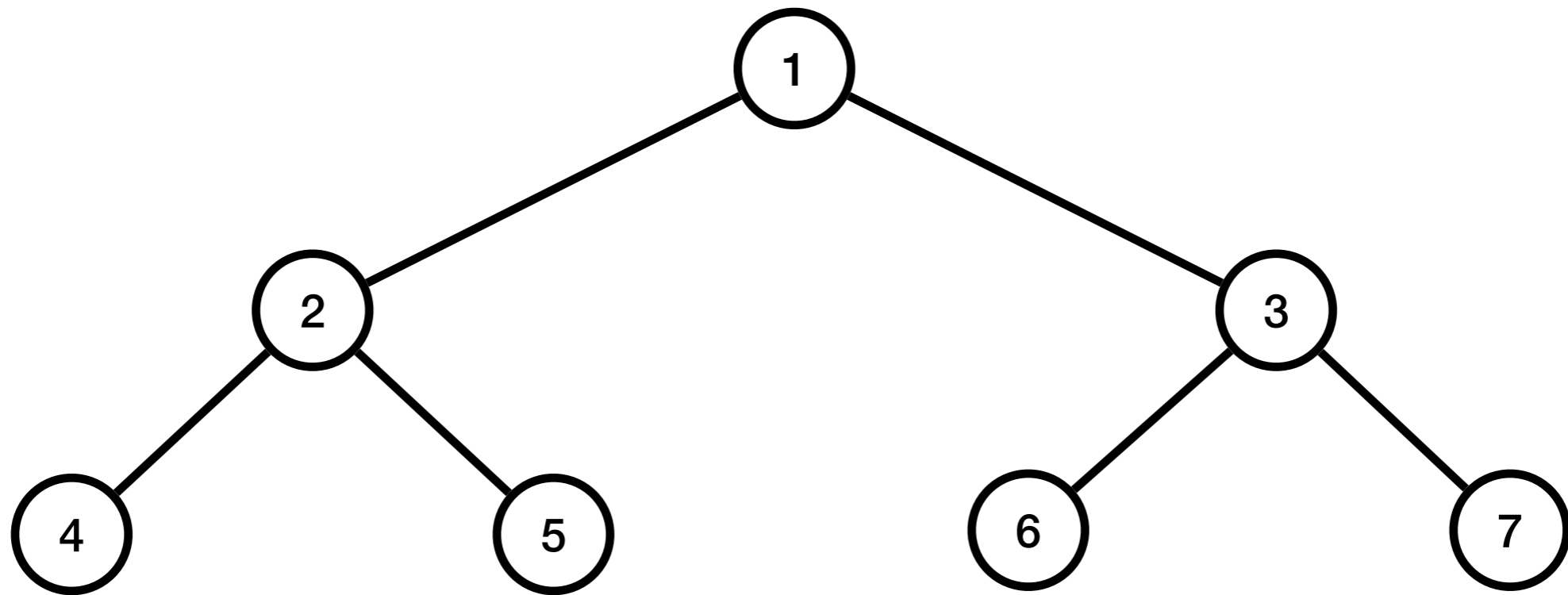
```
def enfant_droit(nœud):  
    return nœud['enfant_droit']
```

Exercice 1 du TD5

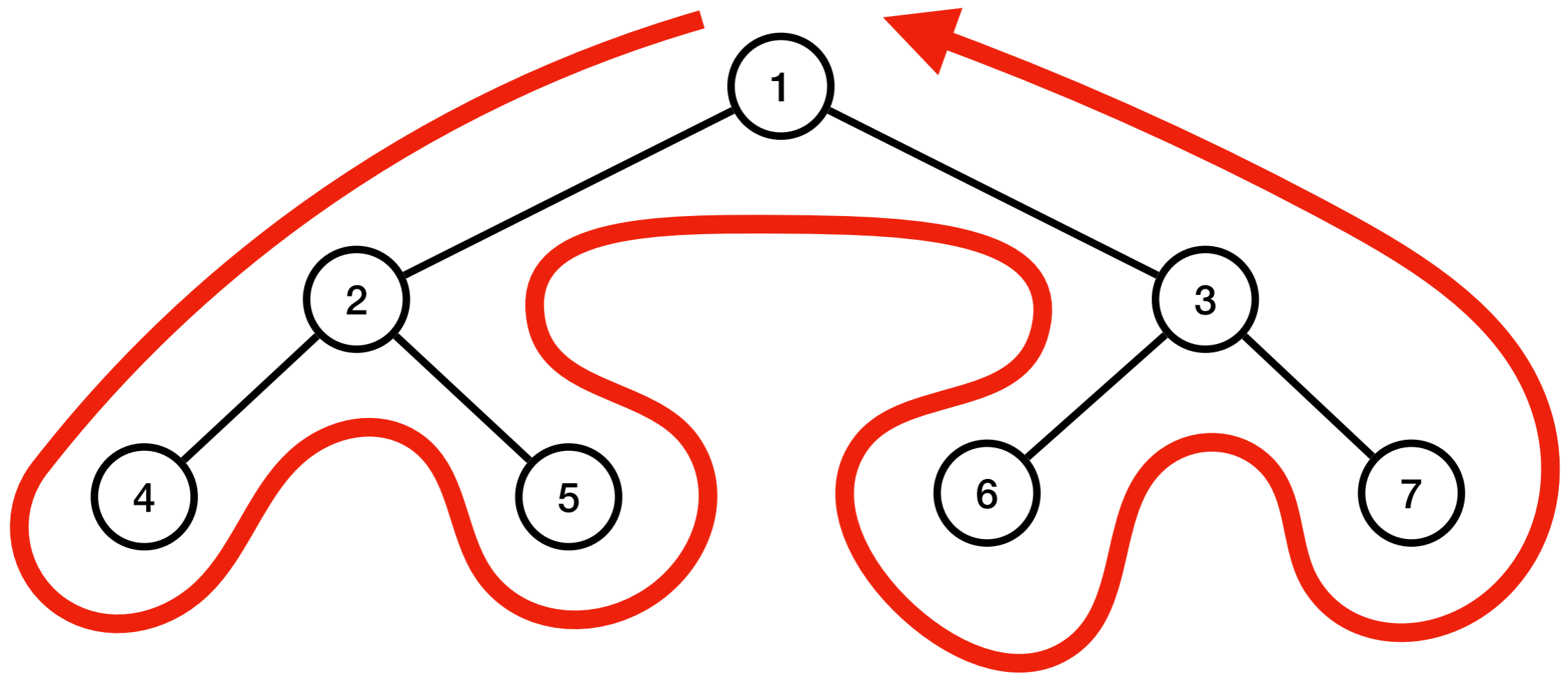
Parcours d'arbres binaires



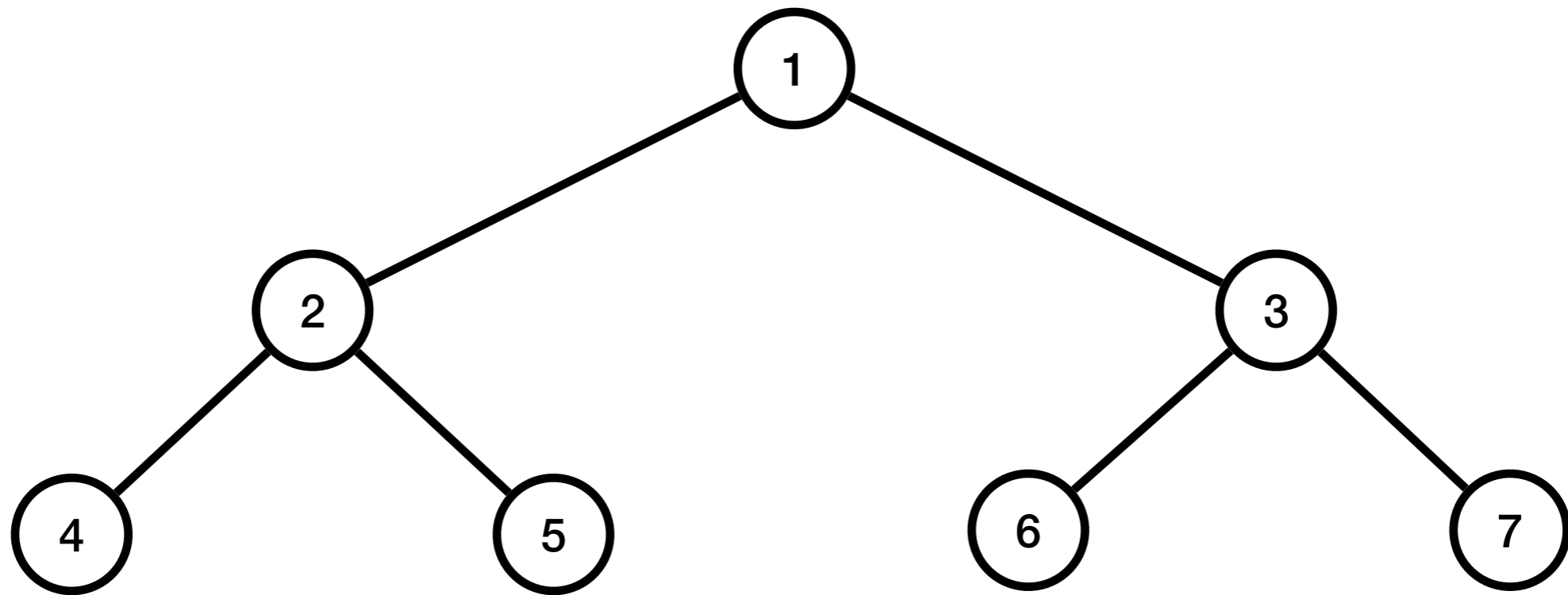
Parcours d'arbres binaires



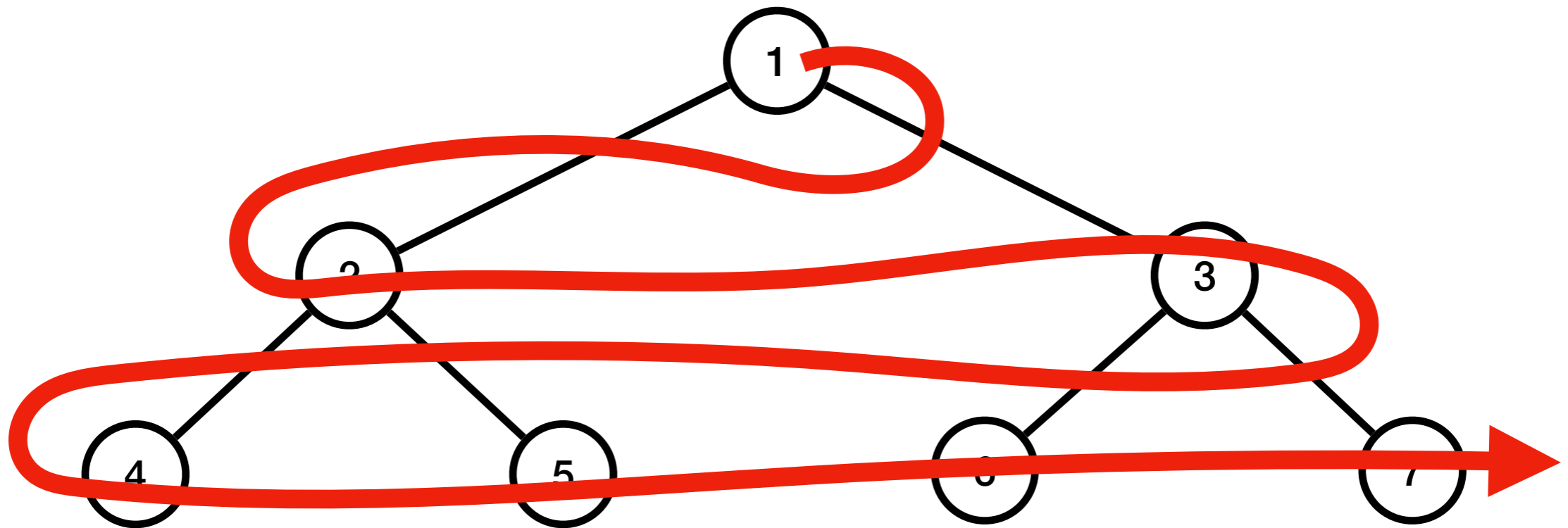
Parcours d'arbres binaires



Parcours d'arbres binaires



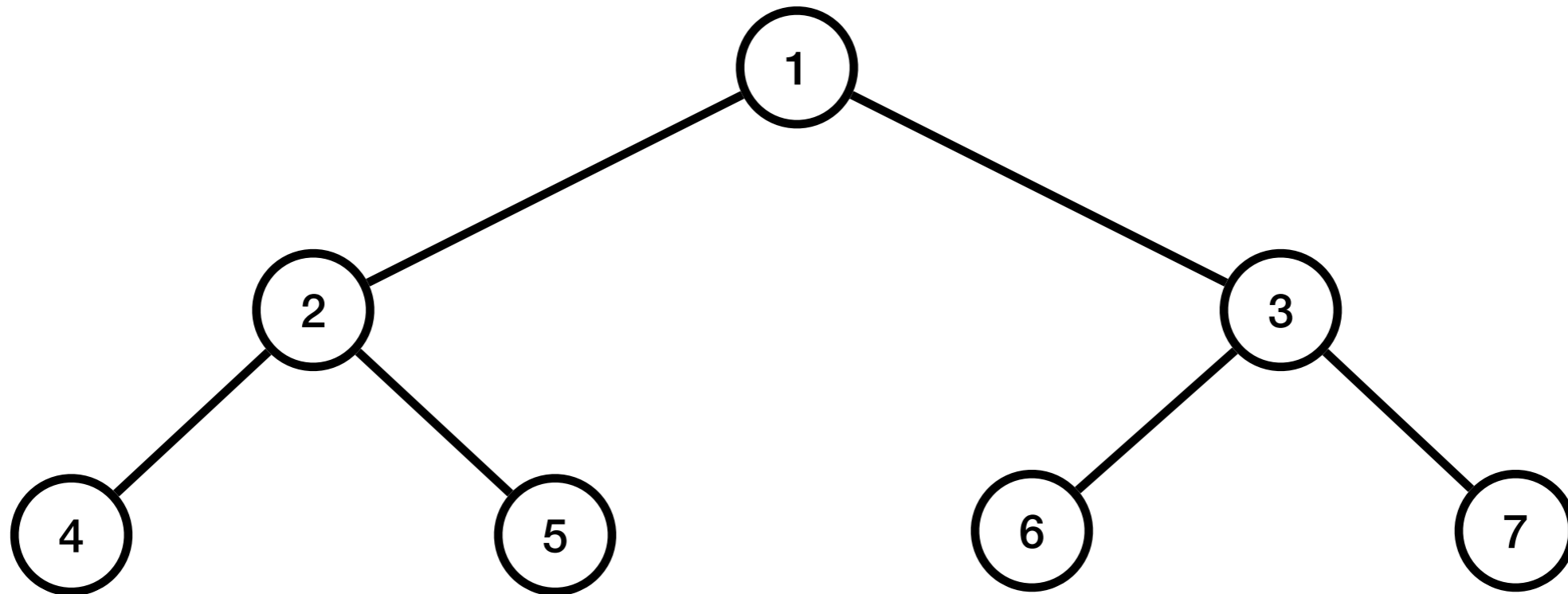
Parcours d'arbres binaires



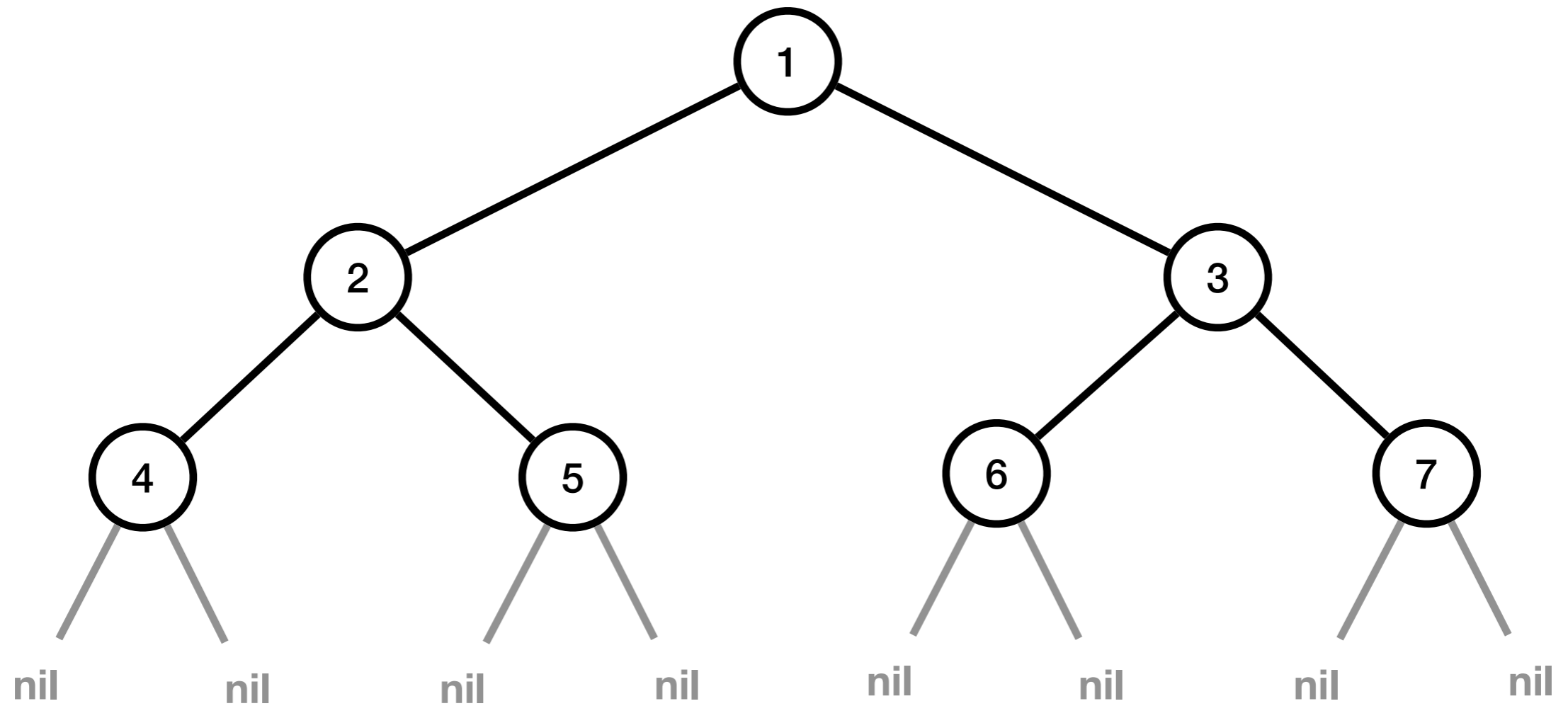
Parcours préfixe

```
def parcours_préfixe(arbre):  
    if not est_vide(arbre):  
        print(valeur(arbre))  
        parcours_préfixe(enfant_gauche(arbre))  
        parcours_préfixe(enfant_droit(arbre))
```

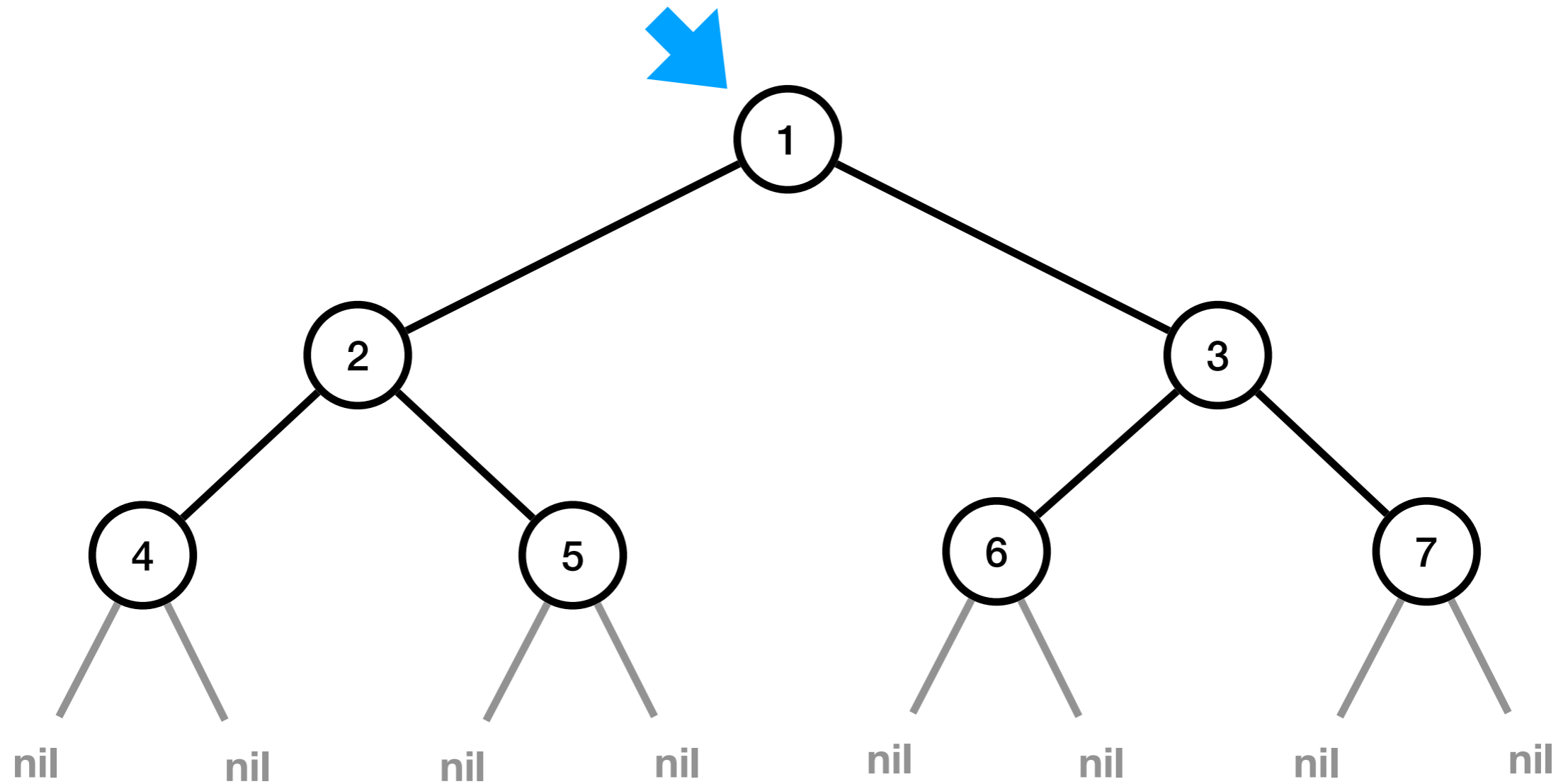

Parcours préfixe



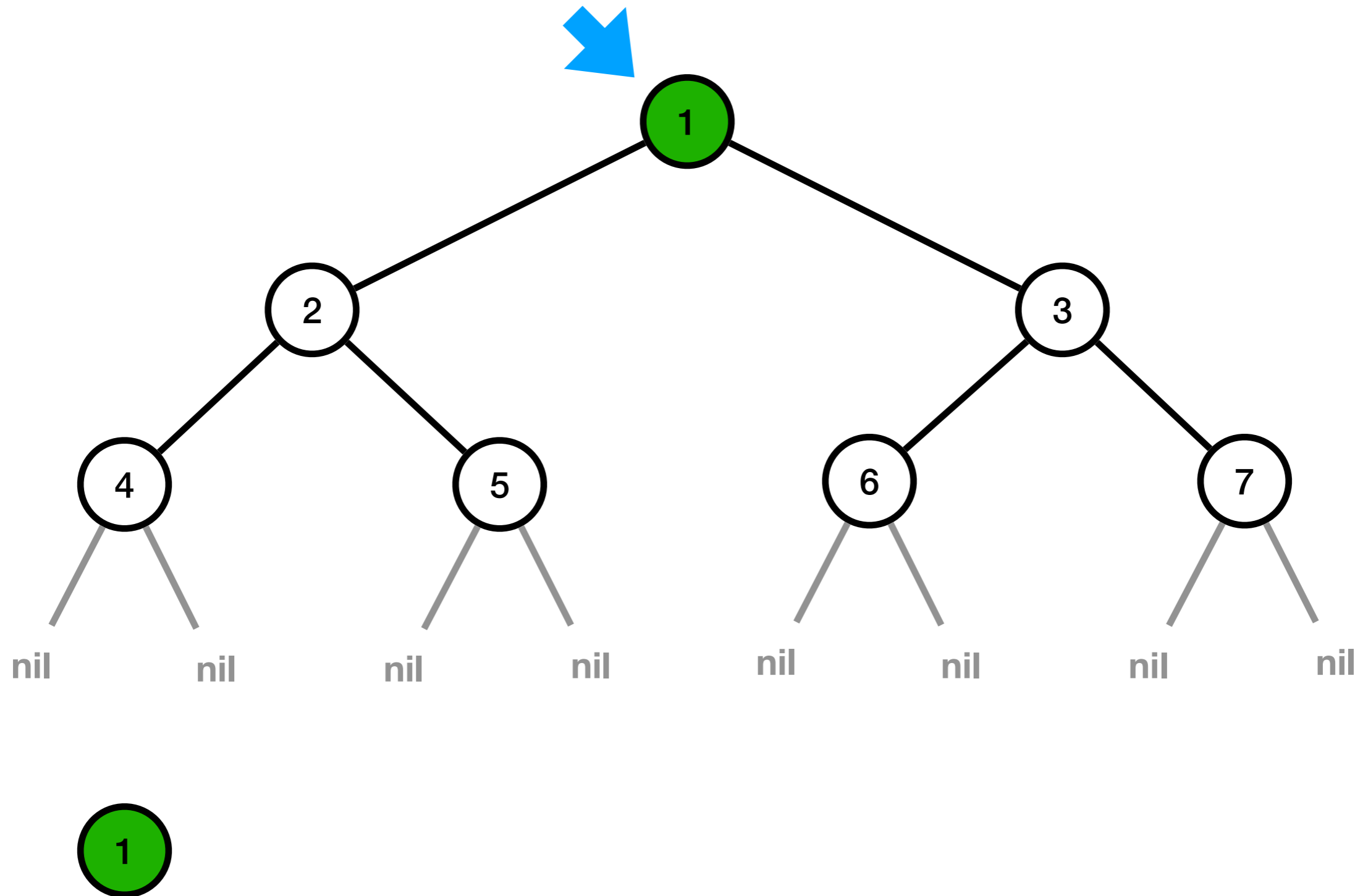
Parcours préfixe



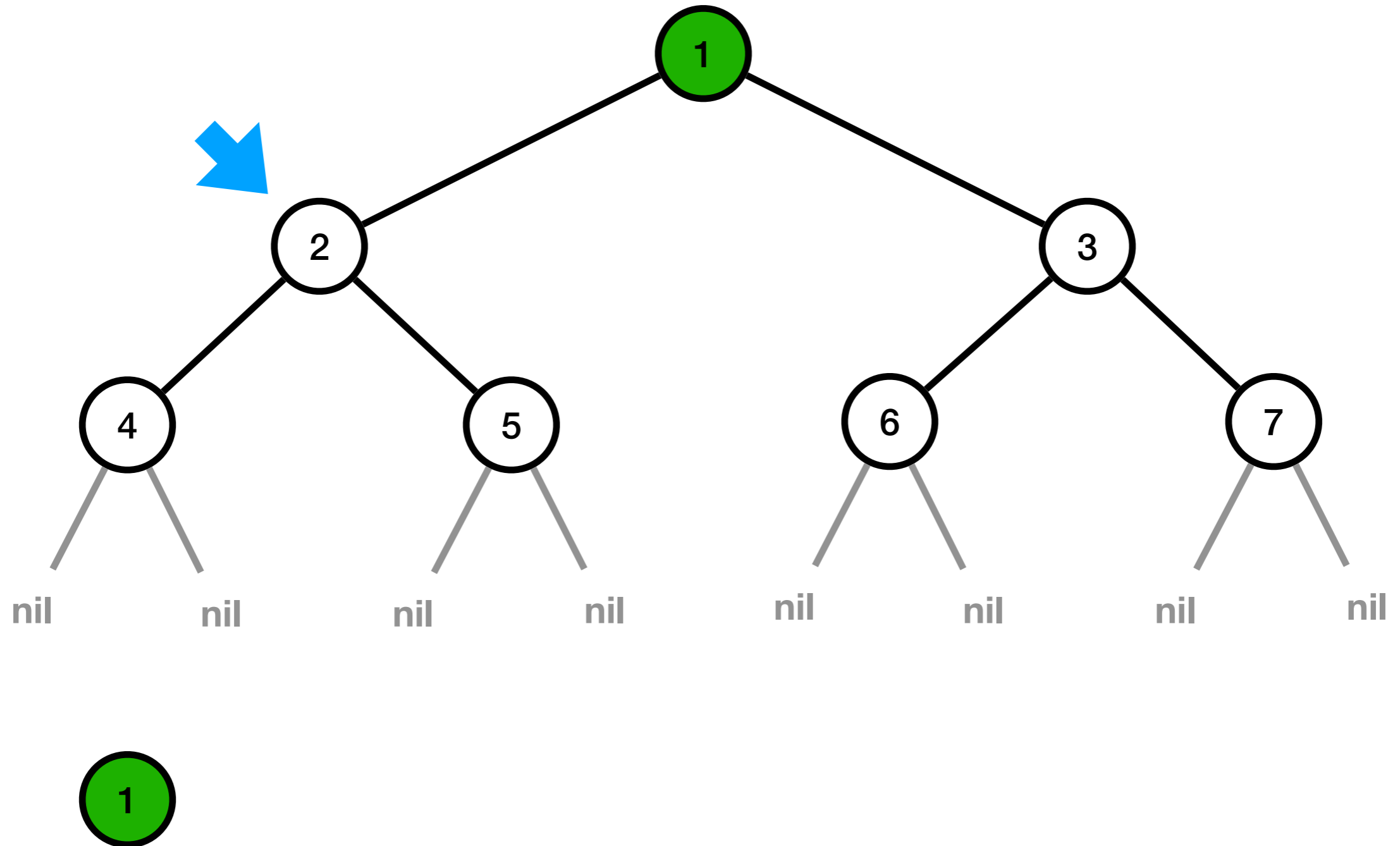
Parcours préfixe



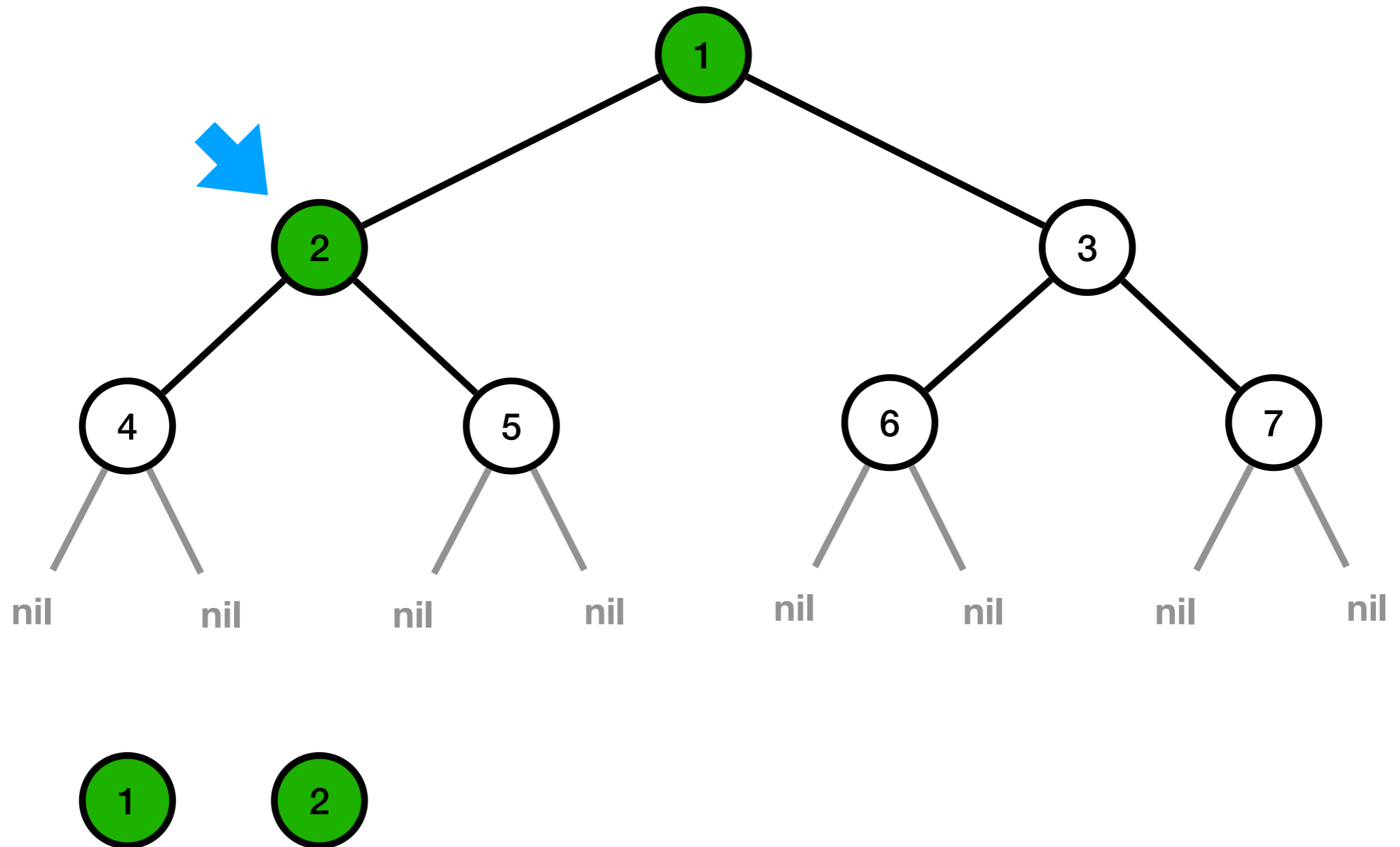
Parcours préfixe



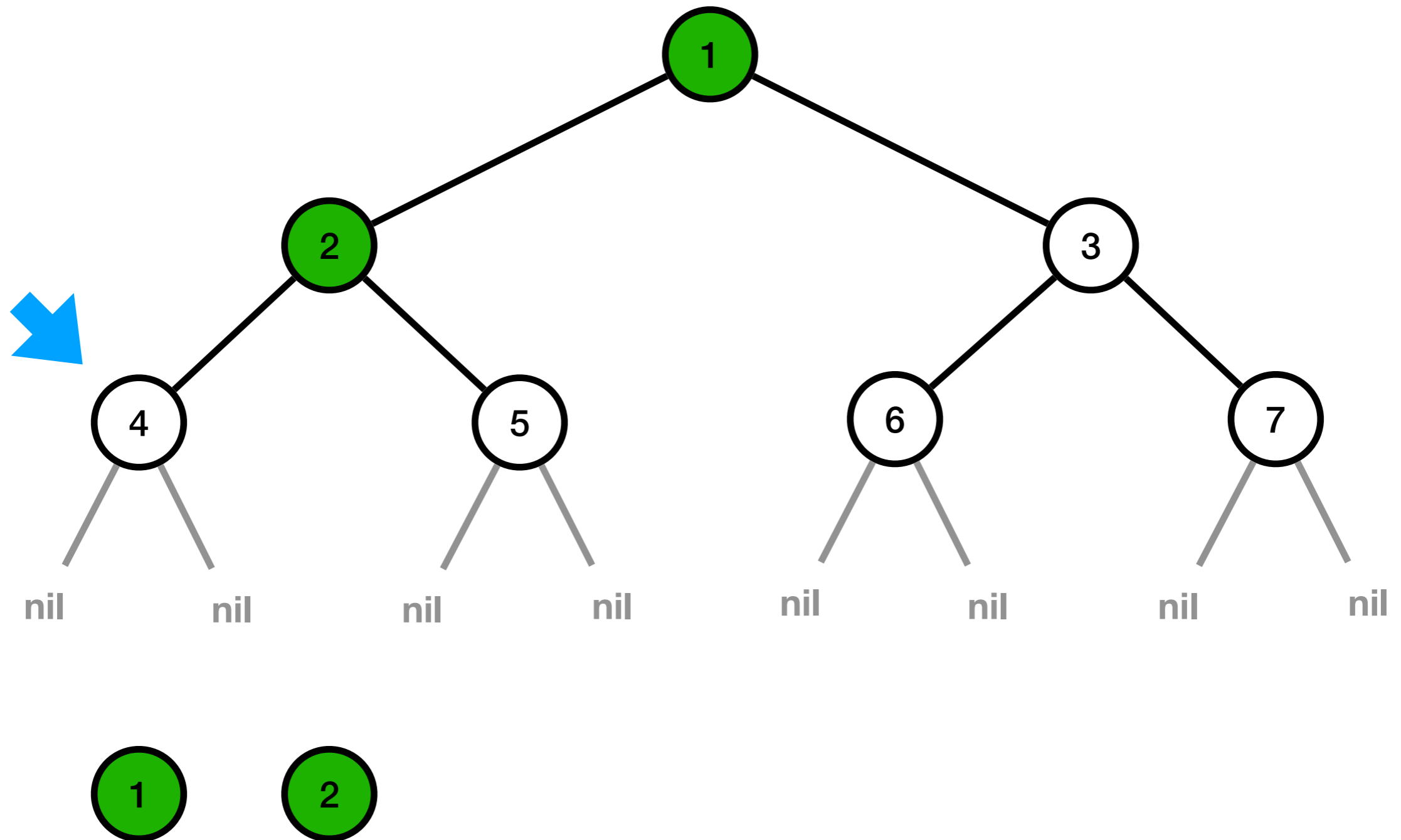
Parcours préfixe



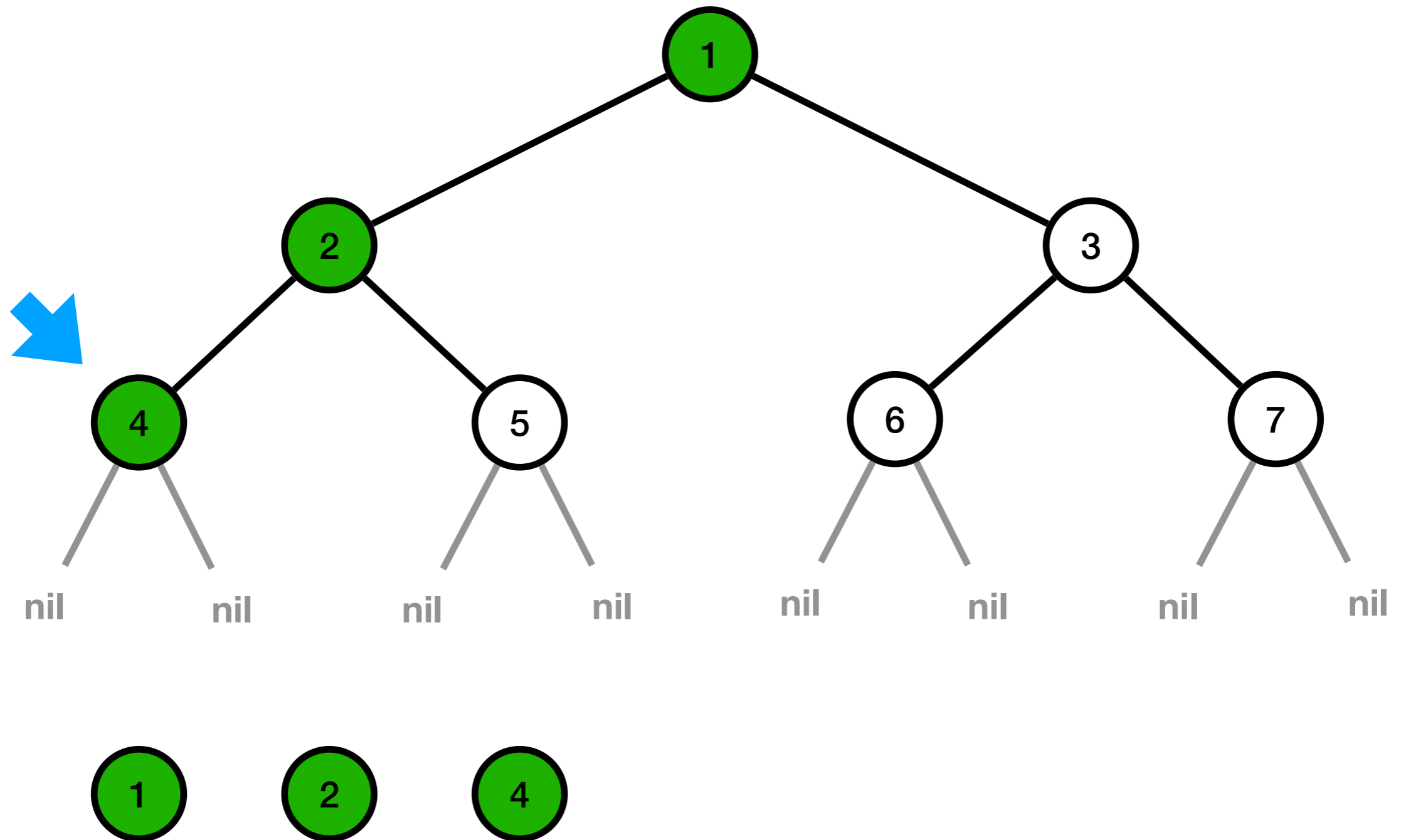
Parcours préfixe



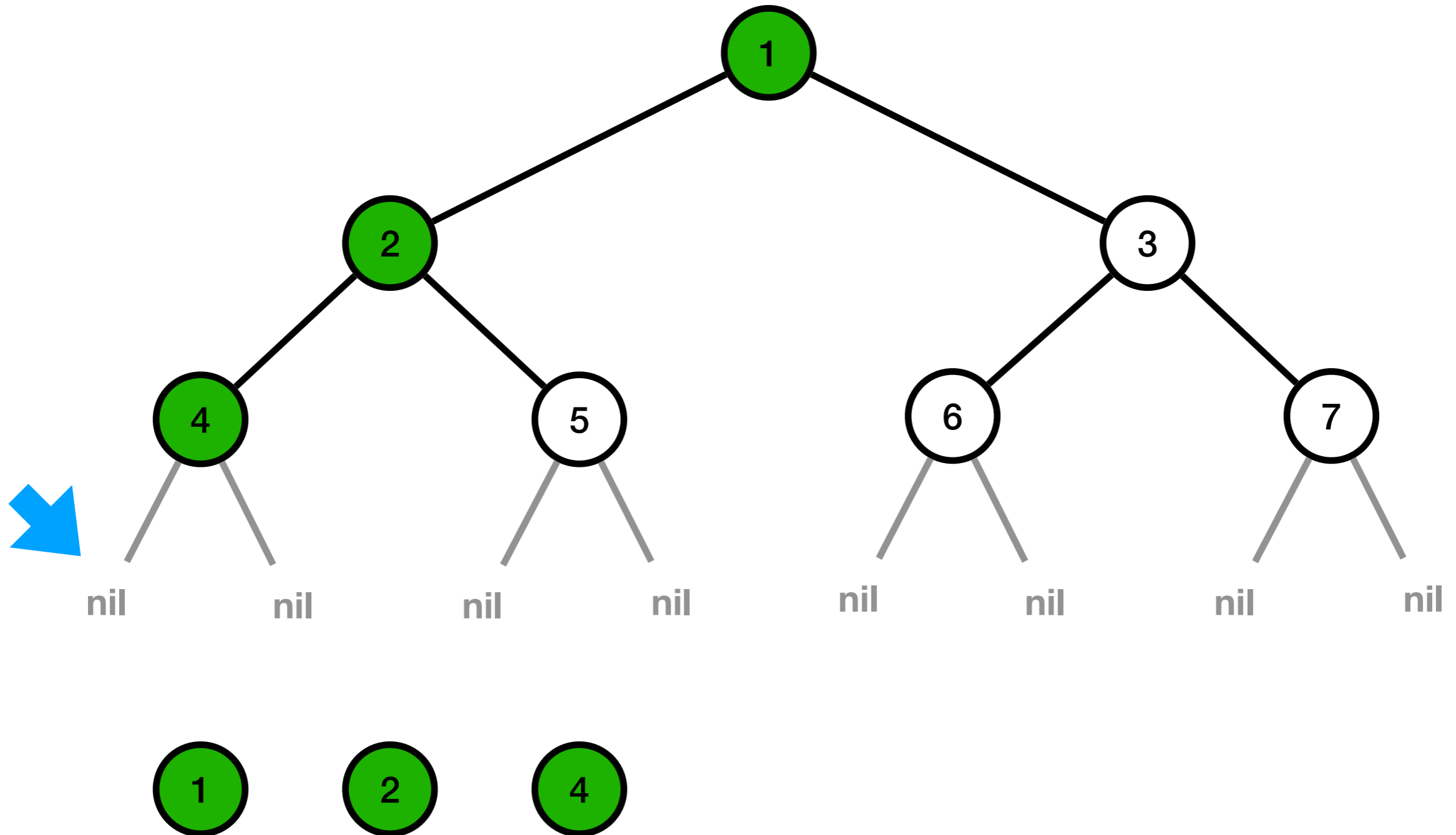
Parcours préfixe



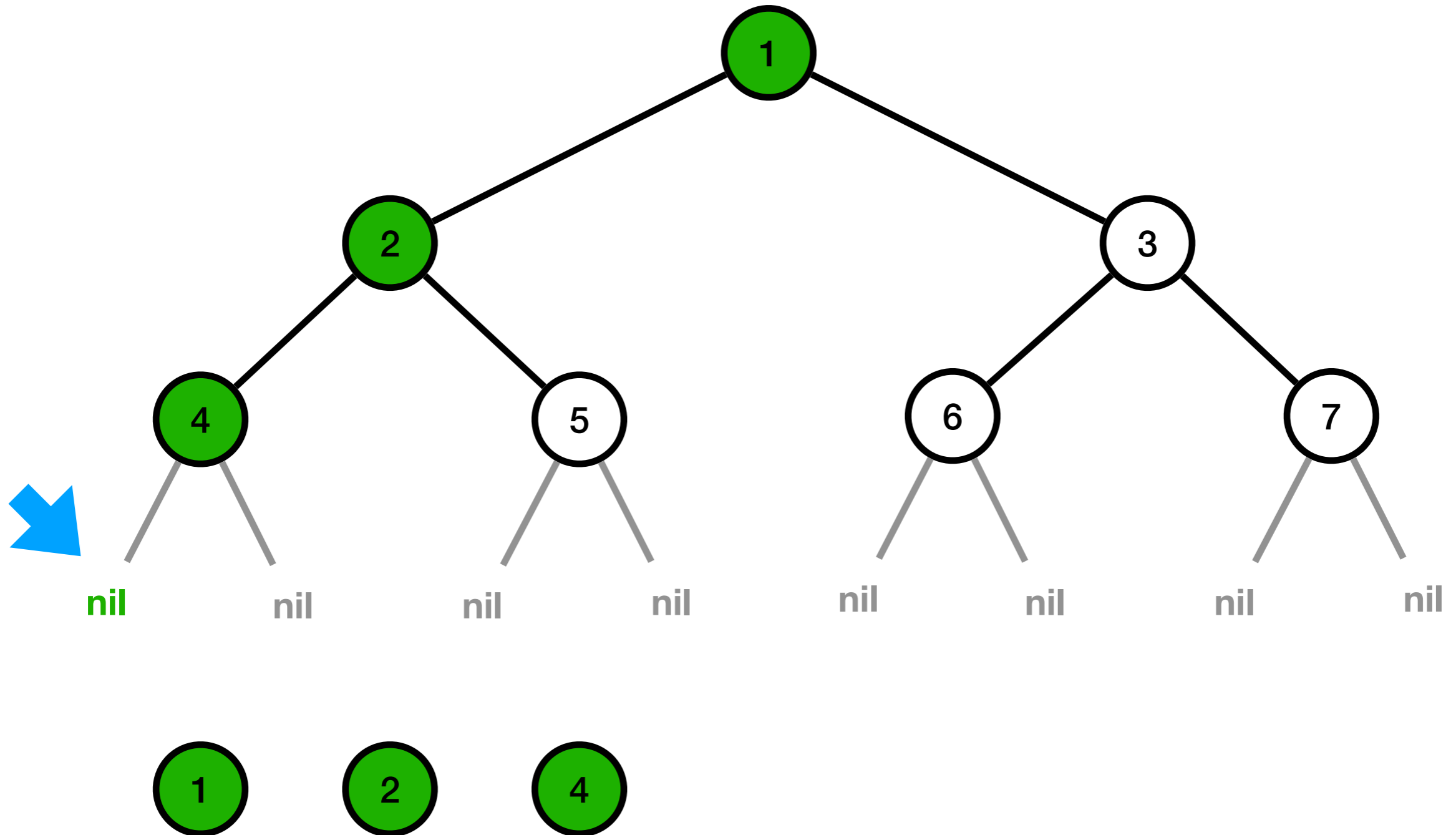
Parcours préfixe



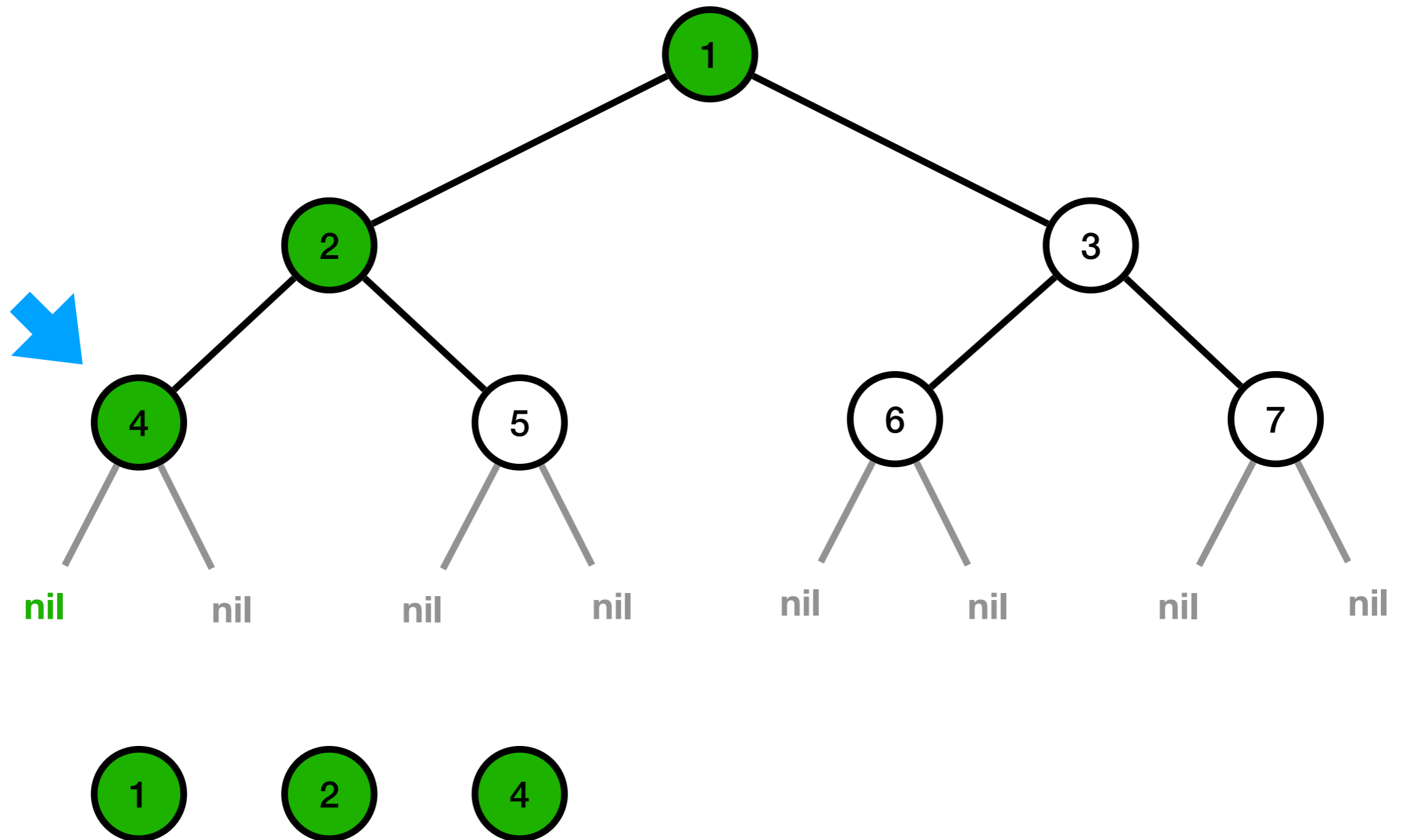
Parcours préfixe



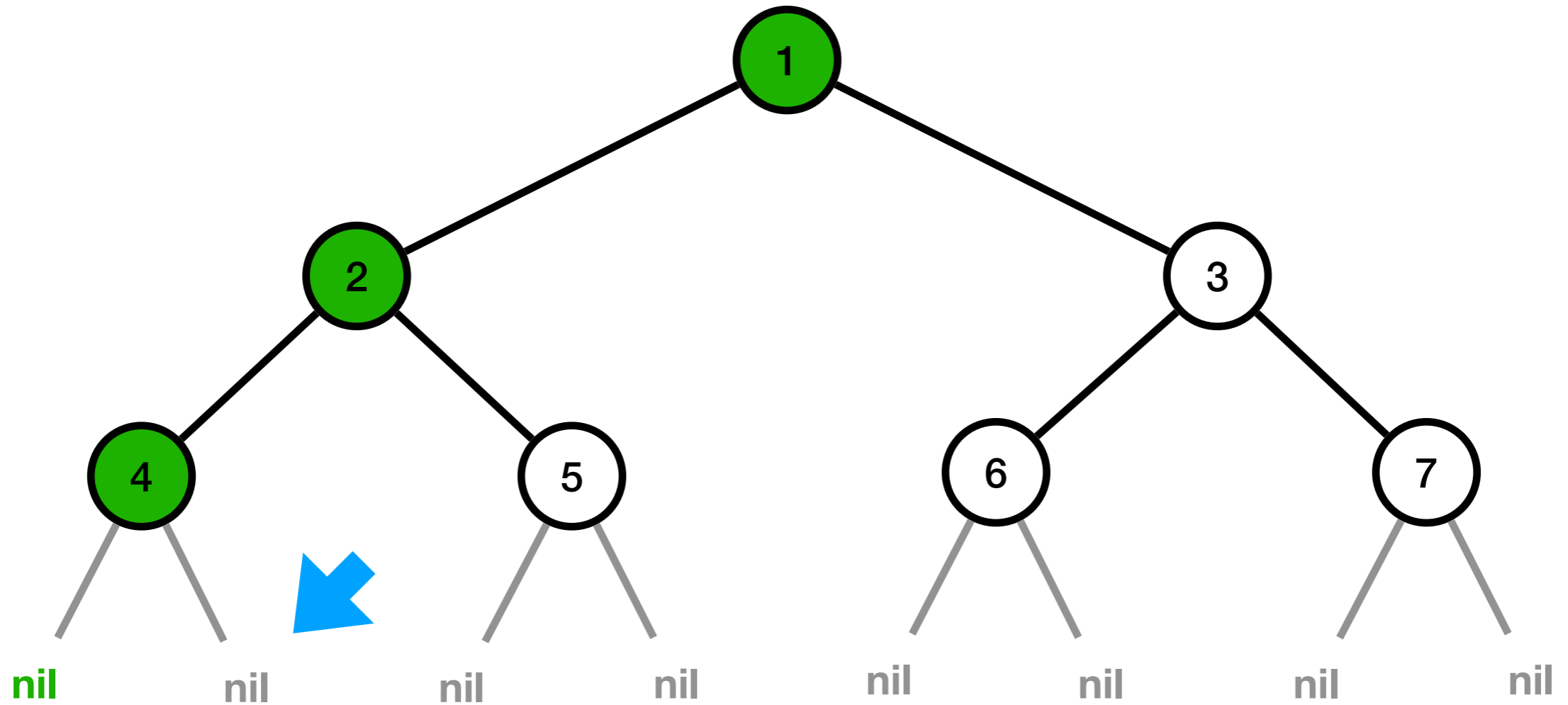
Parcours préfixe



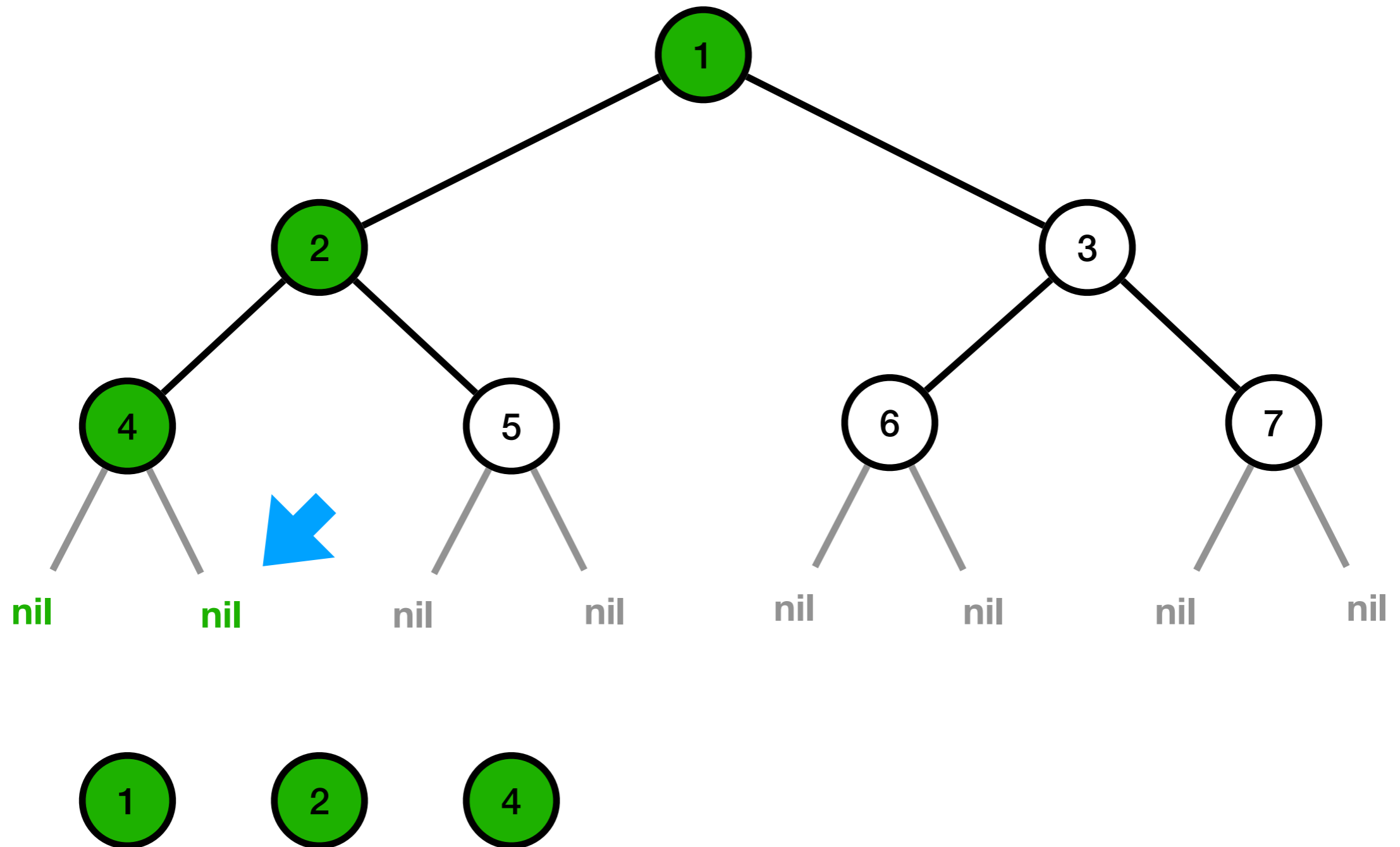
Parcours préfixe



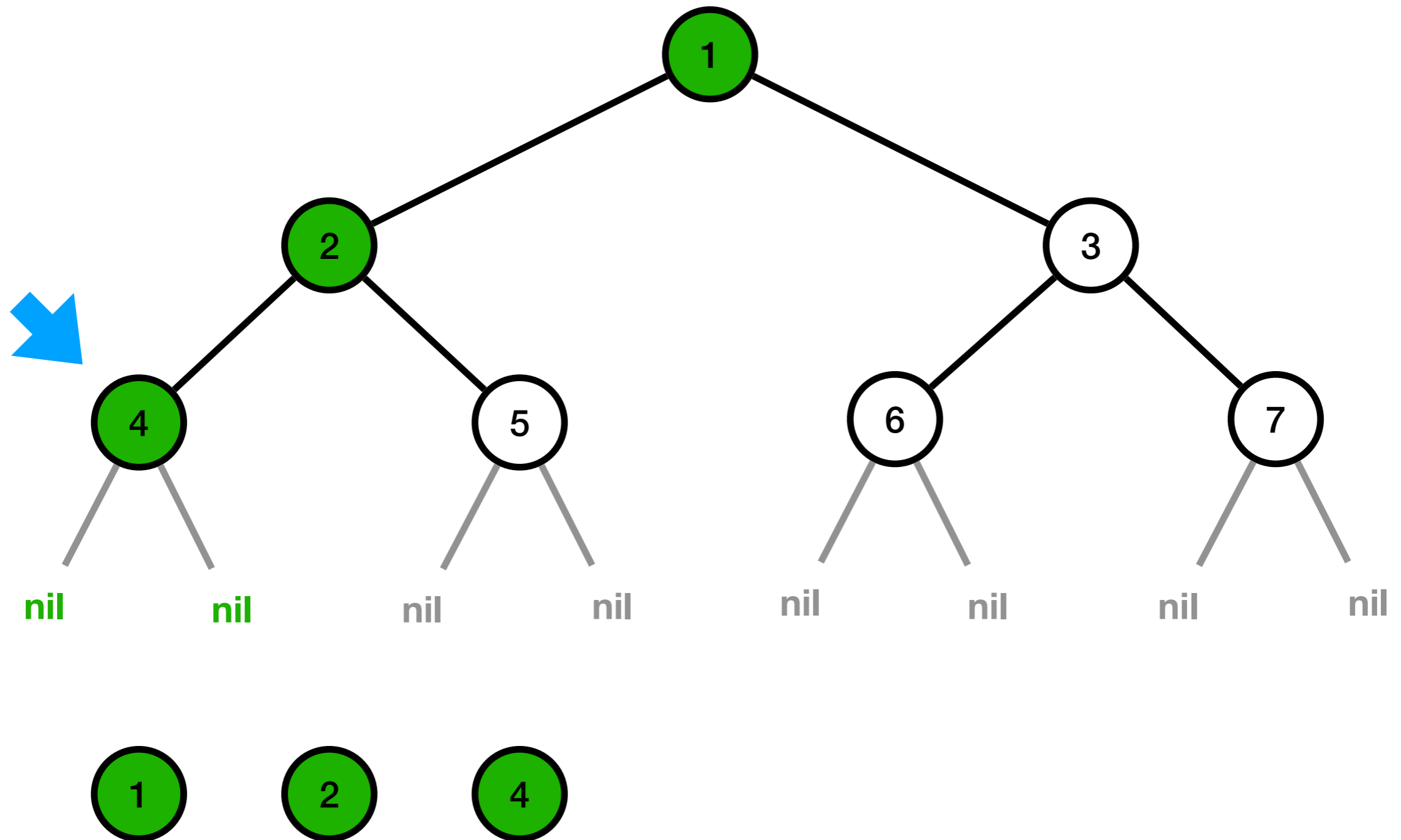
Parcours préfixe



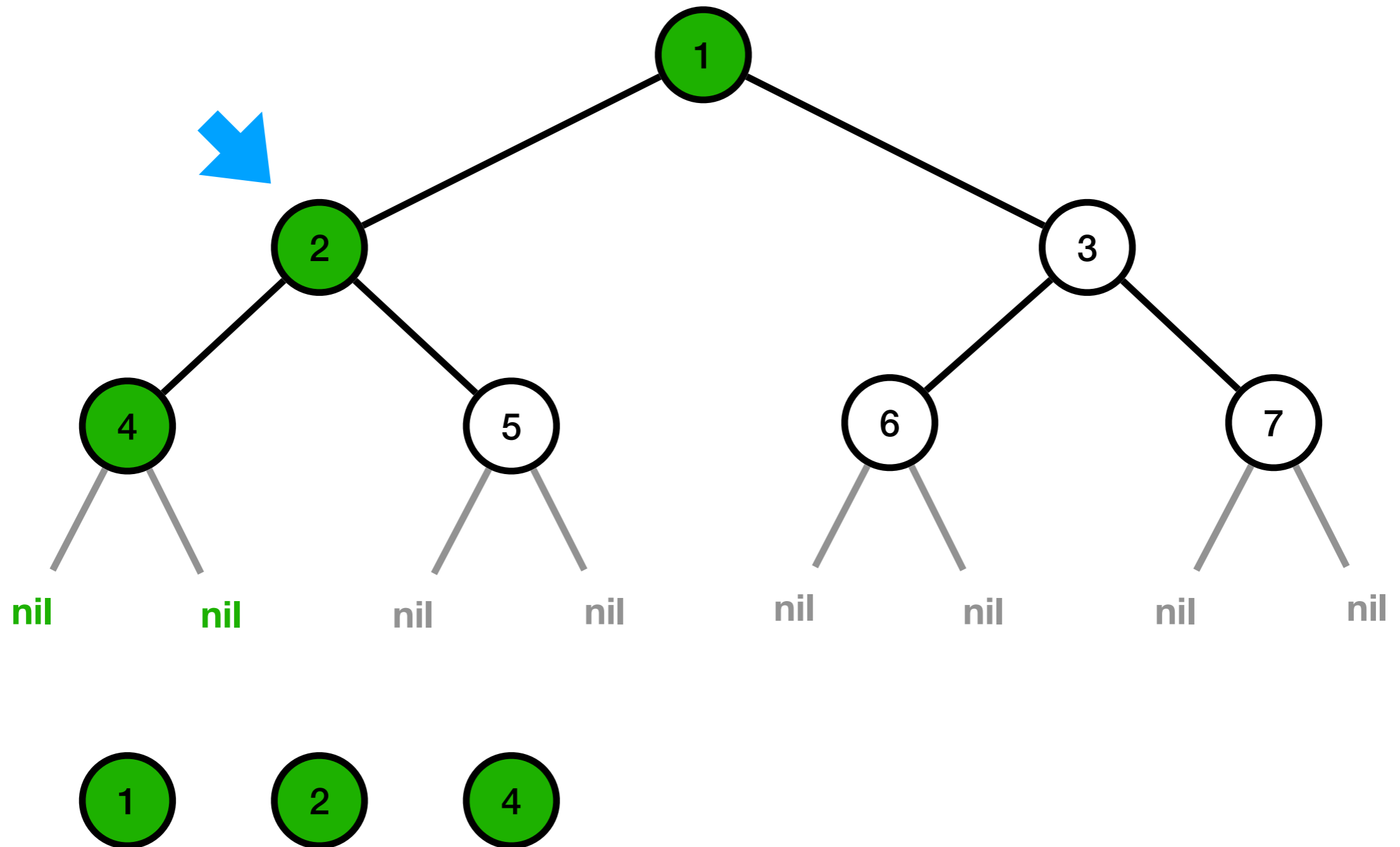
Parcours préfixe



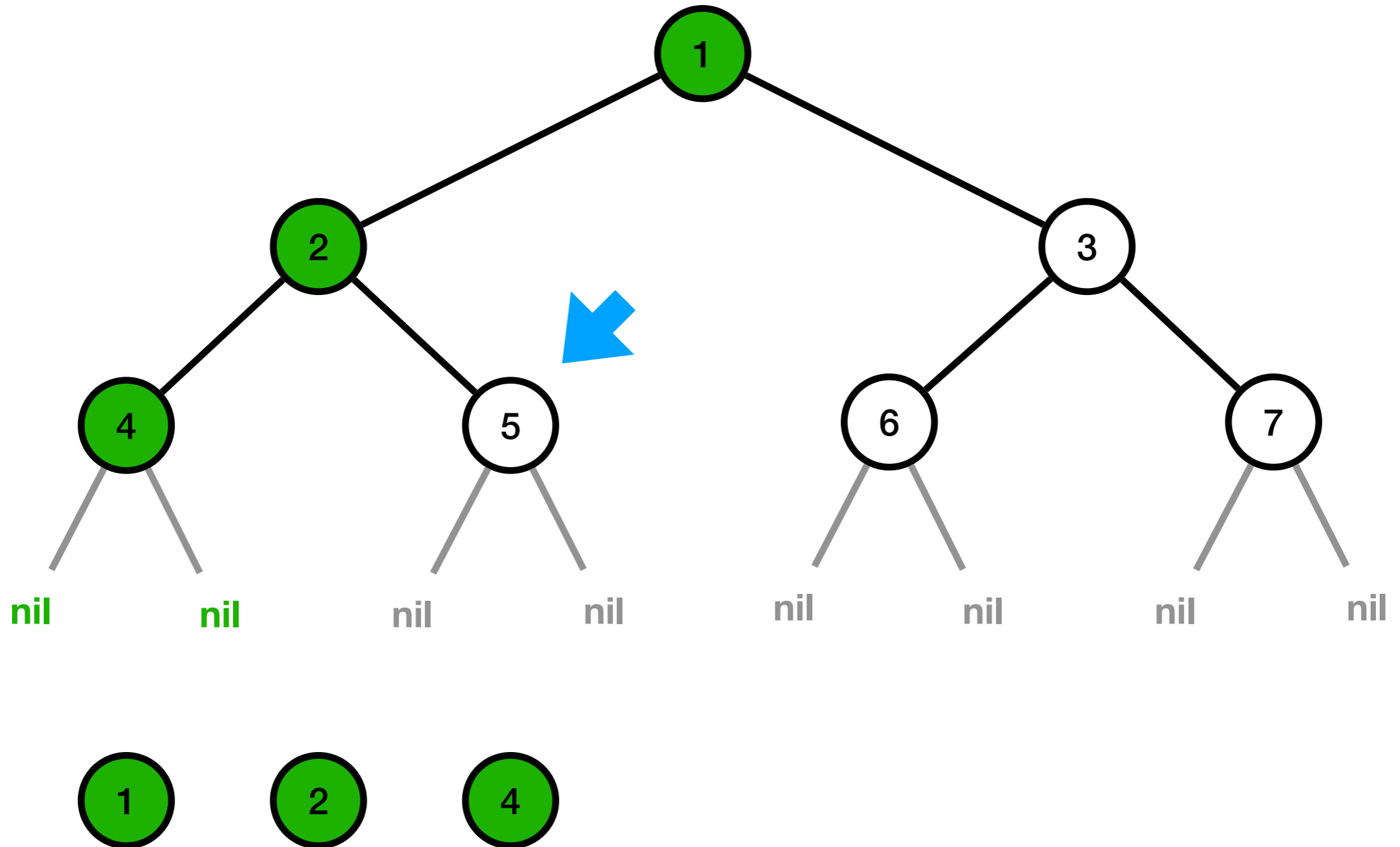
Parcours préfixe



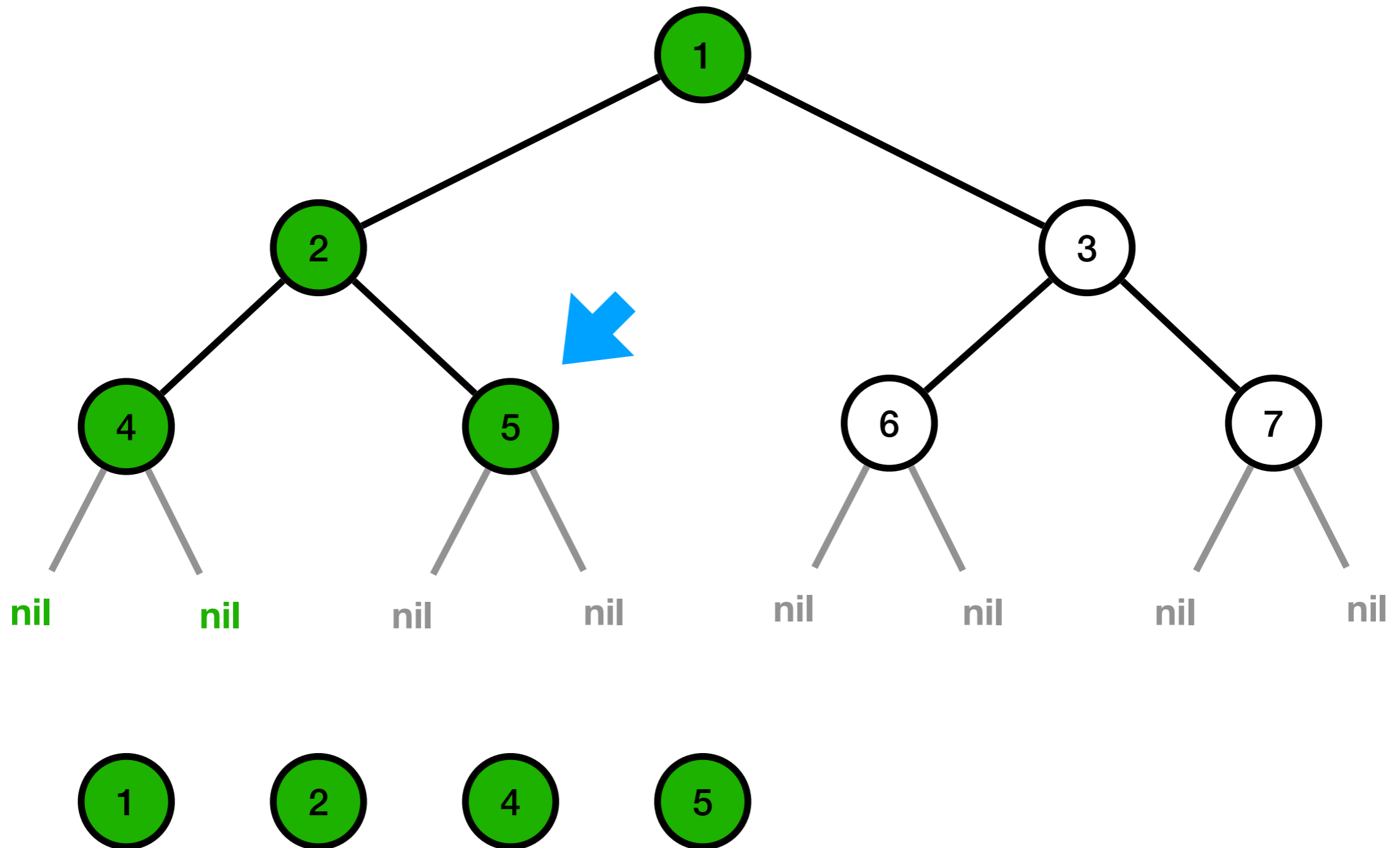
Parcours préfixe



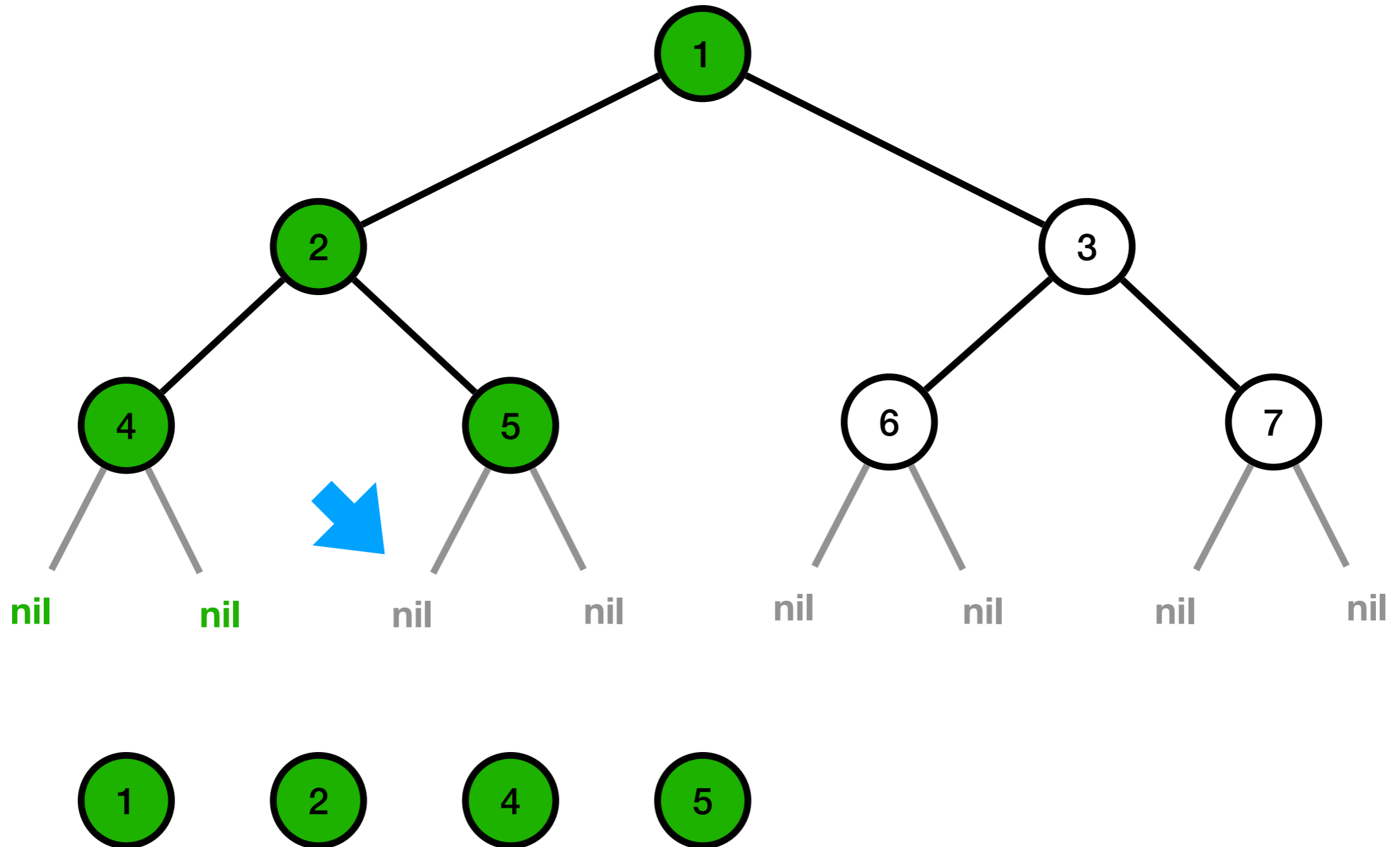
Parcours préfixe



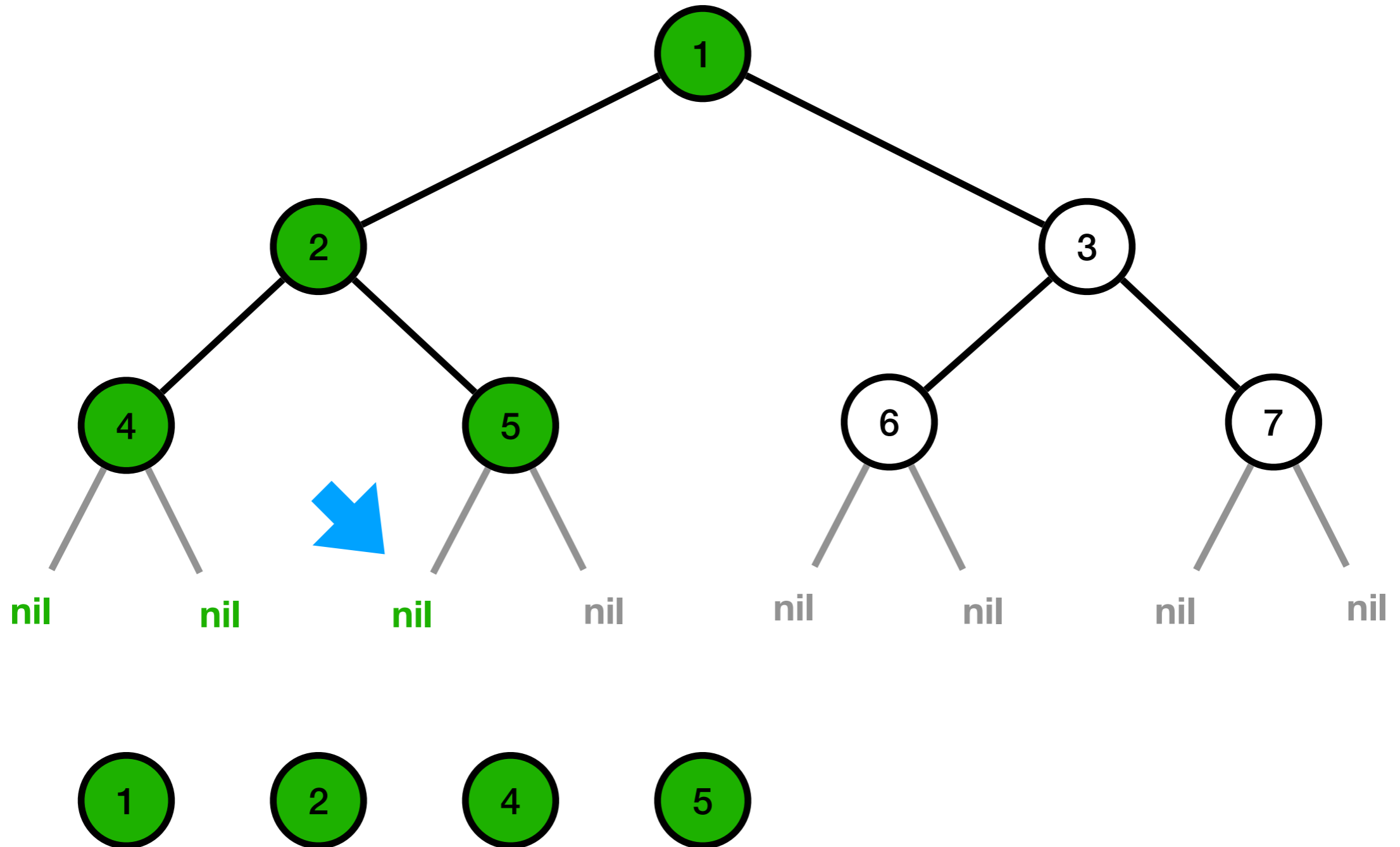
Parcours préfixe



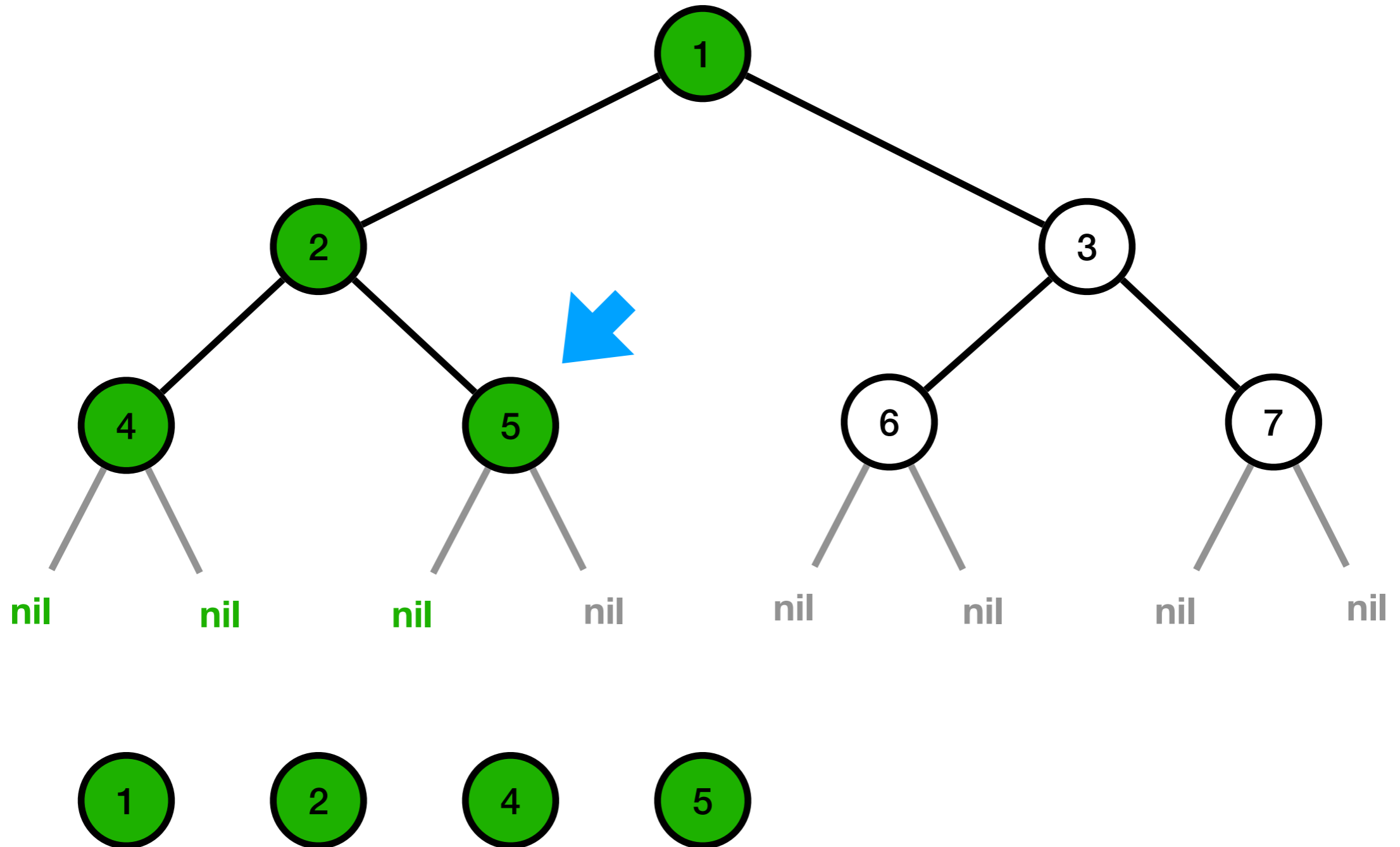
Parcours préfixe



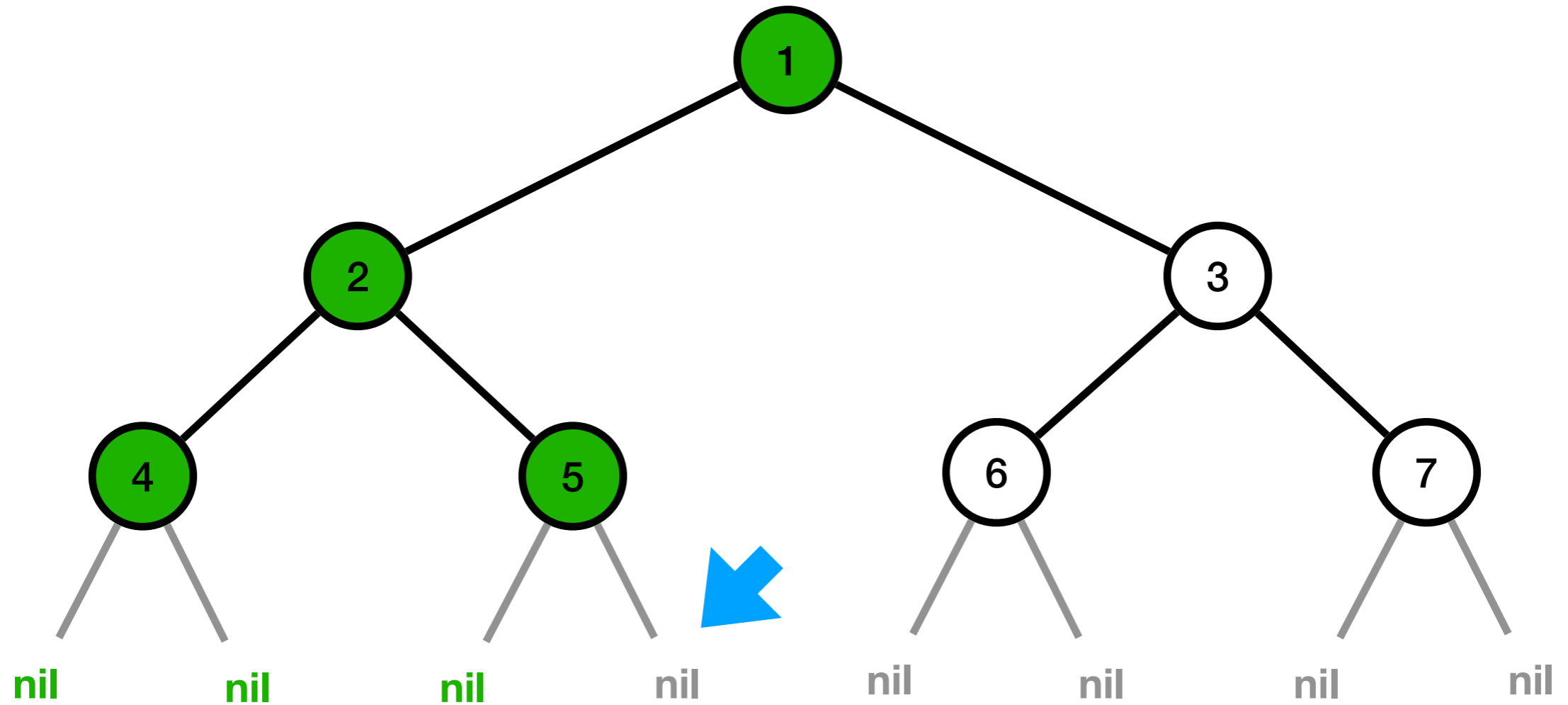
Parcours préfixe



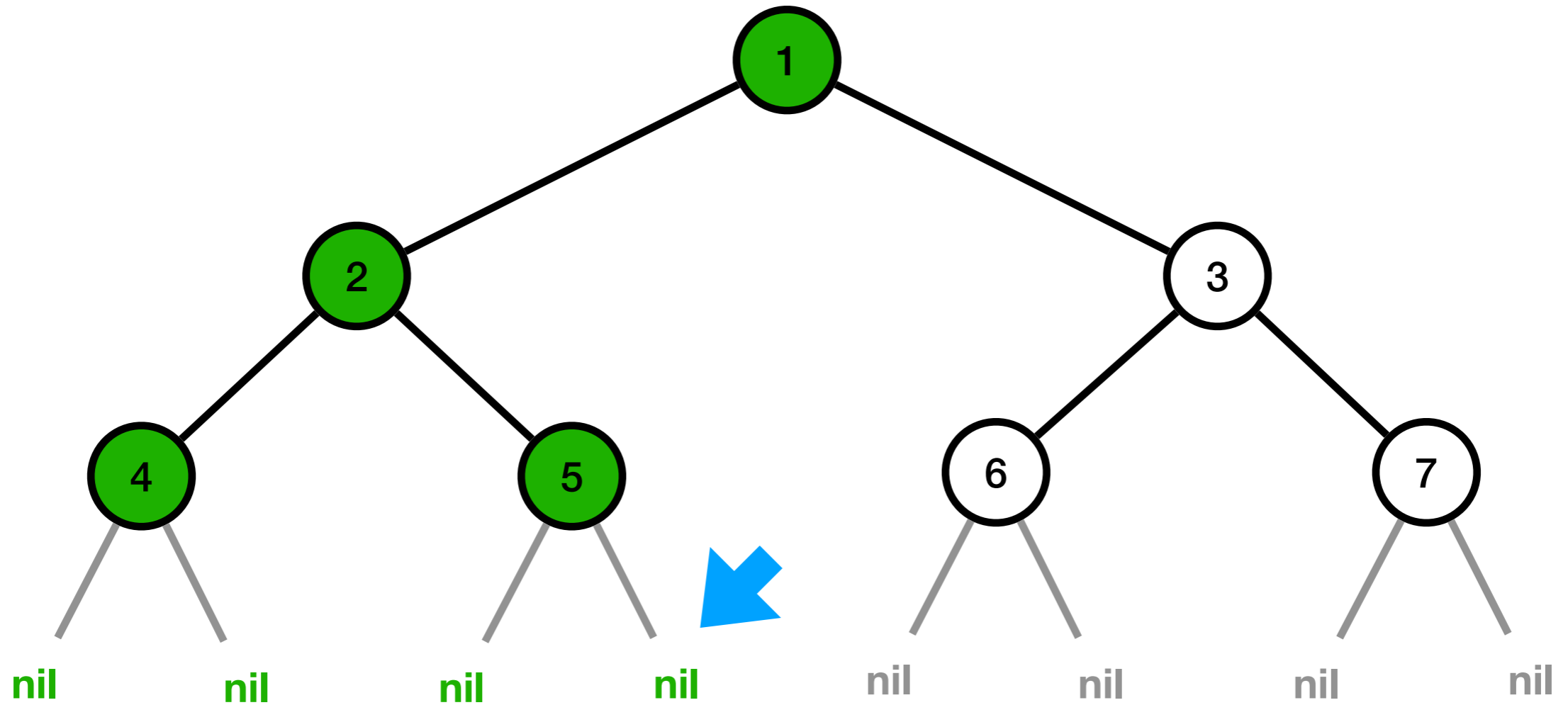
Parcours préfixe



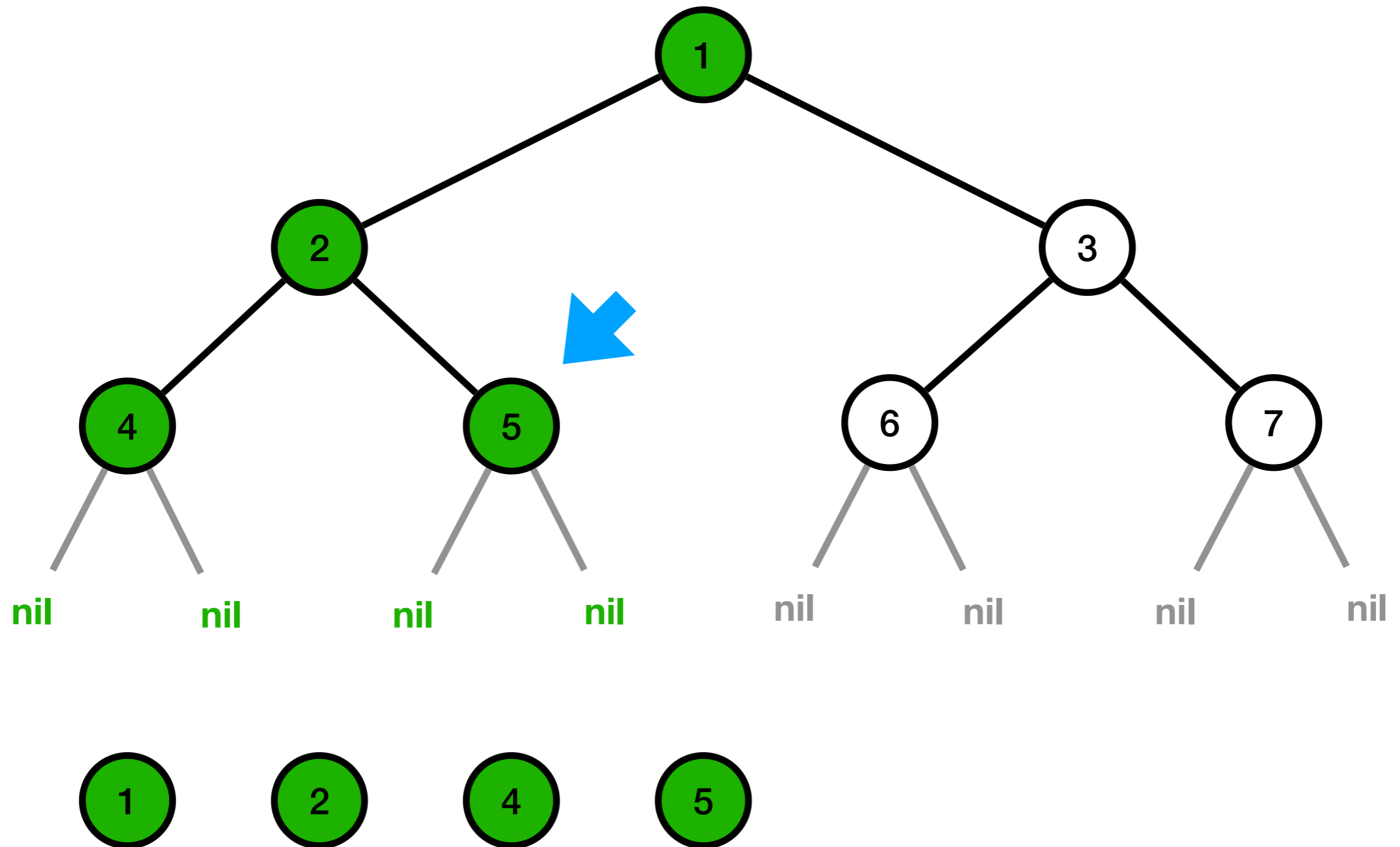
Parcours préfixe



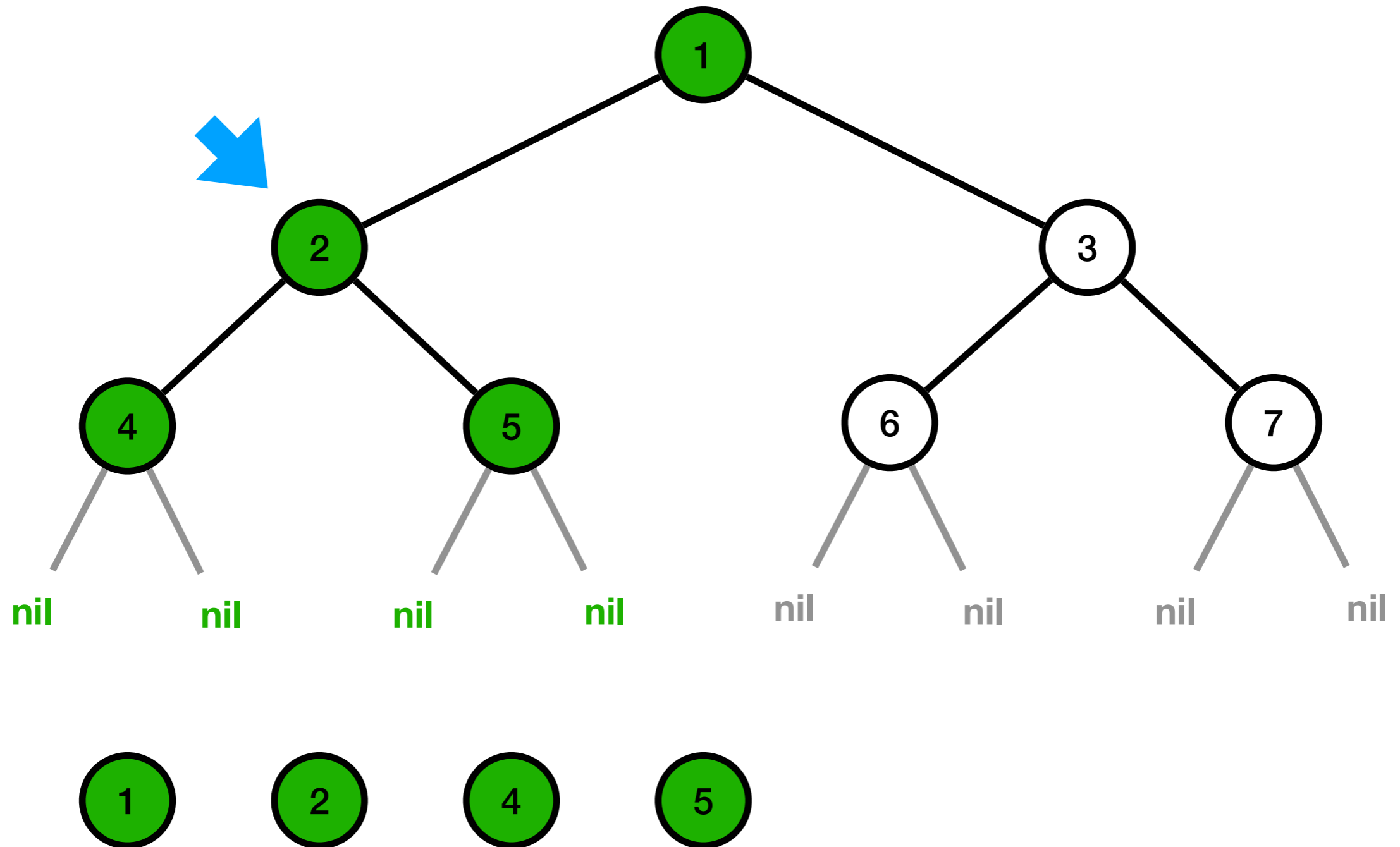
Parcours préfixe



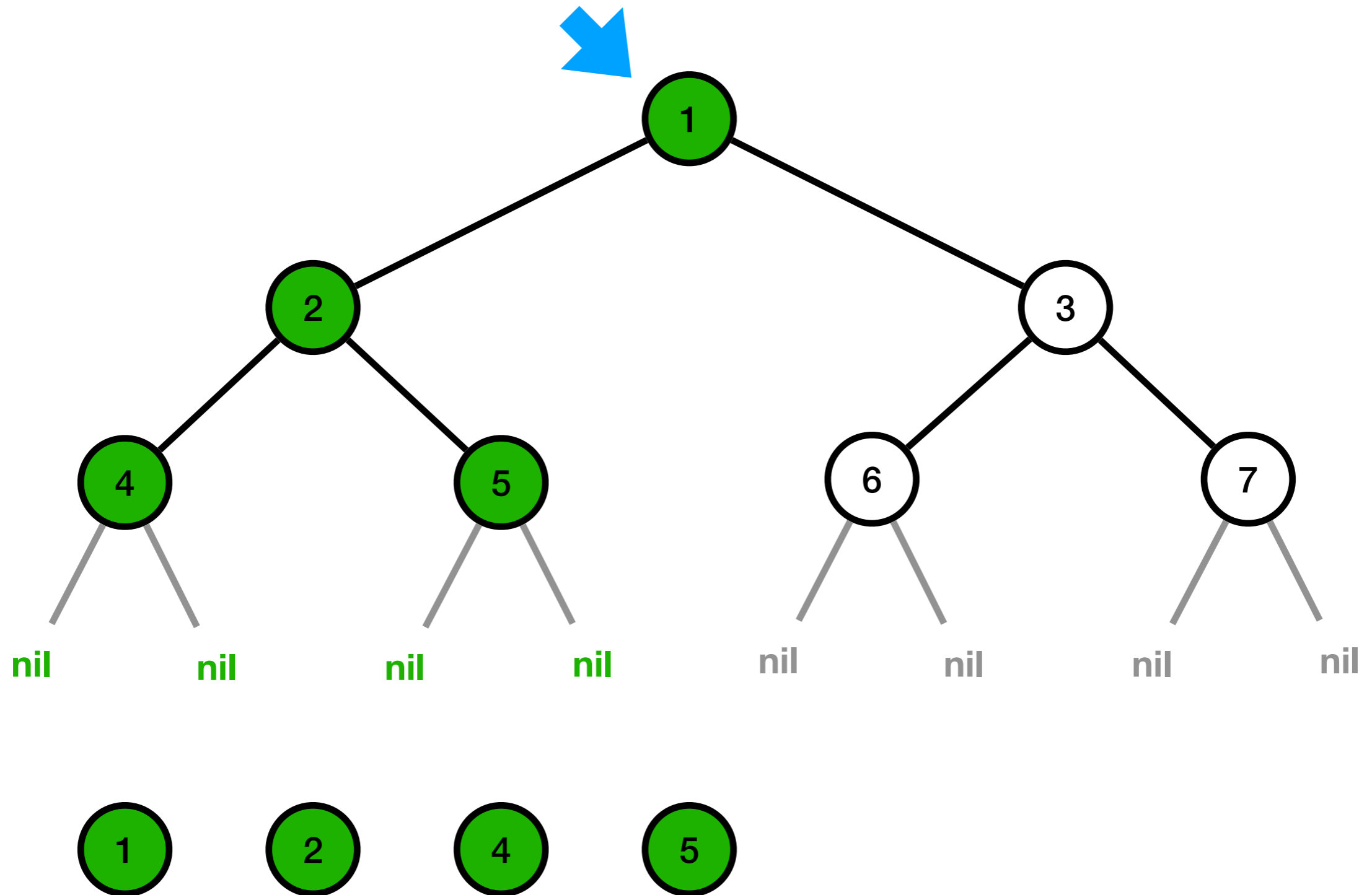
Parcours préfixe



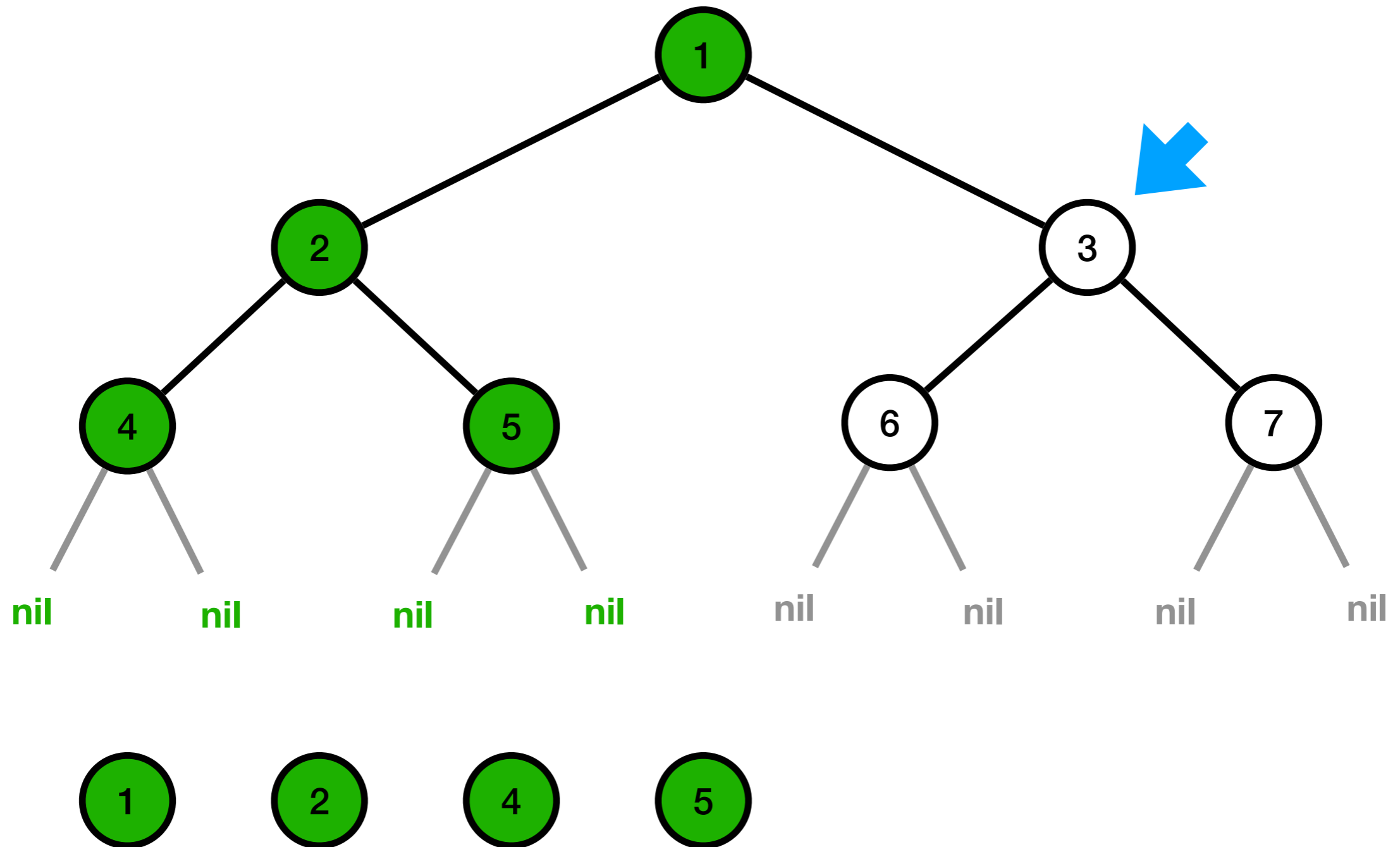
Parcours préfixe



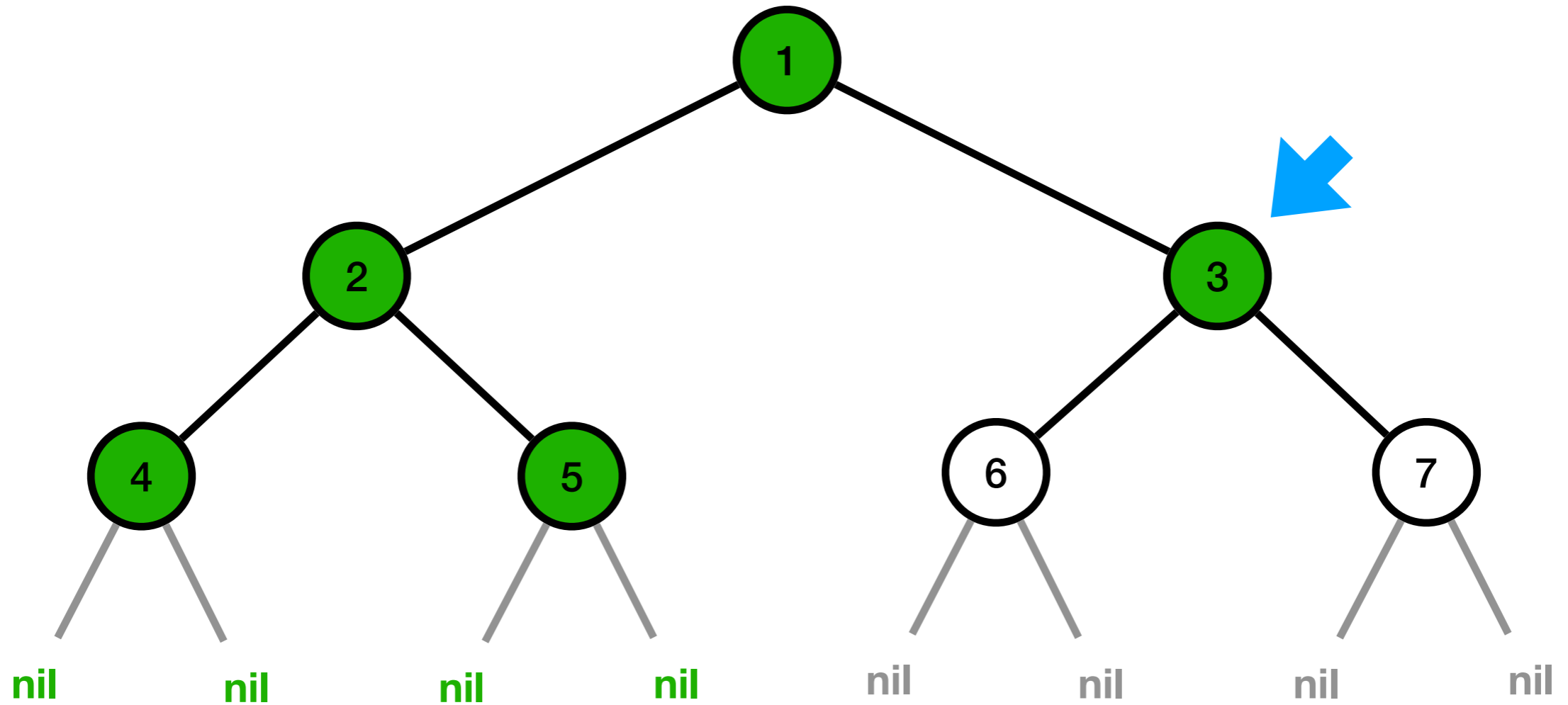
Parcours préfixe



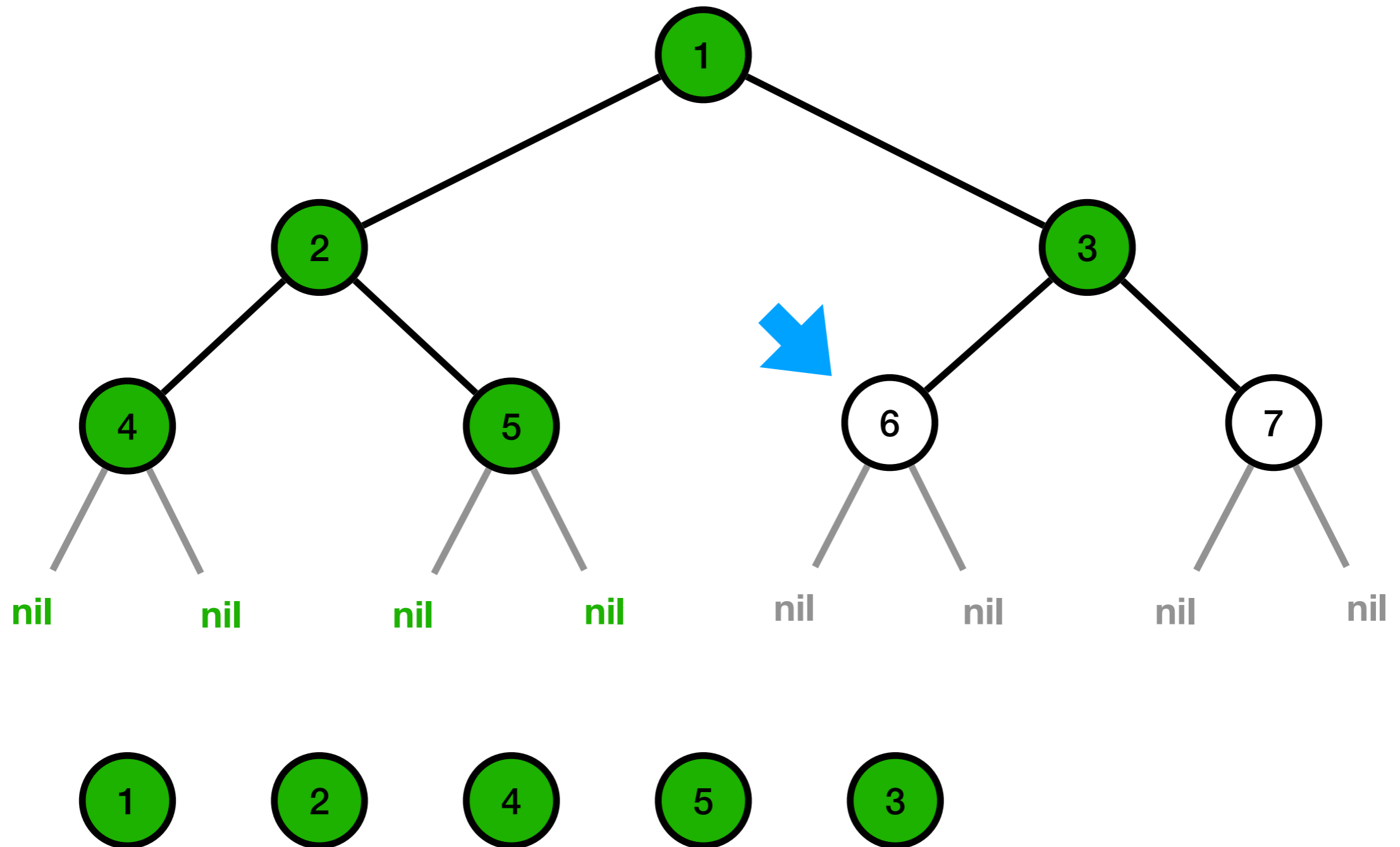
Parcours préfixe



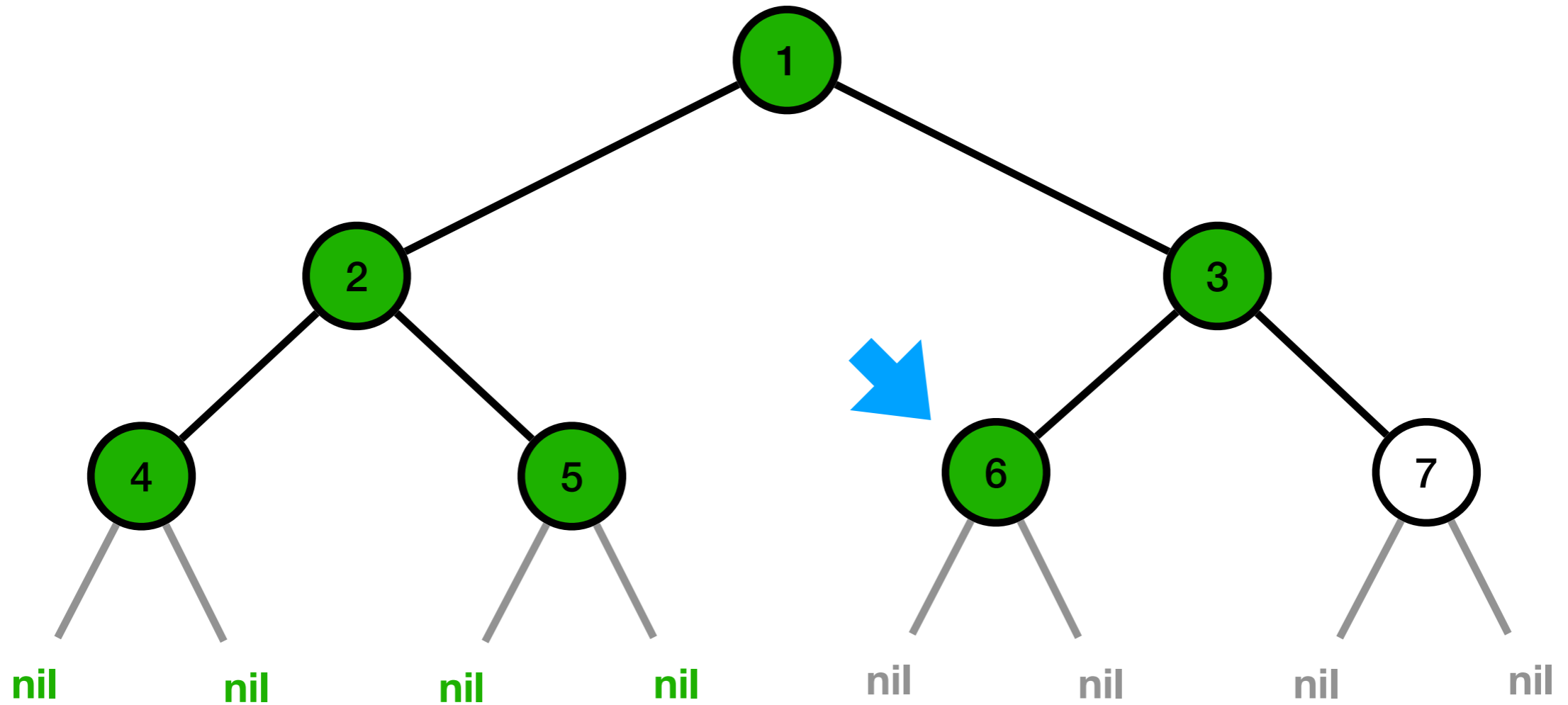
Parcours préfixe



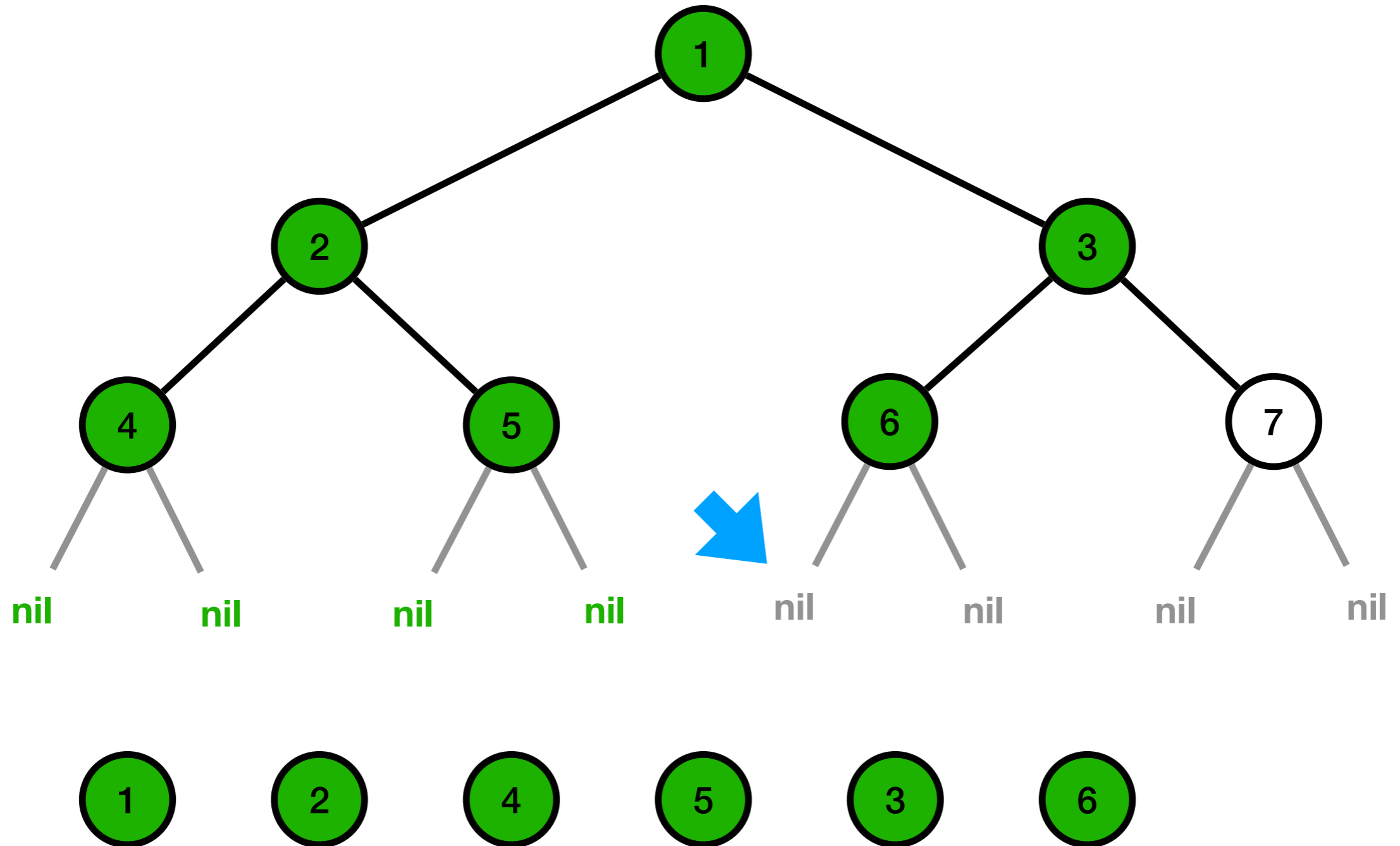
Parcours préfixe



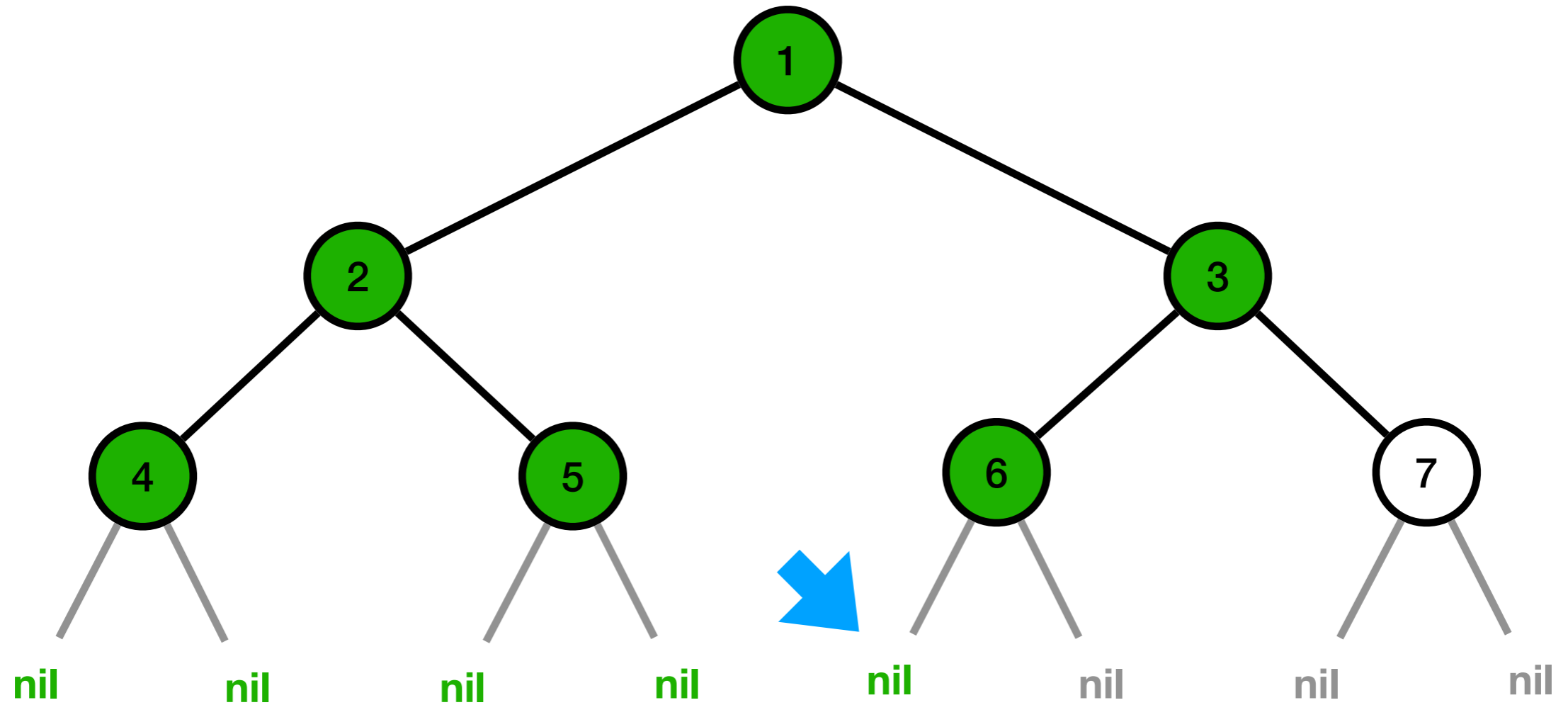
Parcours préfixe



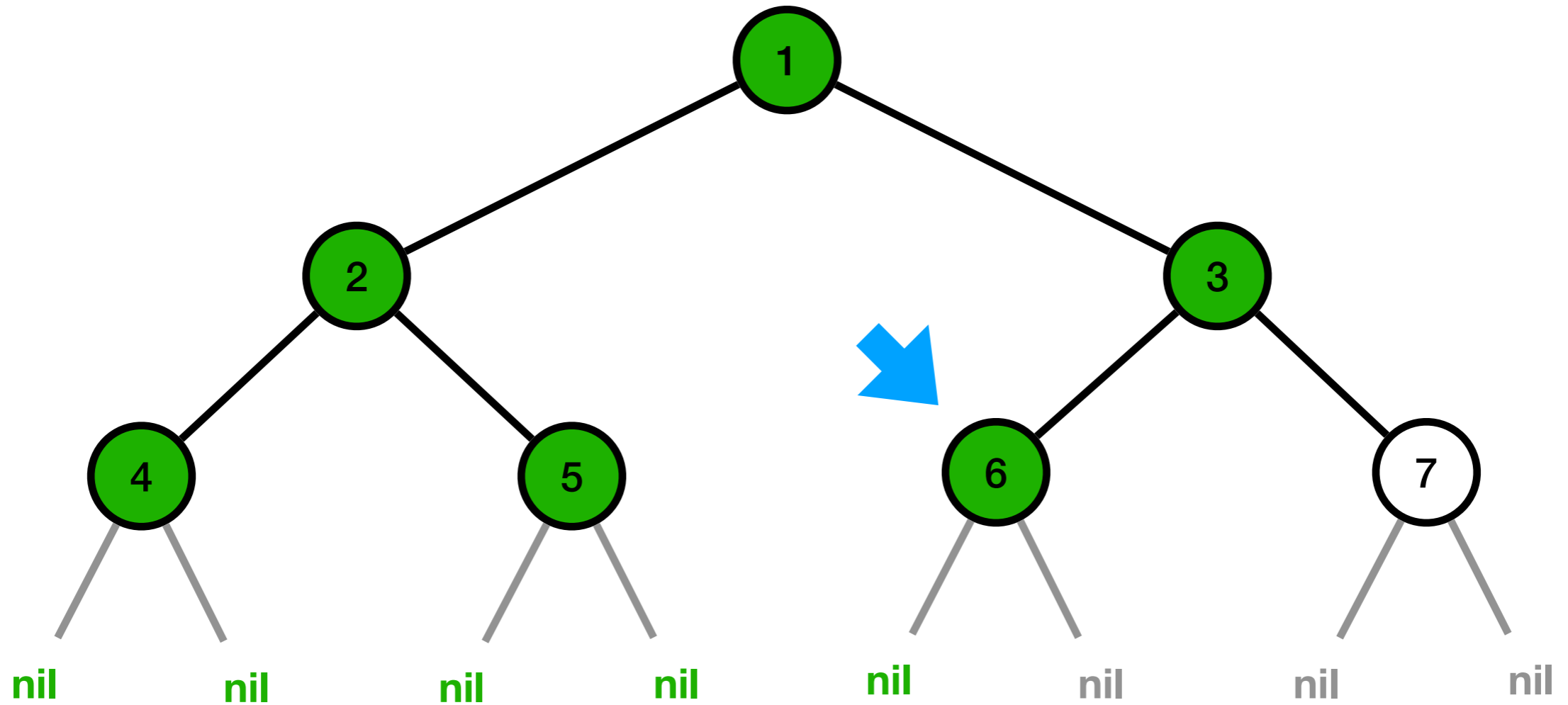
Parcours préfixe



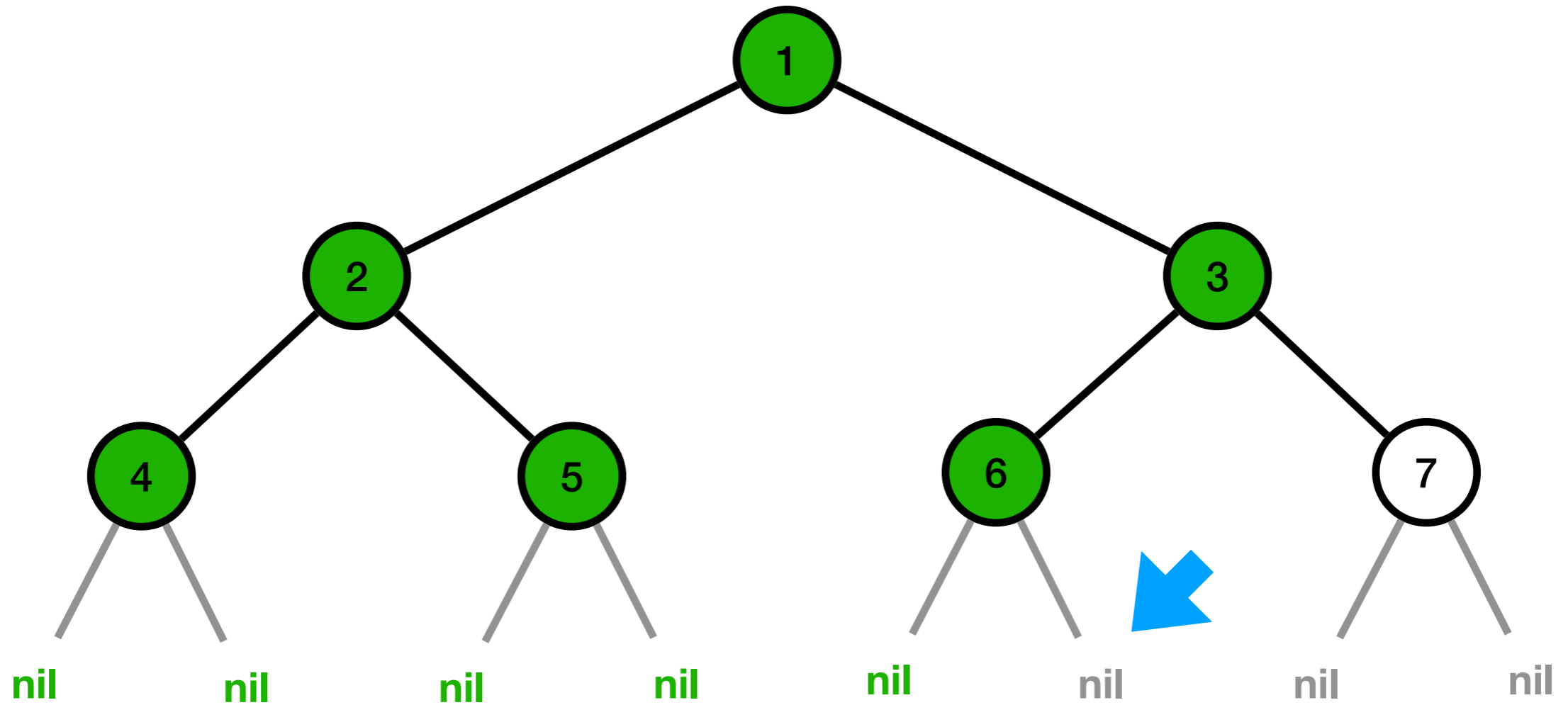
Parcours préfixe



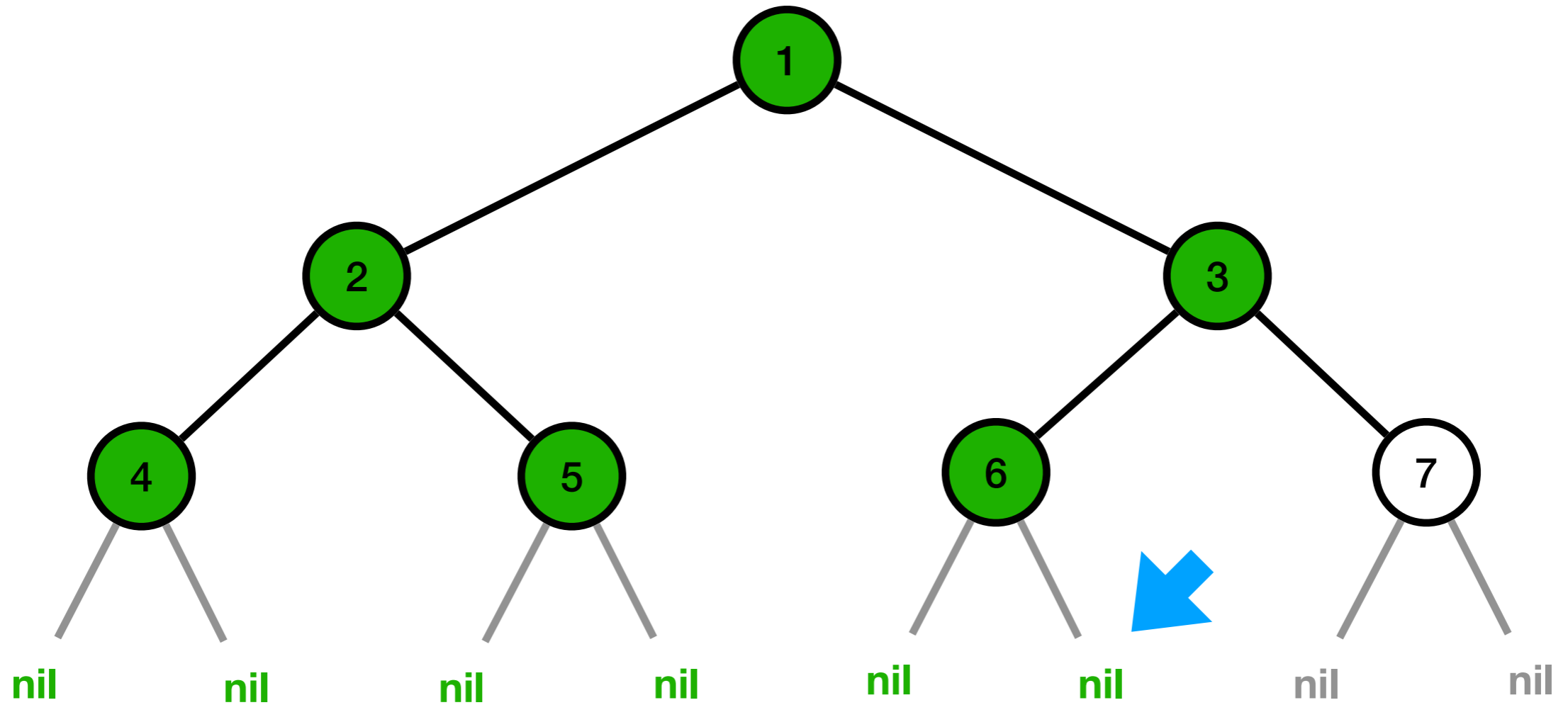
Parcours préfixe



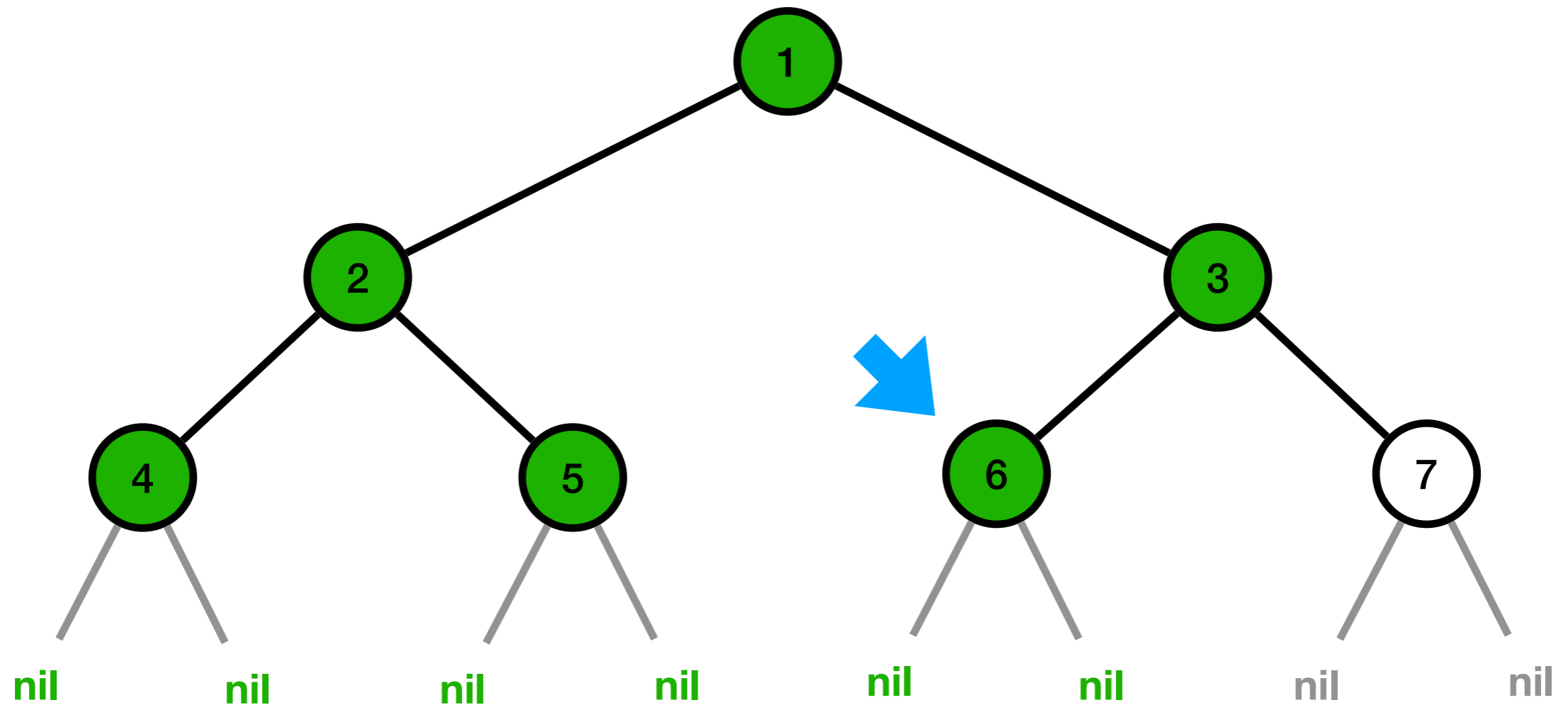
Parcours préfixe



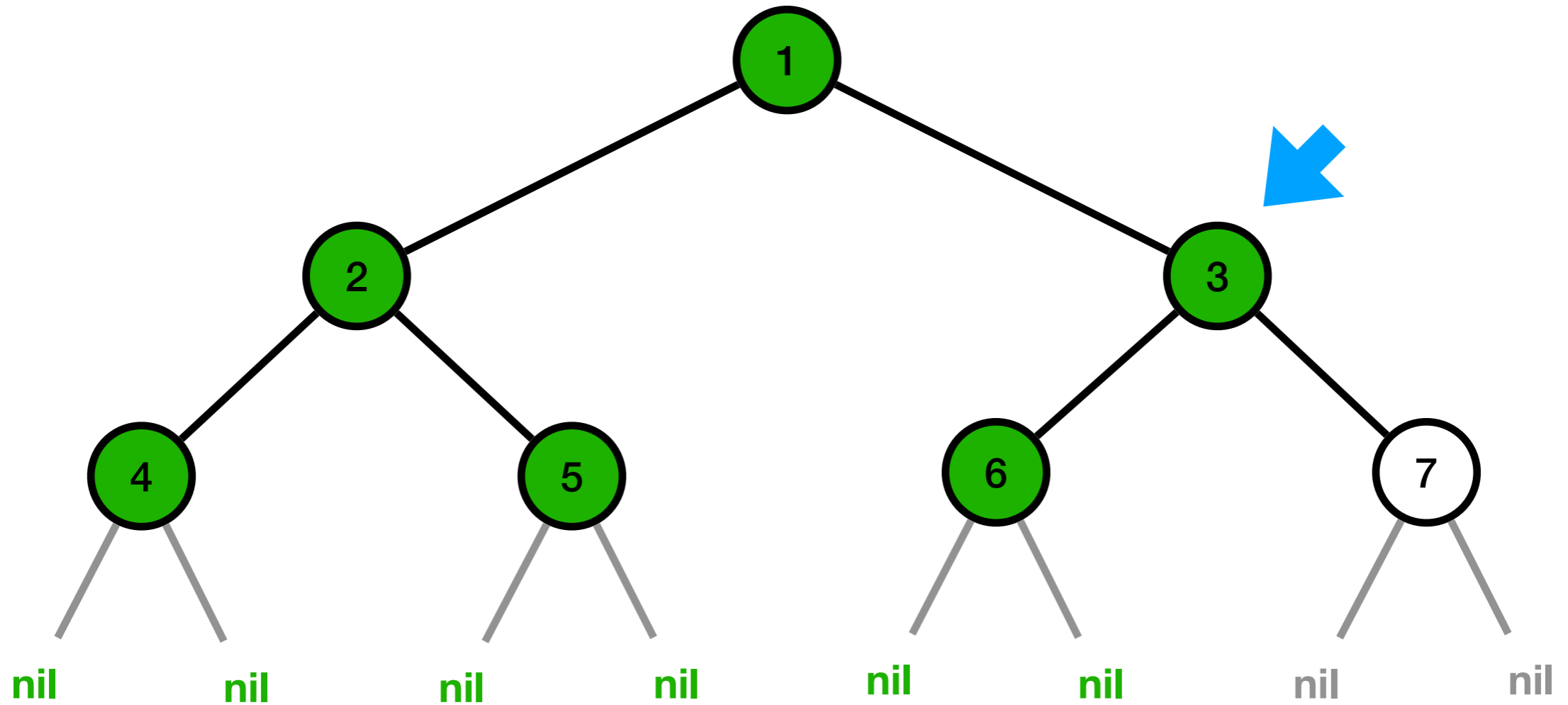
Parcours préfixe



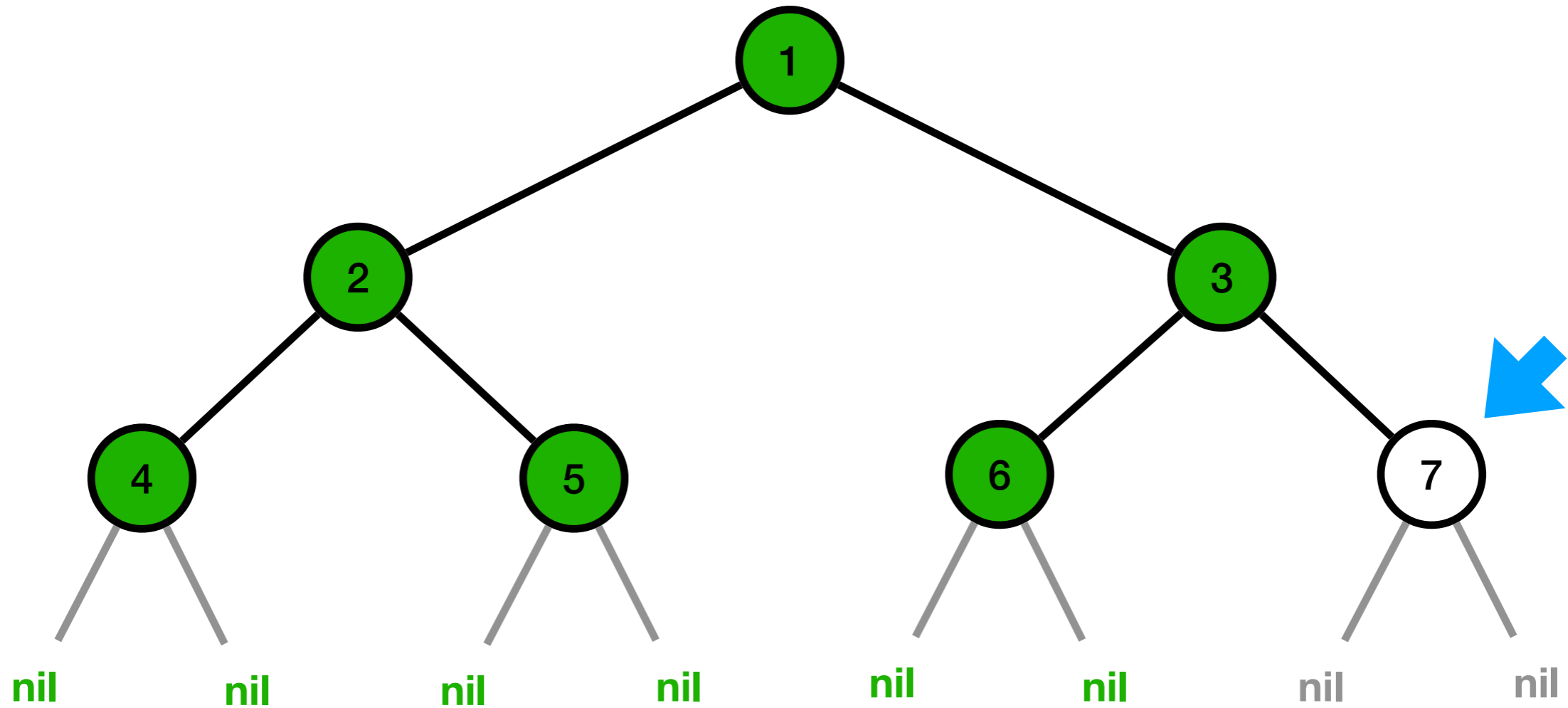
Parcours préfixe



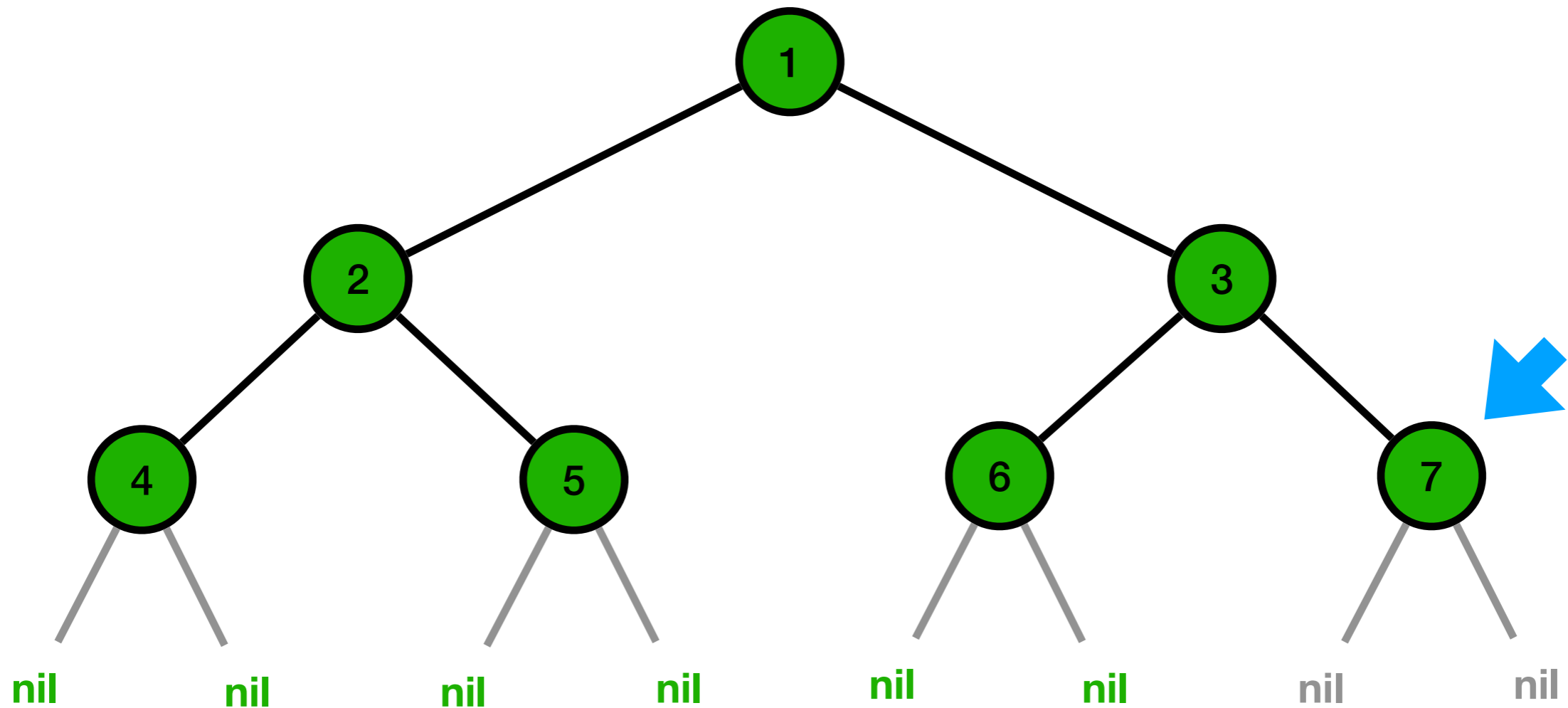
Parcours préfixe



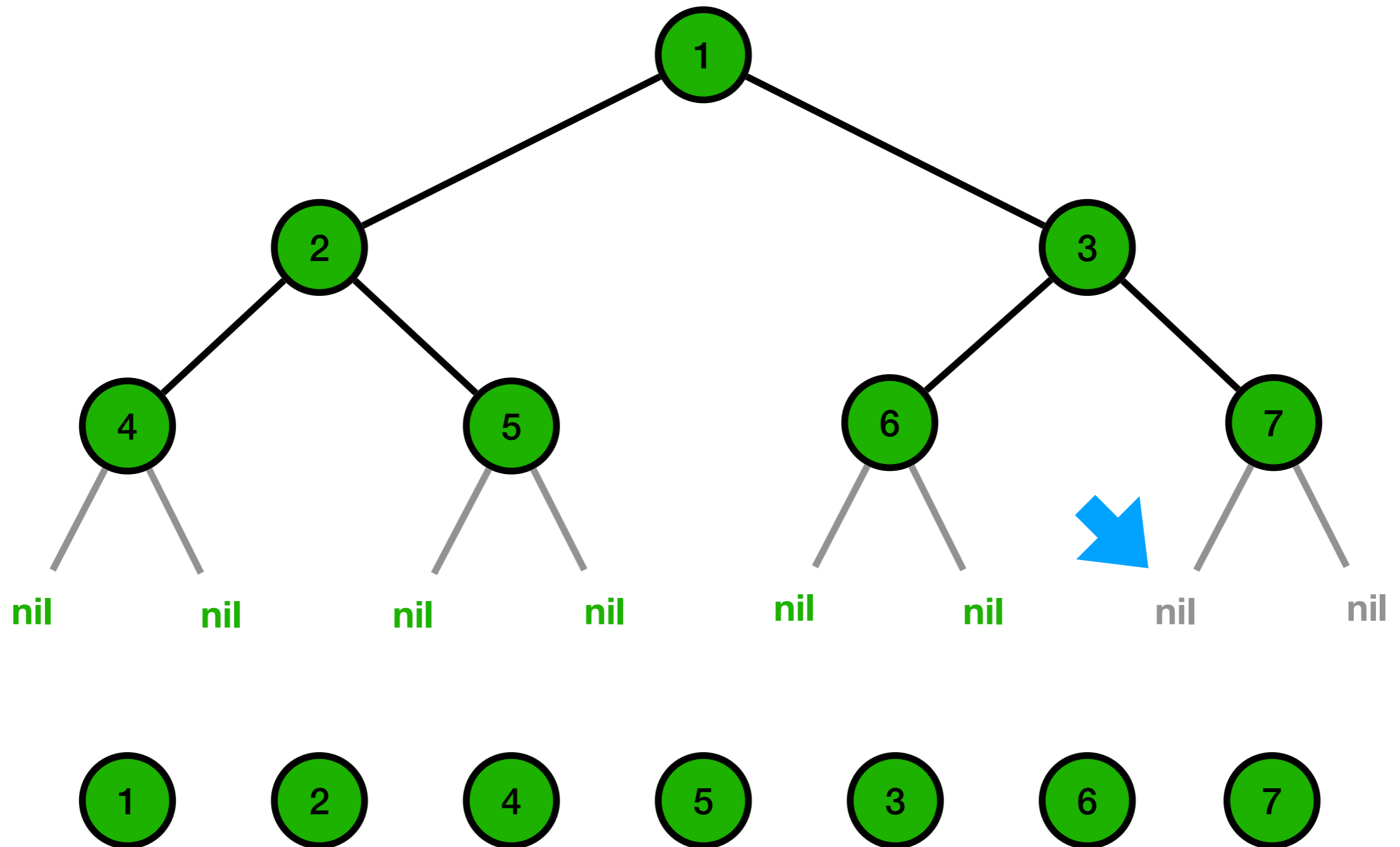
Parcours préfixe



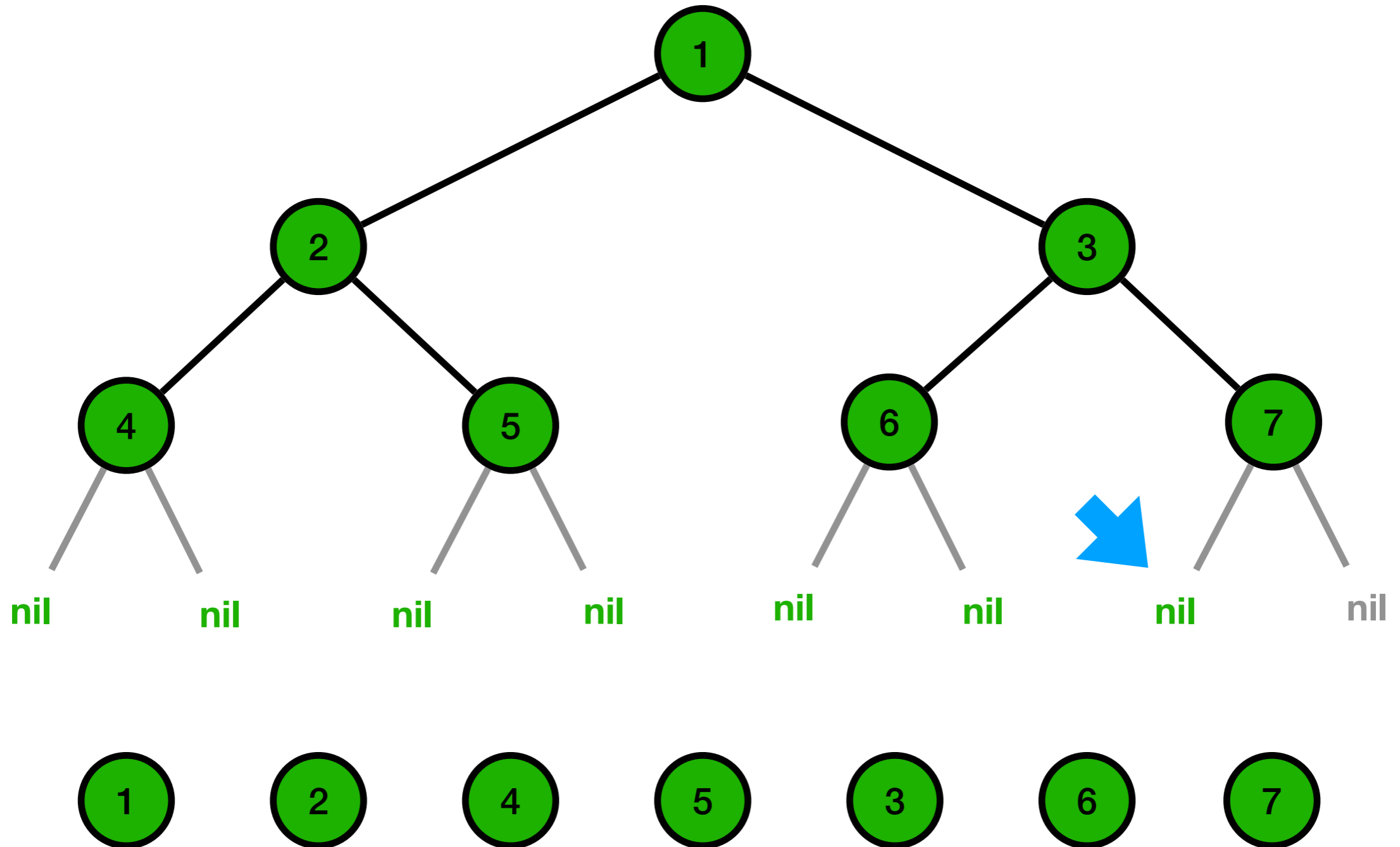
Parcours préfixe



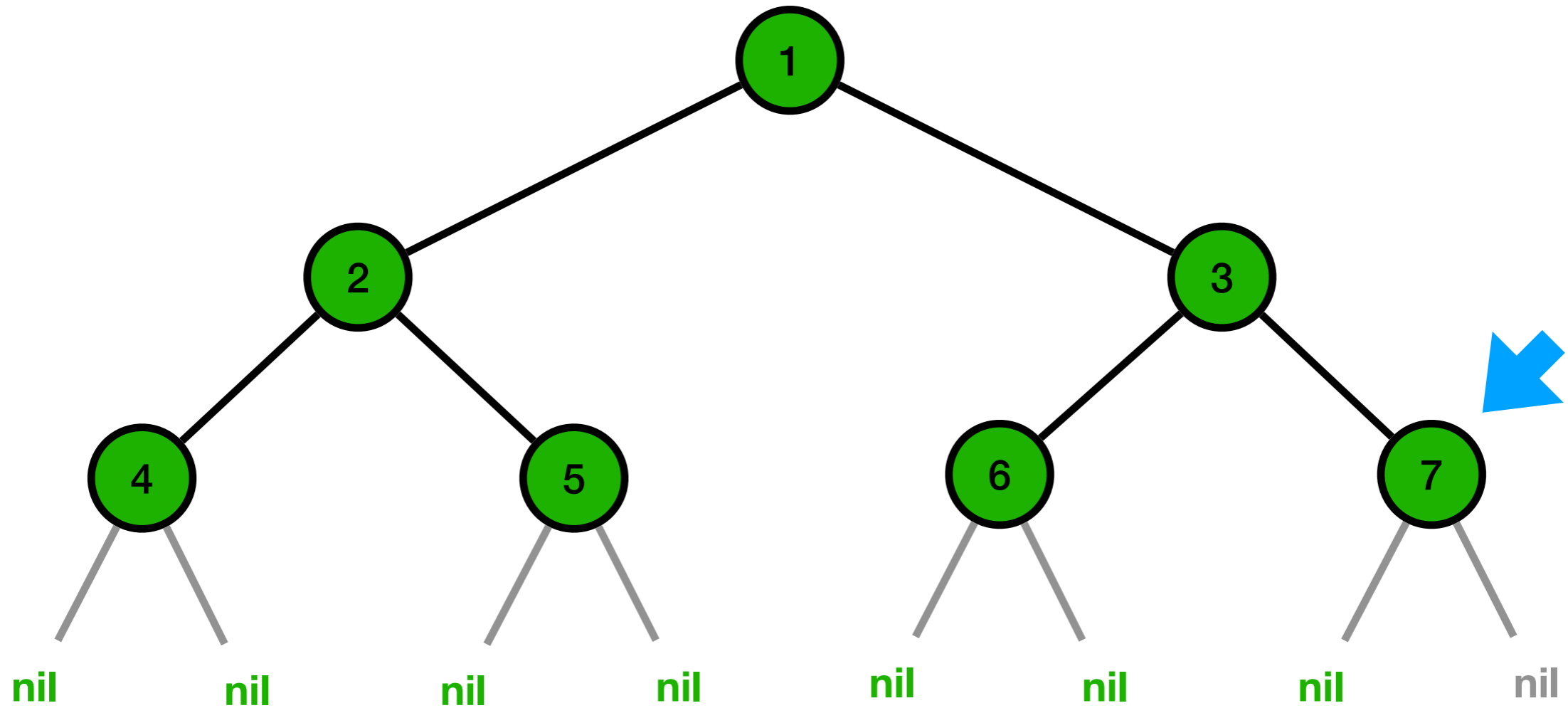
Parcours préfixe



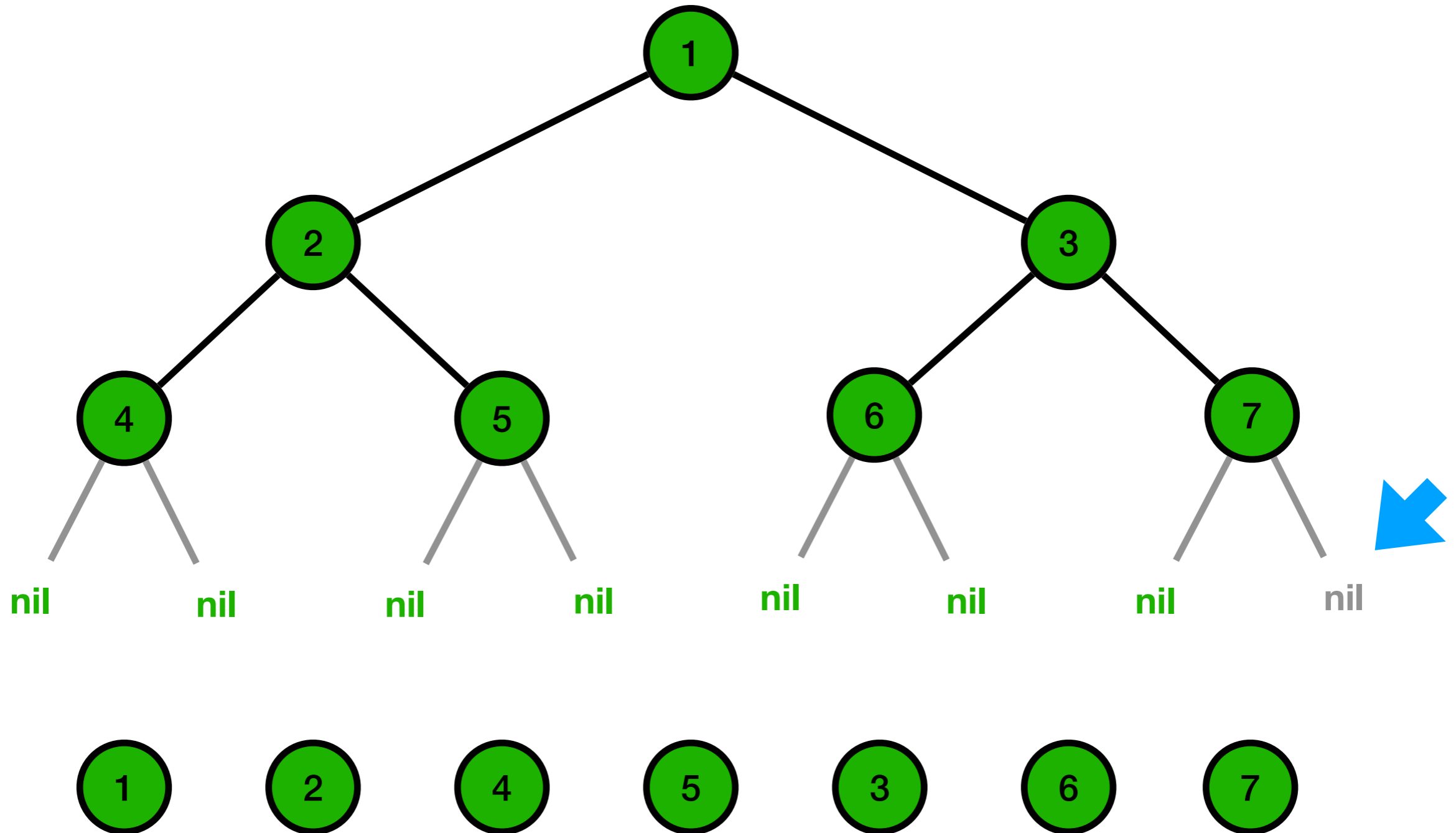
Parcours préfixe



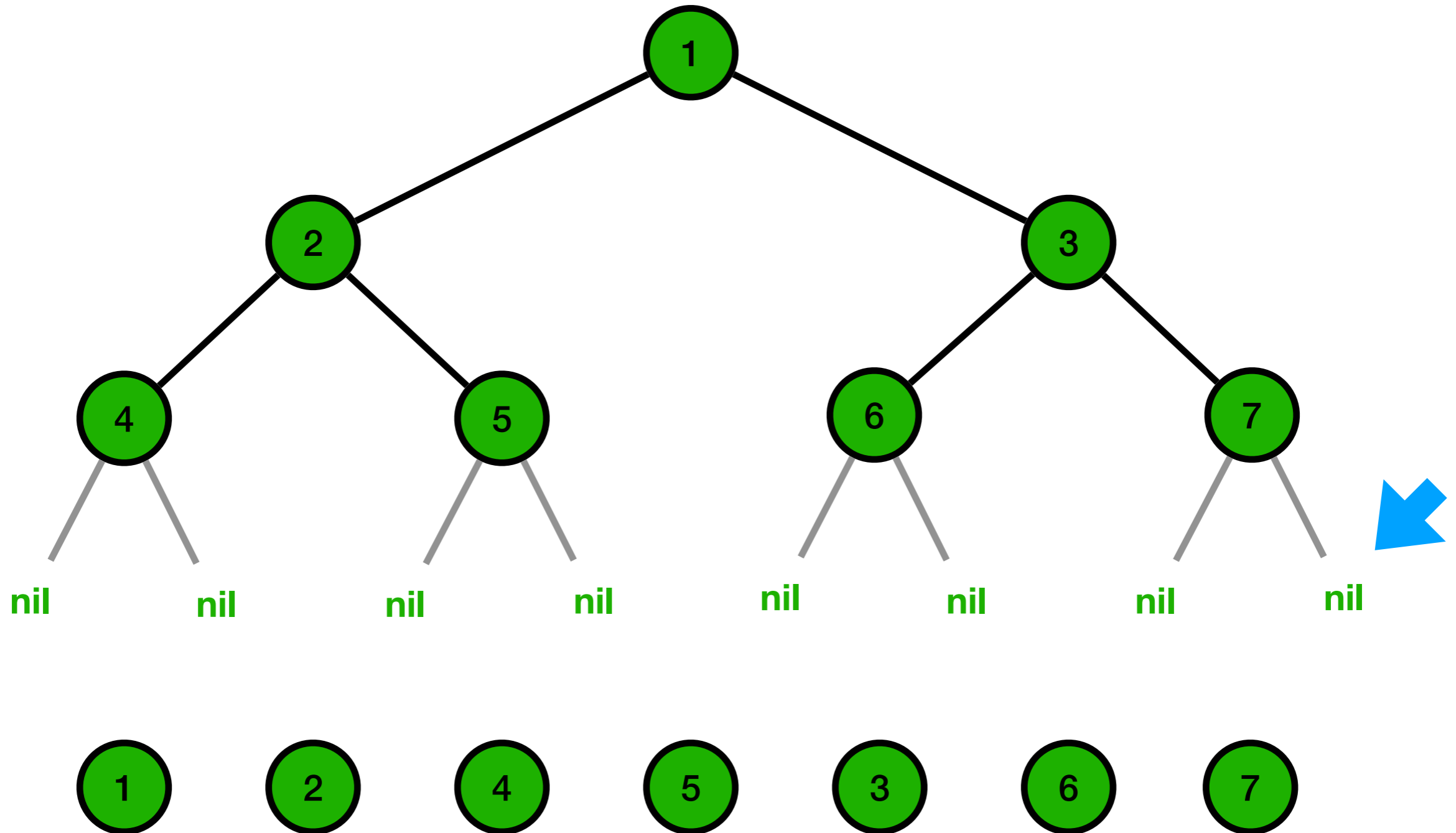
Parcours préfixe



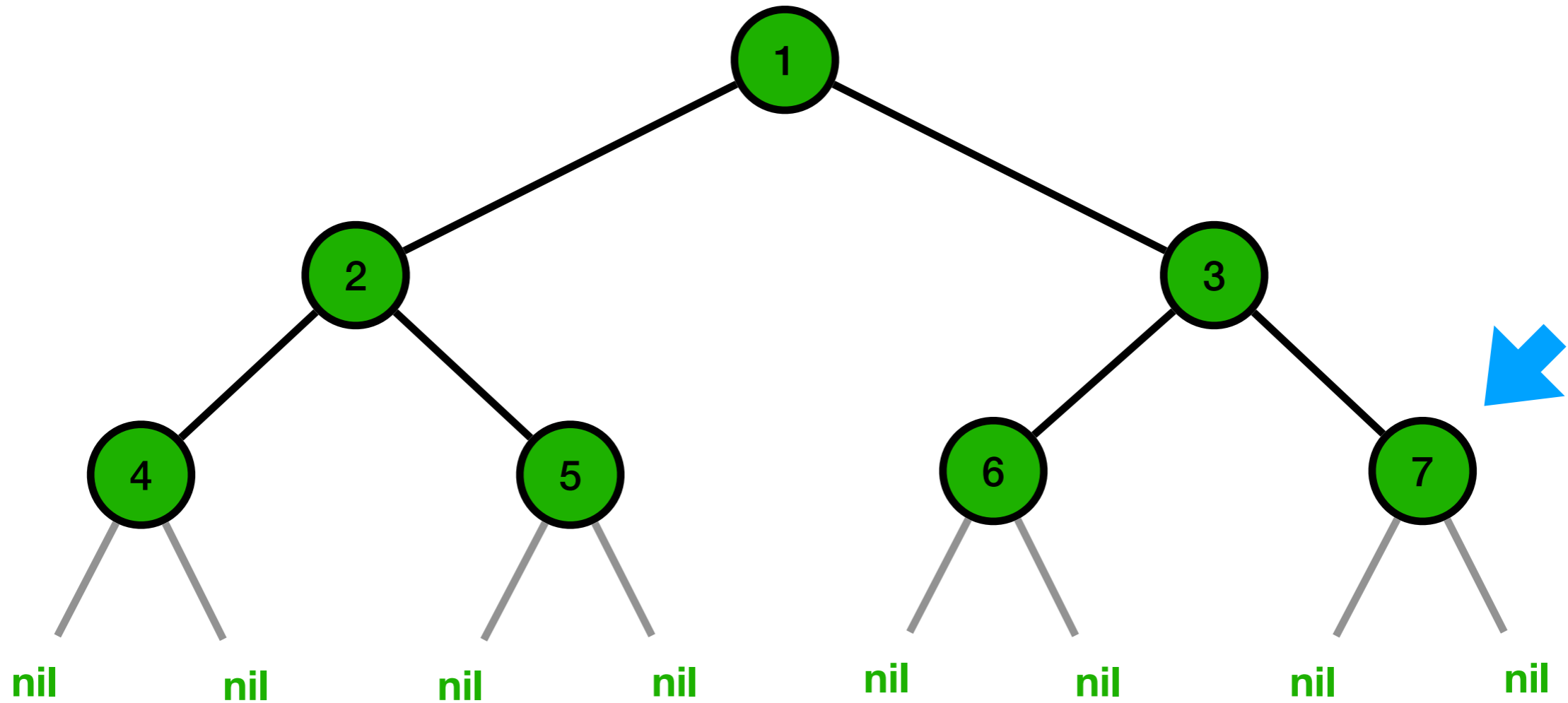
Parcours préfixe



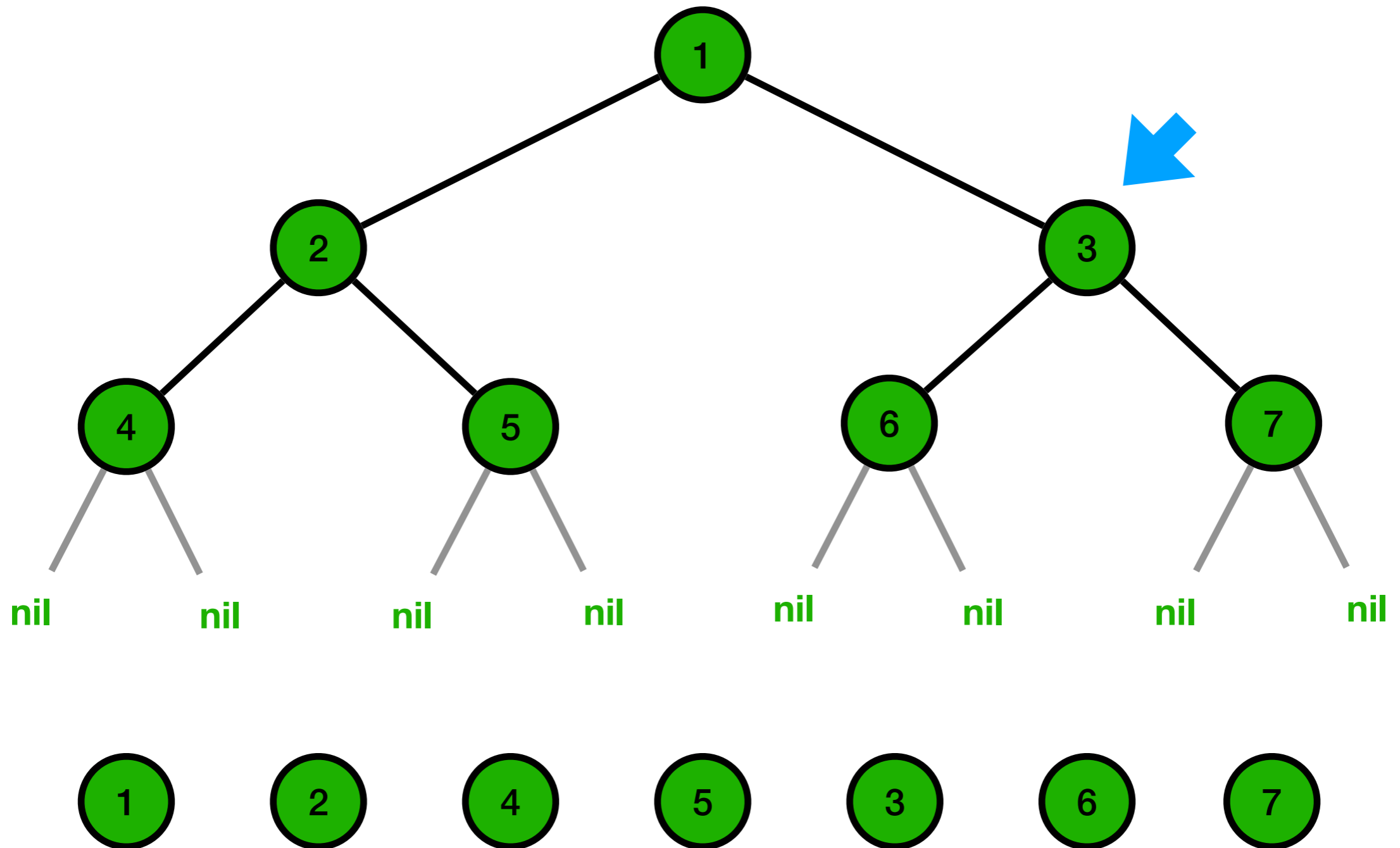
Parcours préfixe



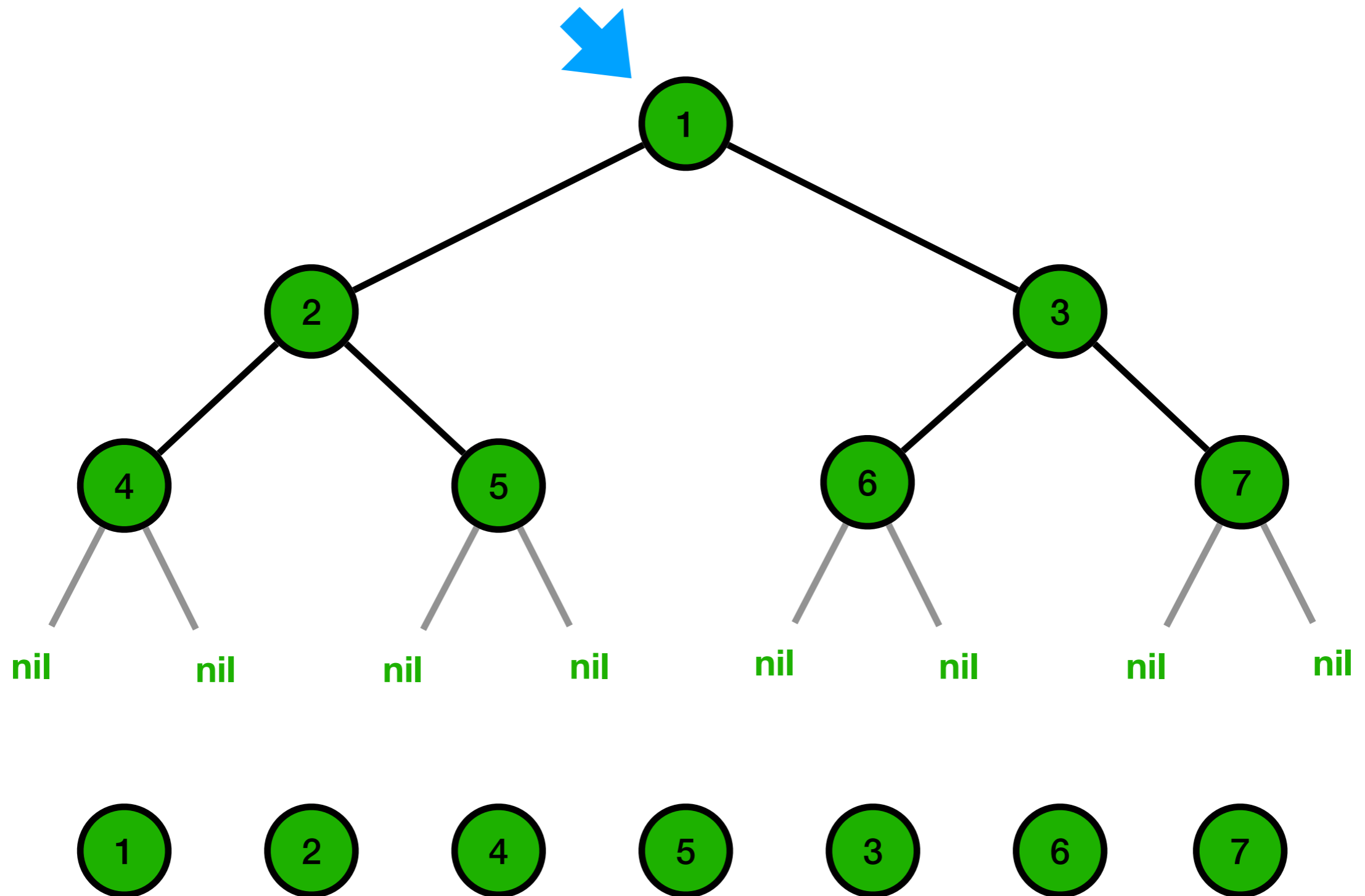
Parcours préfixe



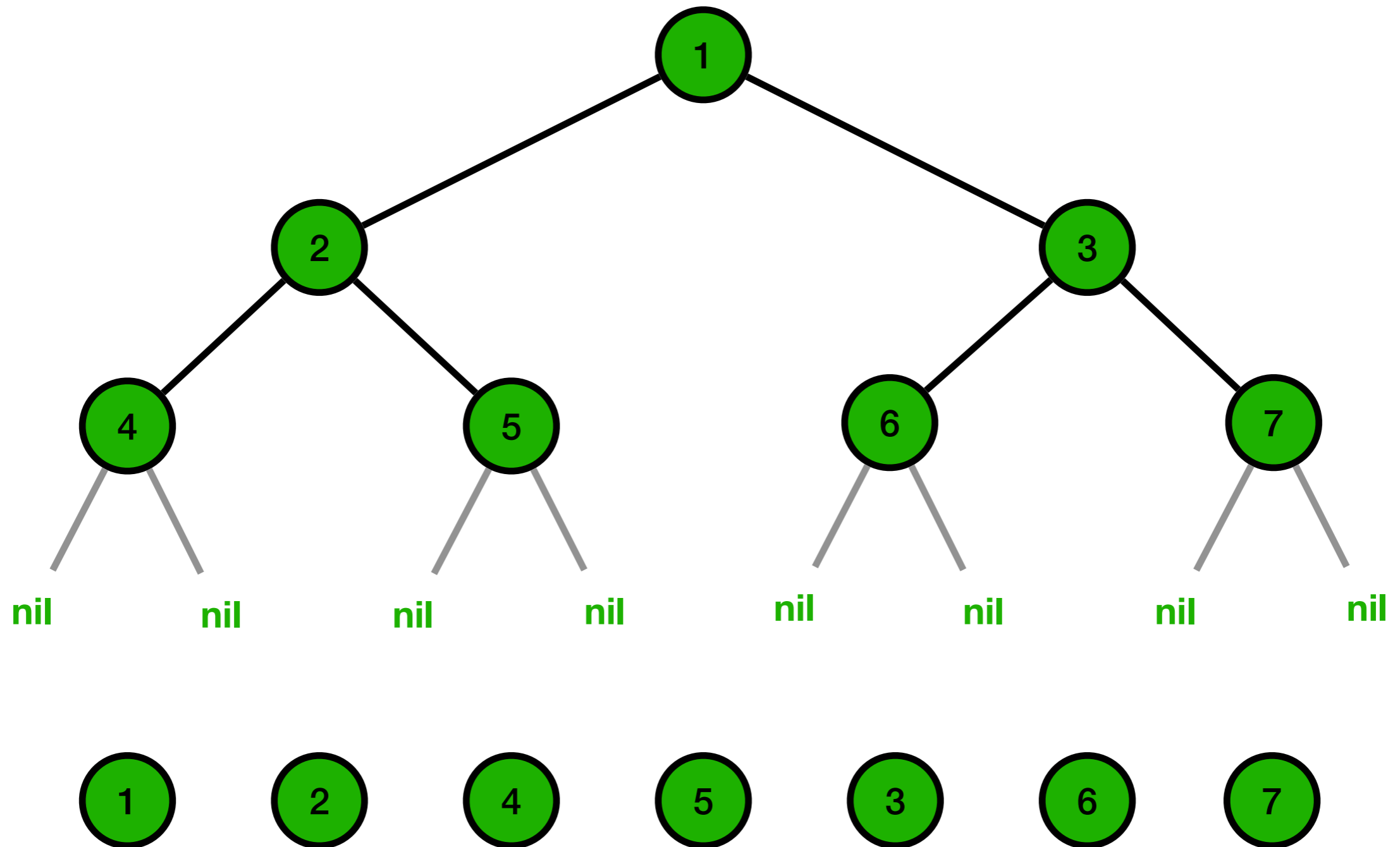
Parcours préfixe



Parcours préfixe



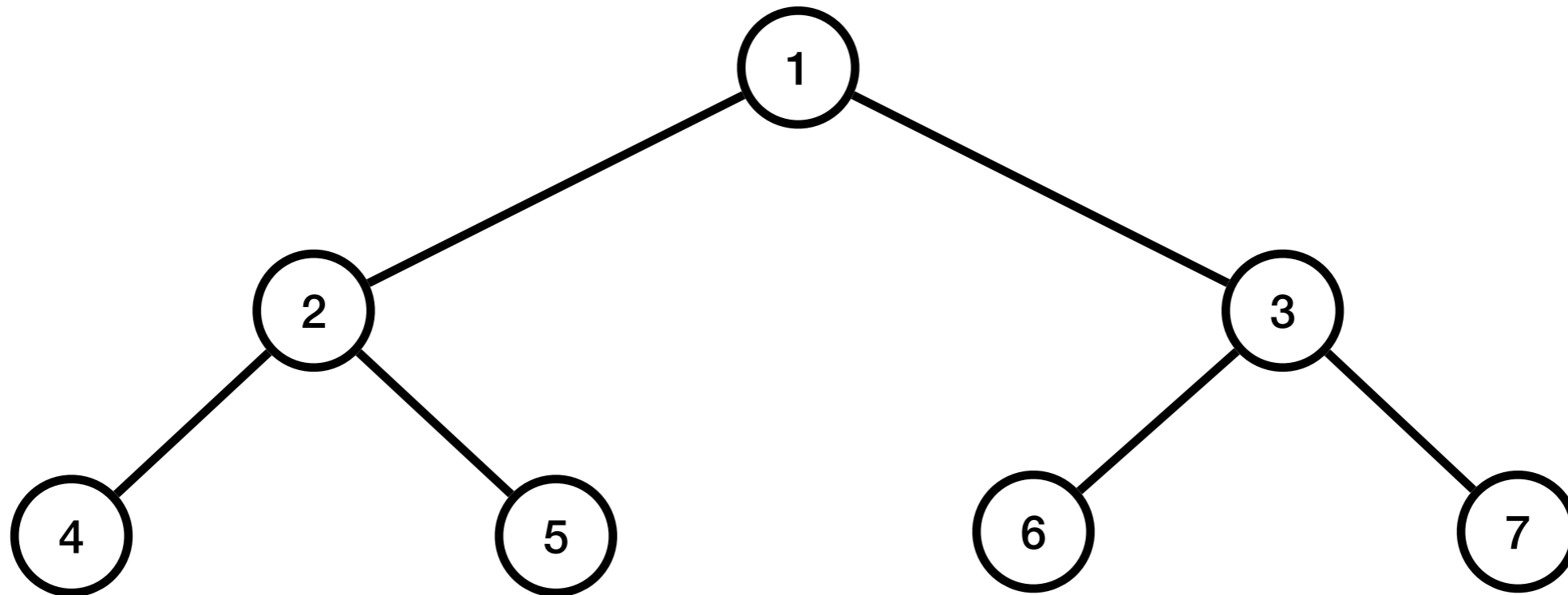
Parcours préfixe



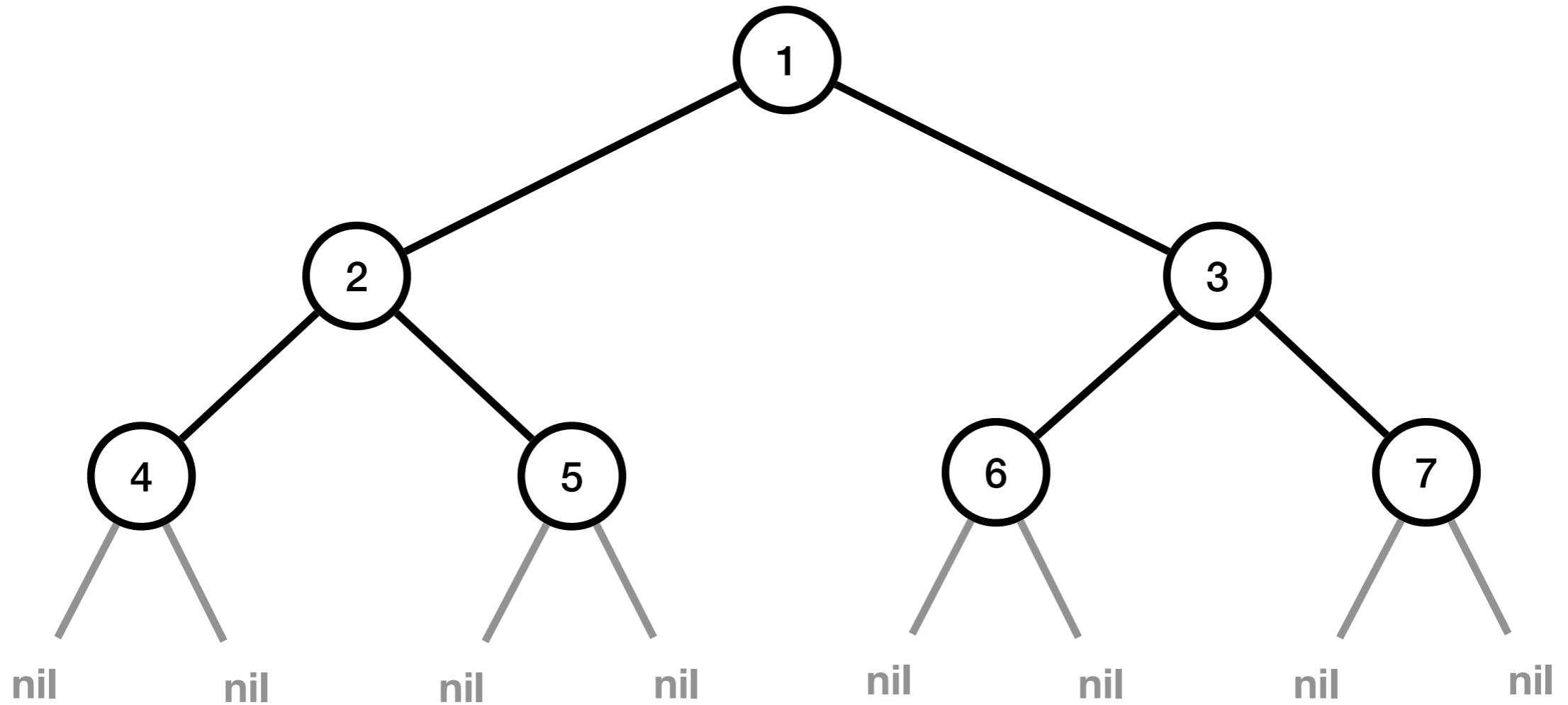
Parcours postfixe (ou suffixe)

```
def parcours_postfixe(arbre):  
    if not est_vide(arbre):  
        parcours_postfixe(enfant_gauche(arbre))  
        parcours_postfixe(enfant_droit(arbre))  
        print(valeur(arbre))
```

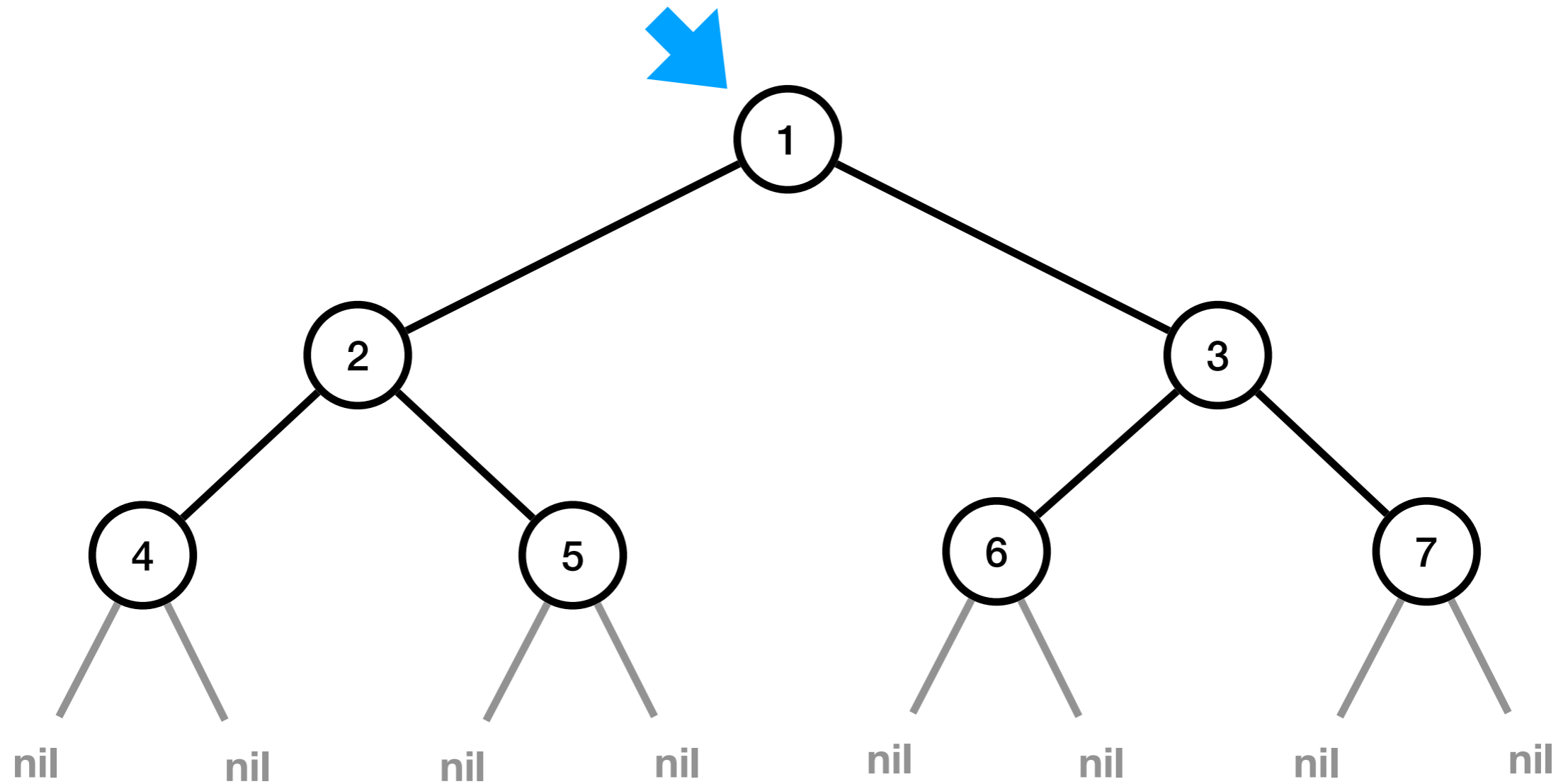

Parcours postfixe



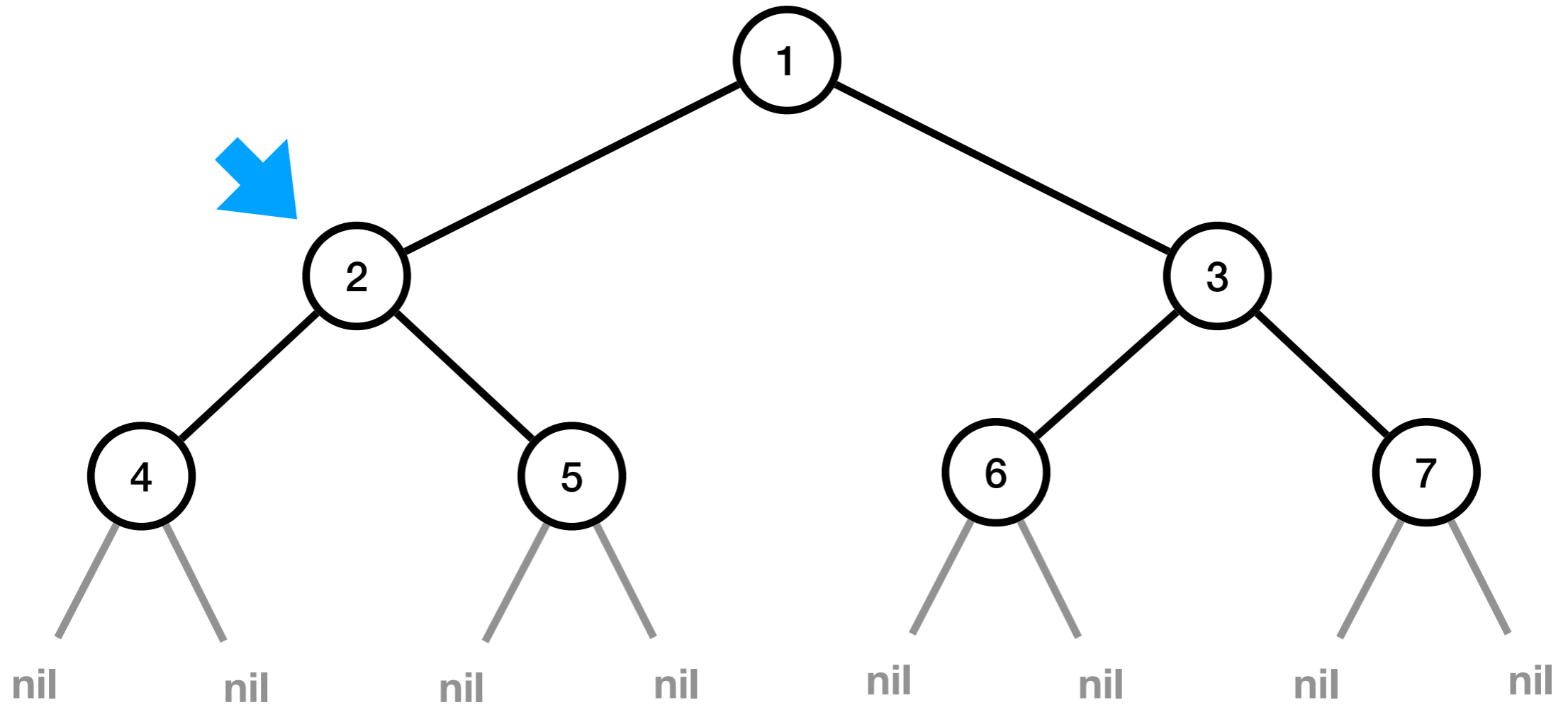
Parcours postfixe



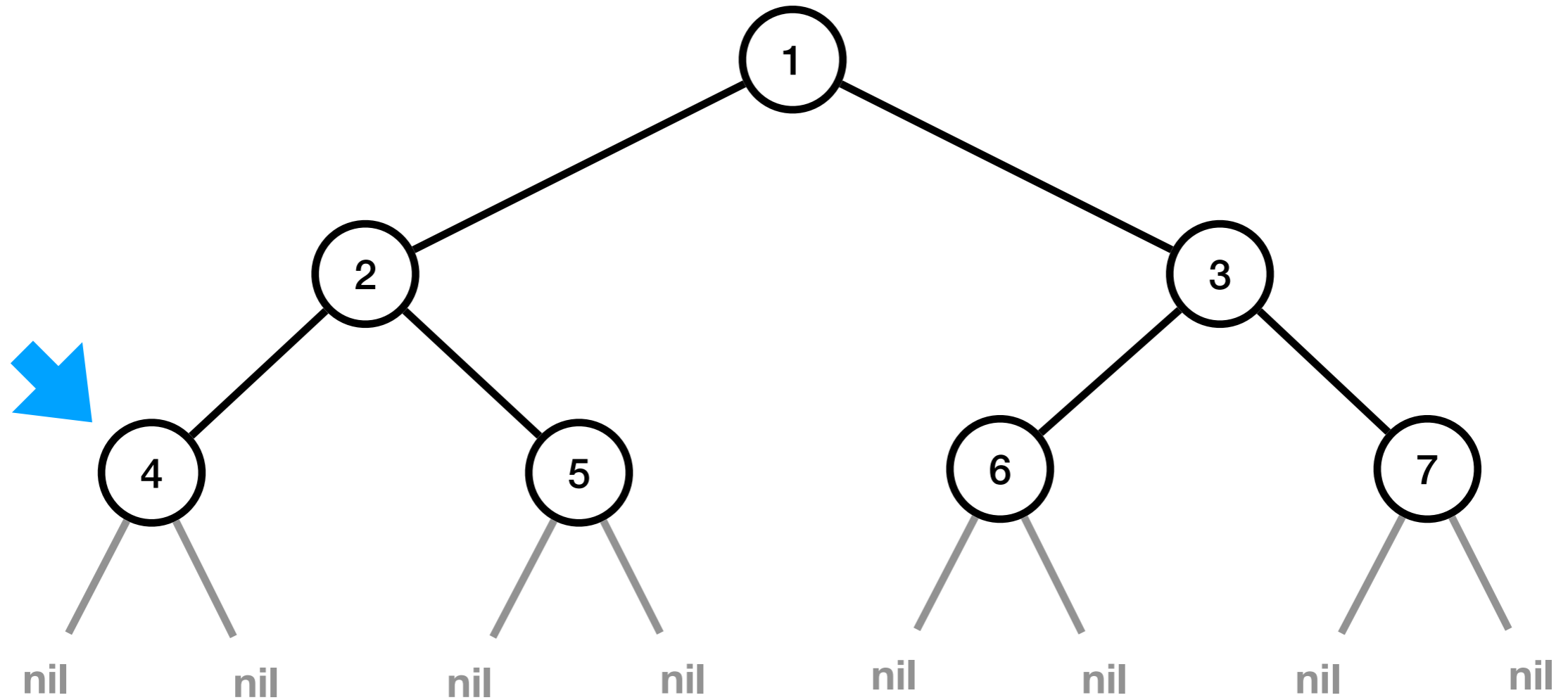
Parcours postfixe



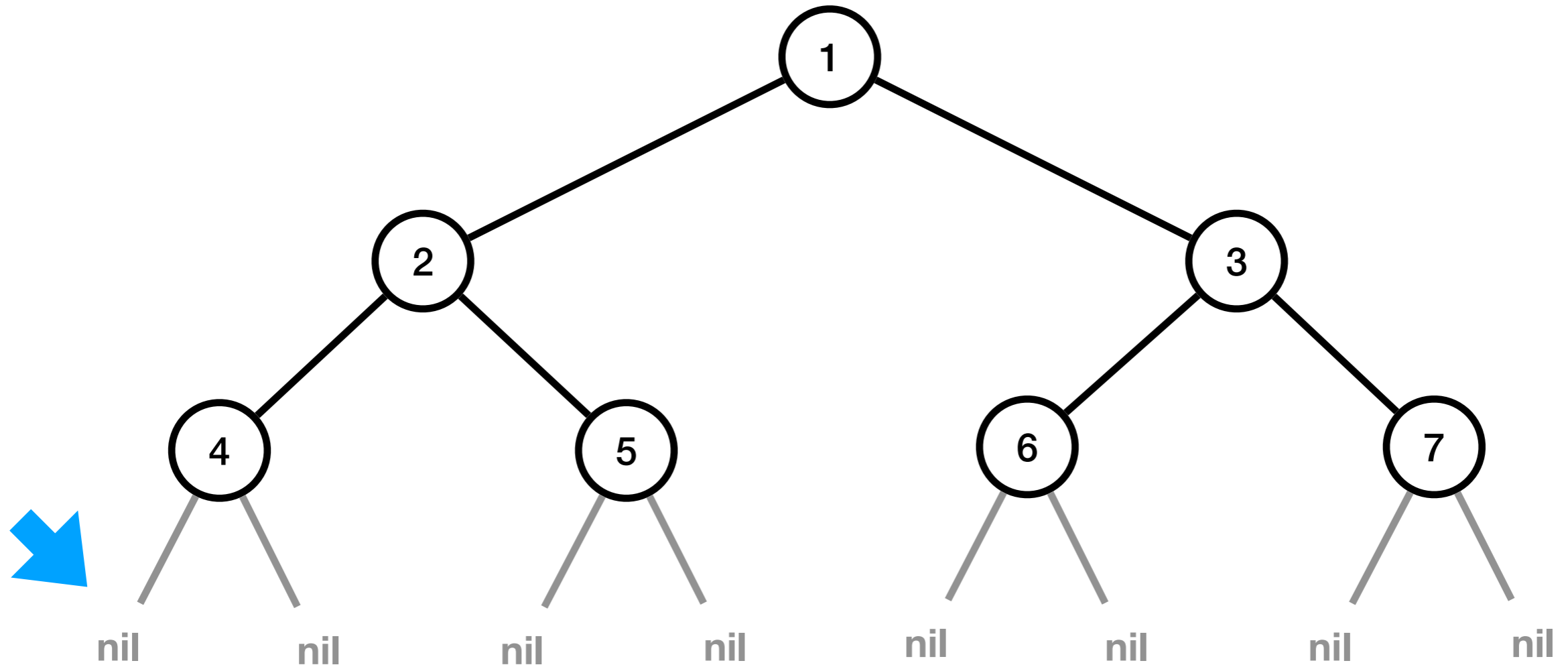
Parcours postfixe



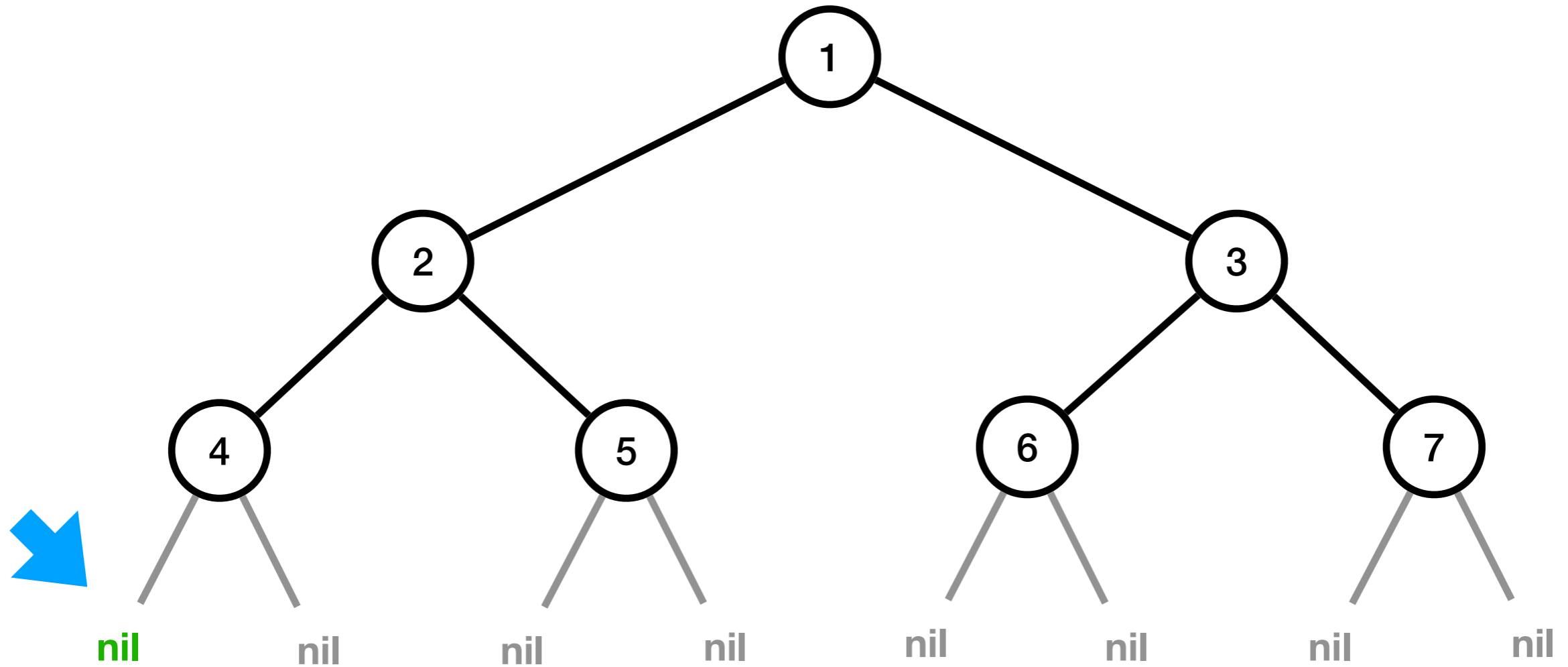
Parcours postfixe



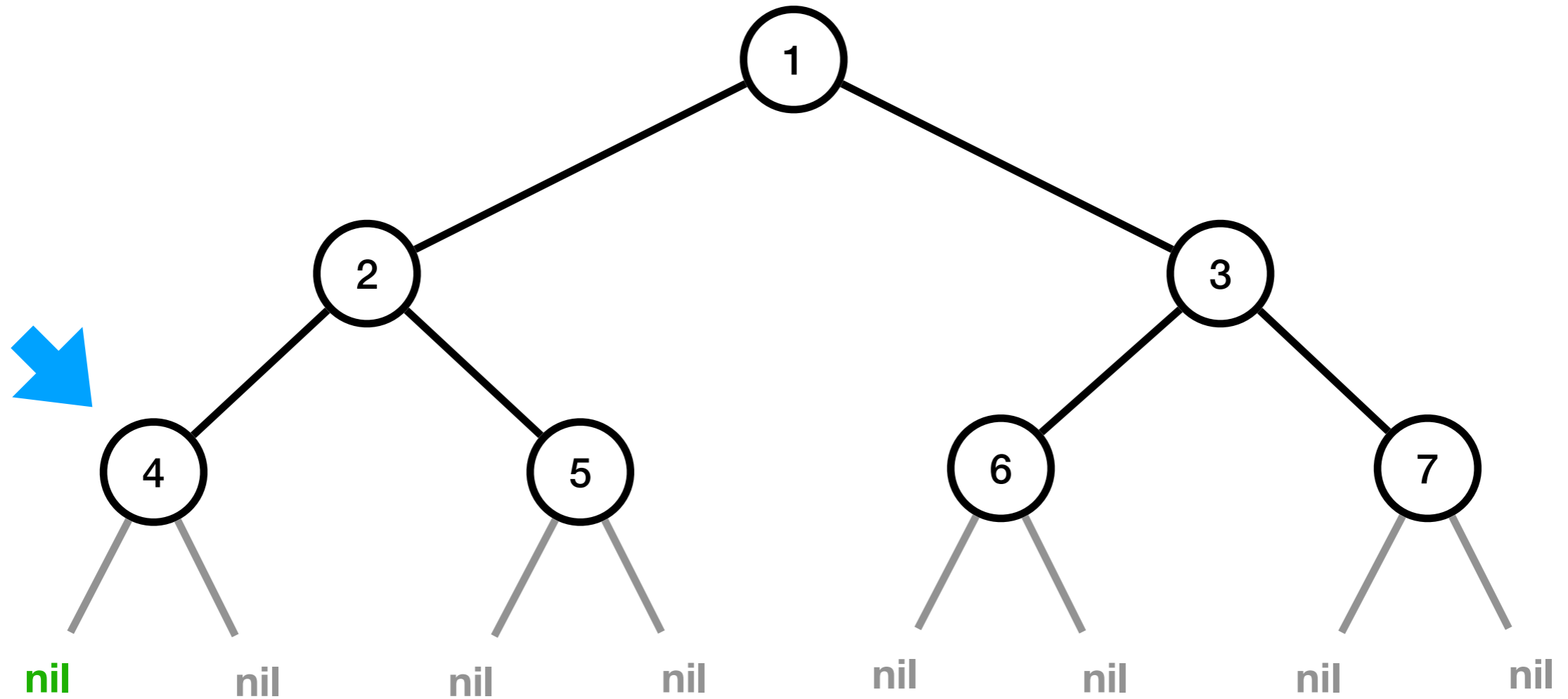
Parcours postfixe



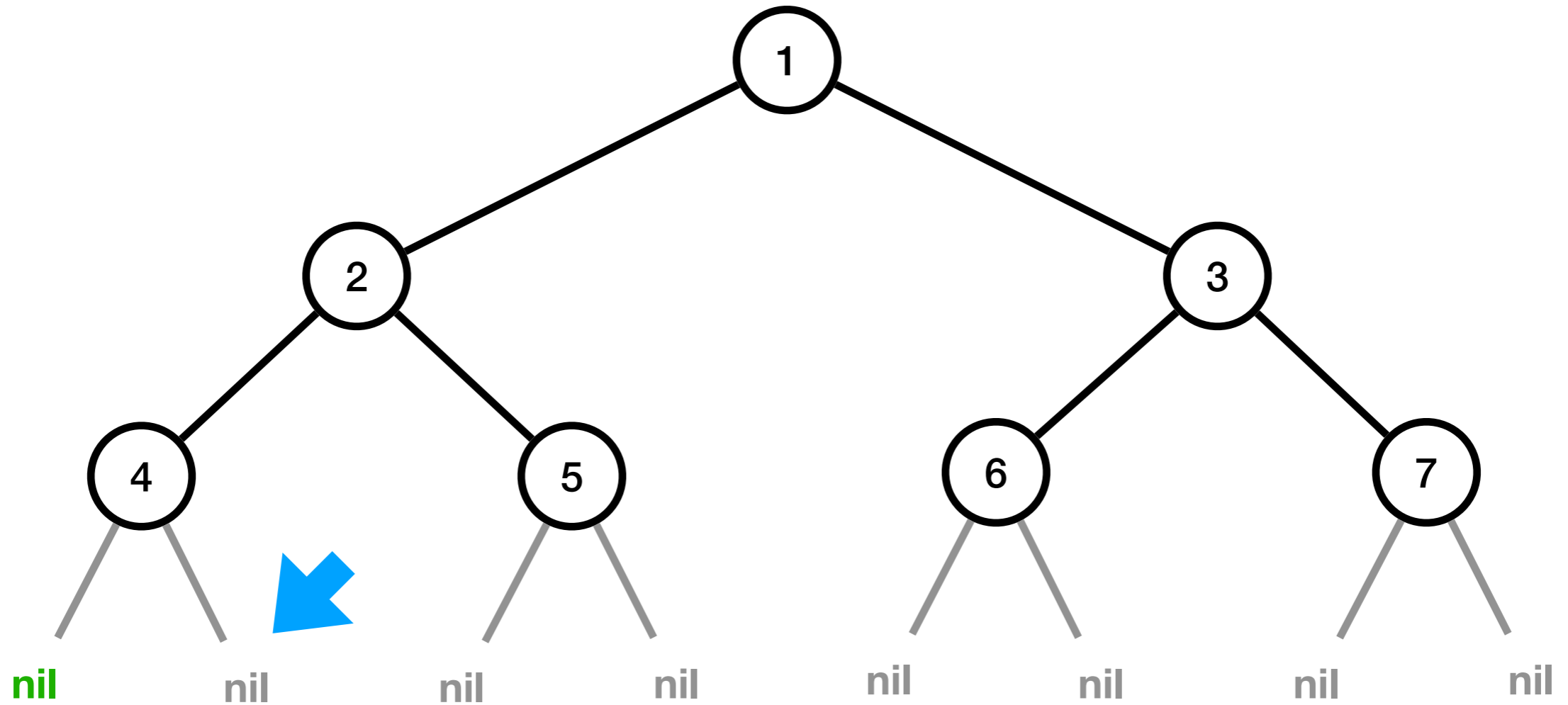
Parcours postfixe



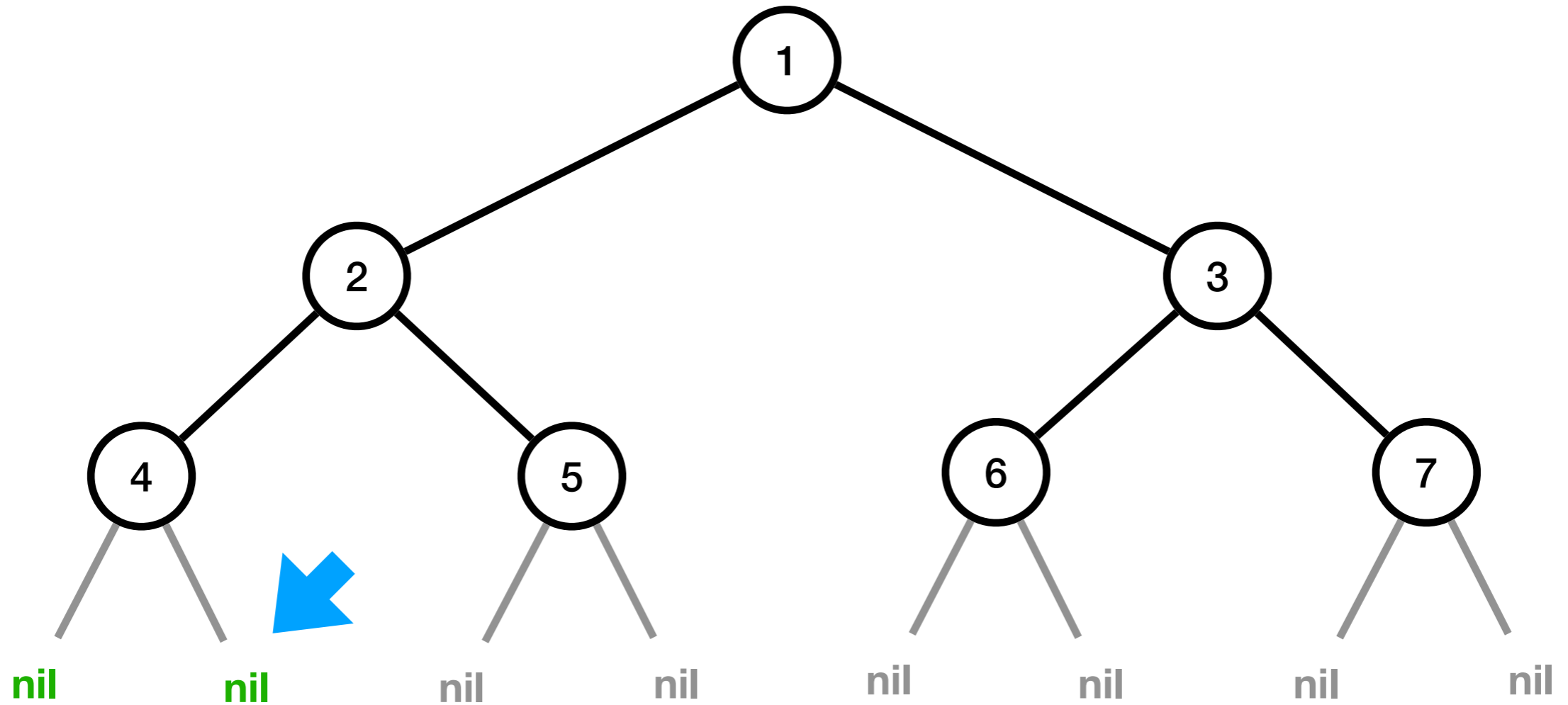
Parcours postfixe



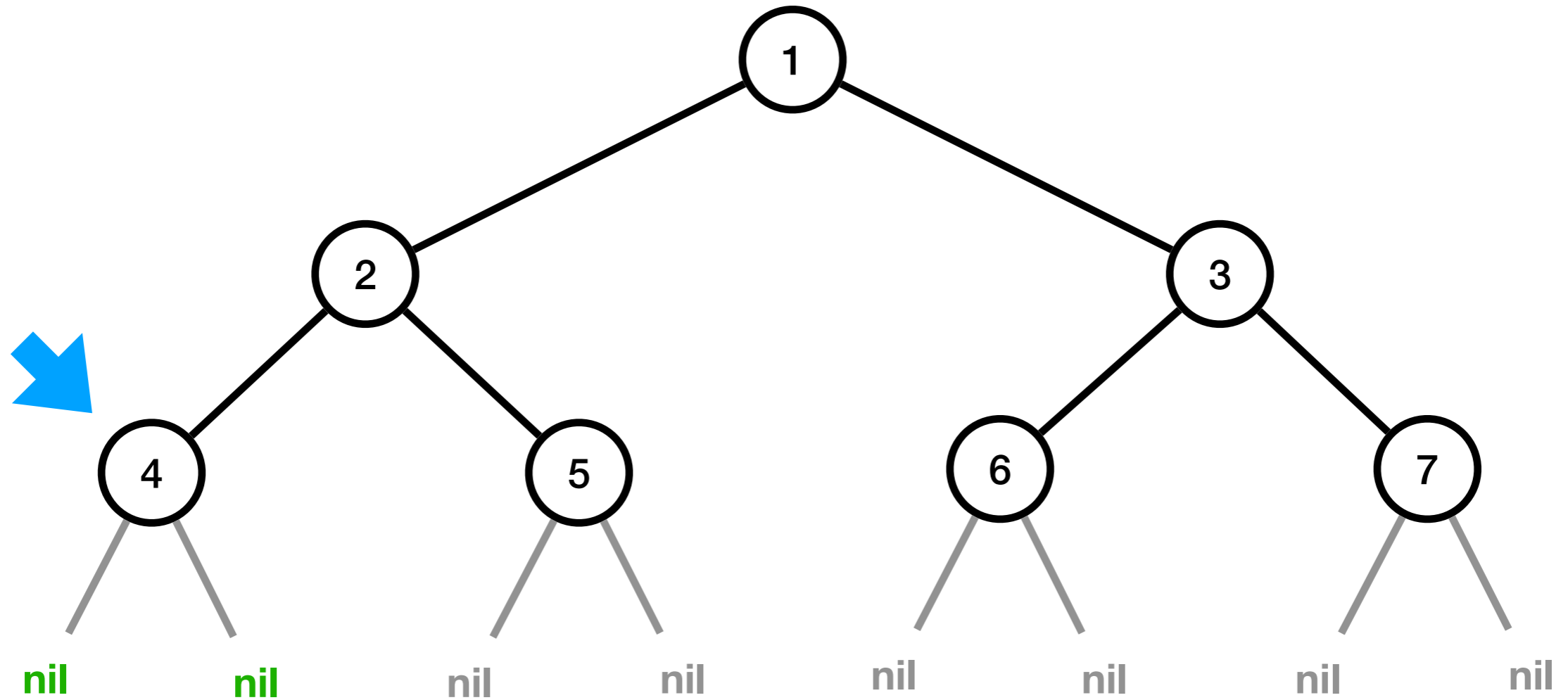
Parcours postfixe



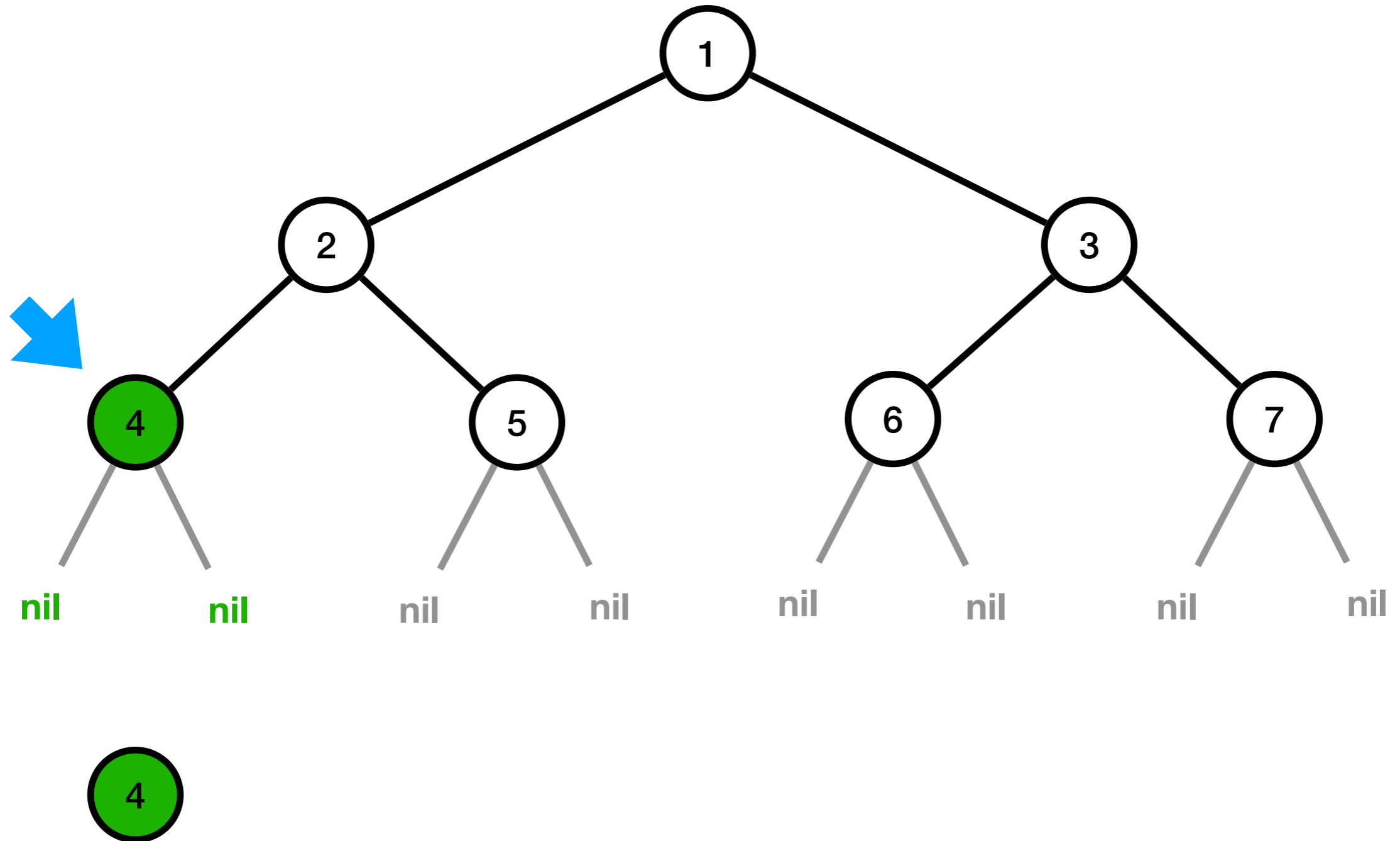
Parcours postfixe



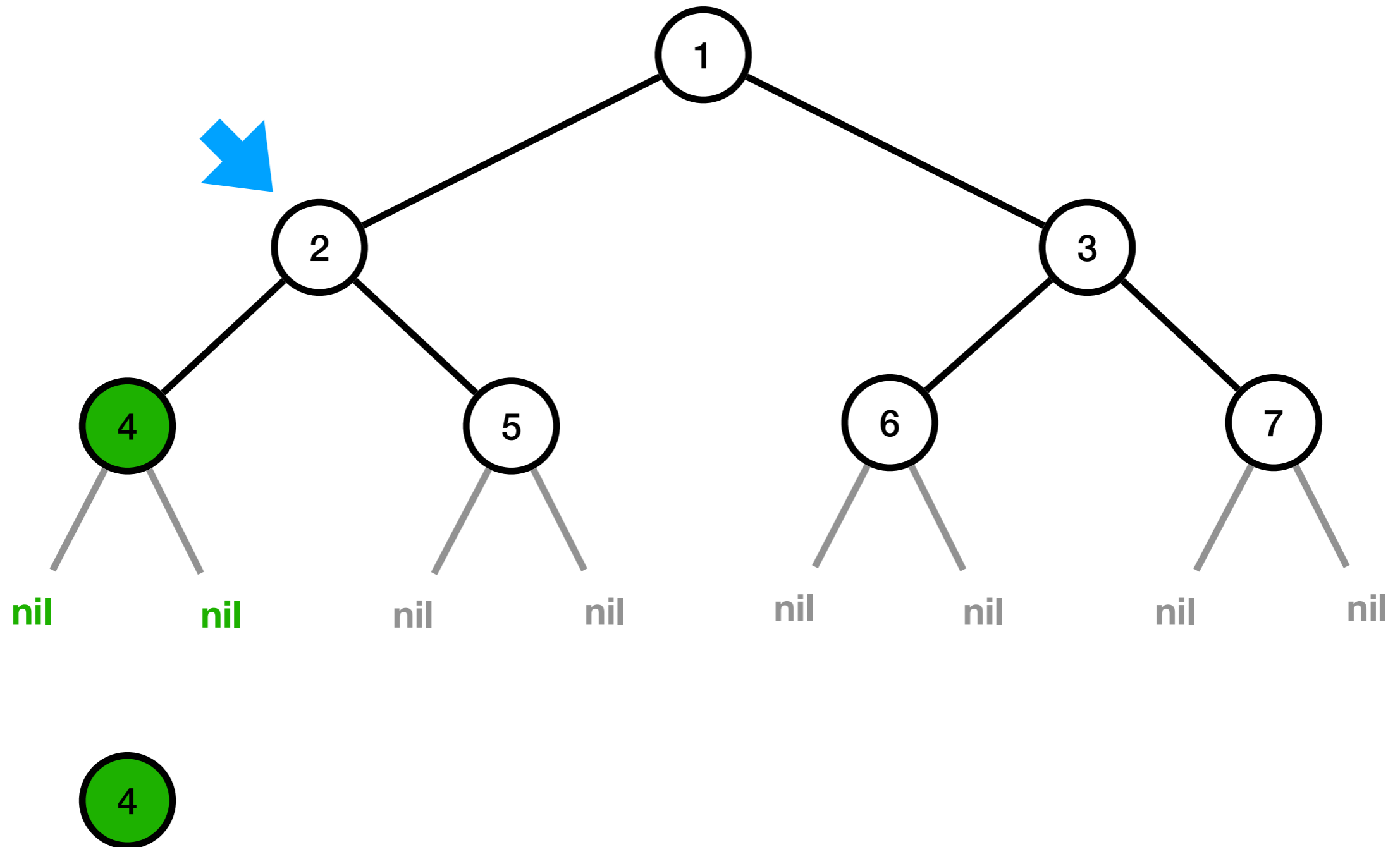
Parcours postfixe



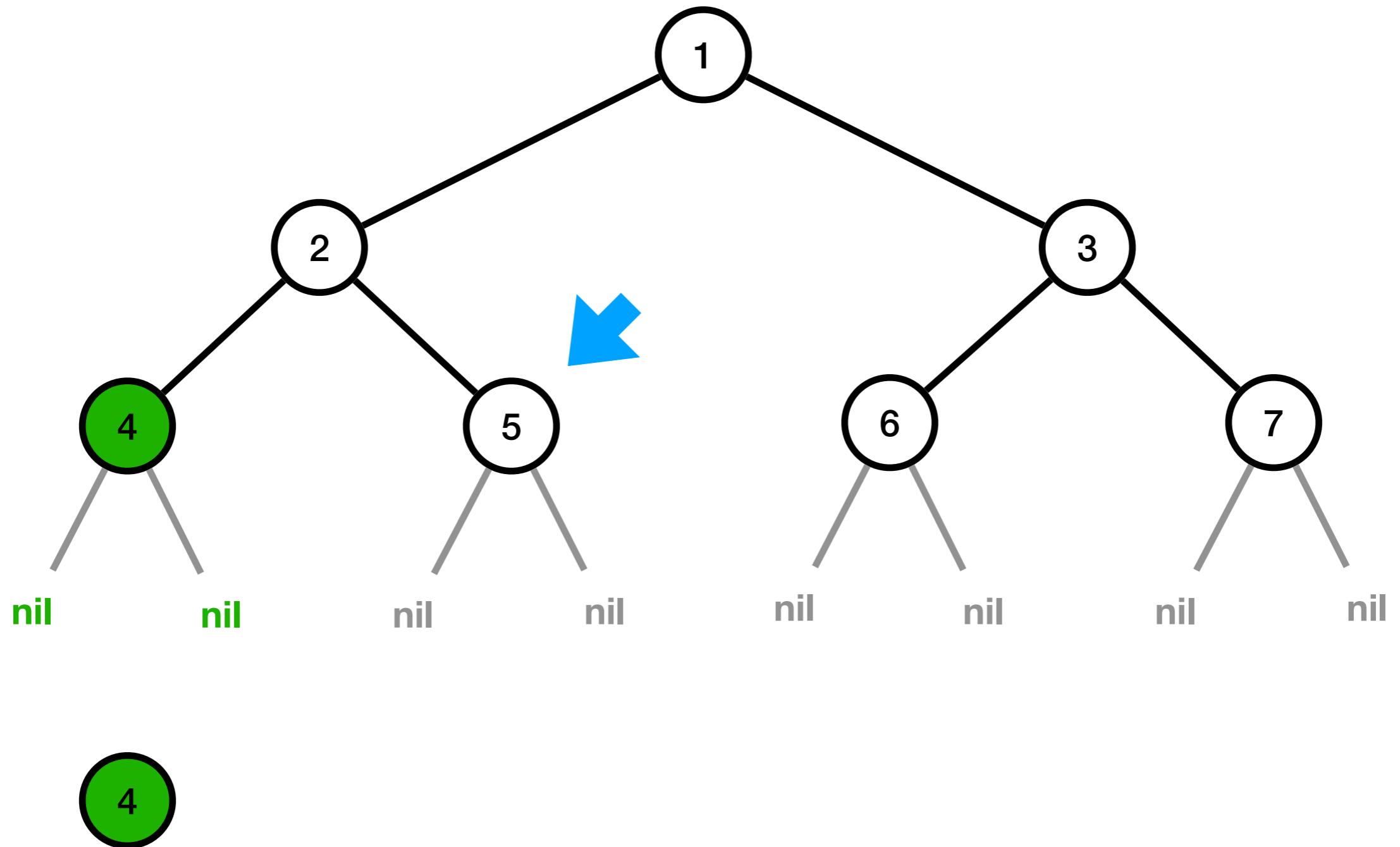
Parcours postfixe



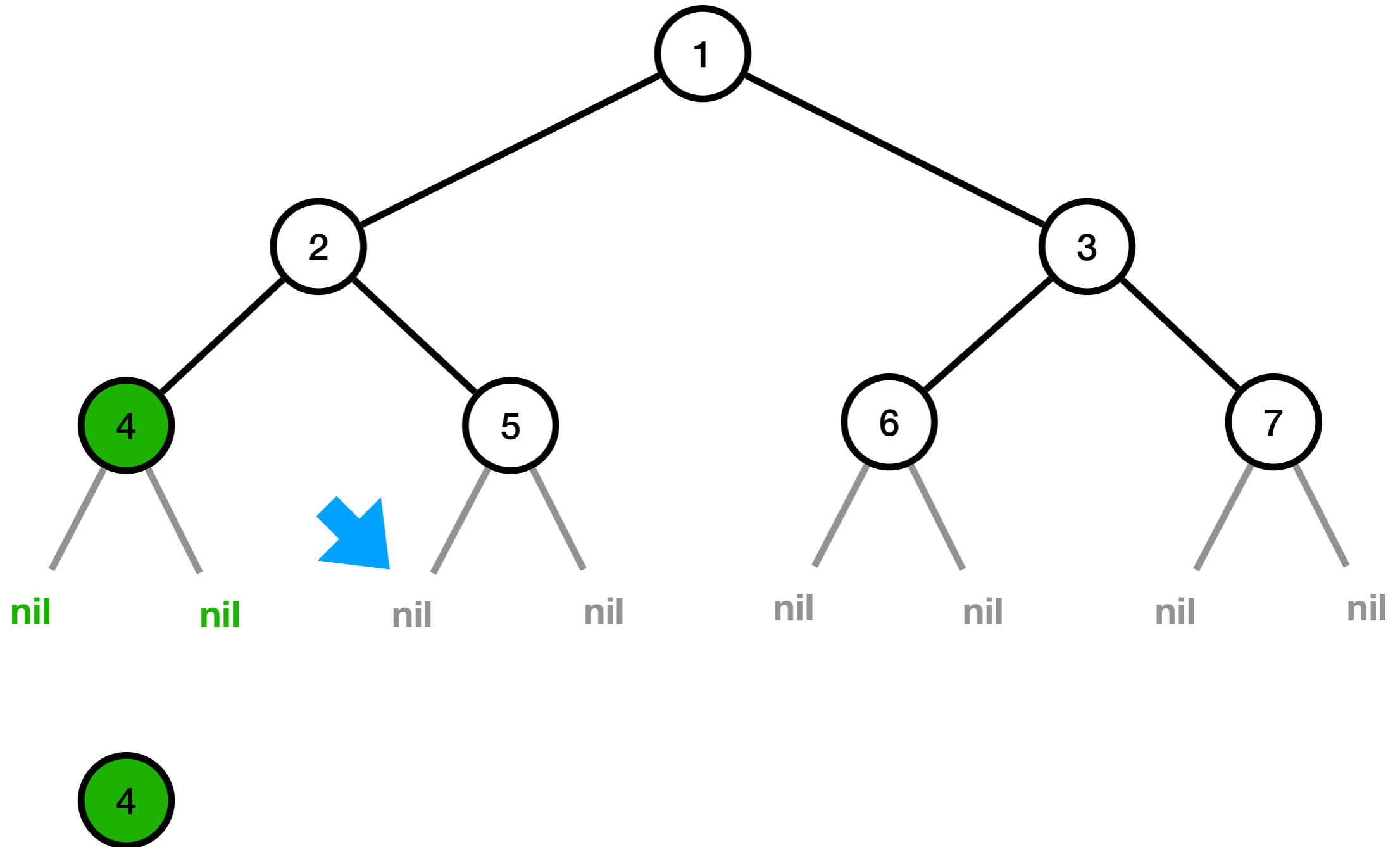
Parcours postfixe



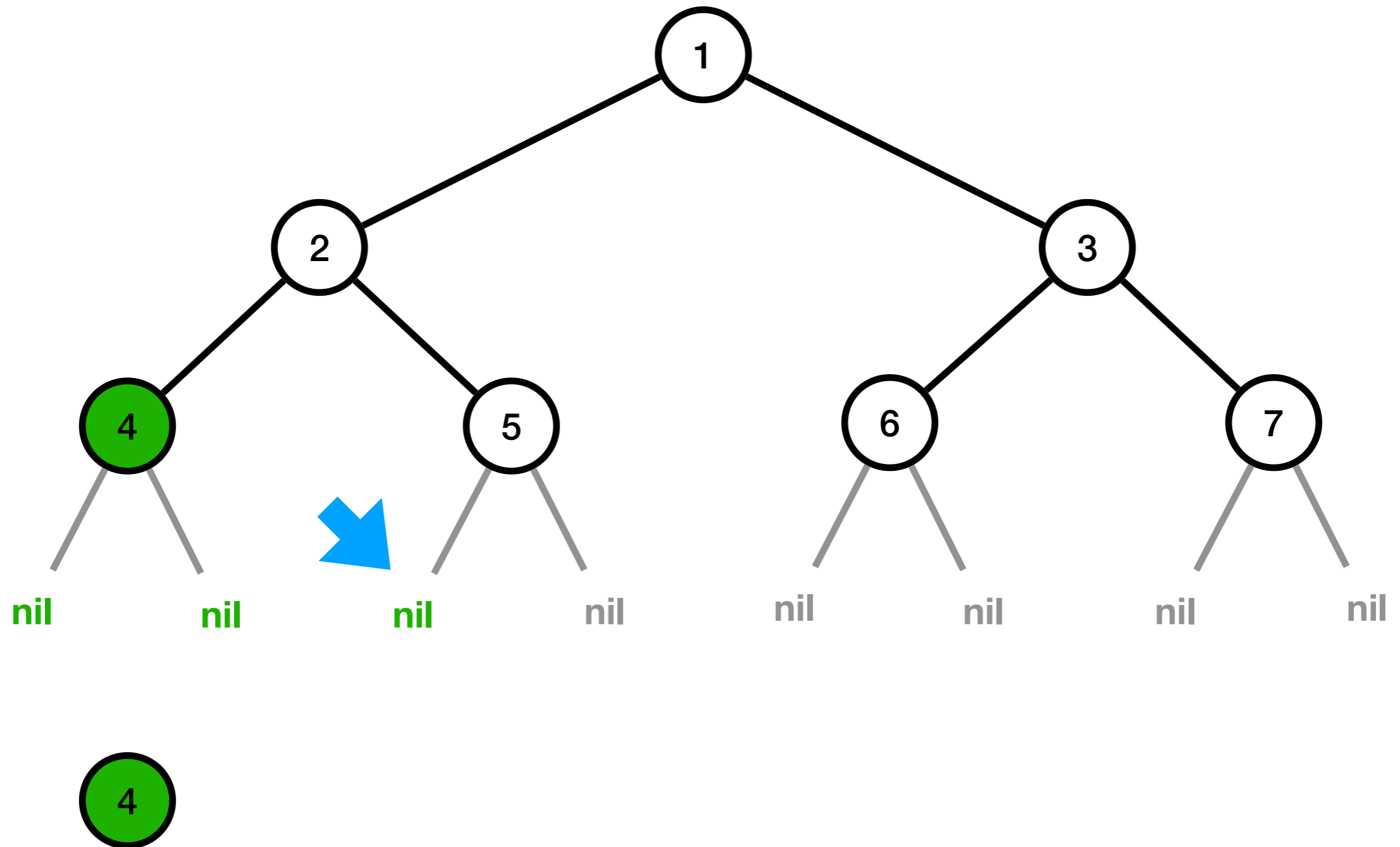
Parcours postfixe



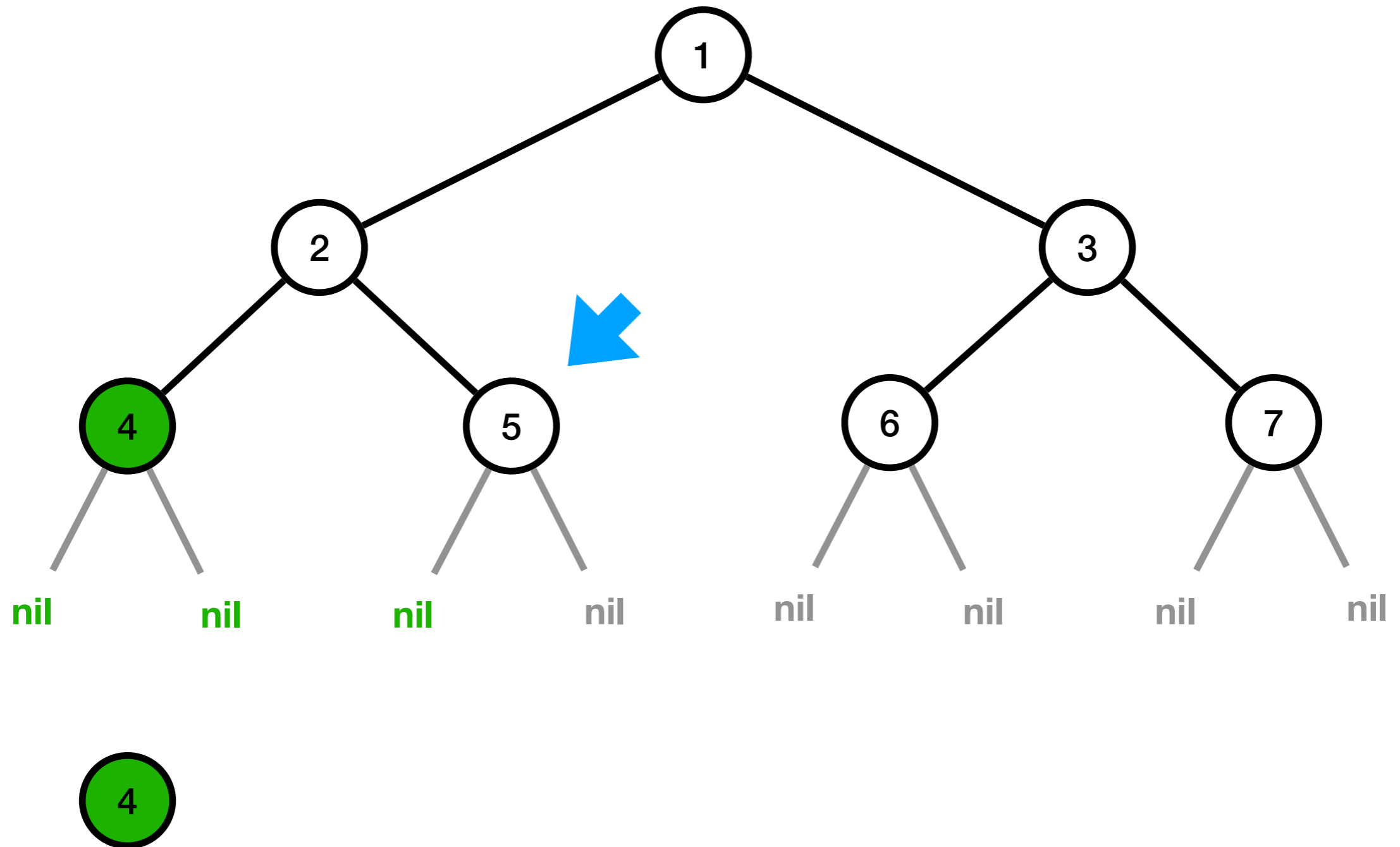
Parcours postfixe



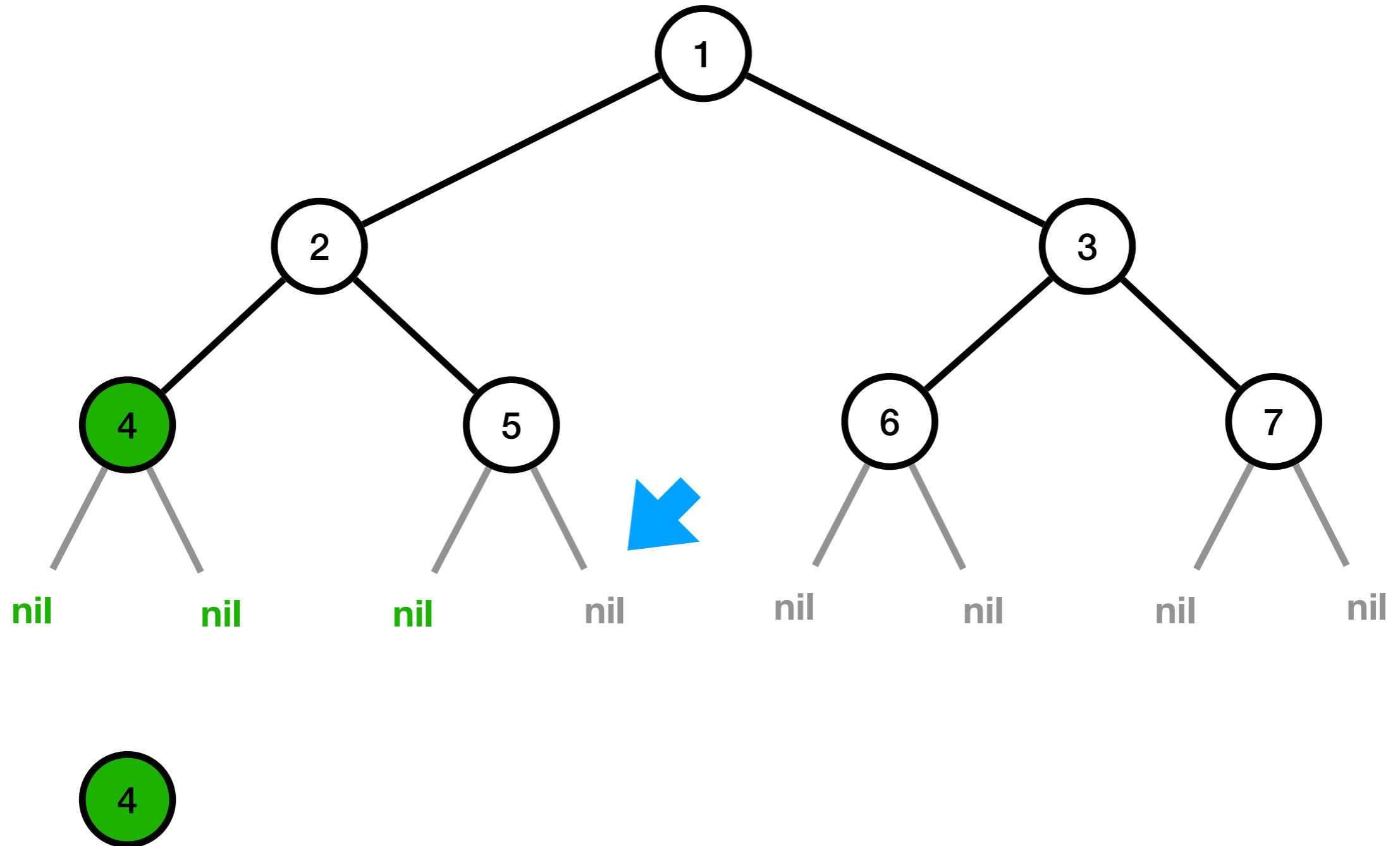
Parcours postfixe



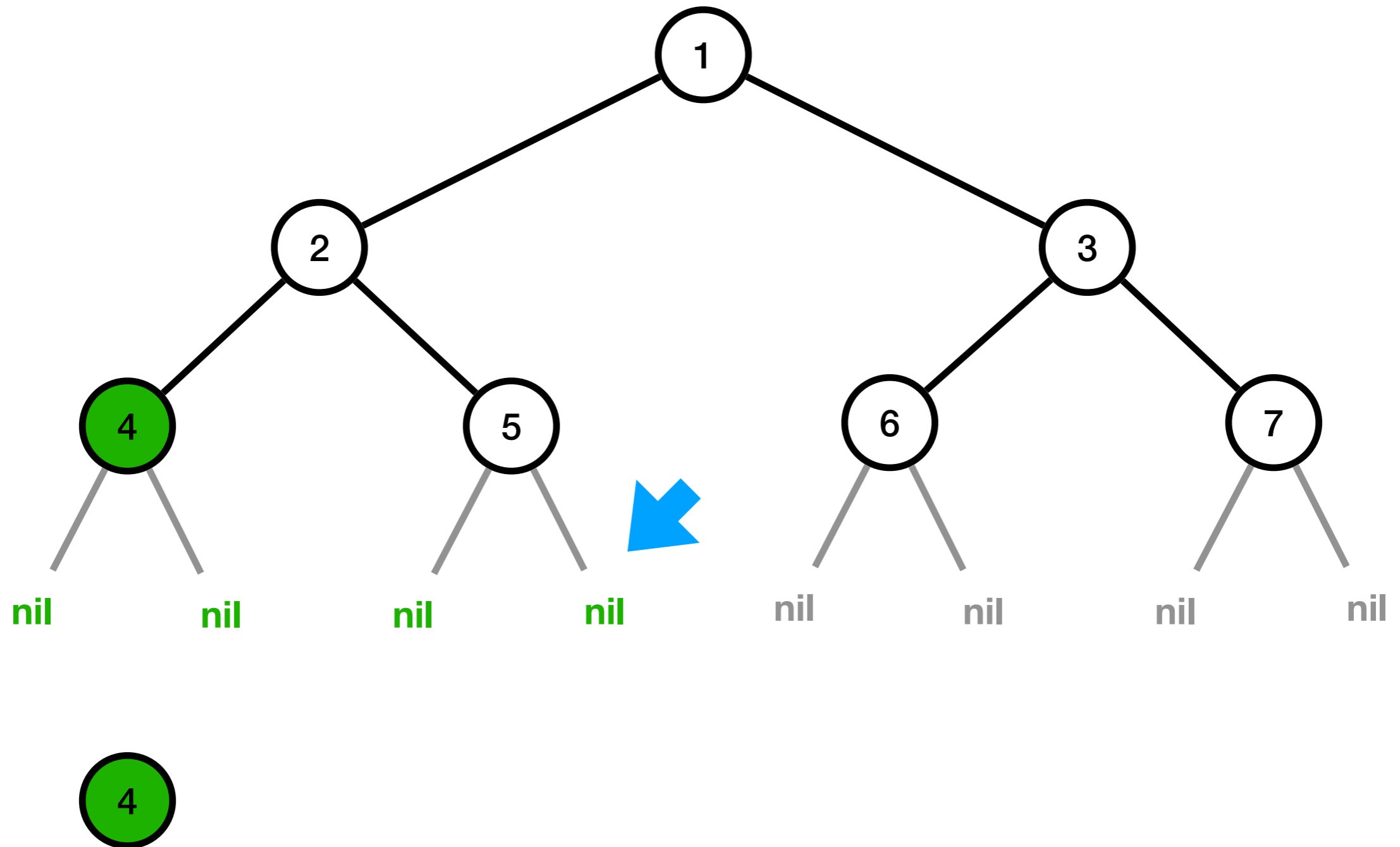
Parcours postfixe



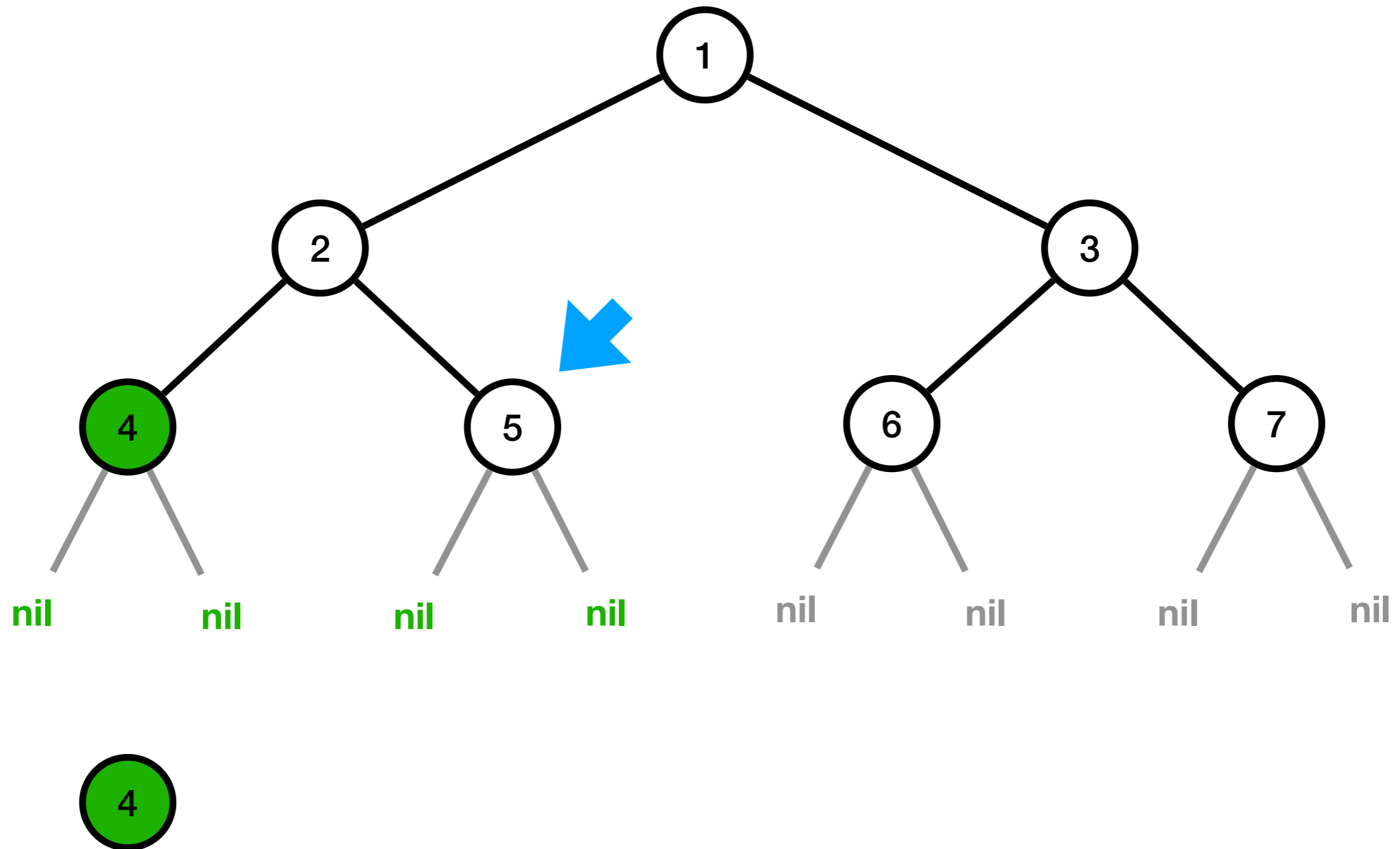
Parcours postfixe



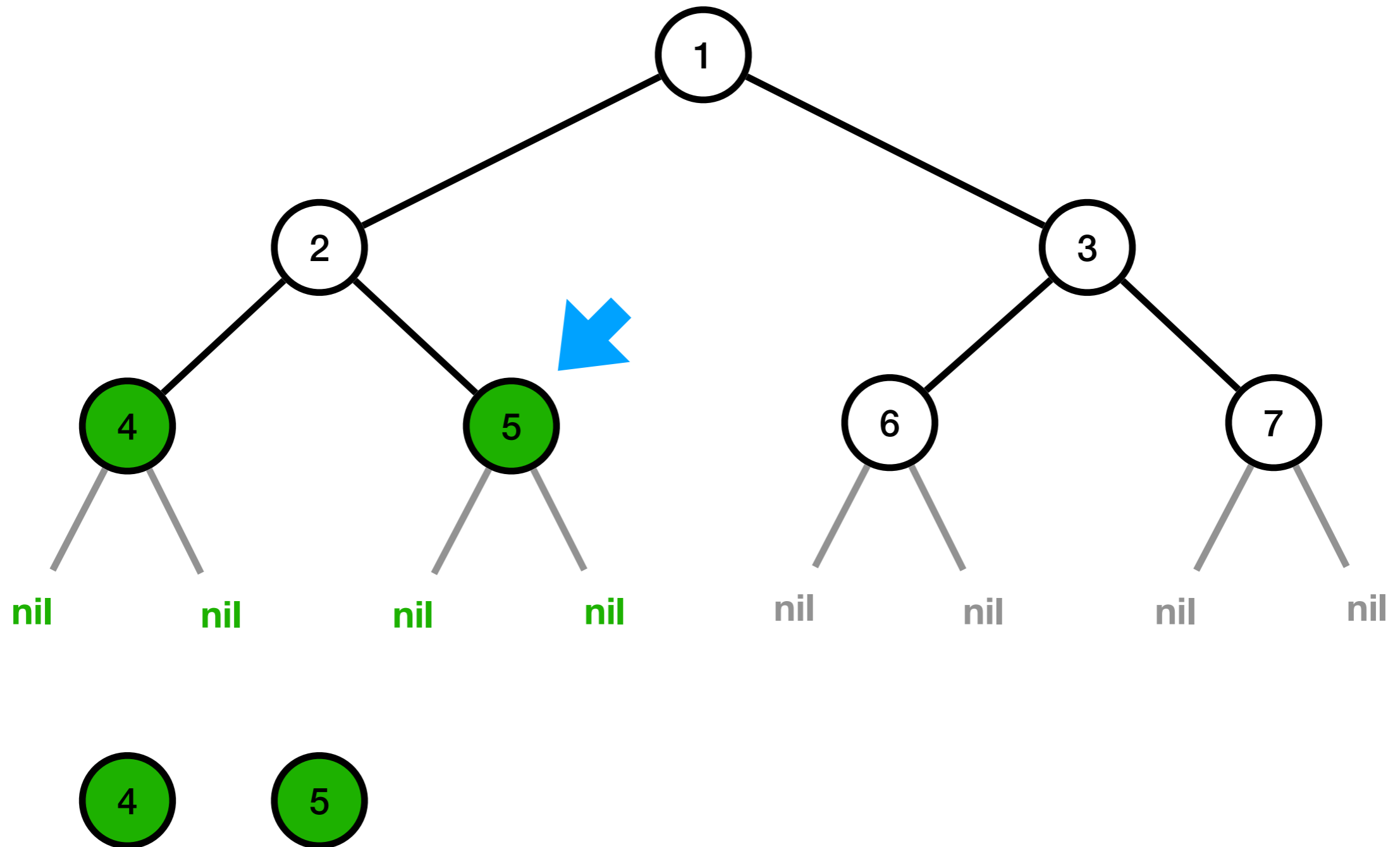
Parcours postfixe



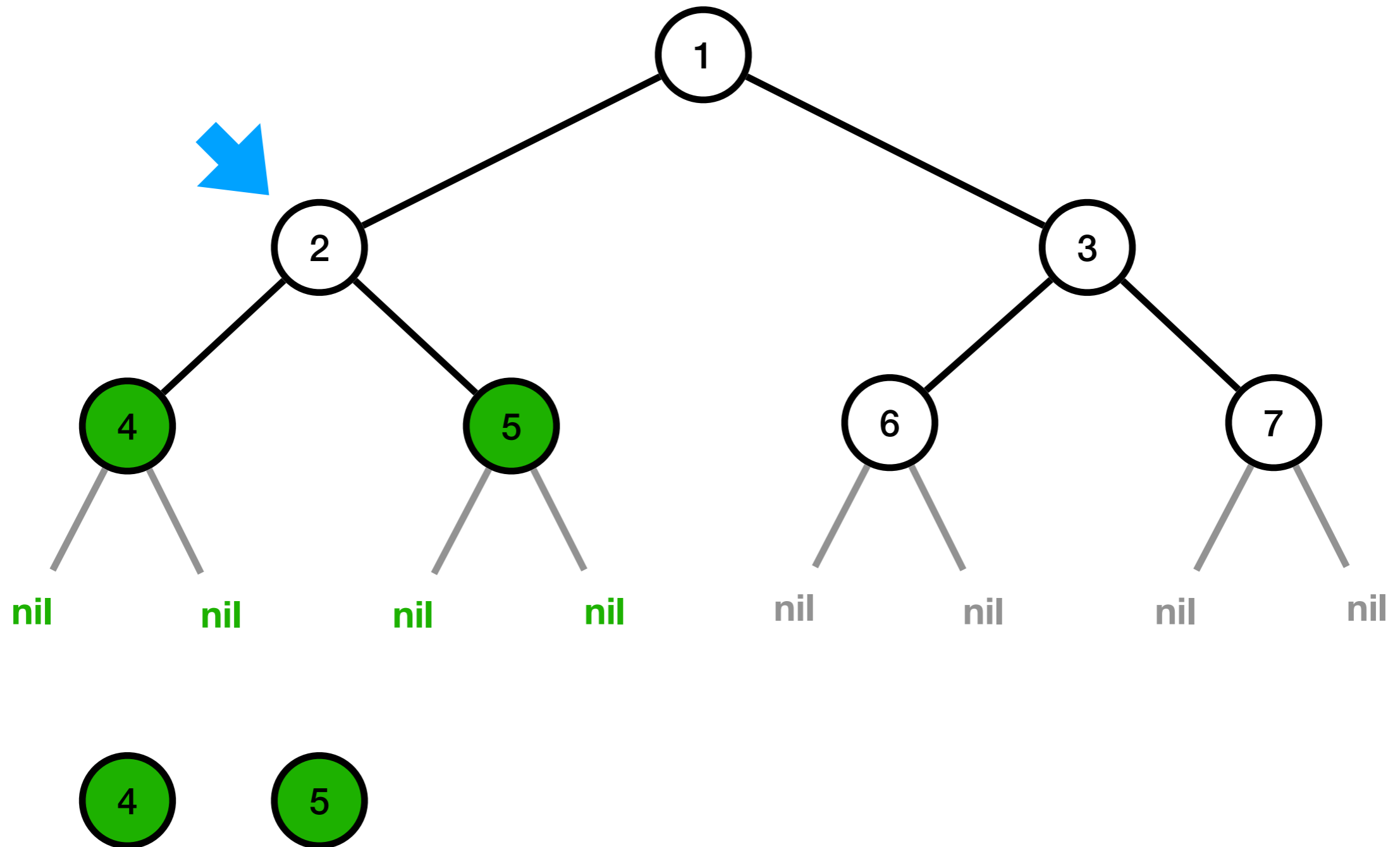
Parcours postfixe



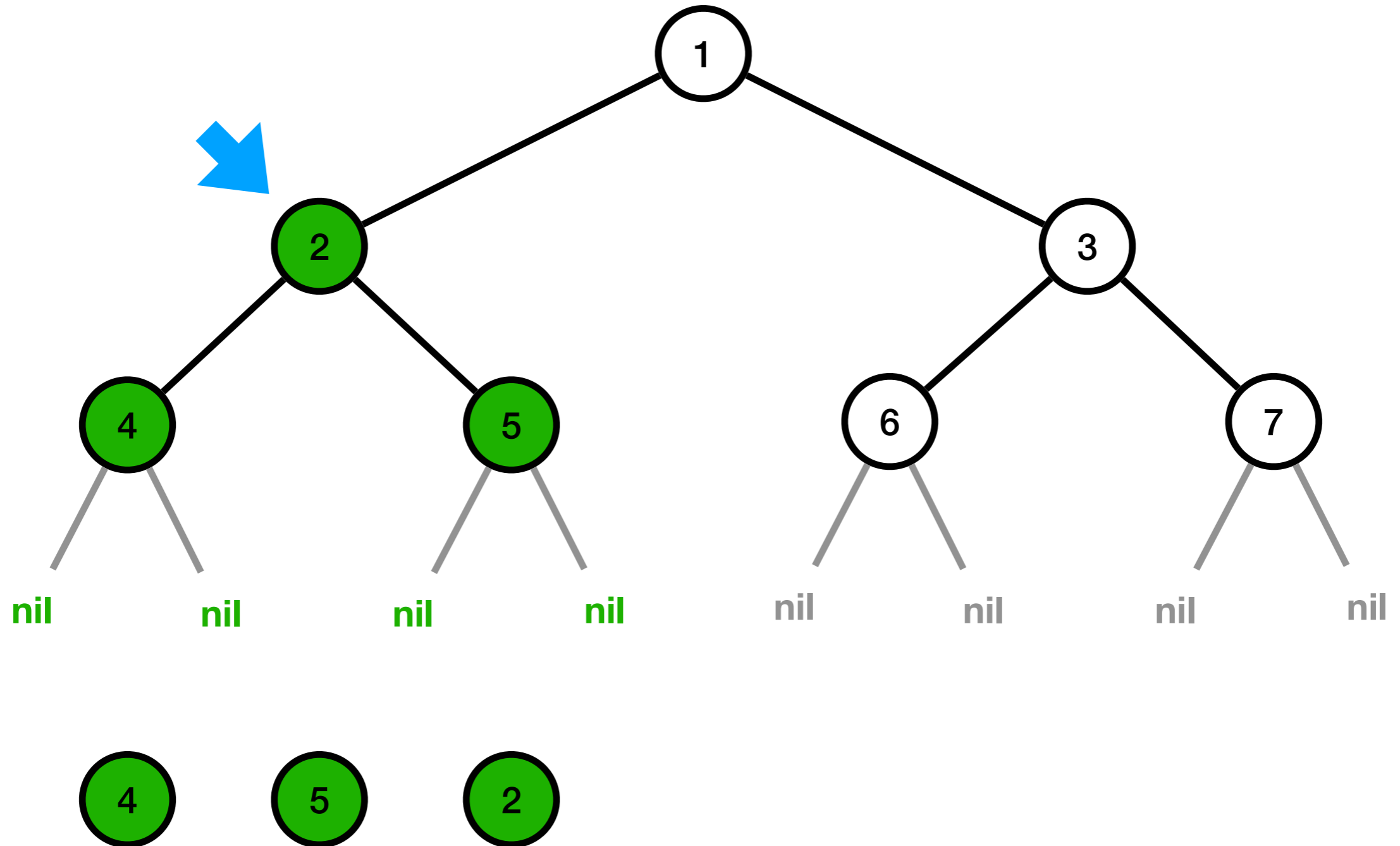
Parcours postfixe



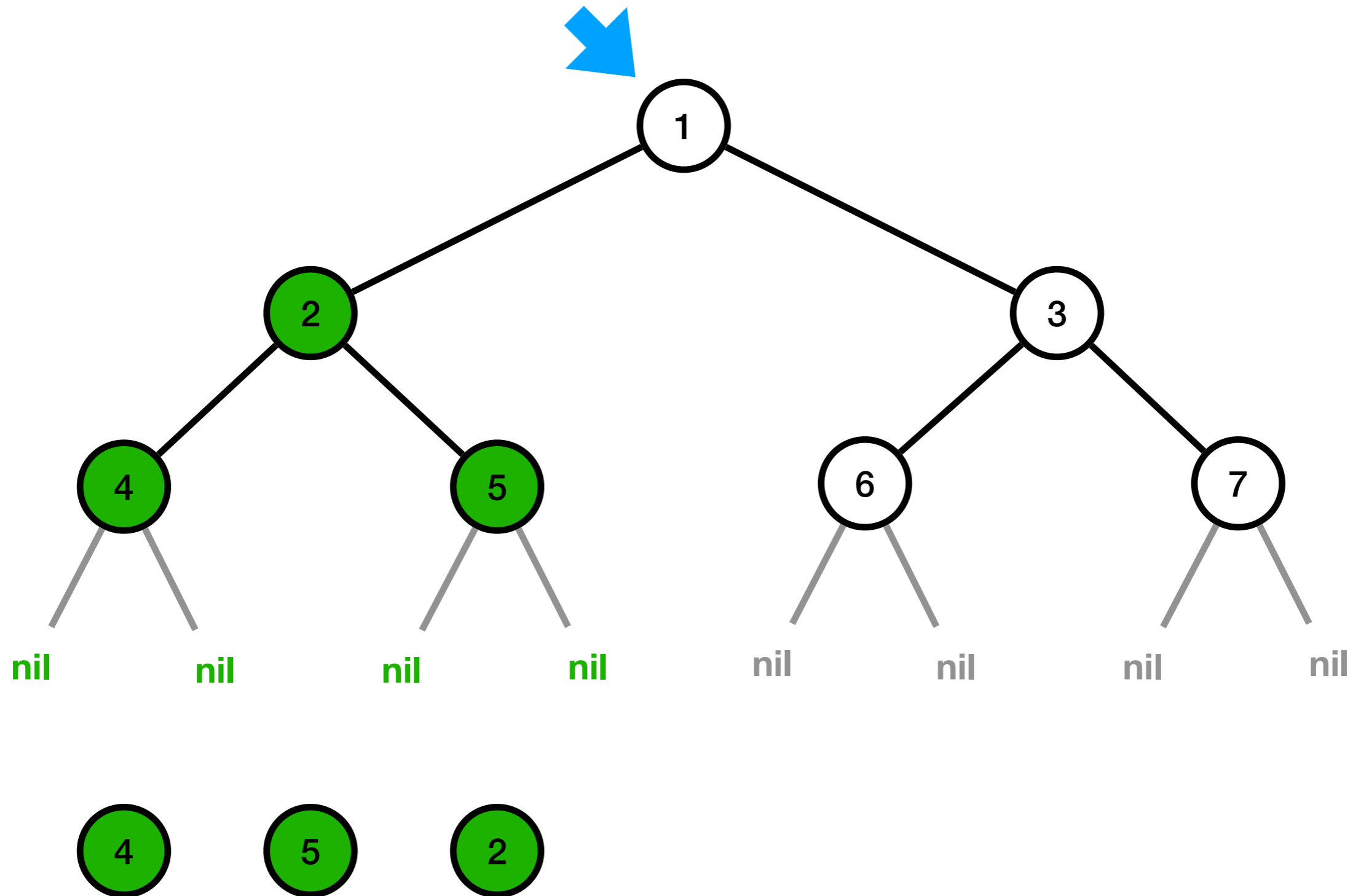
Parcours postfixe



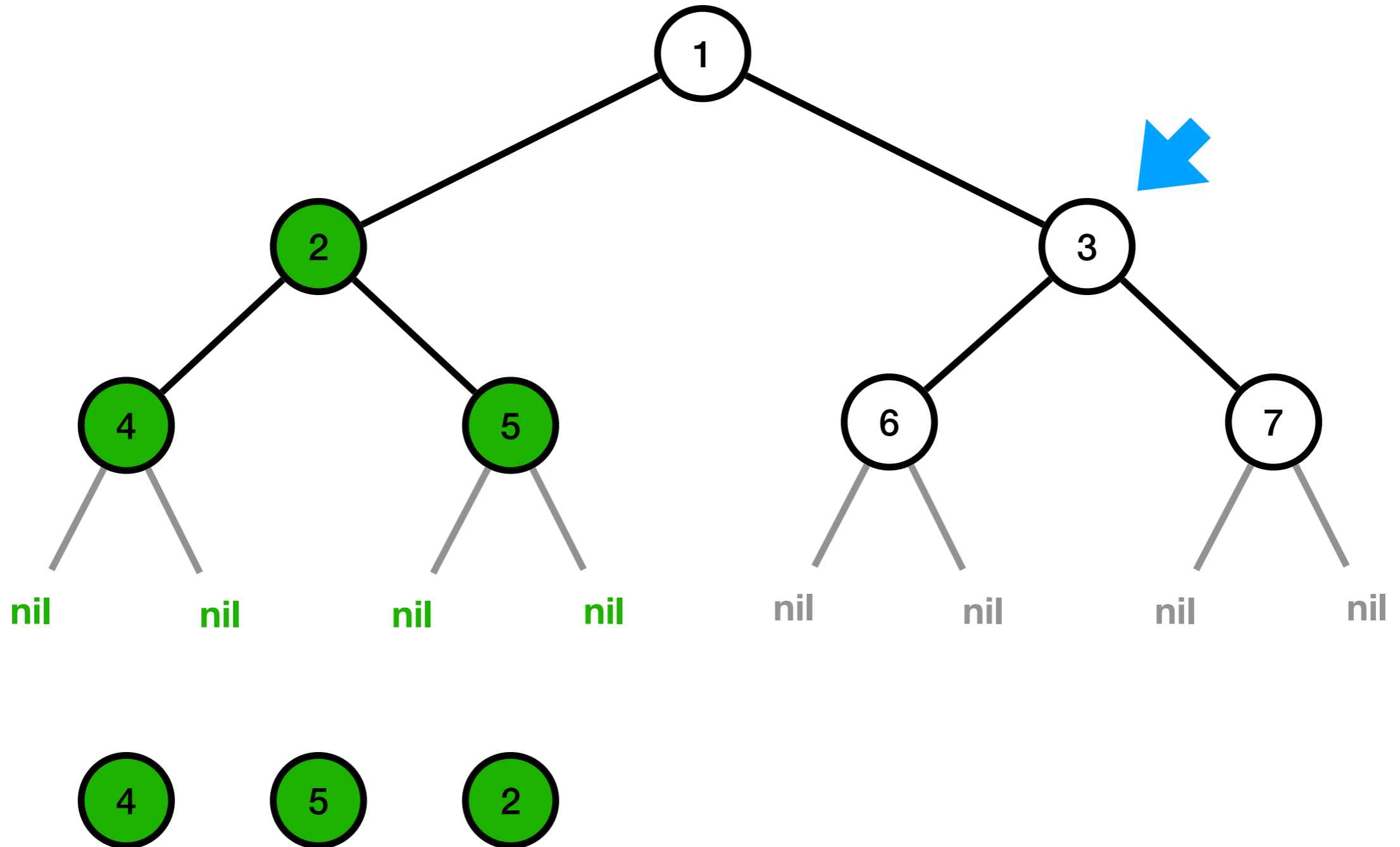
Parcours postfixe



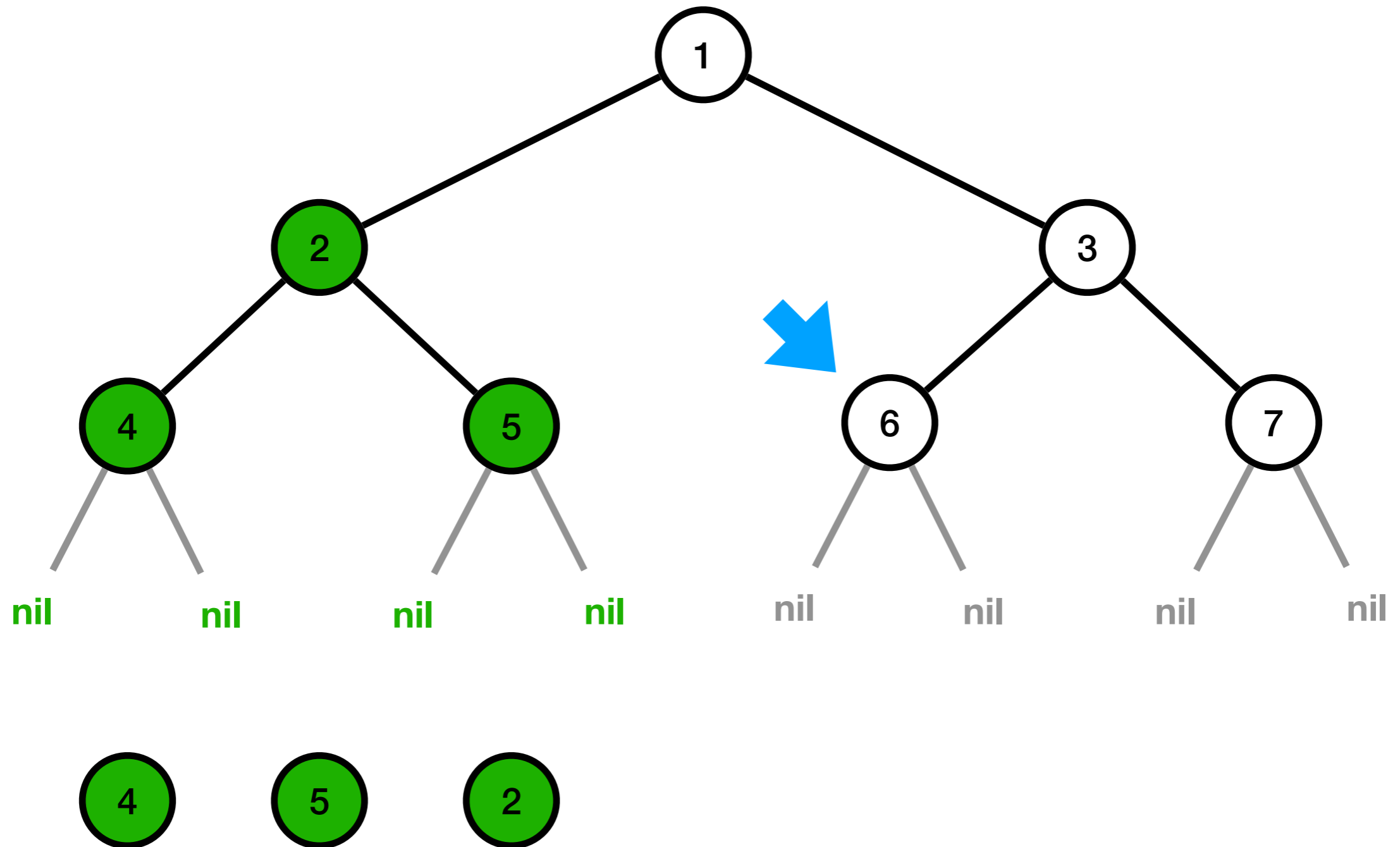
Parcours postfixe



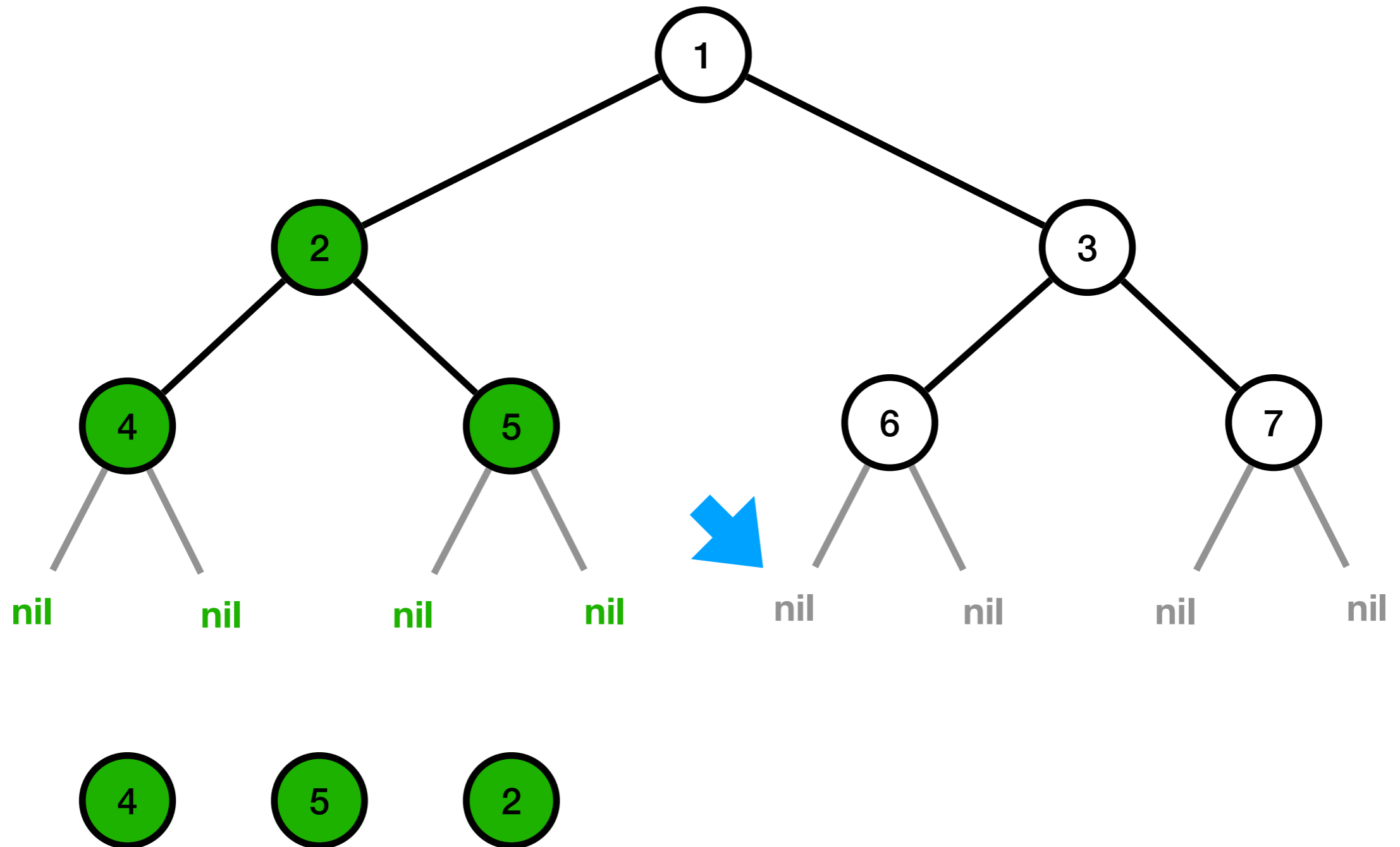
Parcours postfixe



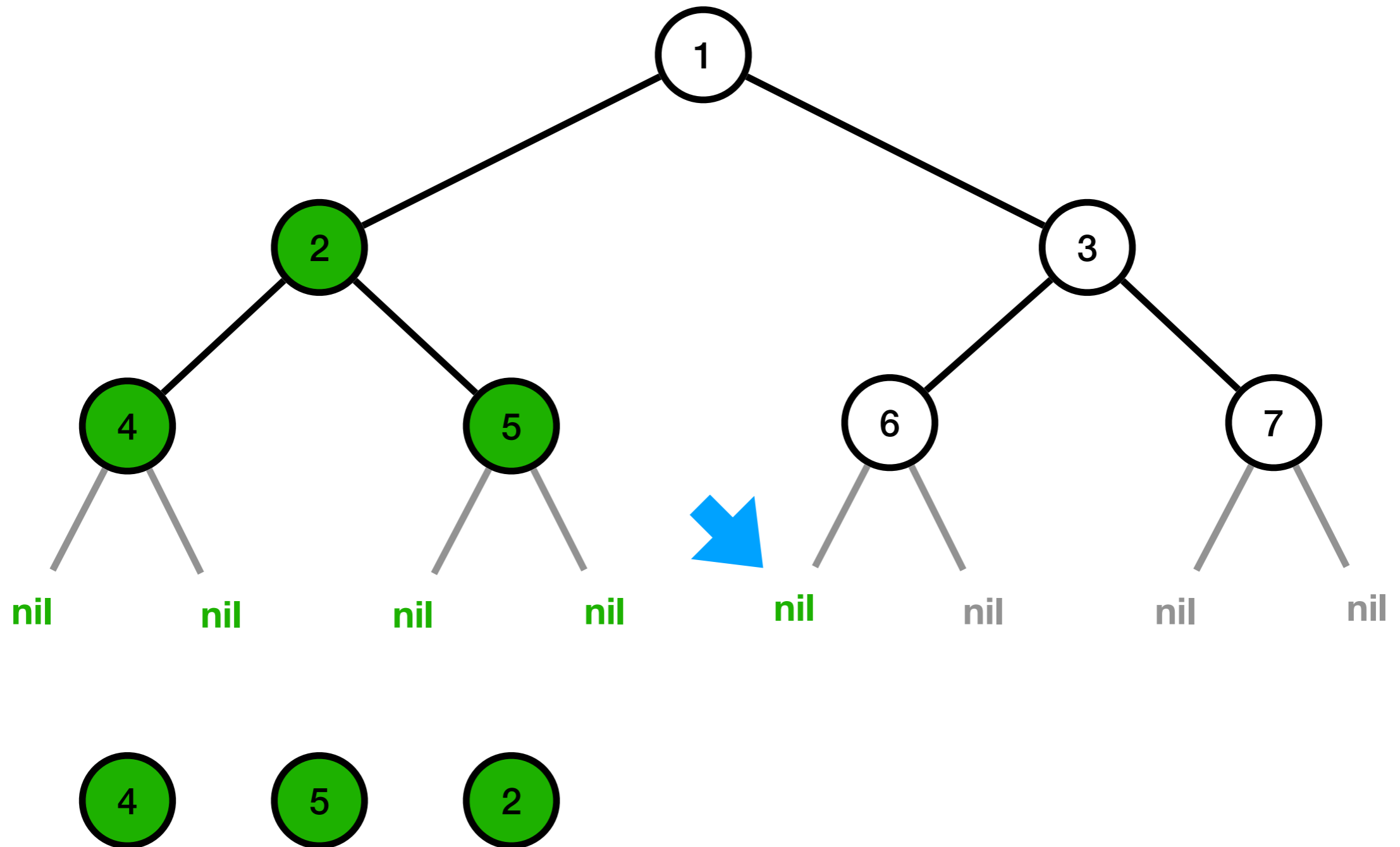
Parcours postfixe



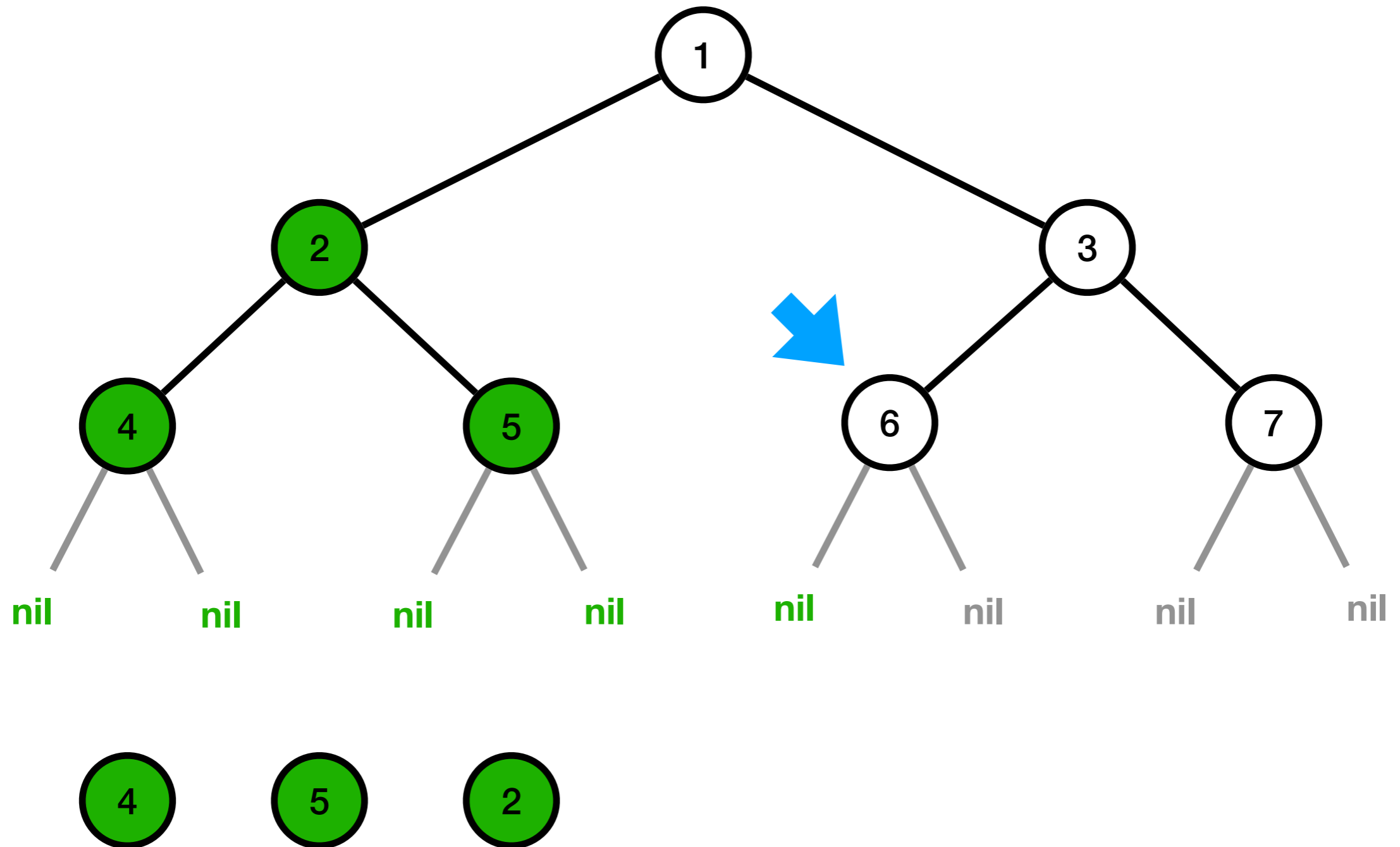
Parcours postfixe



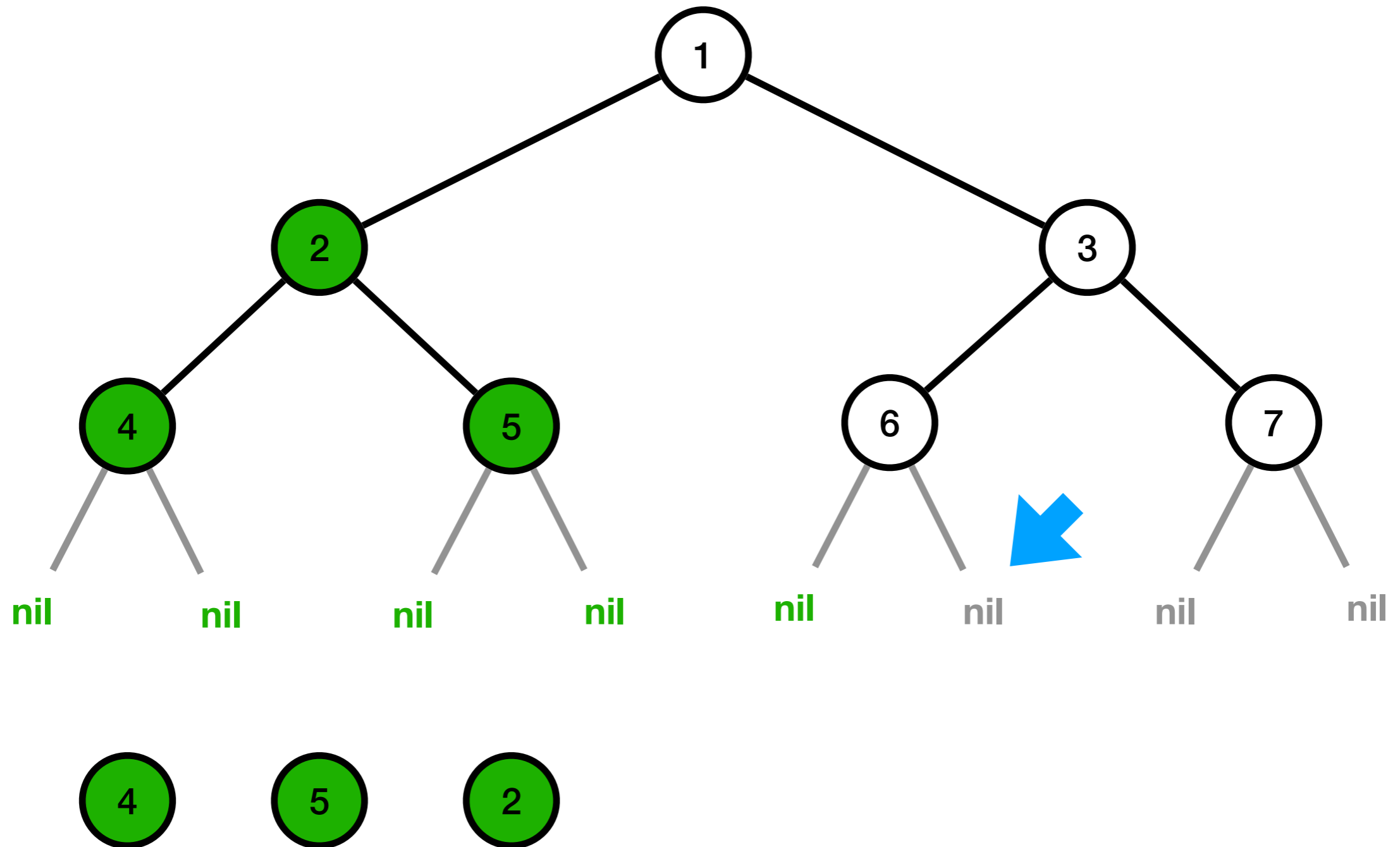
Parcours postfixe



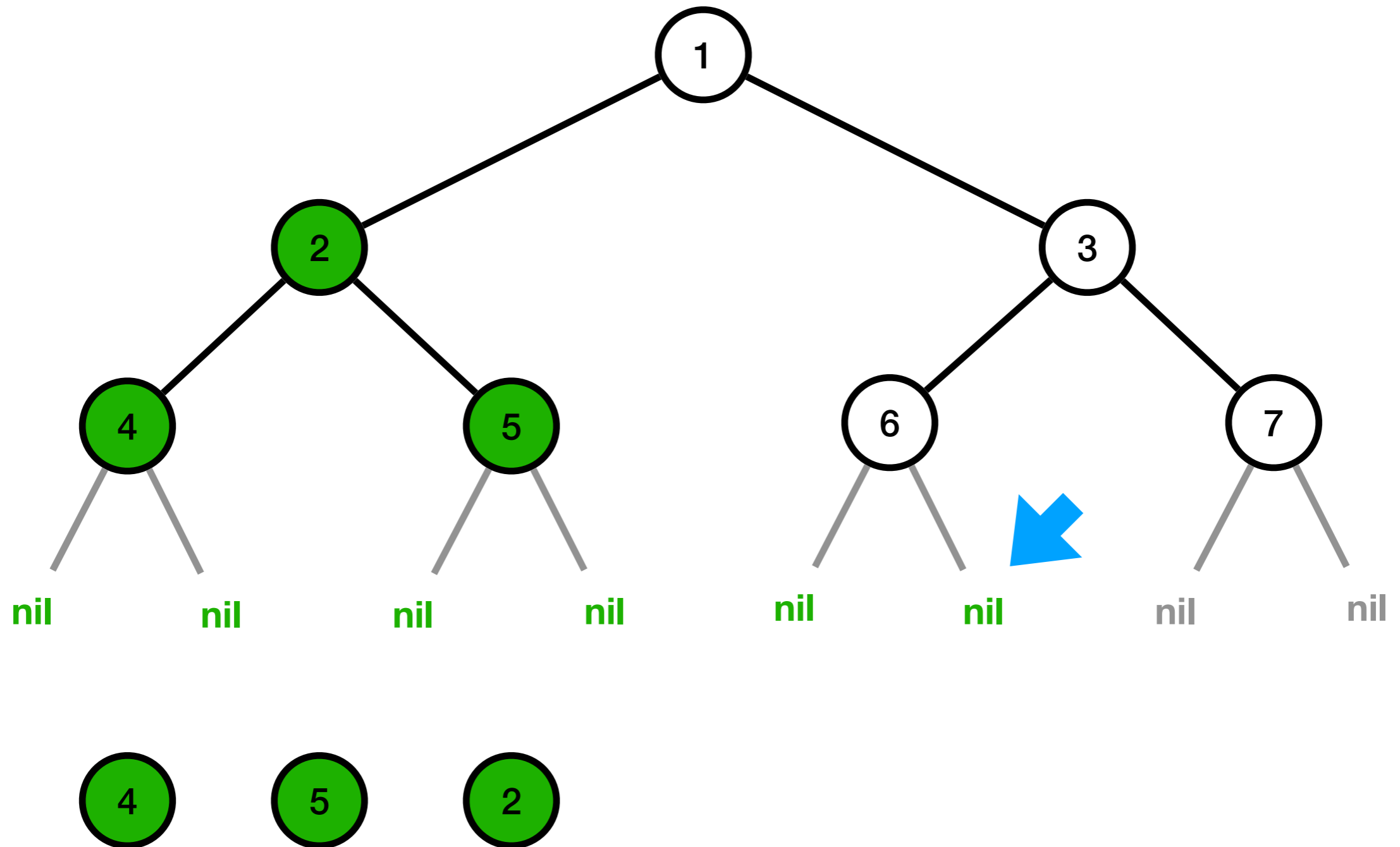
Parcours postfixe



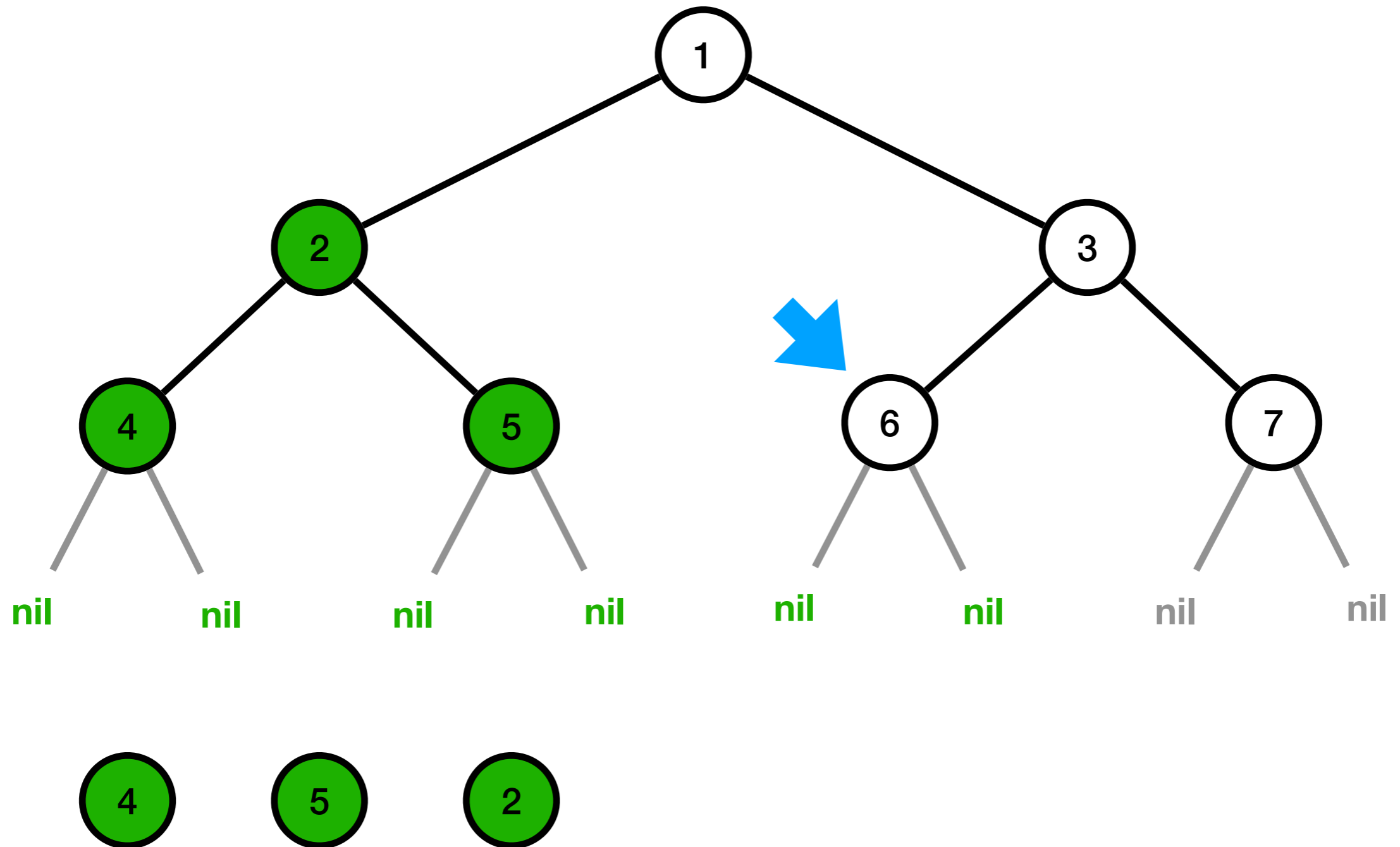
Parcours postfixe



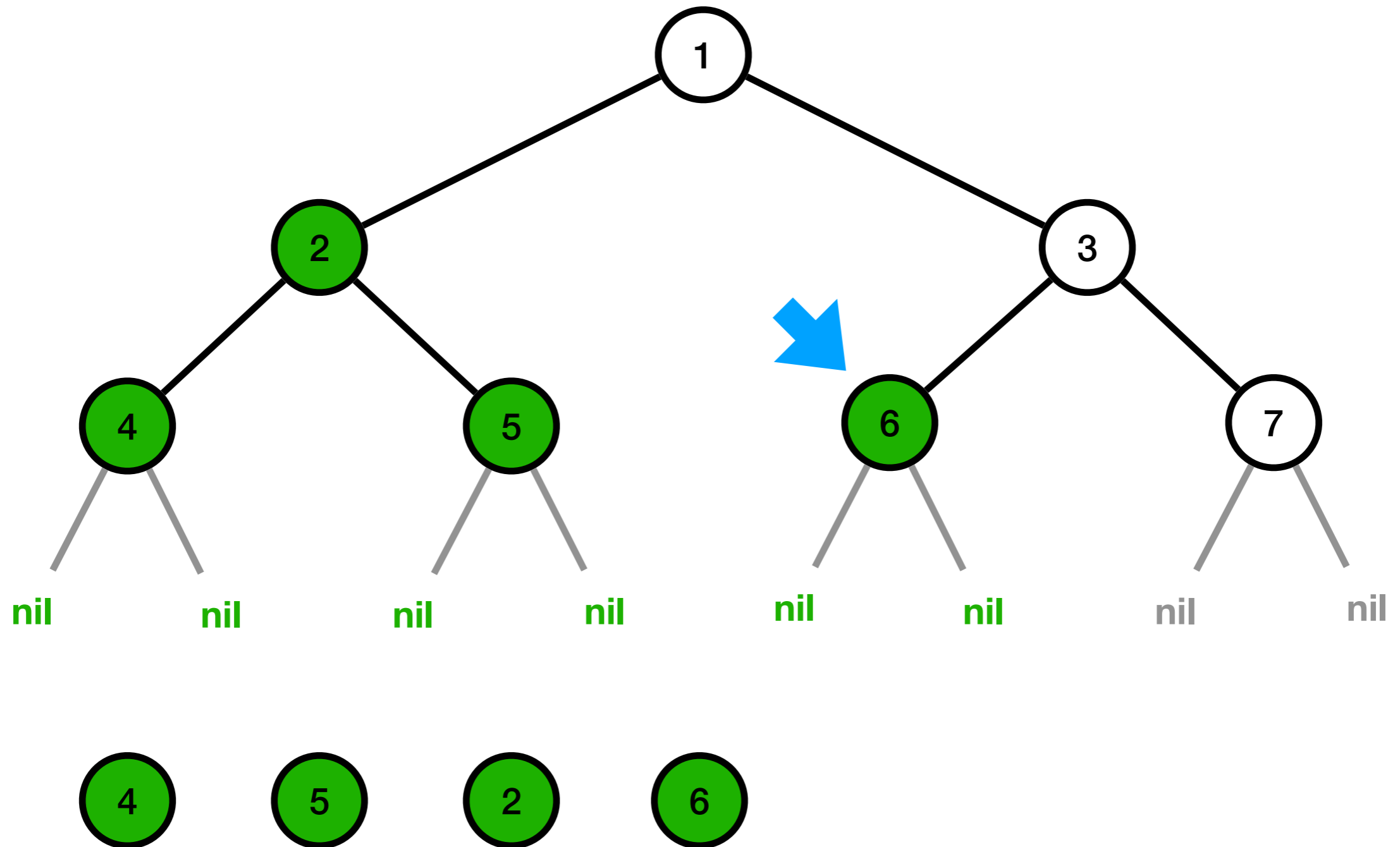
Parcours postfixe



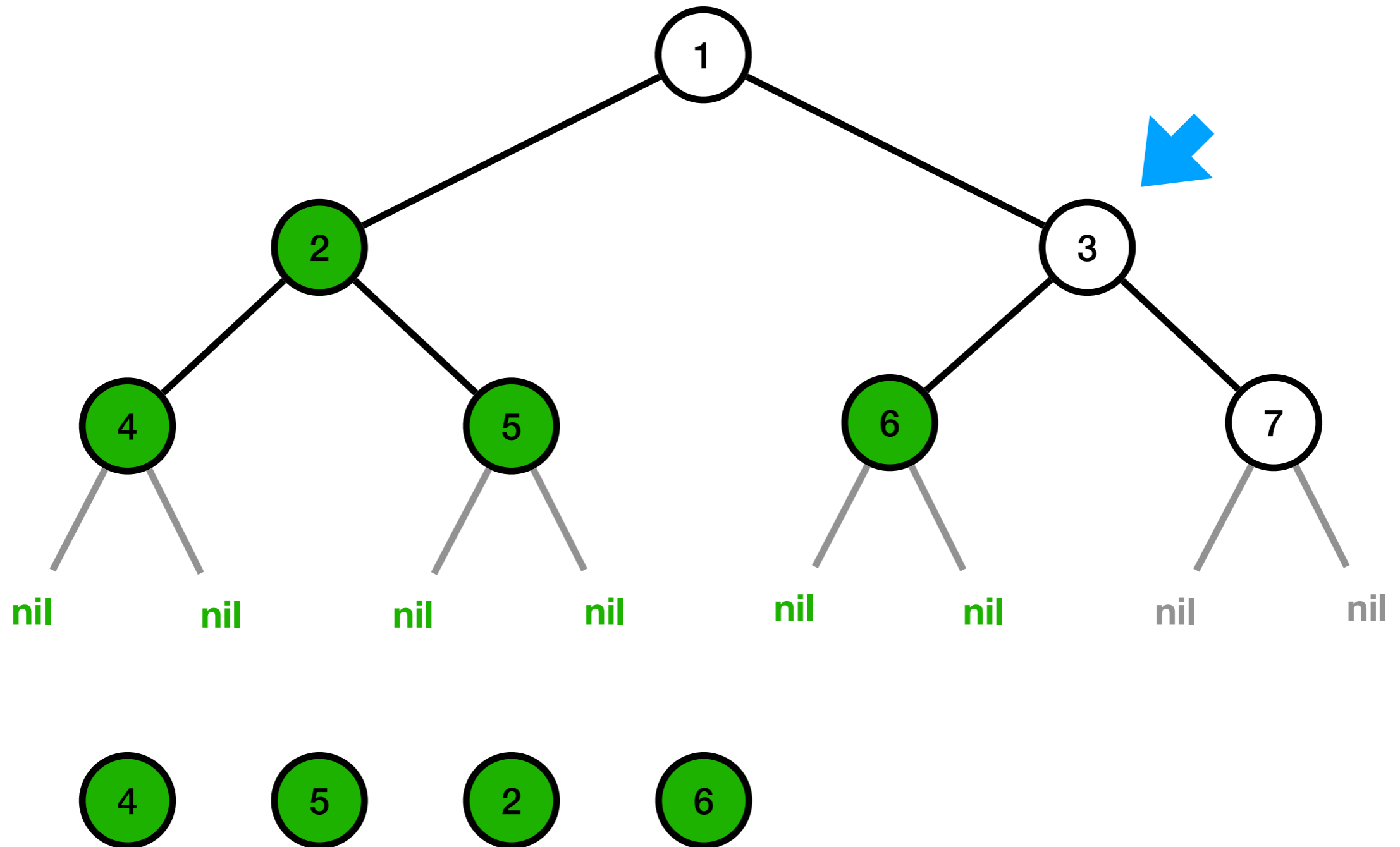
Parcours postfixe



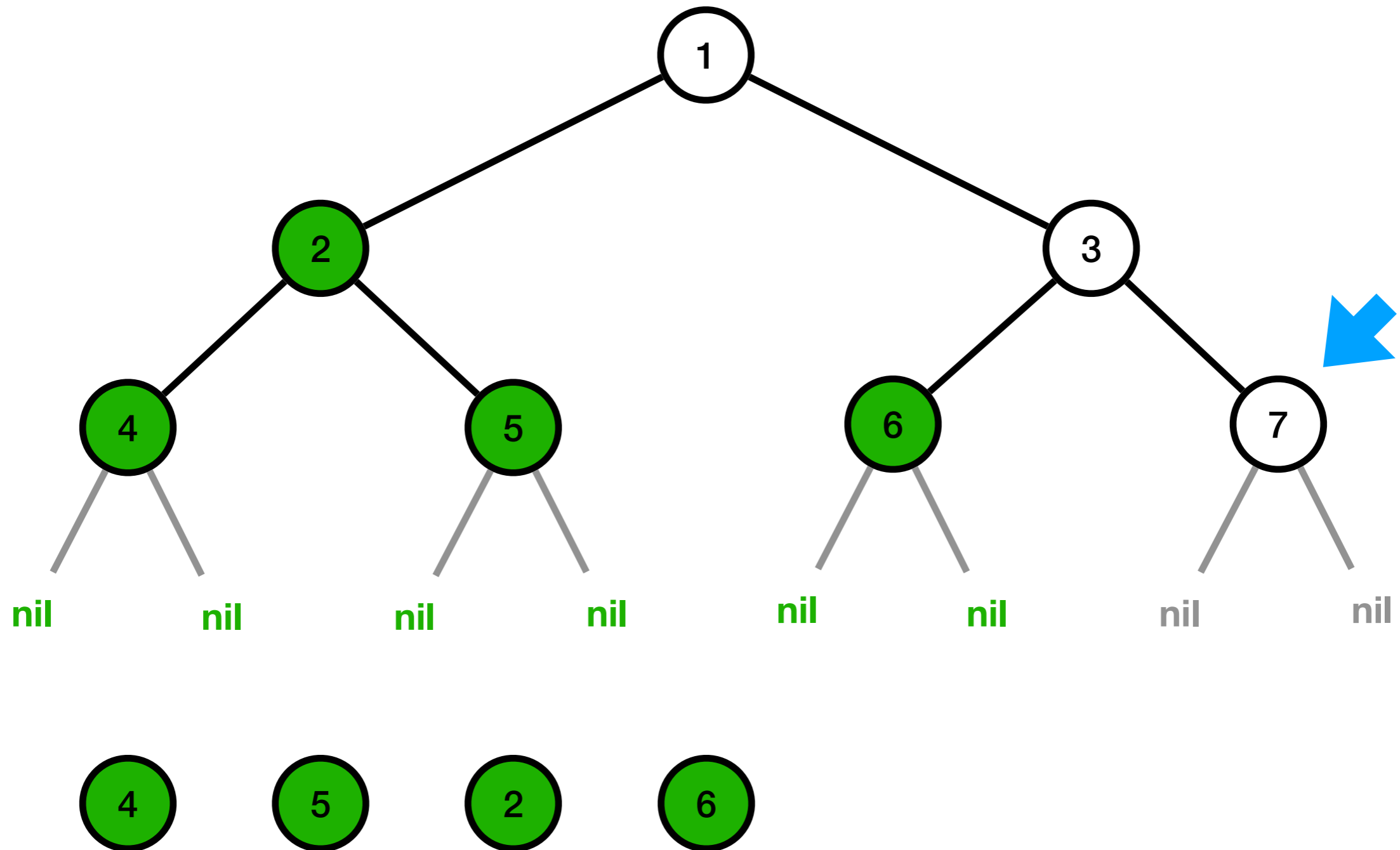
Parcours postfixe



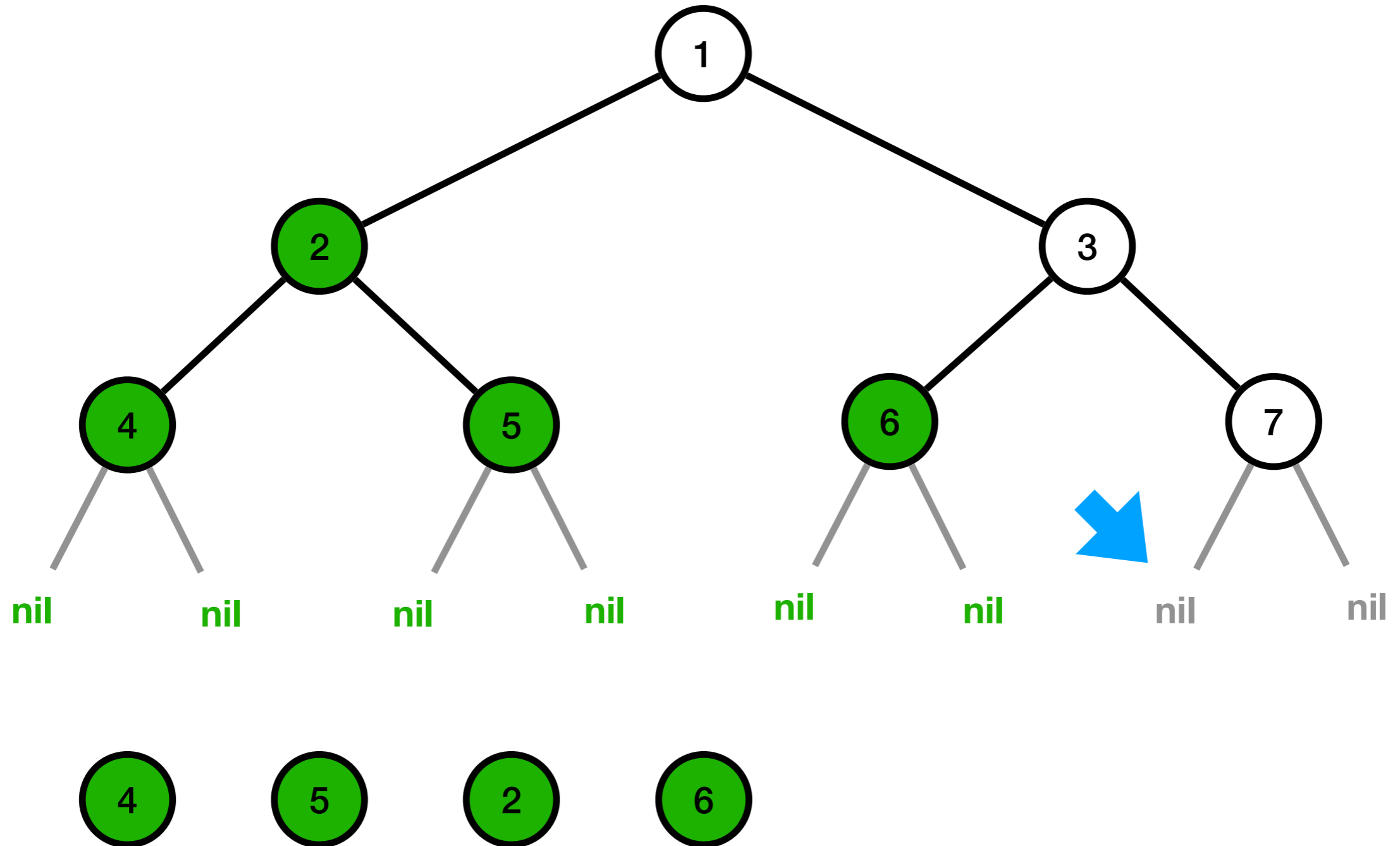
Parcours postfixe



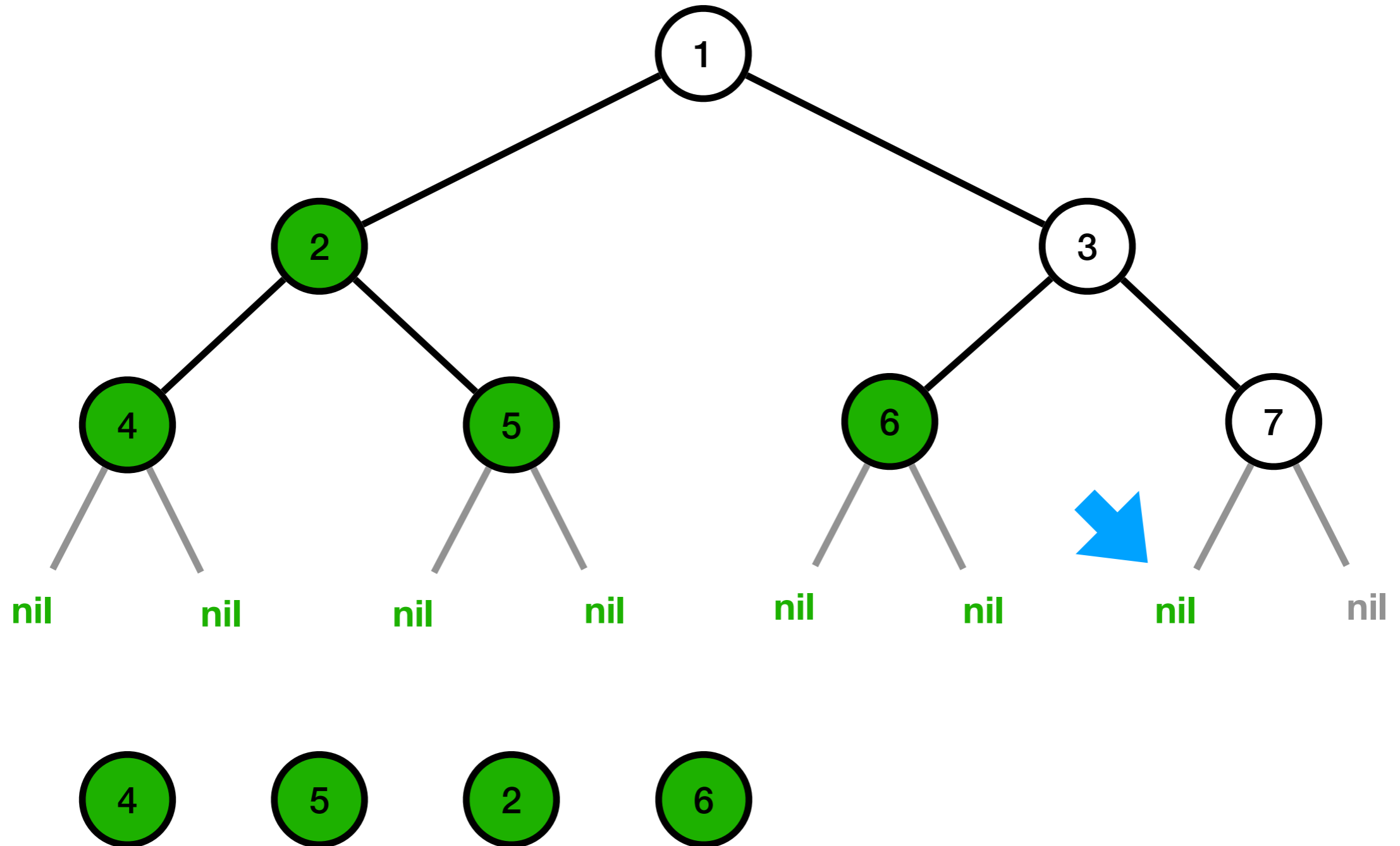
Parcours postfixe



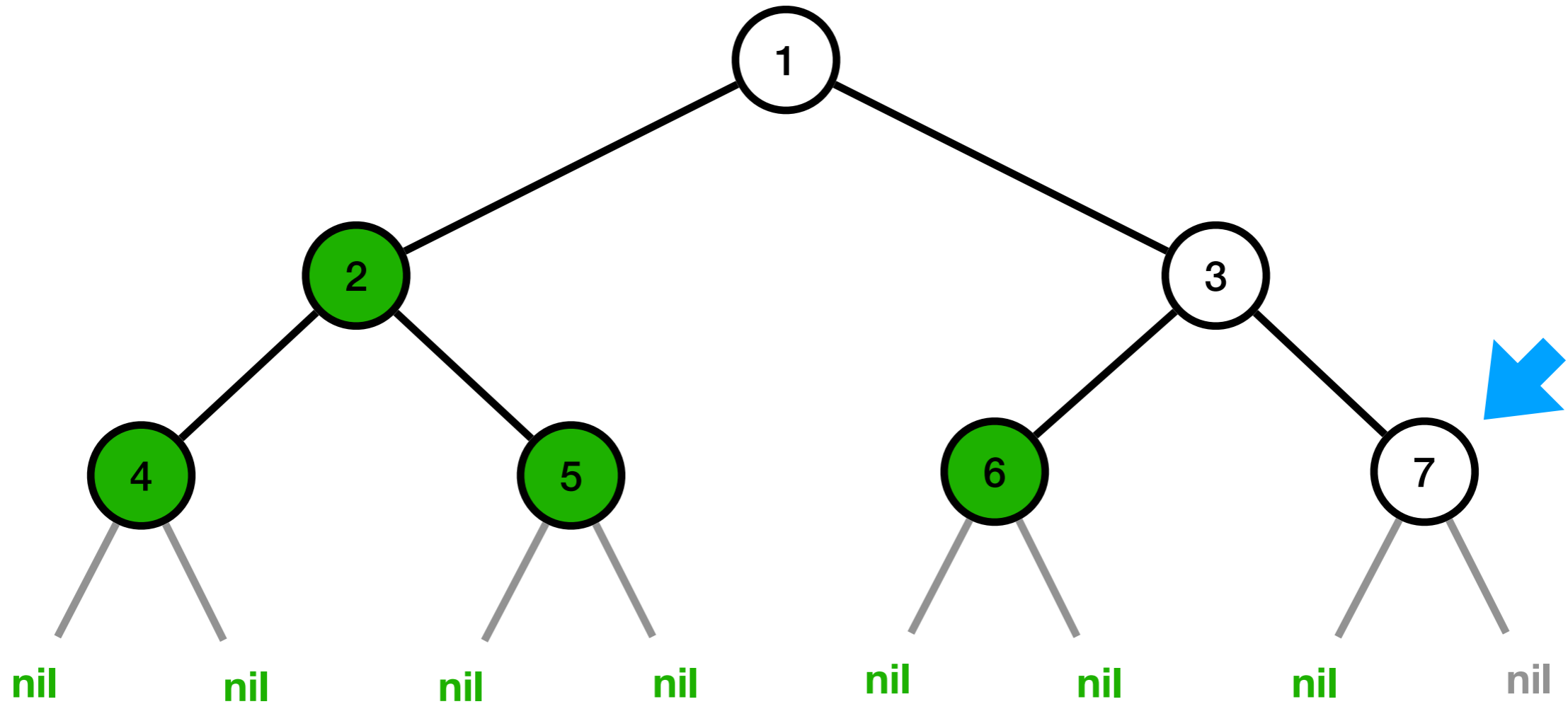
Parcours postfixe



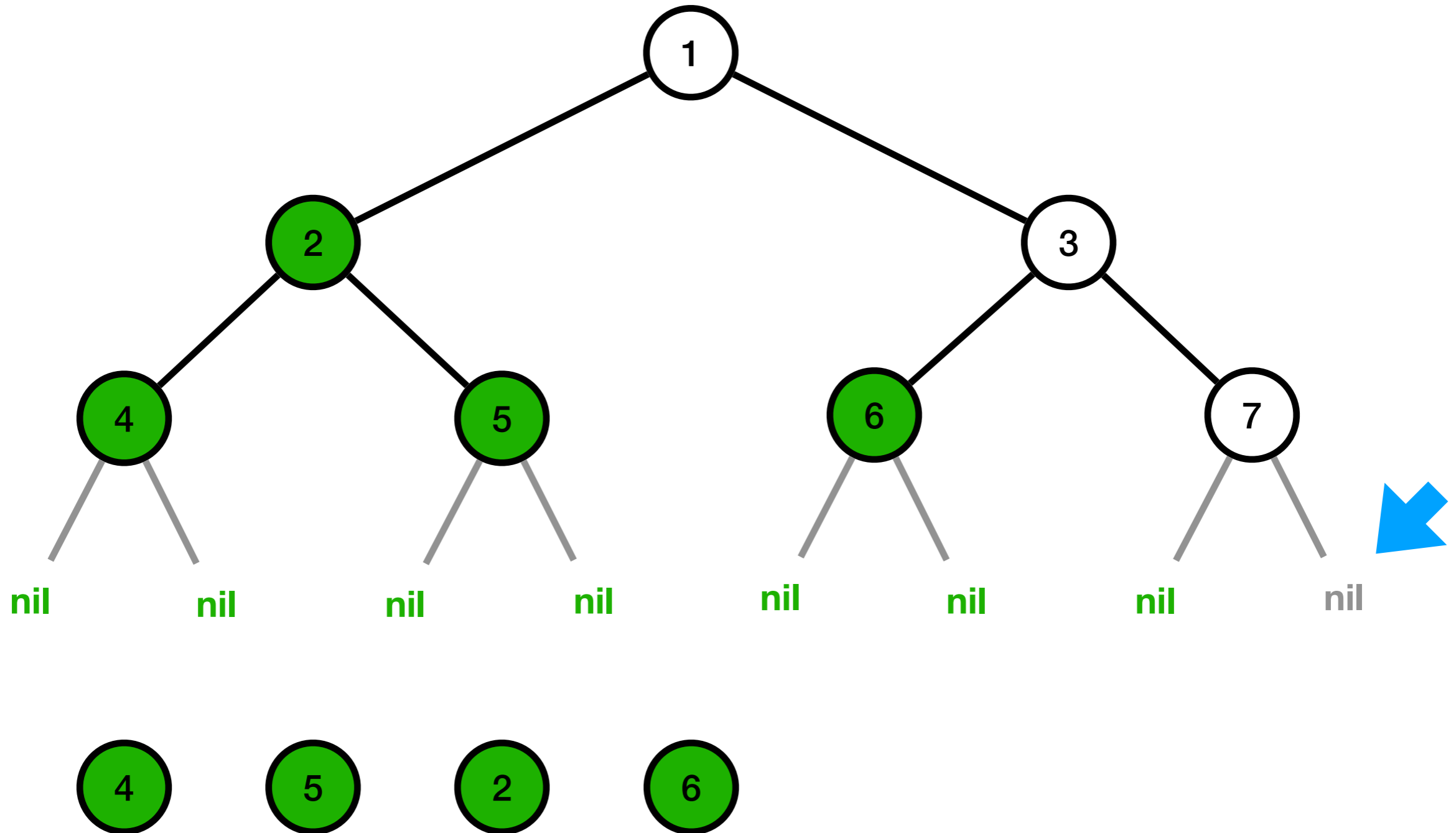
Parcours postfixe



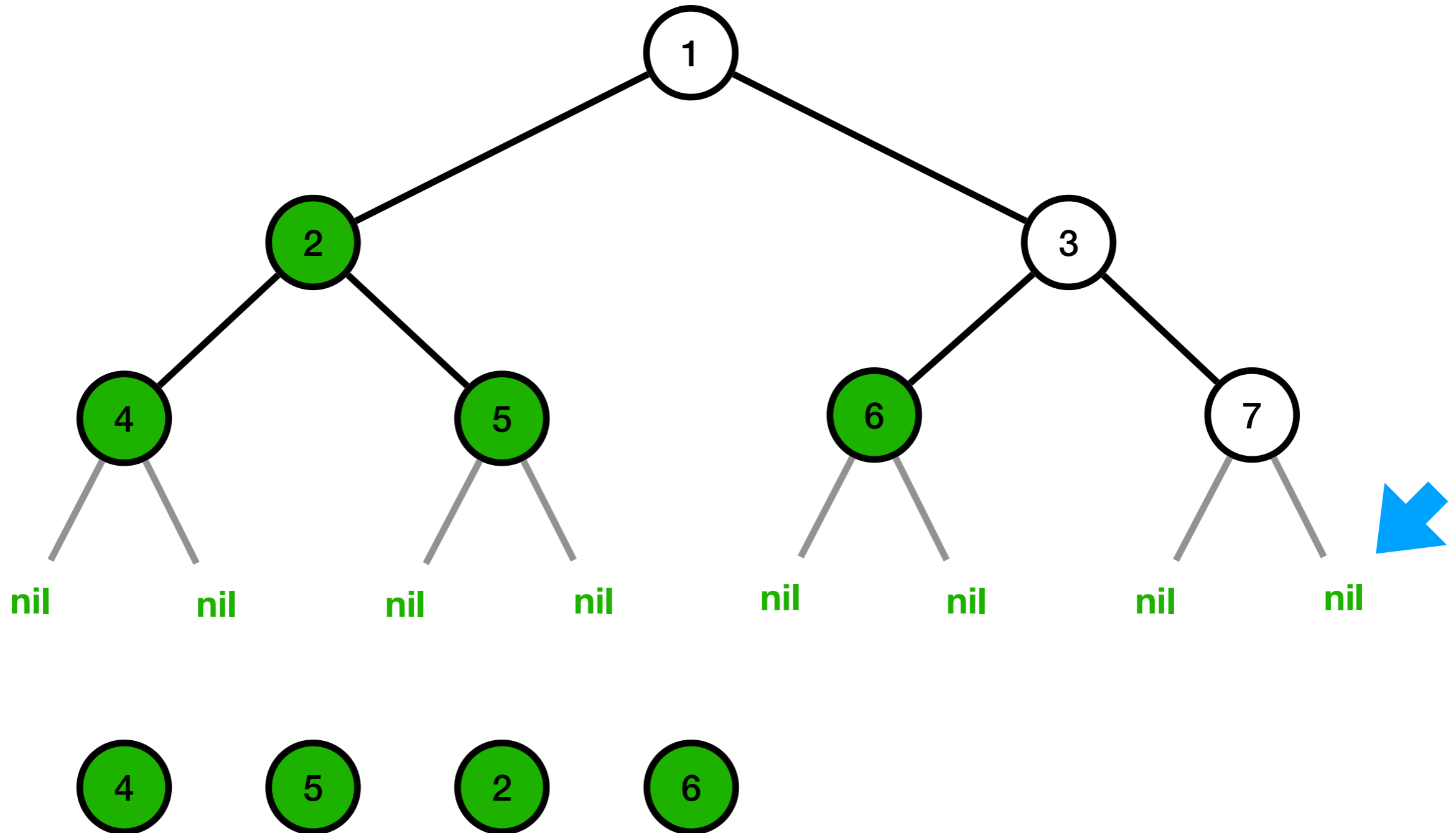
Parcours postfixe



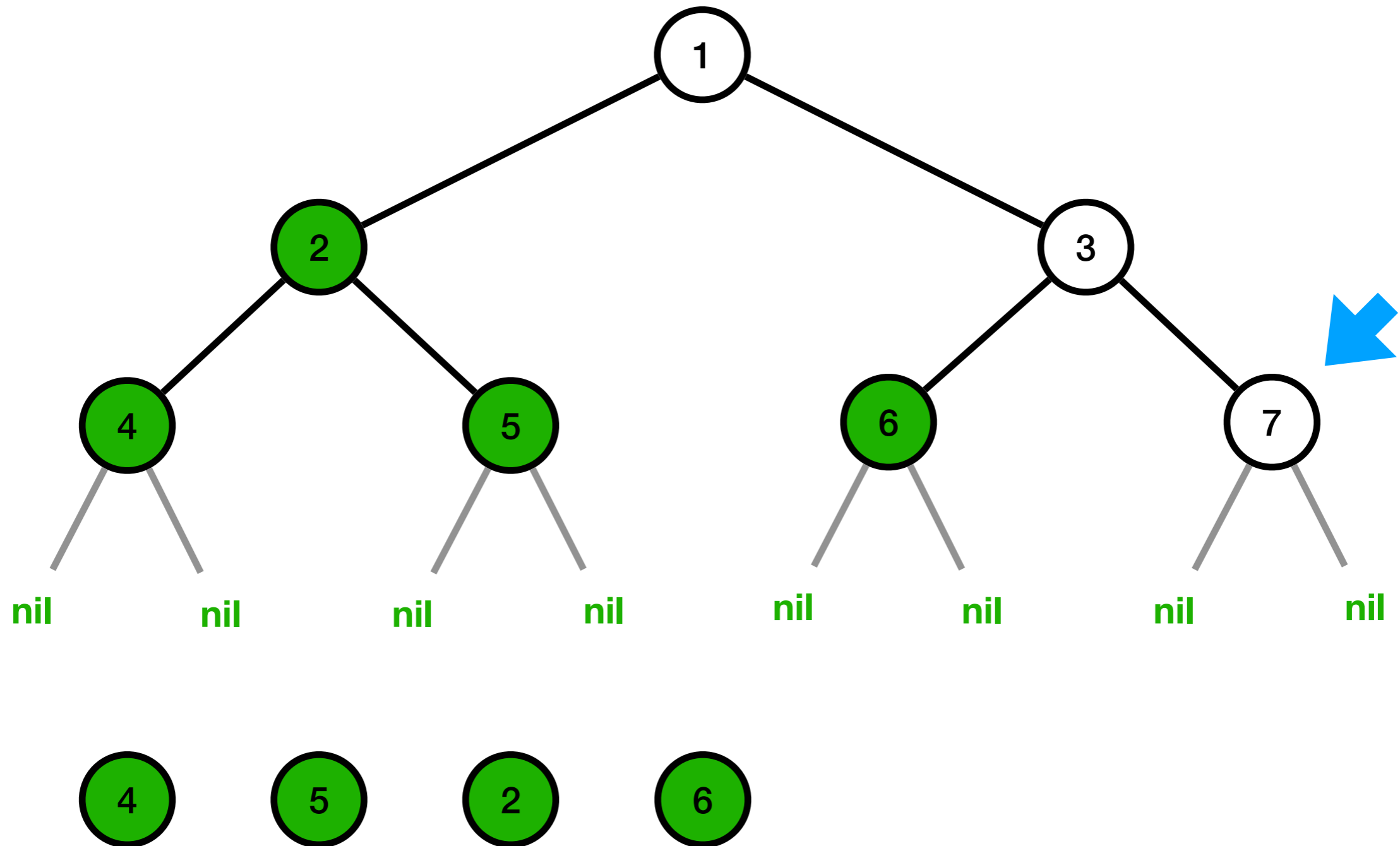
Parcours postfixe



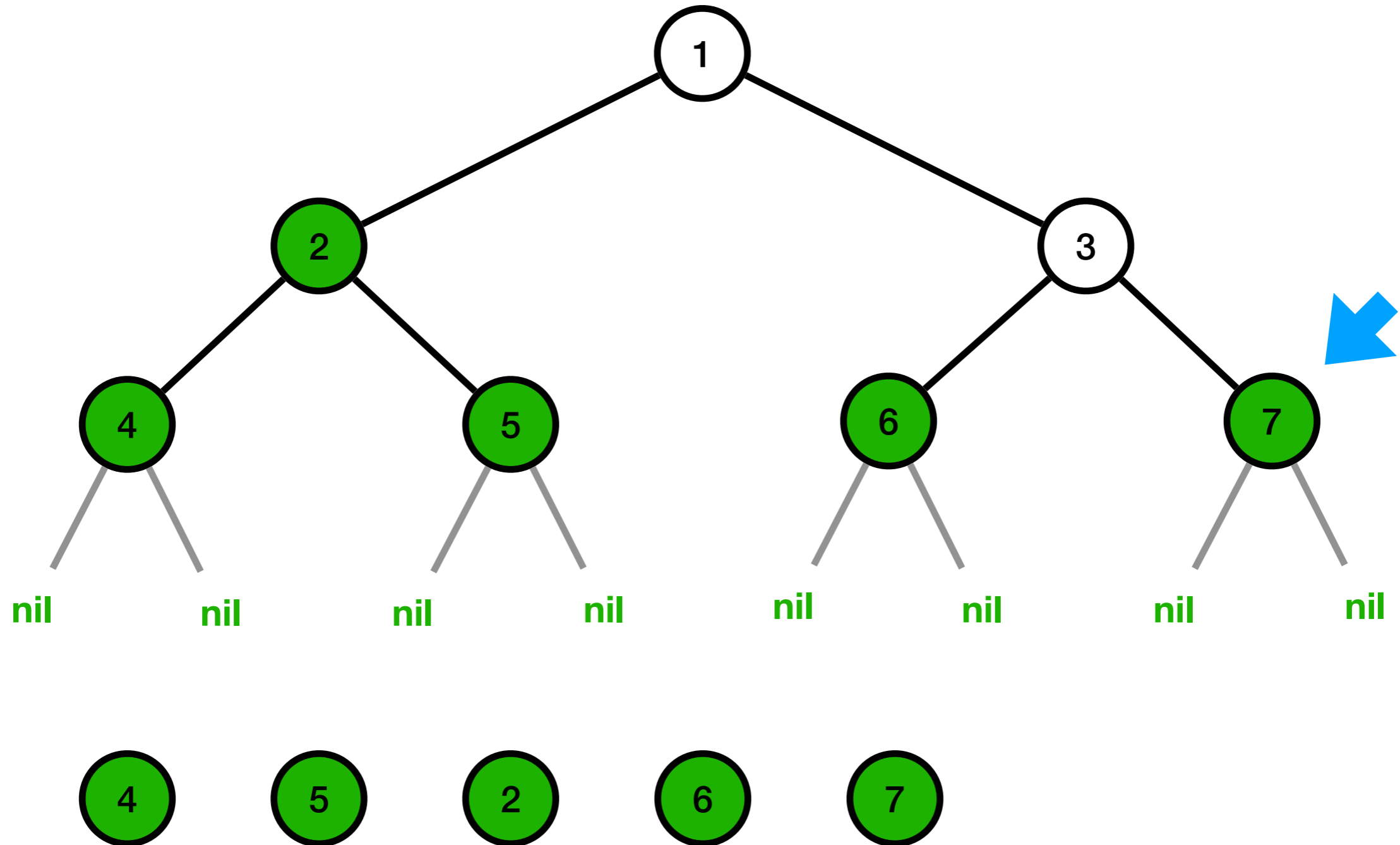
Parcours postfixe



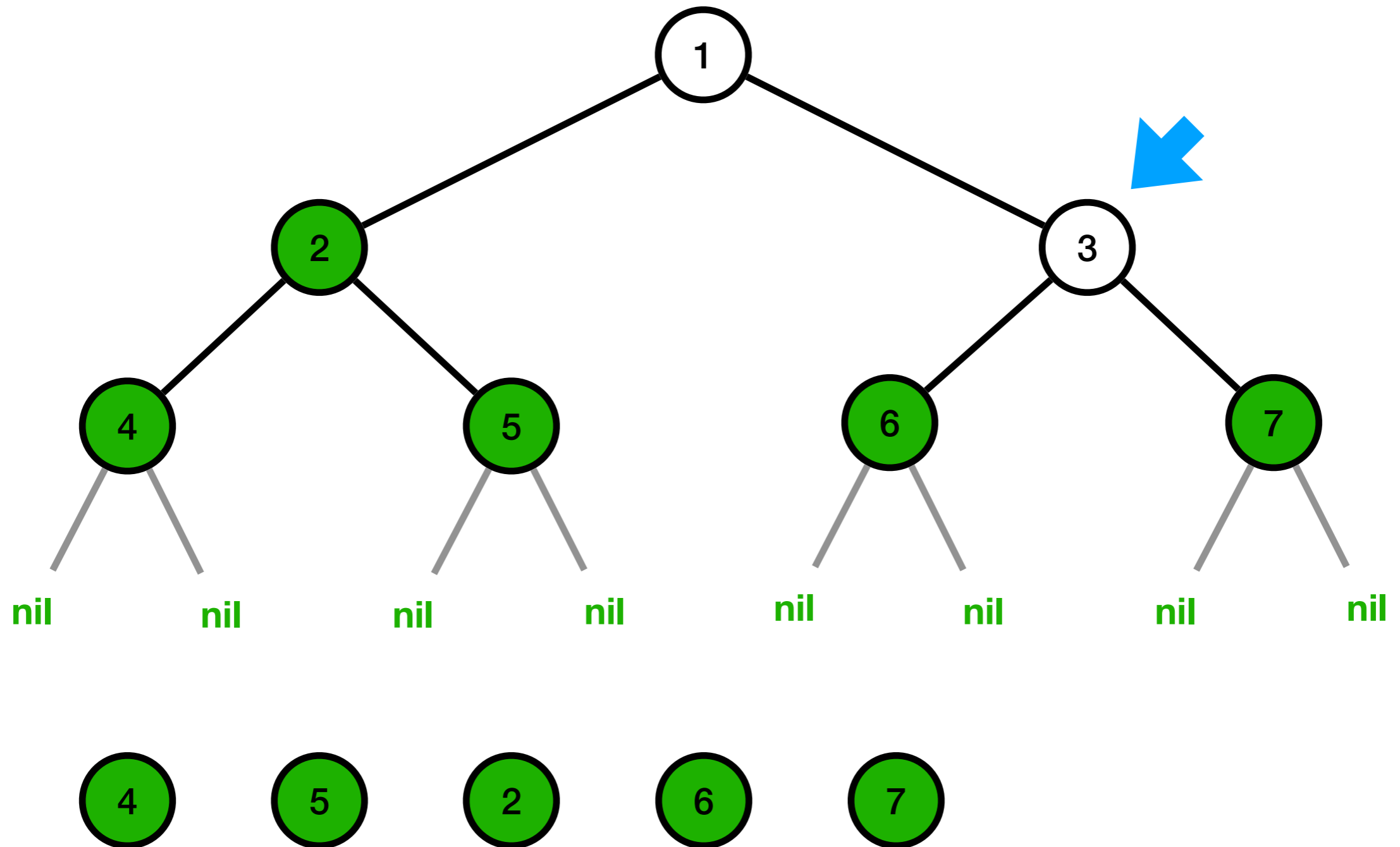
Parcours postfixe



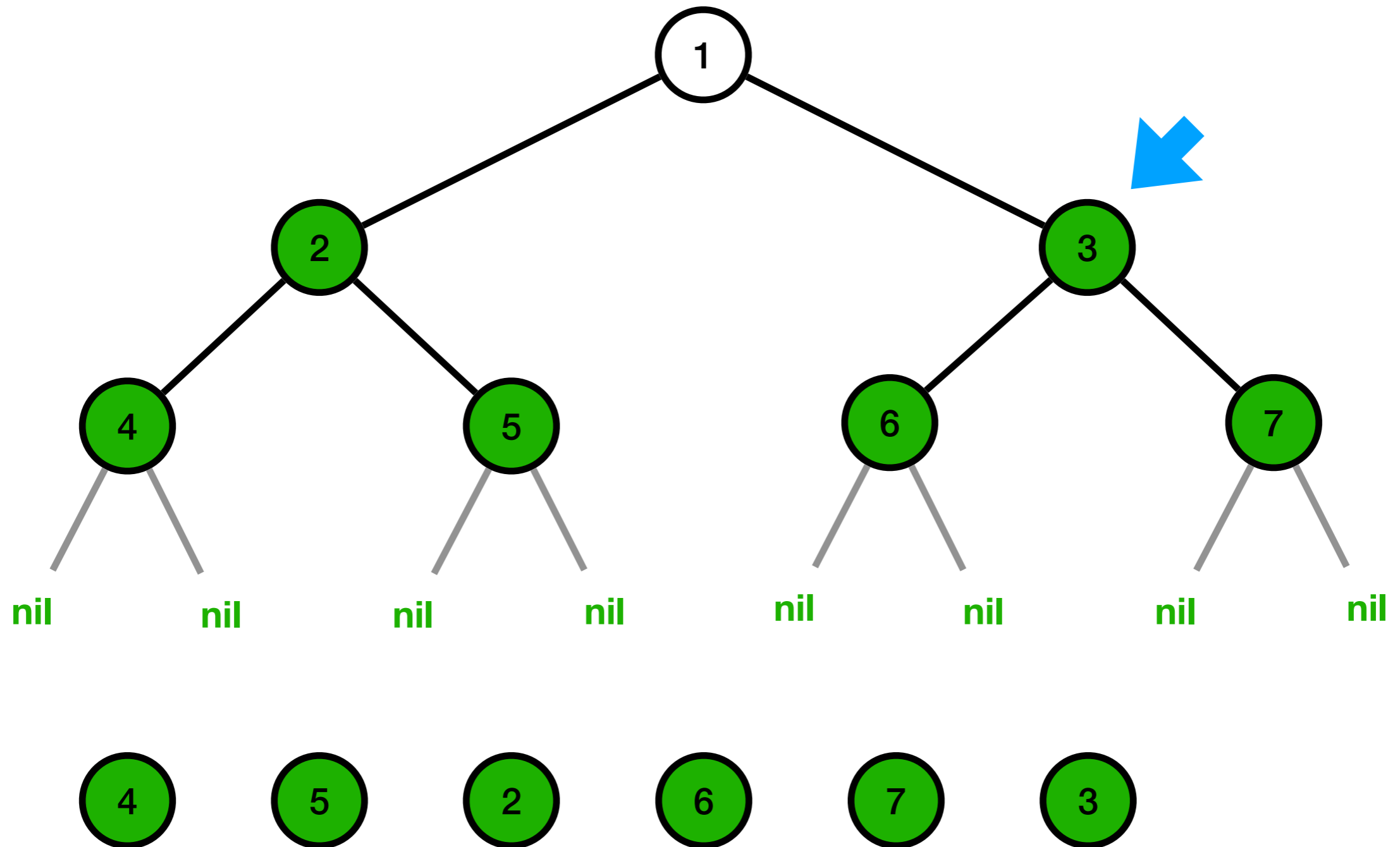
Parcours postfixe



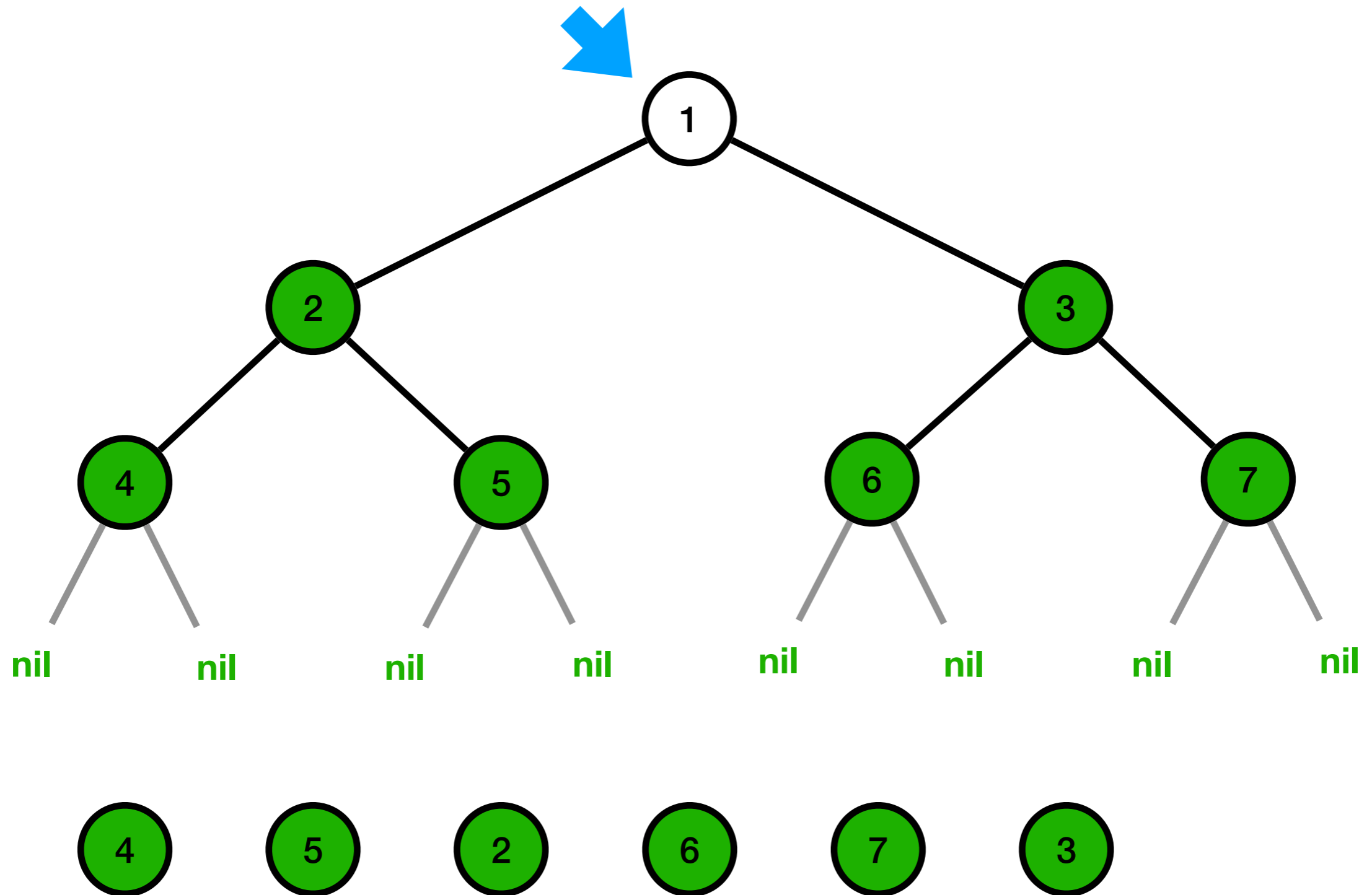
Parcours postfixe



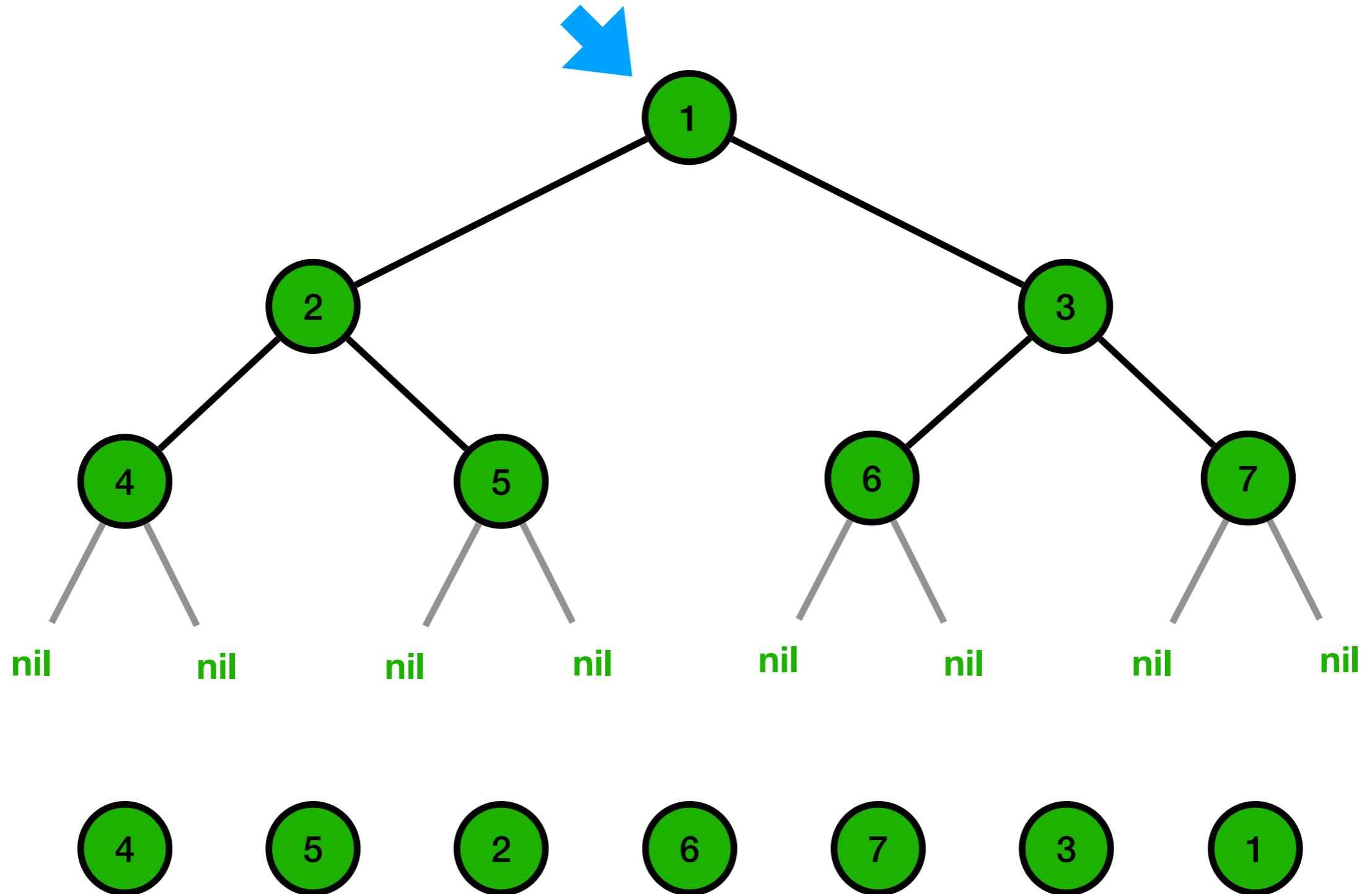
Parcours postfixe



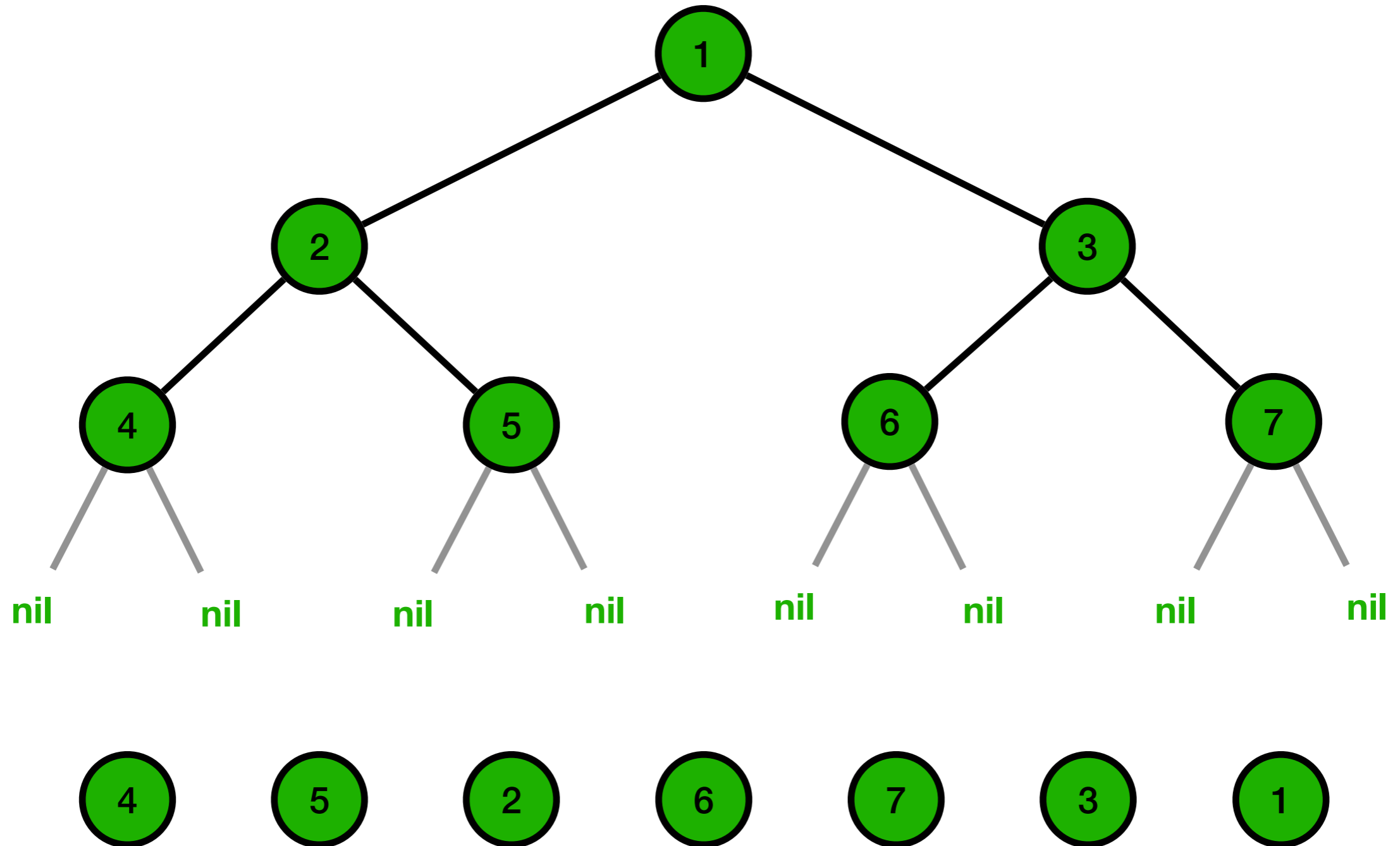
Parcours postfixe



Parcours postfixe



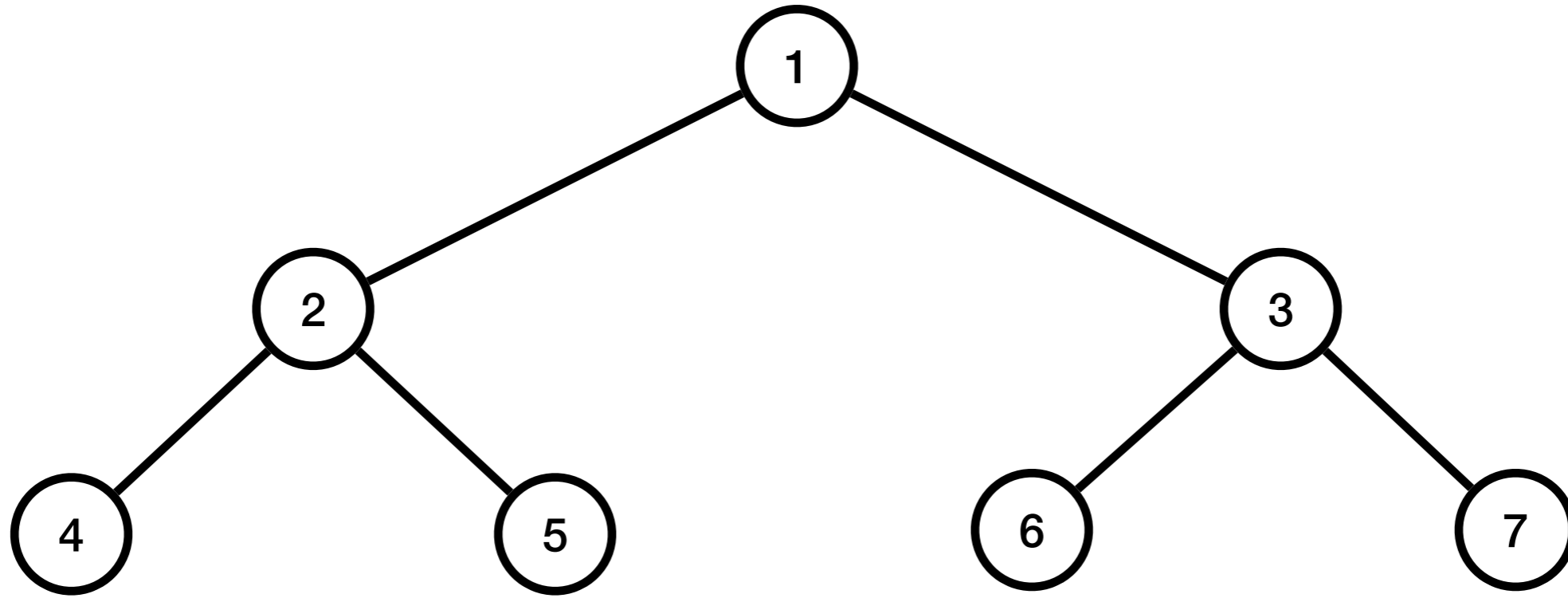
Parcours postfixe



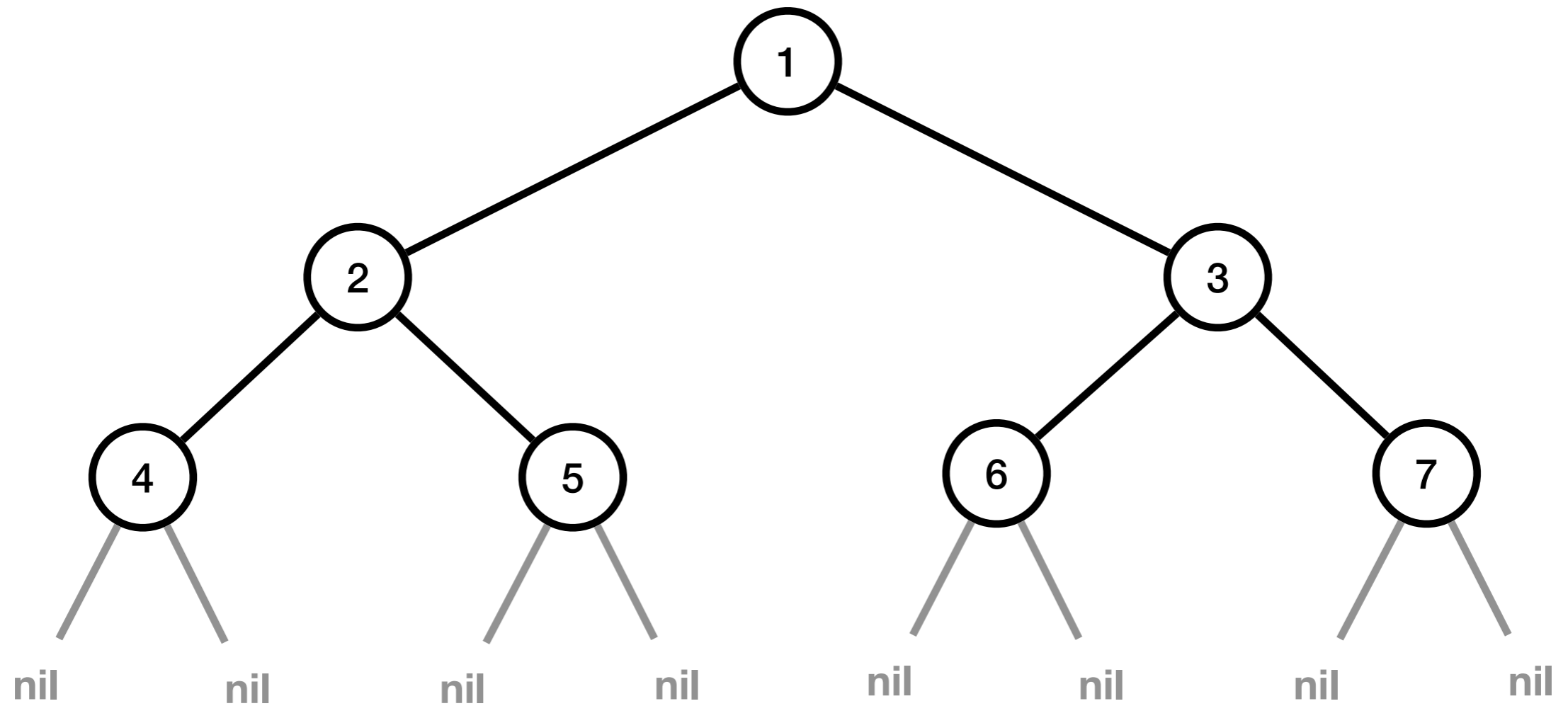
Parcours infixe

```
def parcours_infixe(arbre):  
    if not est_vide(arbre):  
        parcours_infixe(enfant_gauche(arbre))  
        print(valeur(arbre))  
        parcours_infixe(enfant_droit(arbre))
```

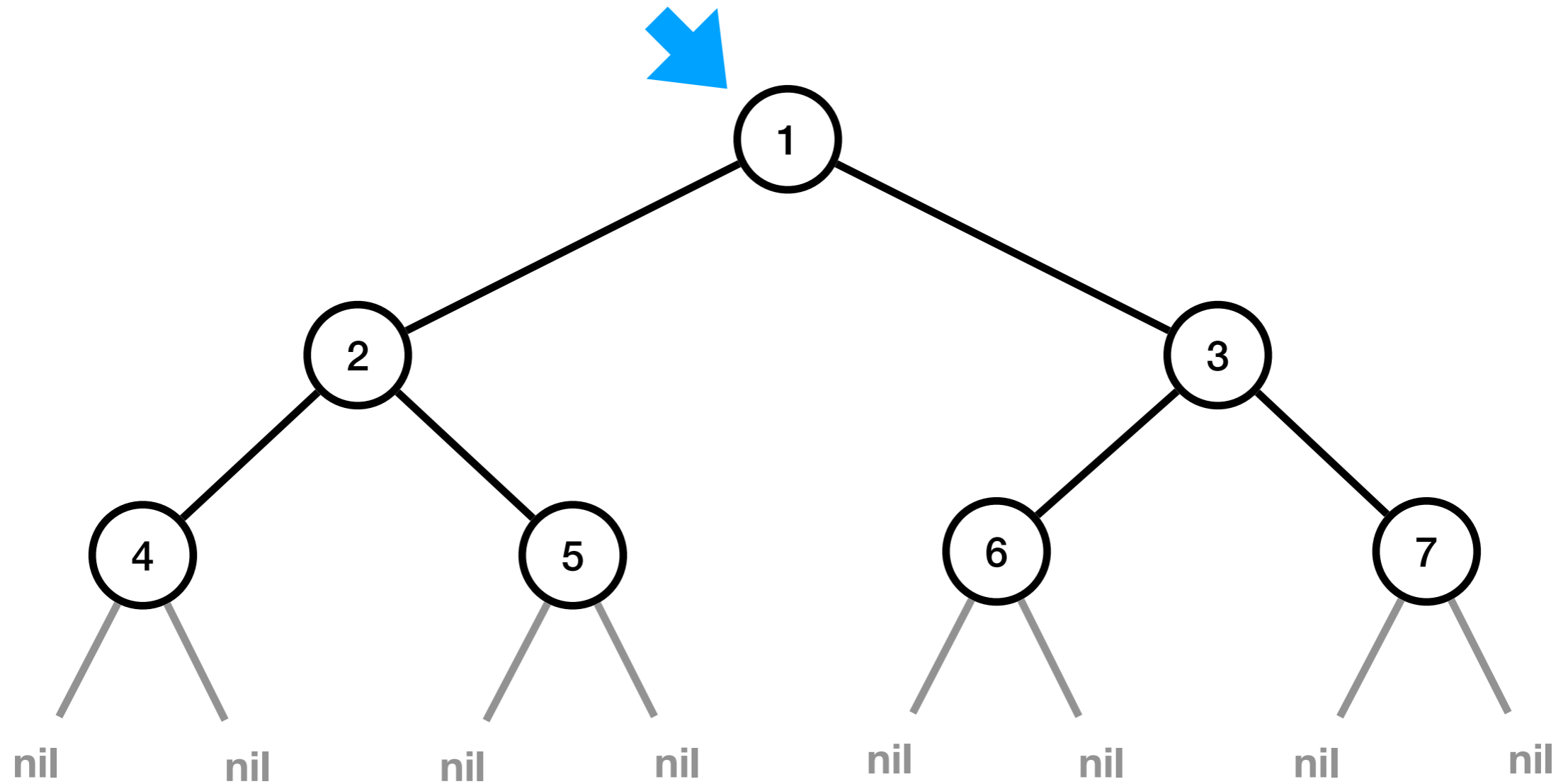

Parcours infixe



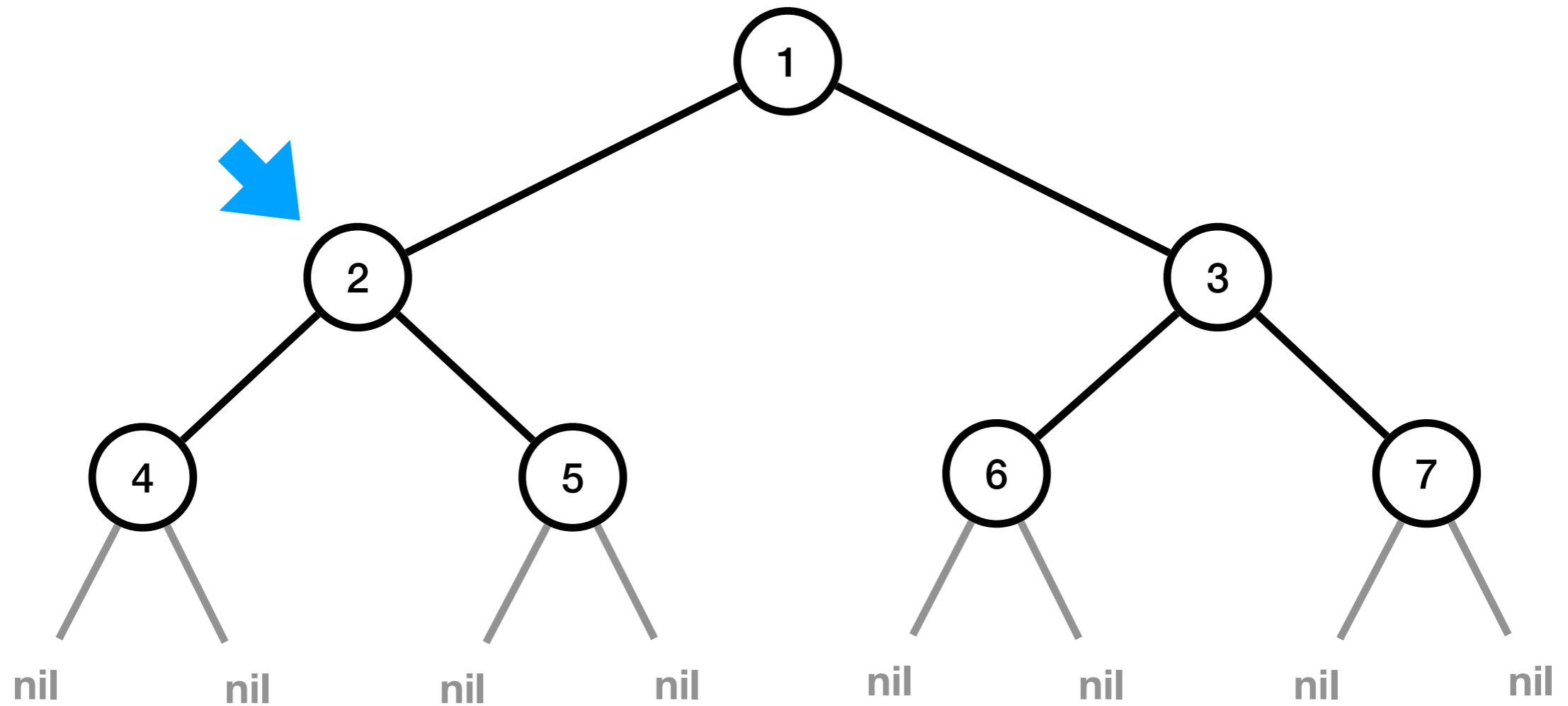
Parcours infixe



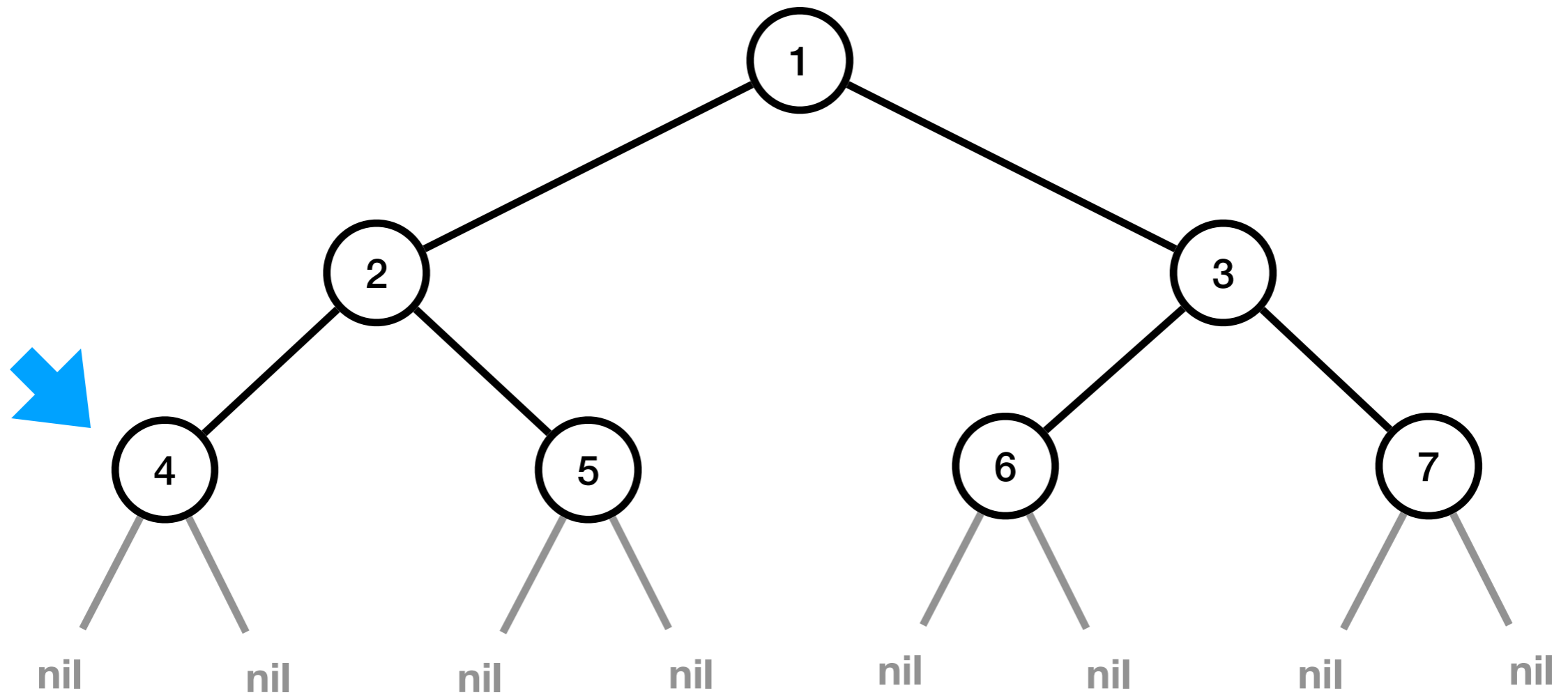
Parcours infixe



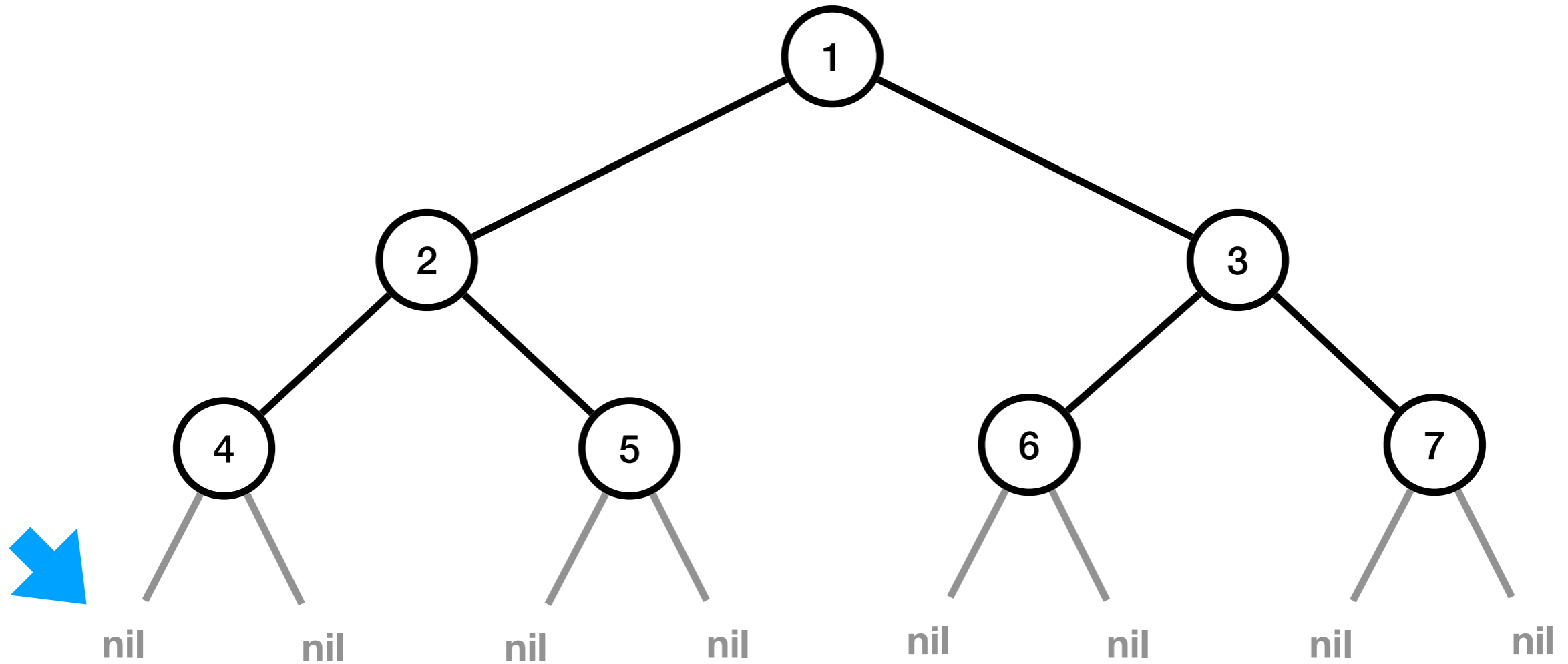
Parcours infixe



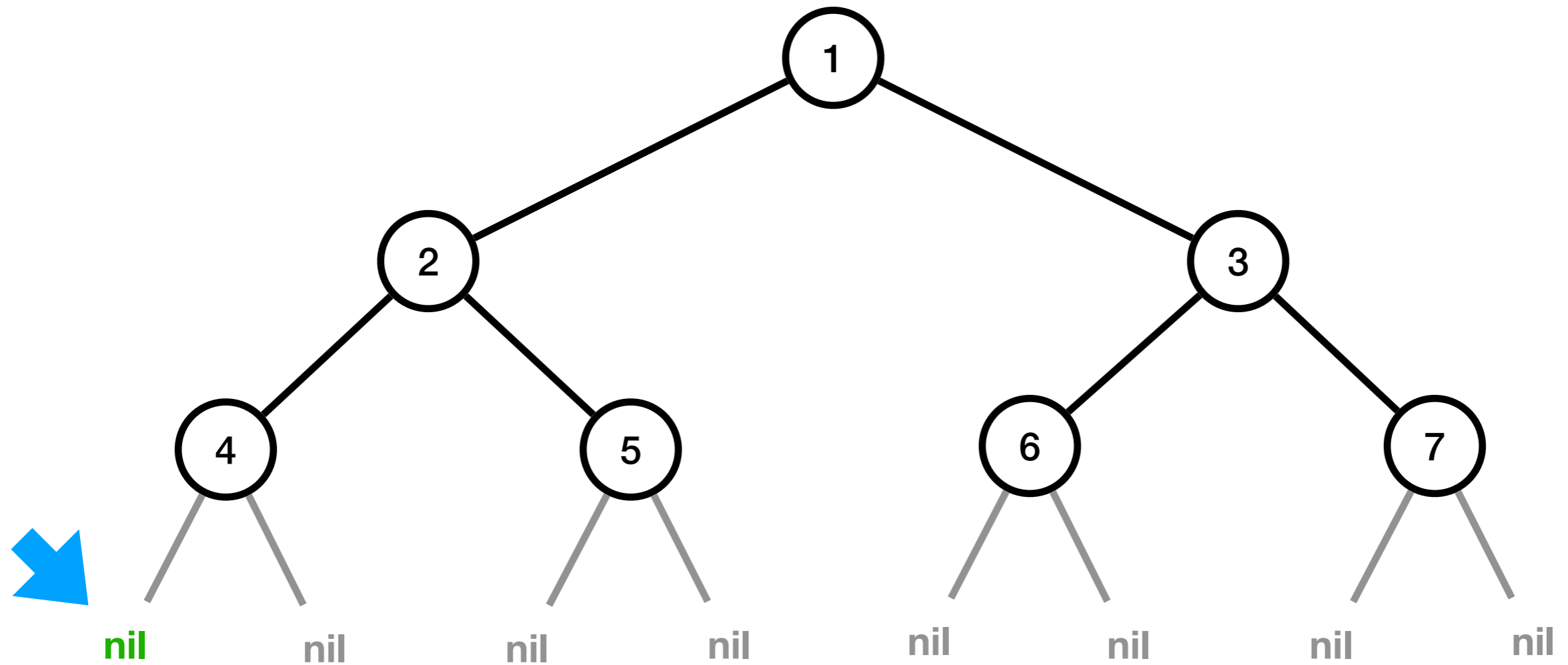
Parcours infixe



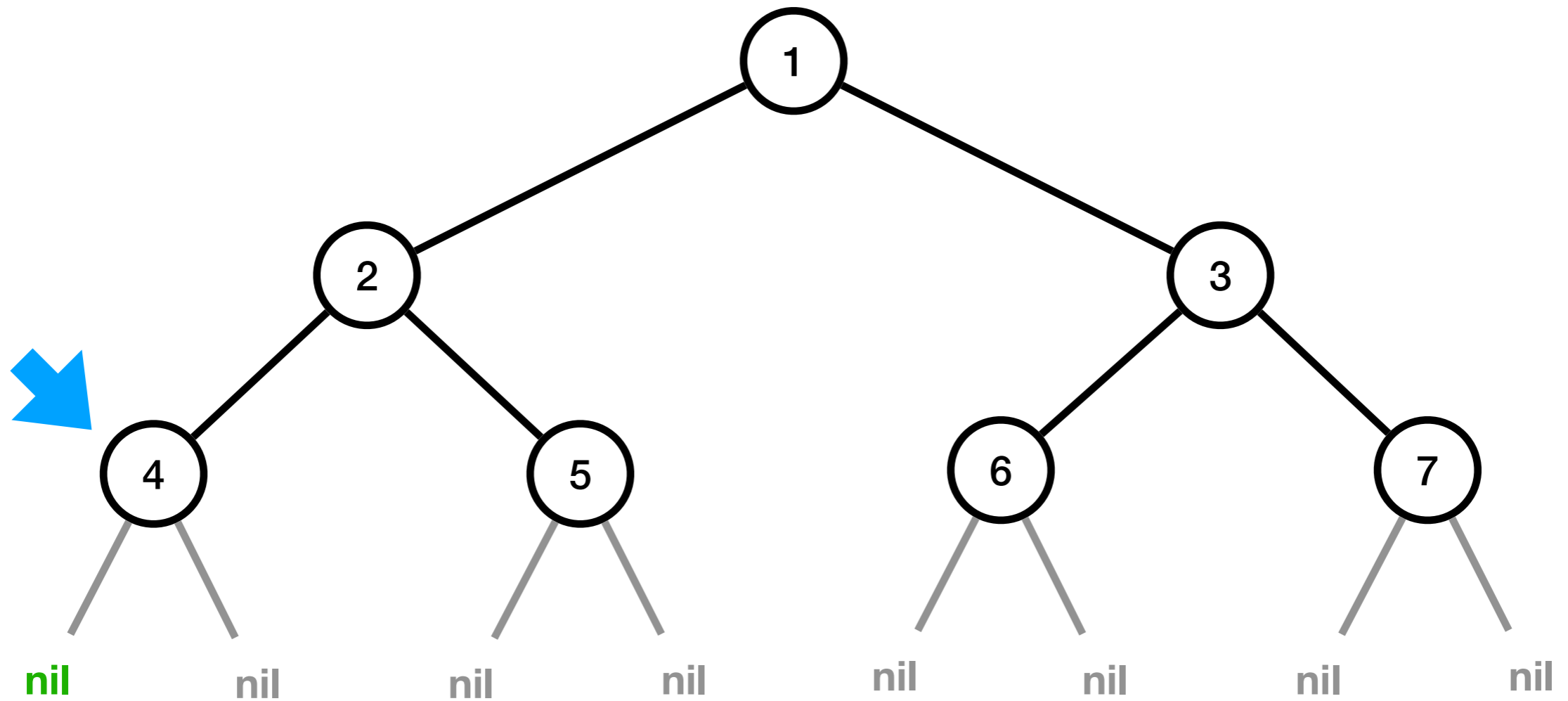
Parcours infixe



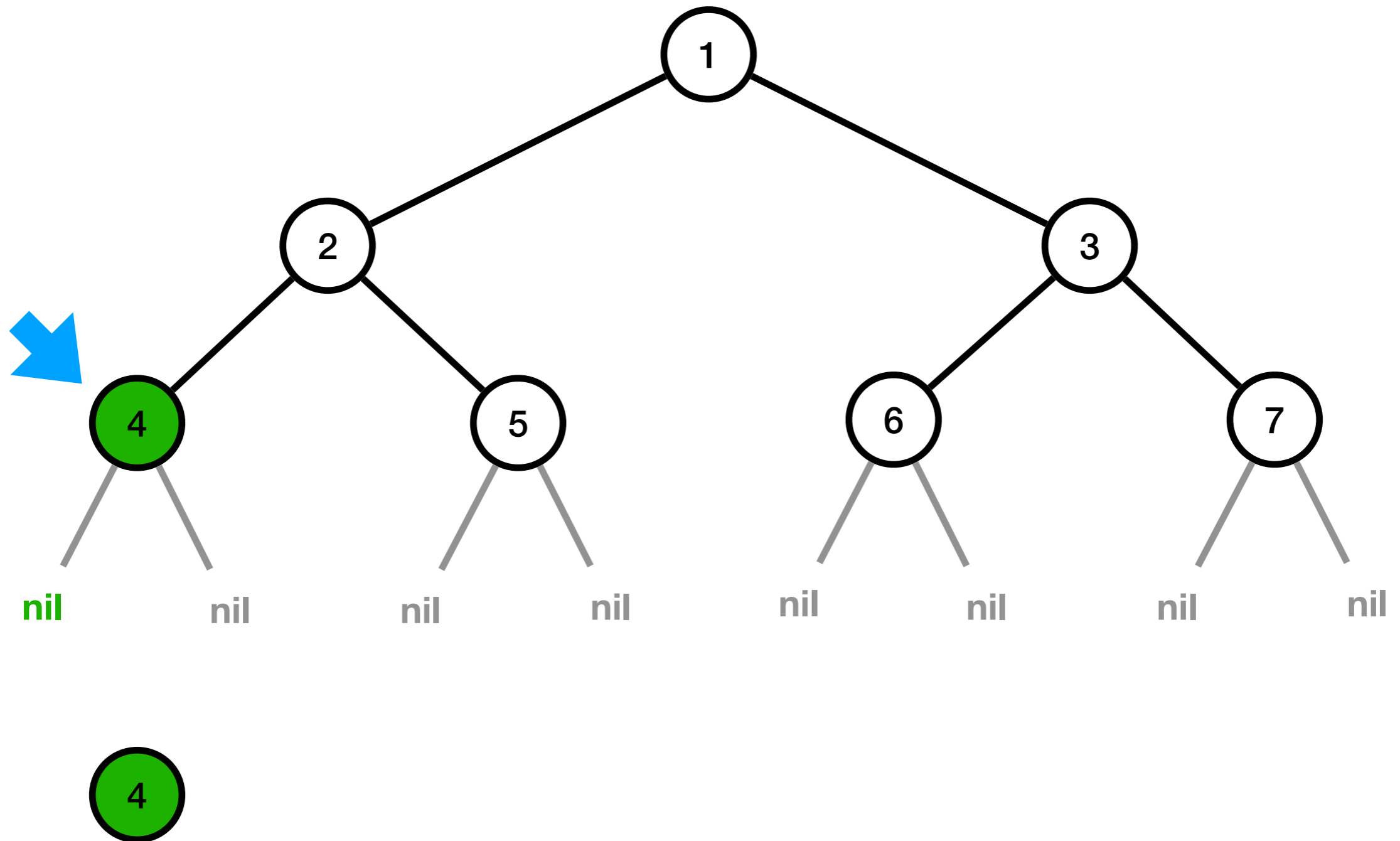
Parcours infixe



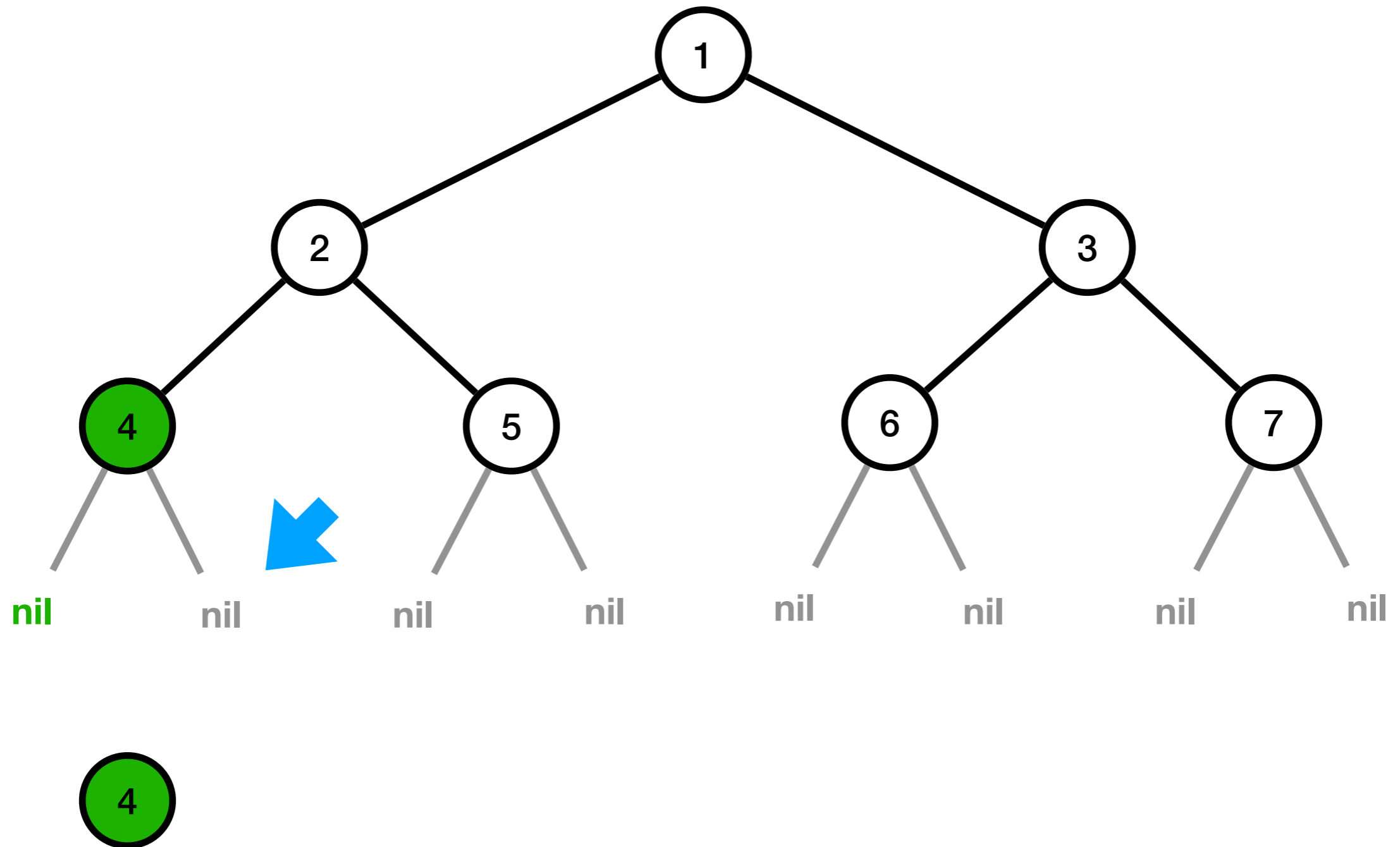
Parcours infixe



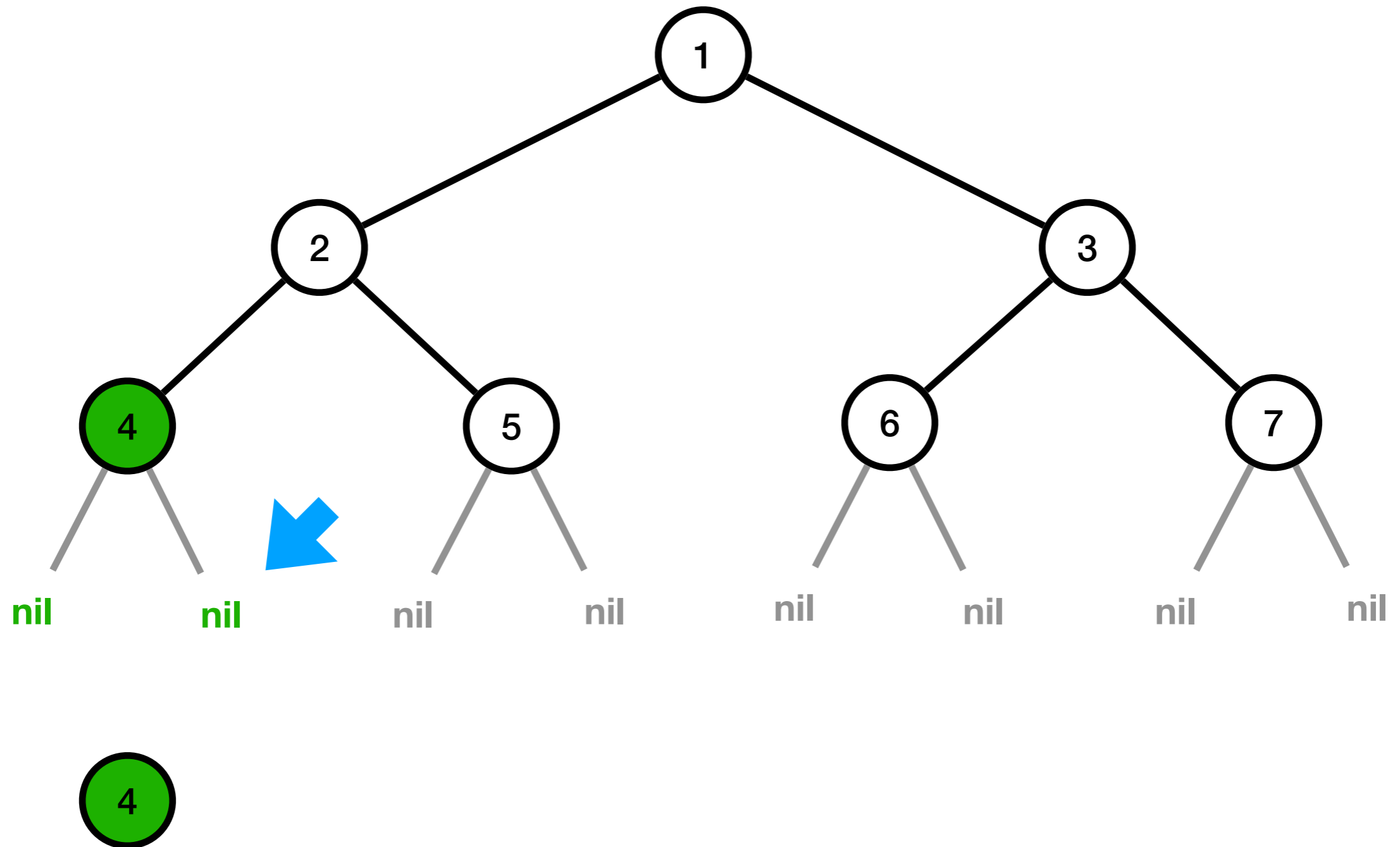
Parcours infixe



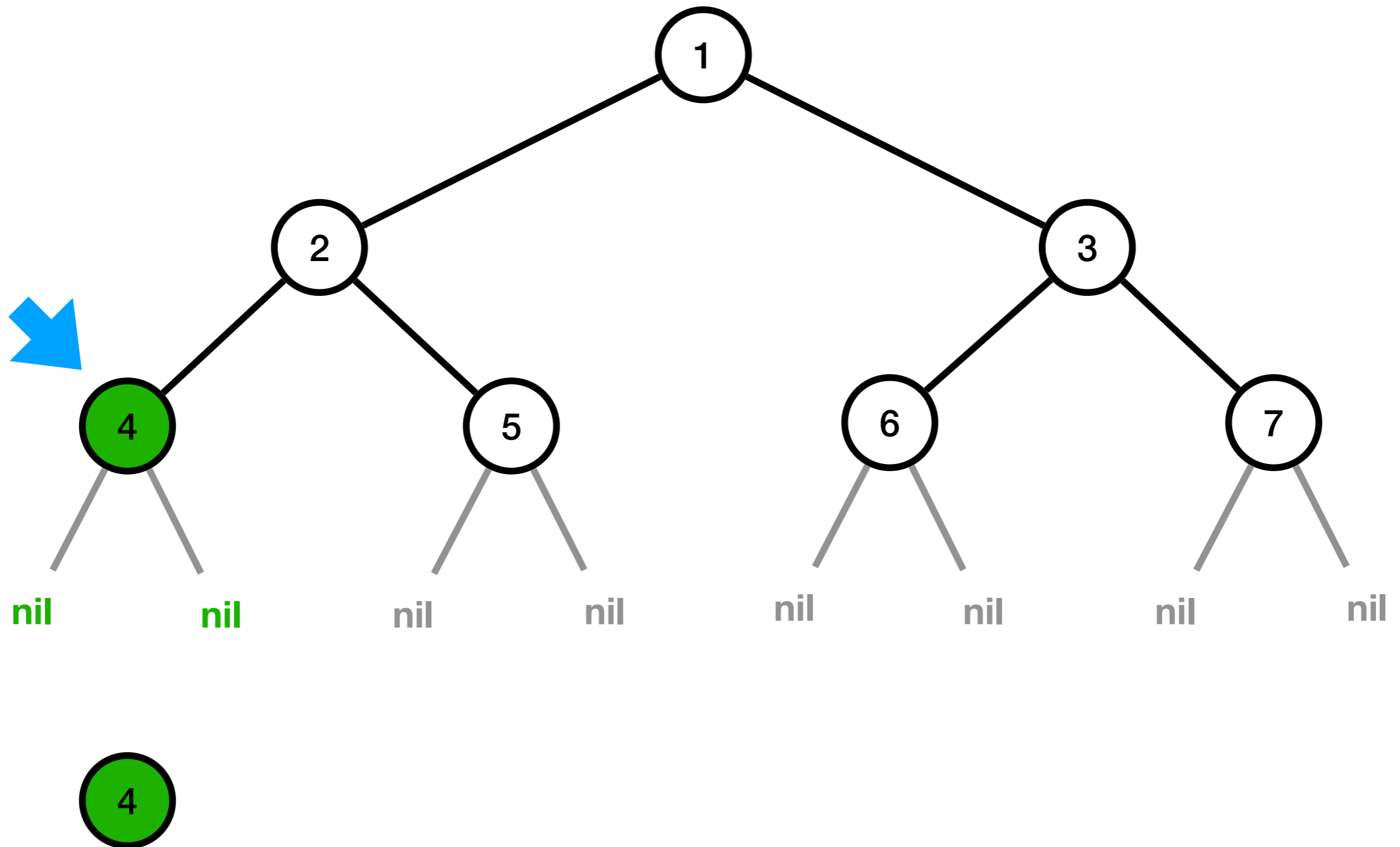
Parcours infixe



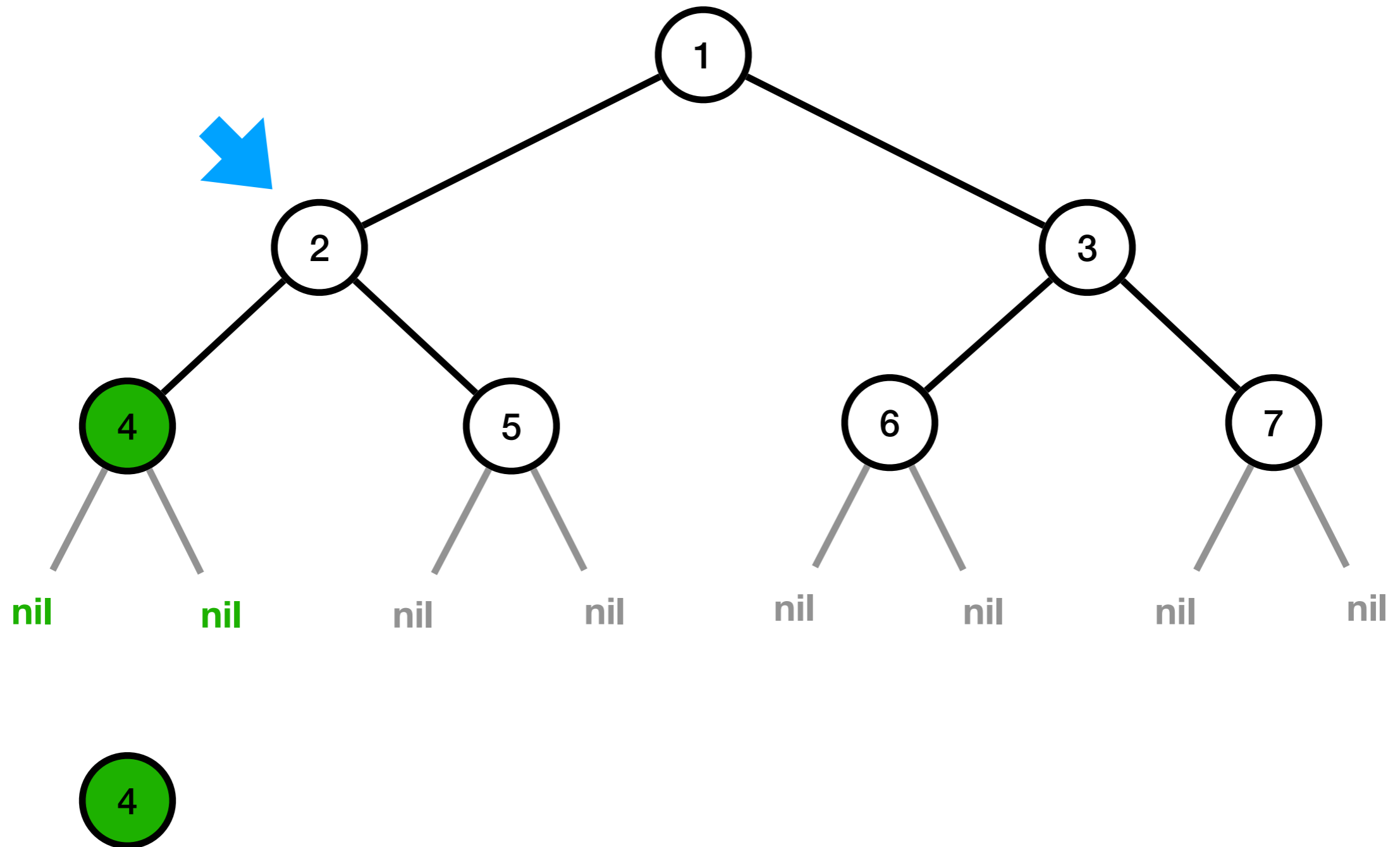
Parcours infixe



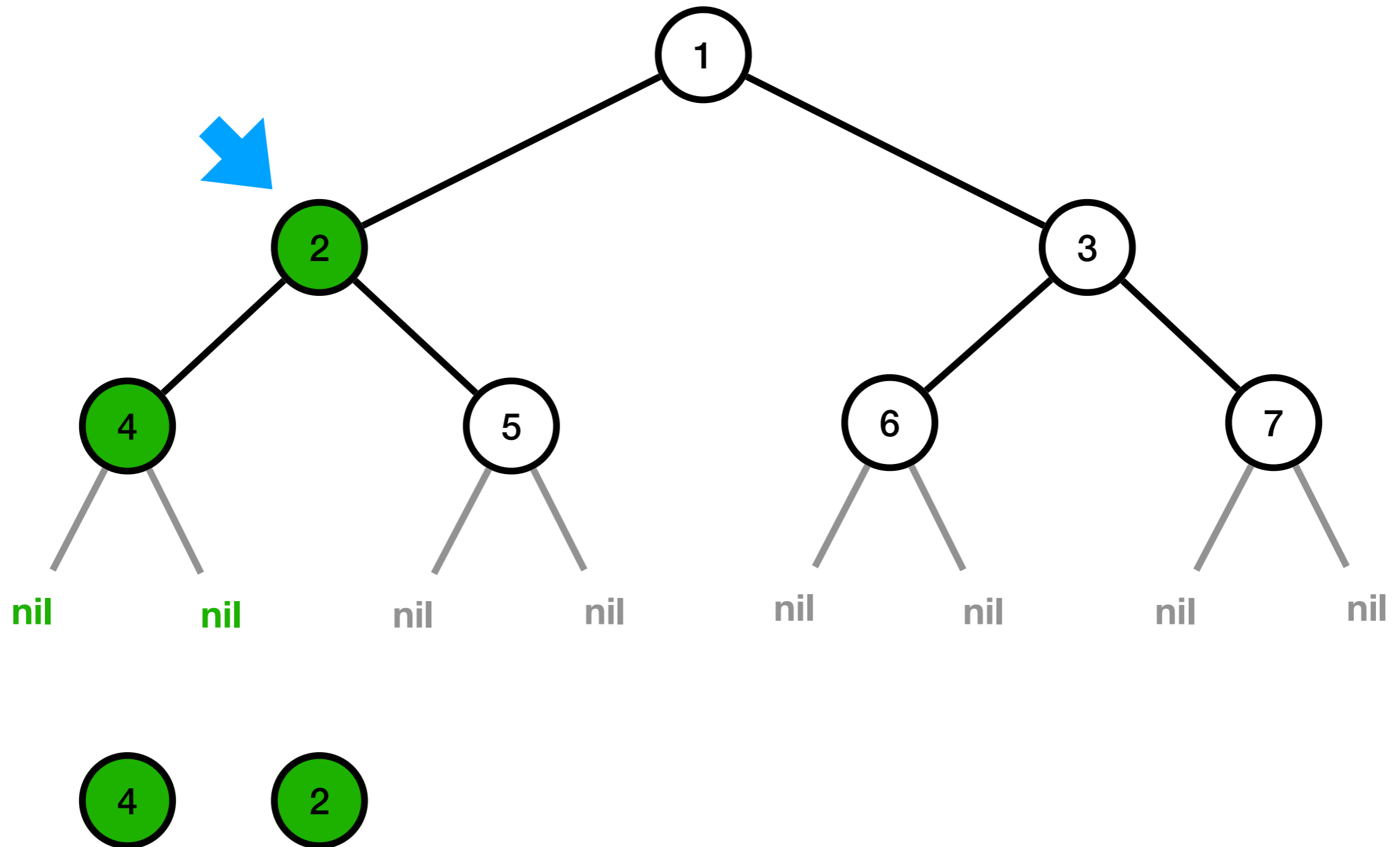
Parcours infixe



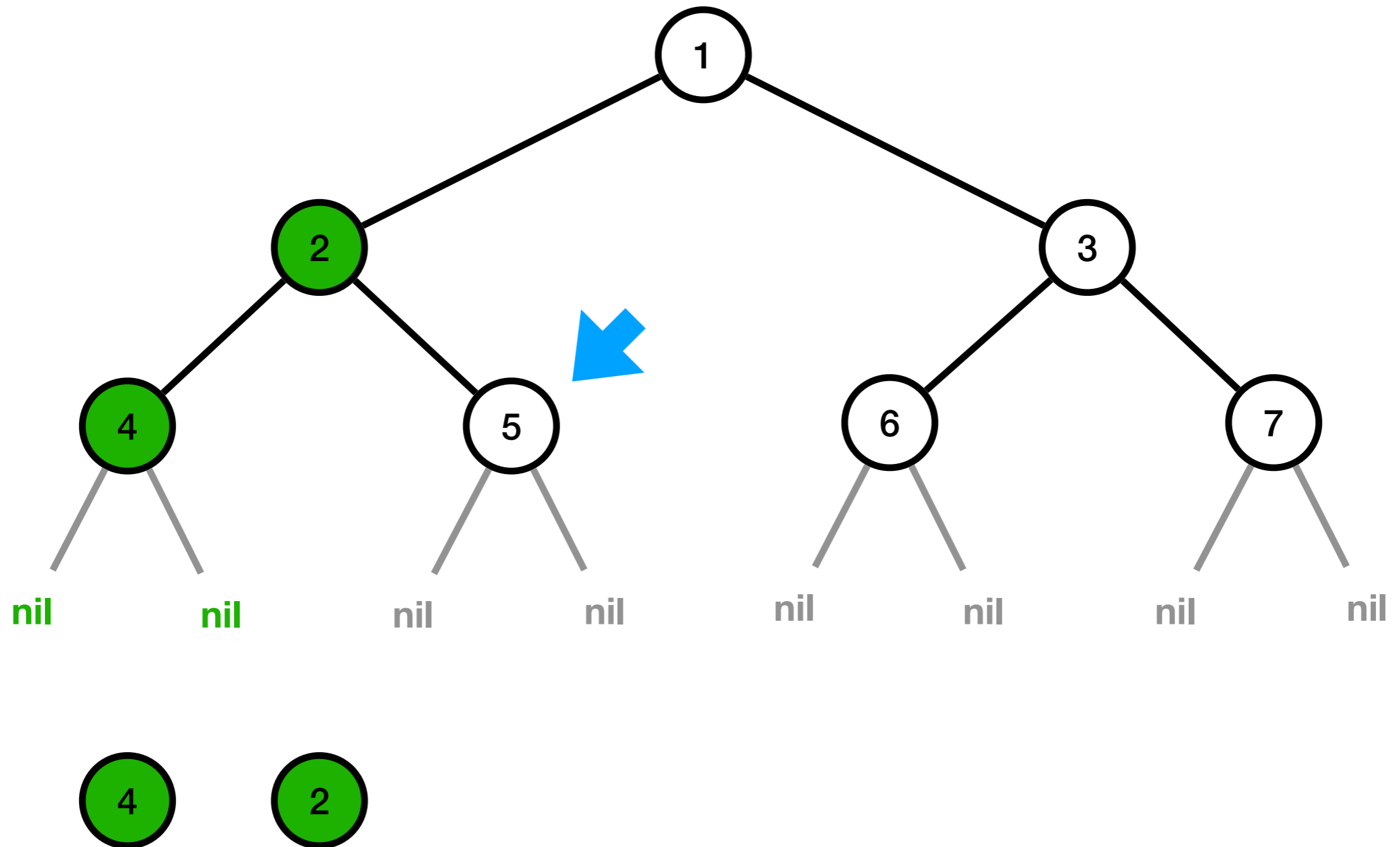
Parcours infixe



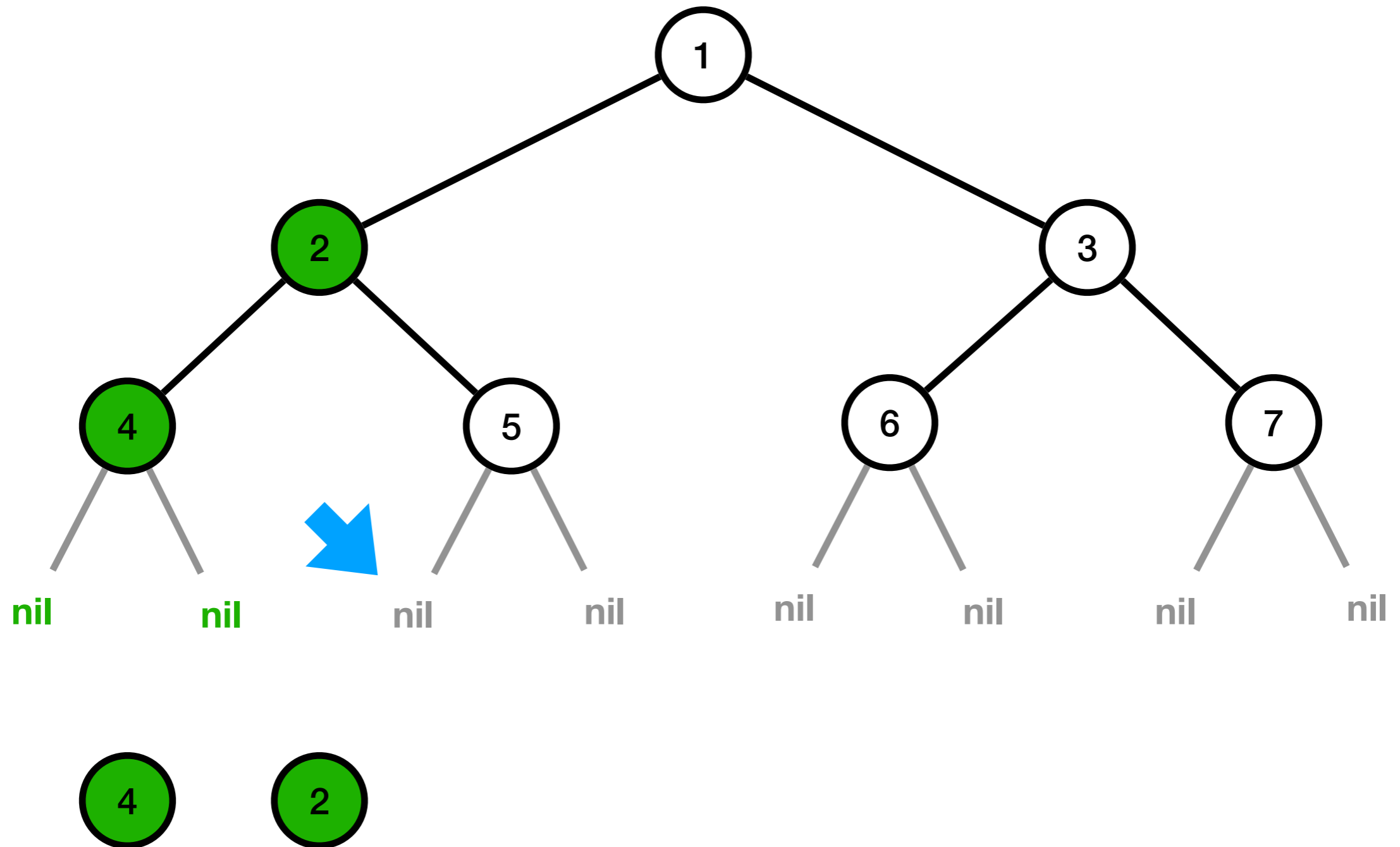
Parcours infixe



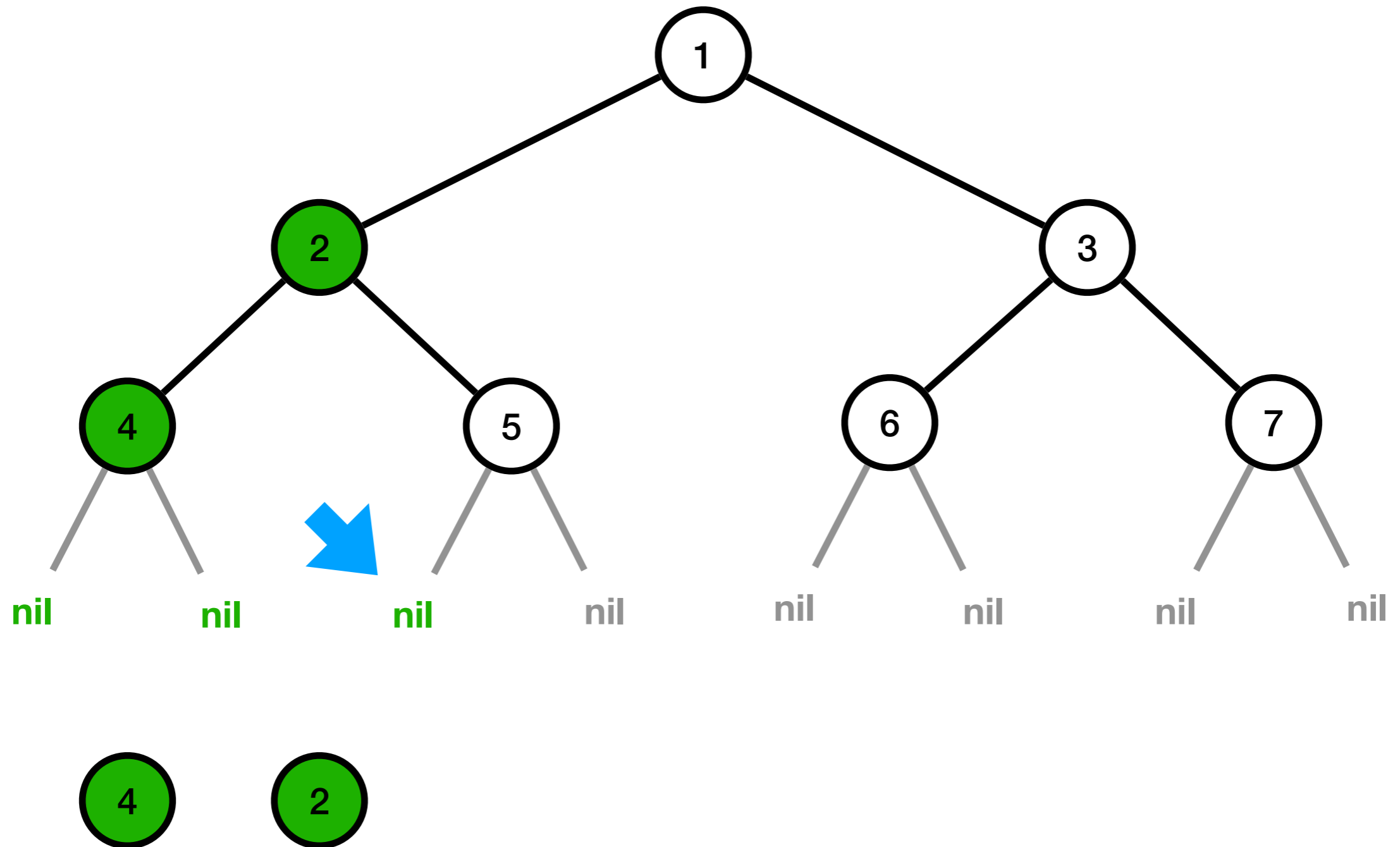
Parcours infixe



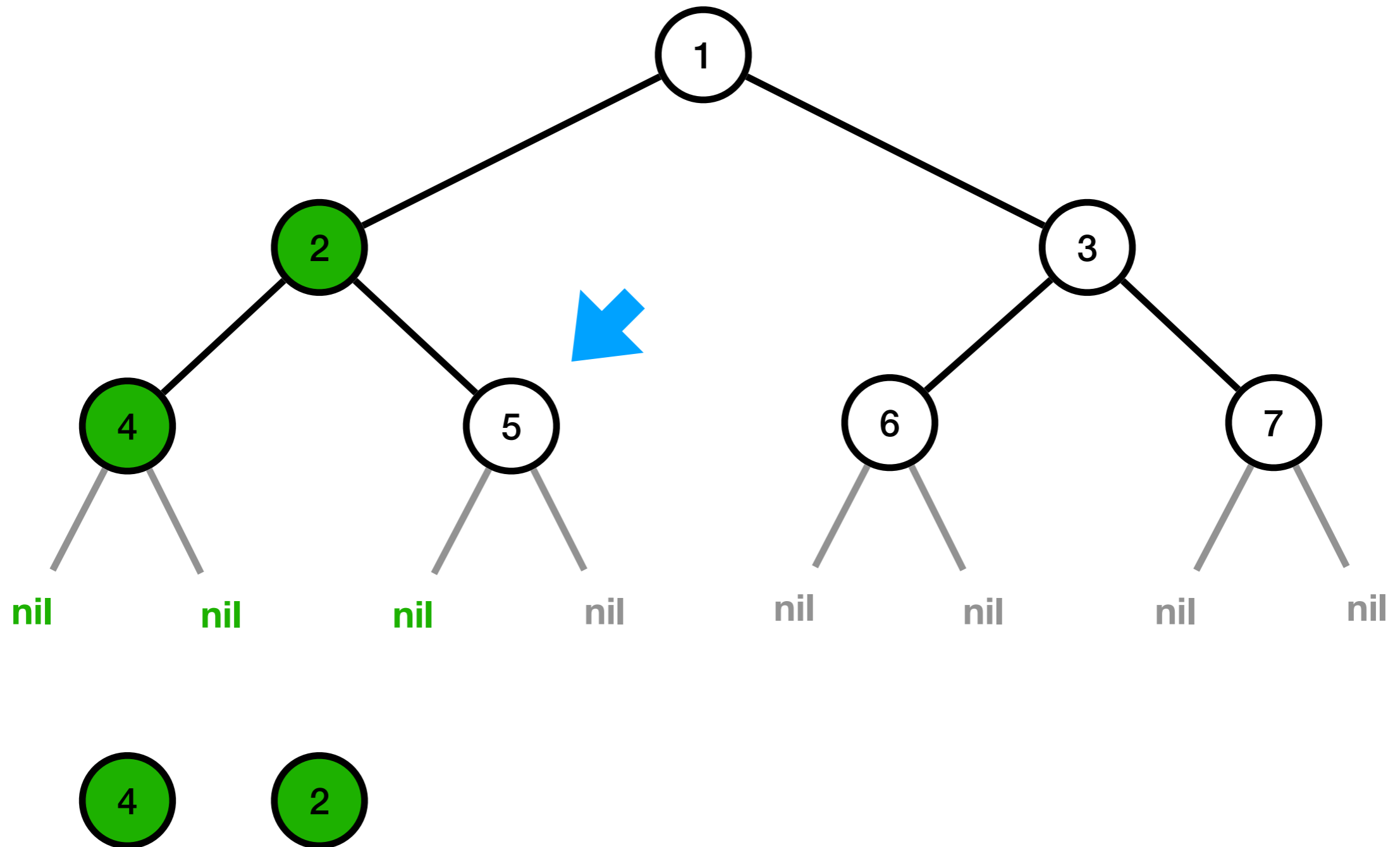
Parcours infixe



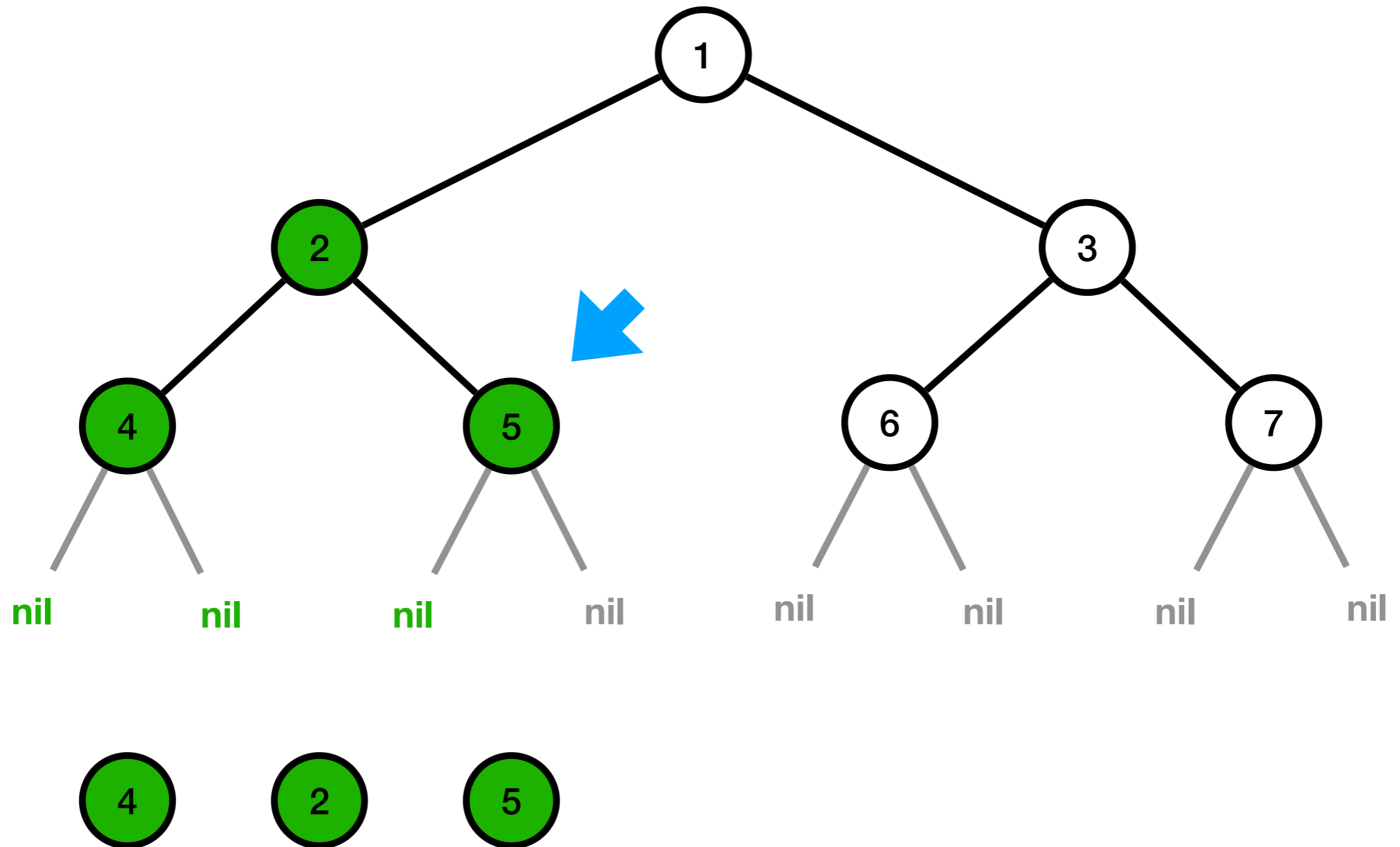
Parcours infixe



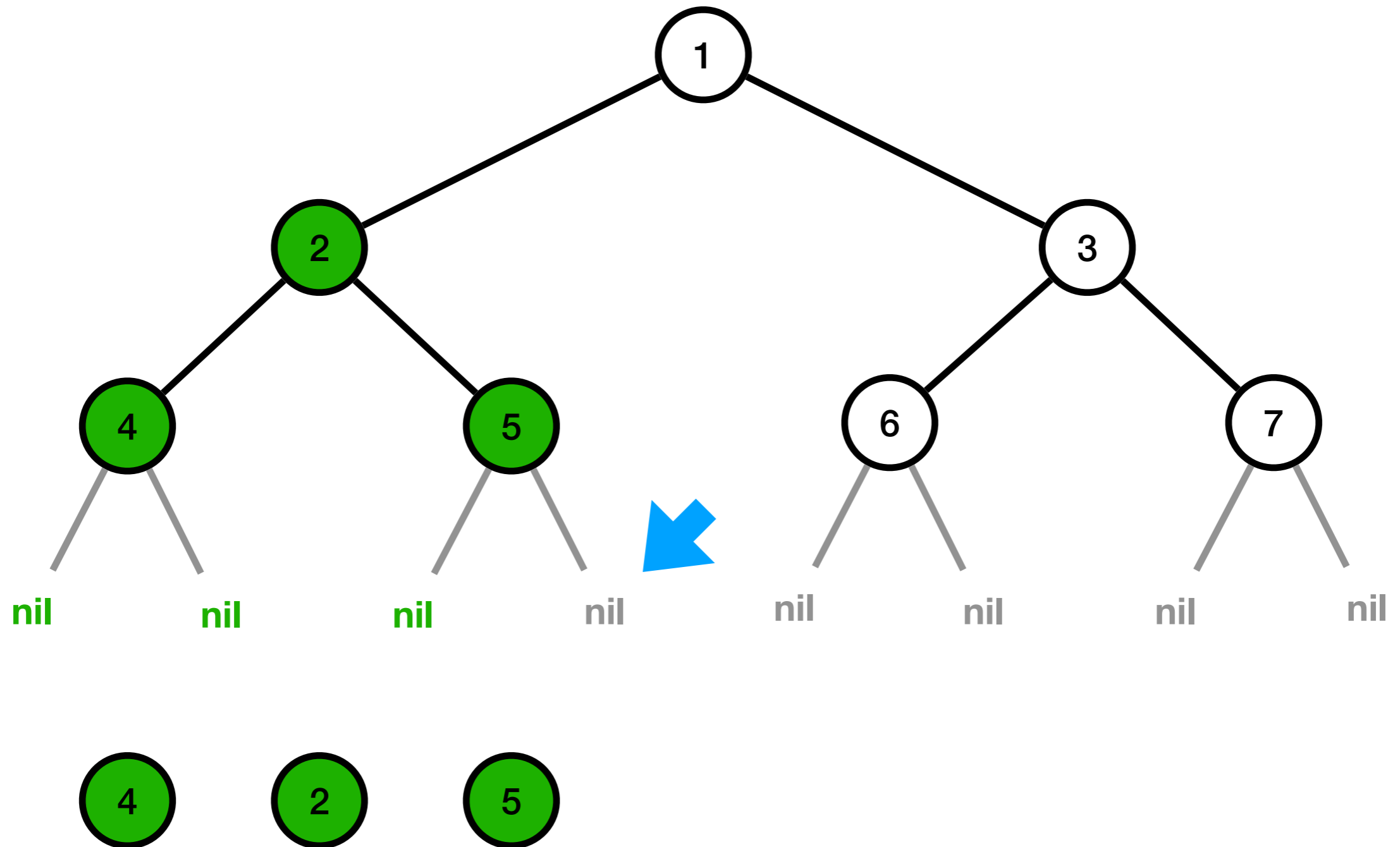
Parcours infixe



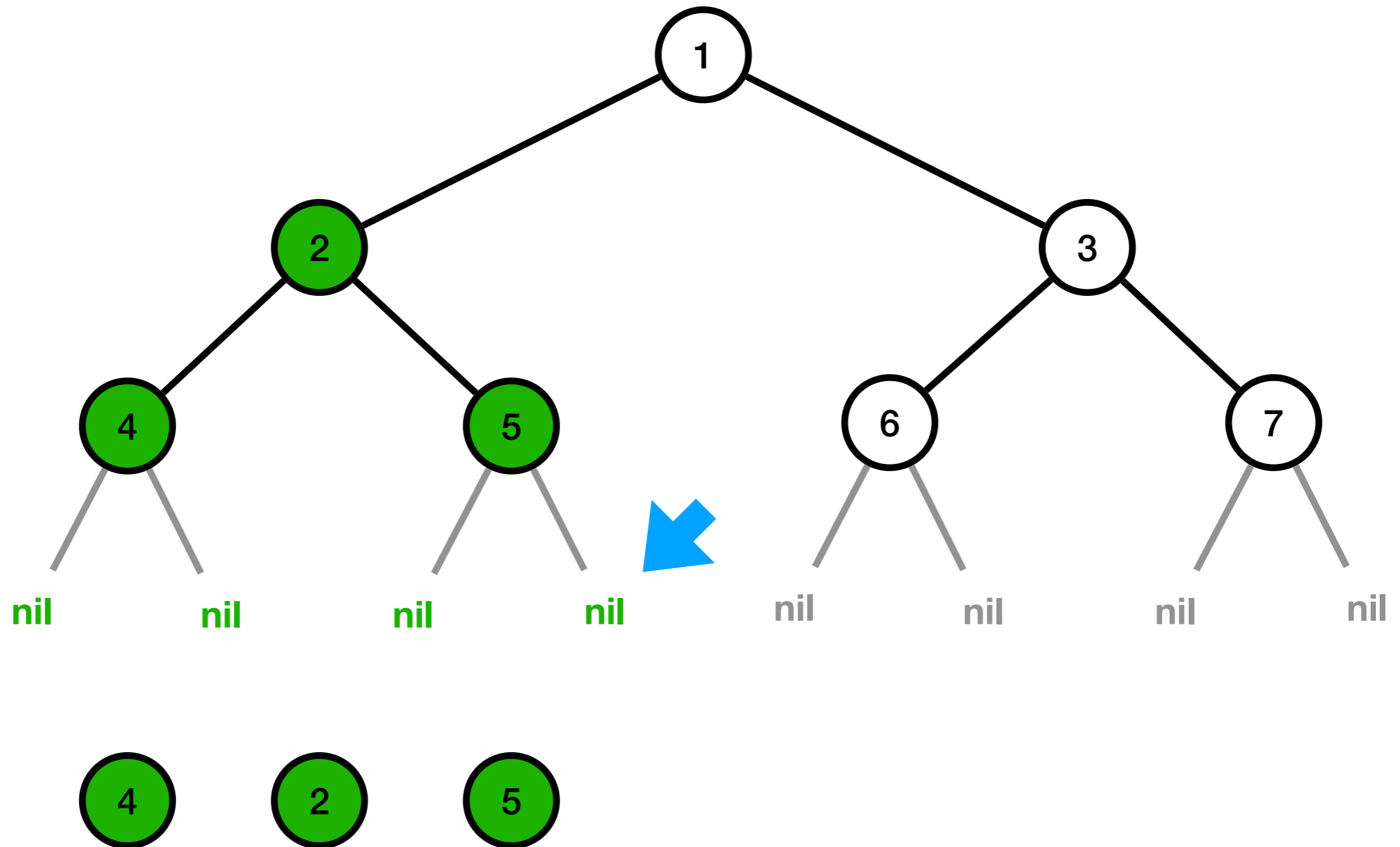
Parcours infixe



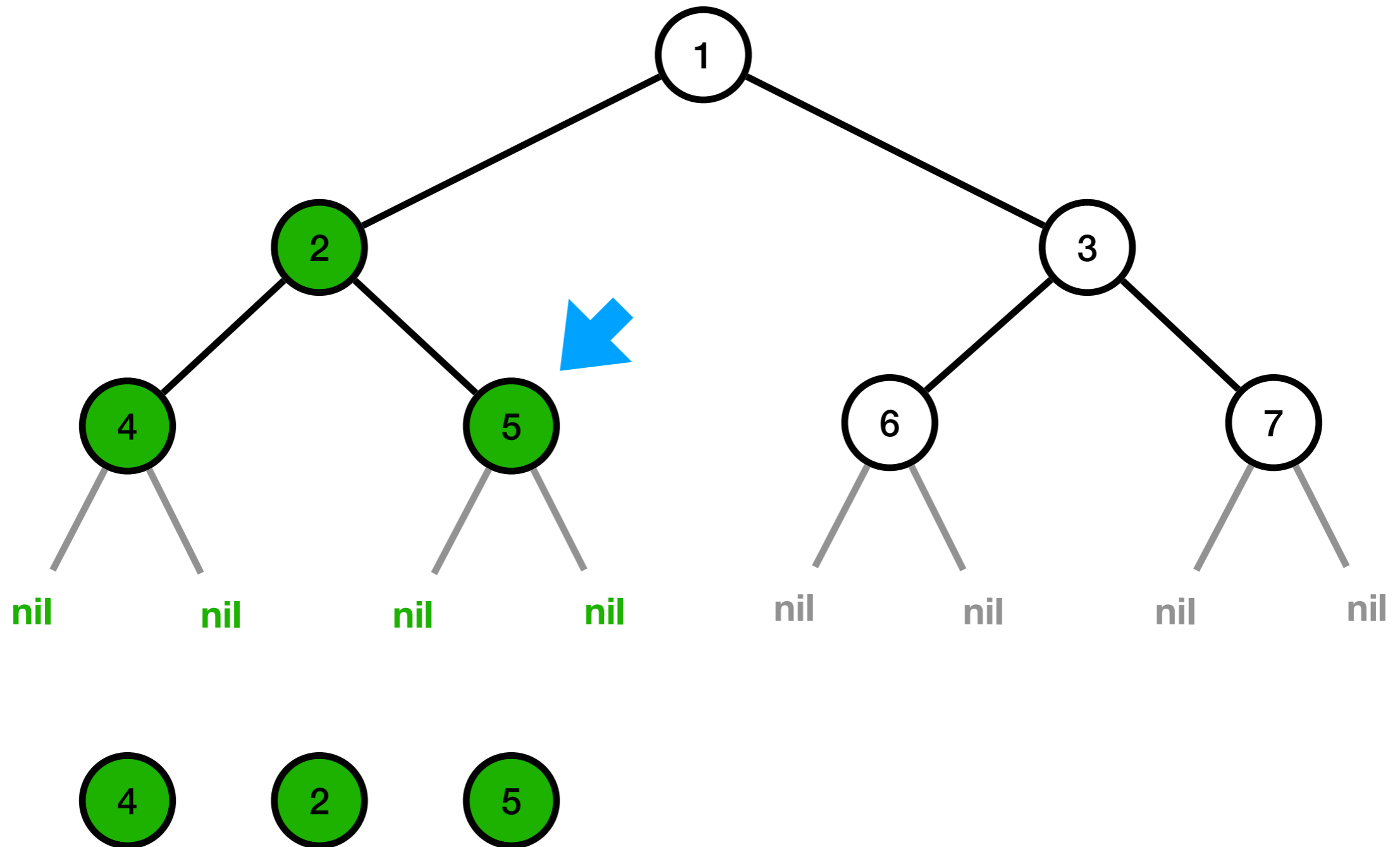
Parcours infixe



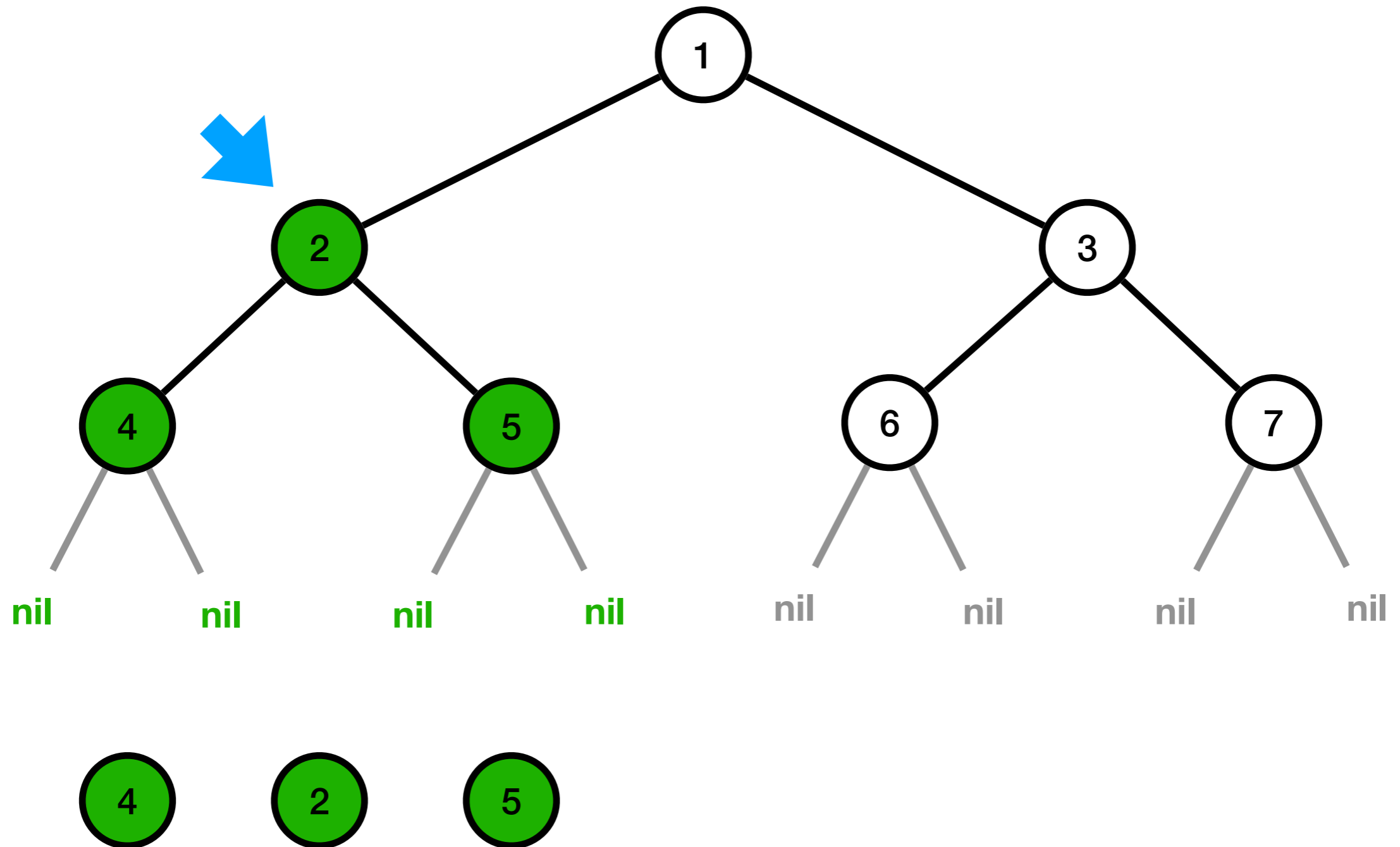
Parcours infixe



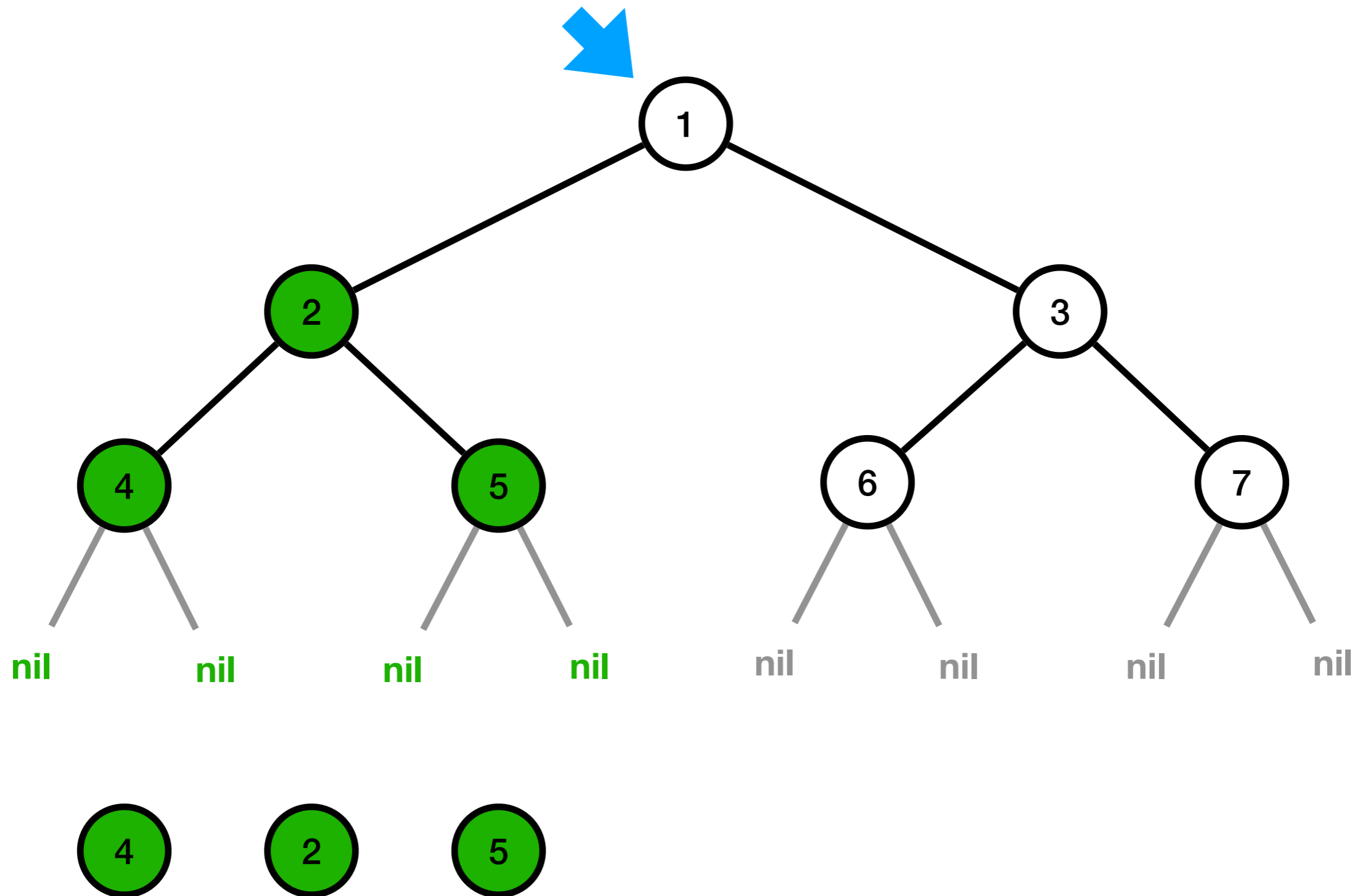
Parcours infixe



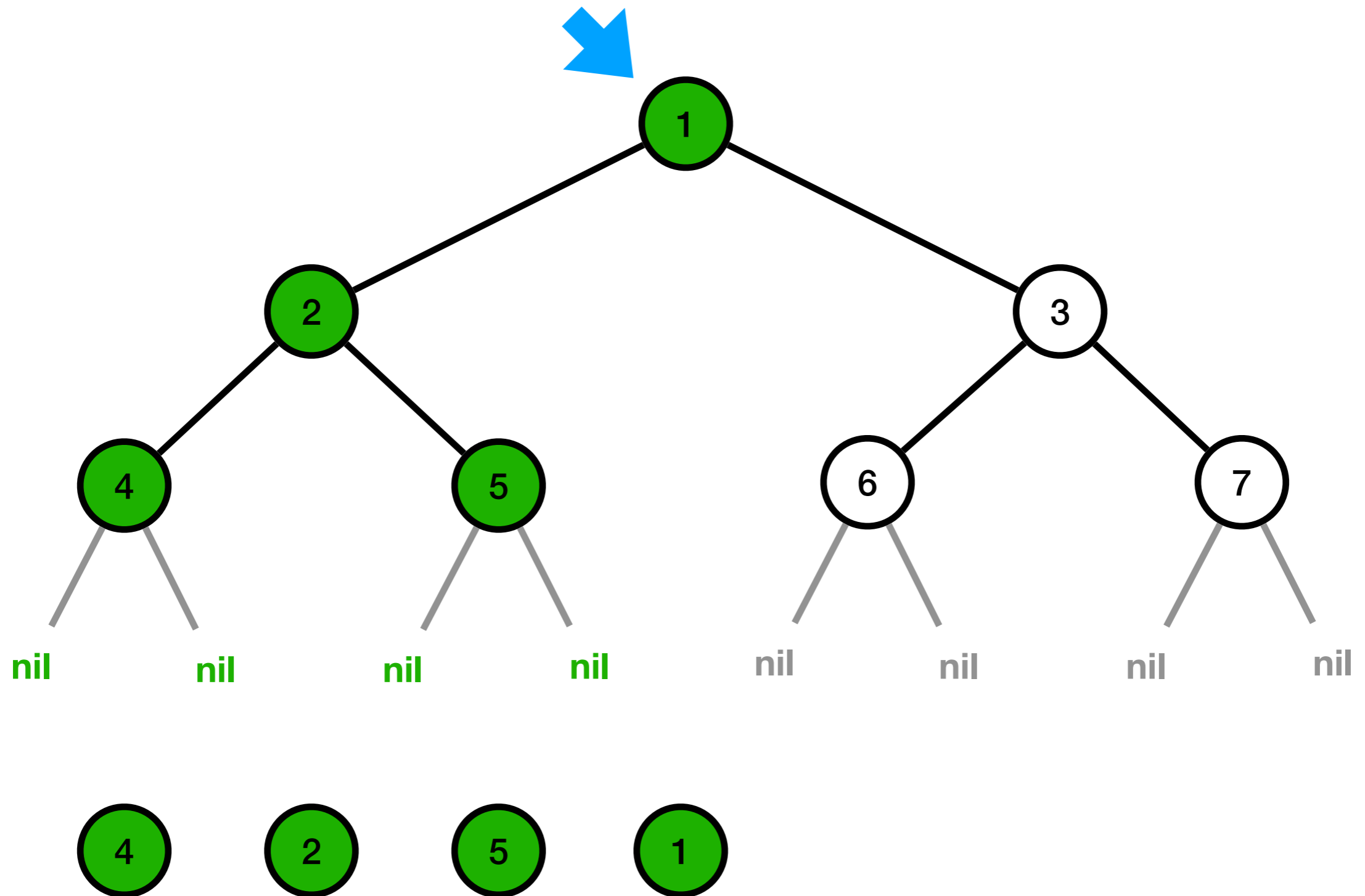
Parcours infixe



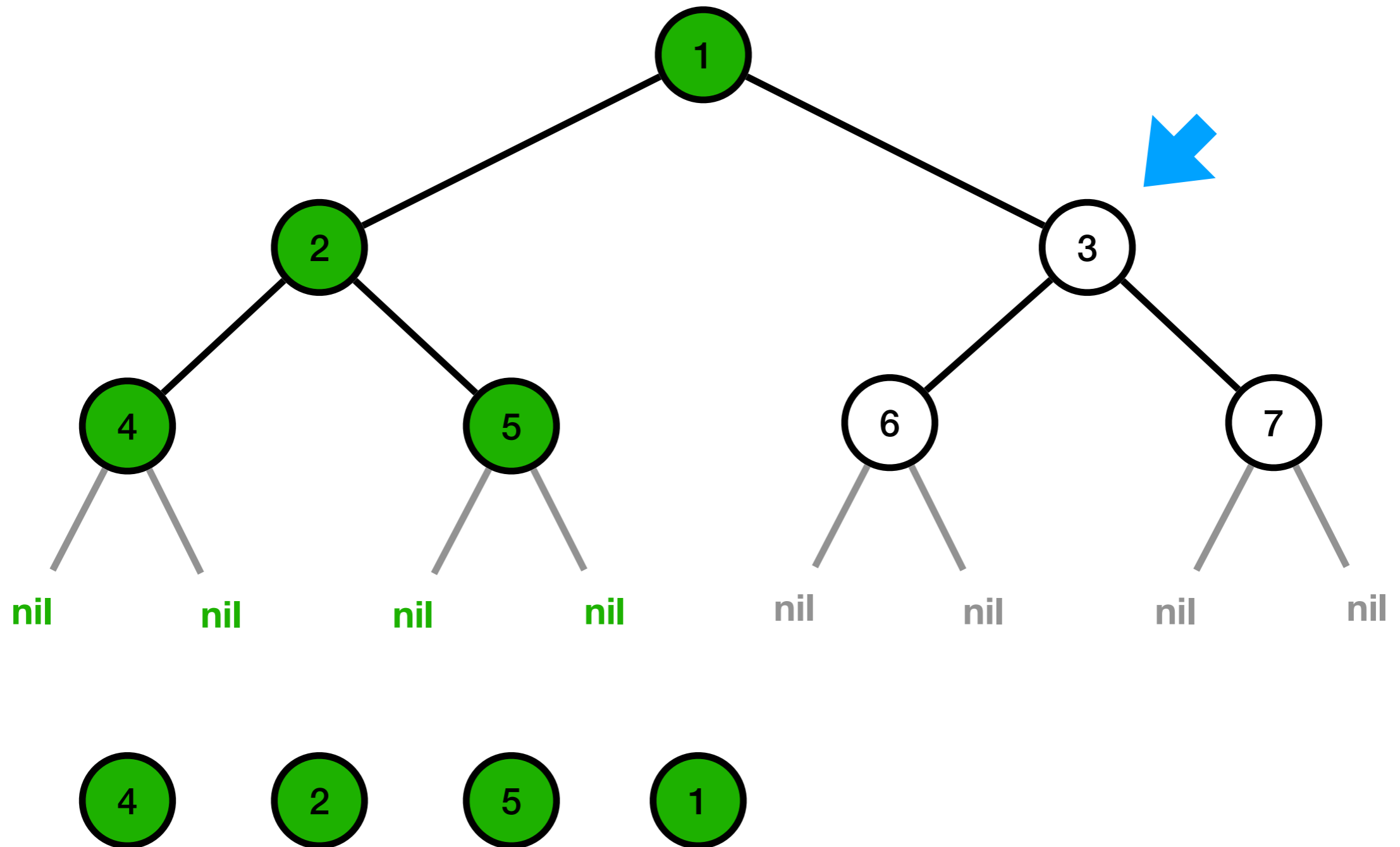
Parcours infixe



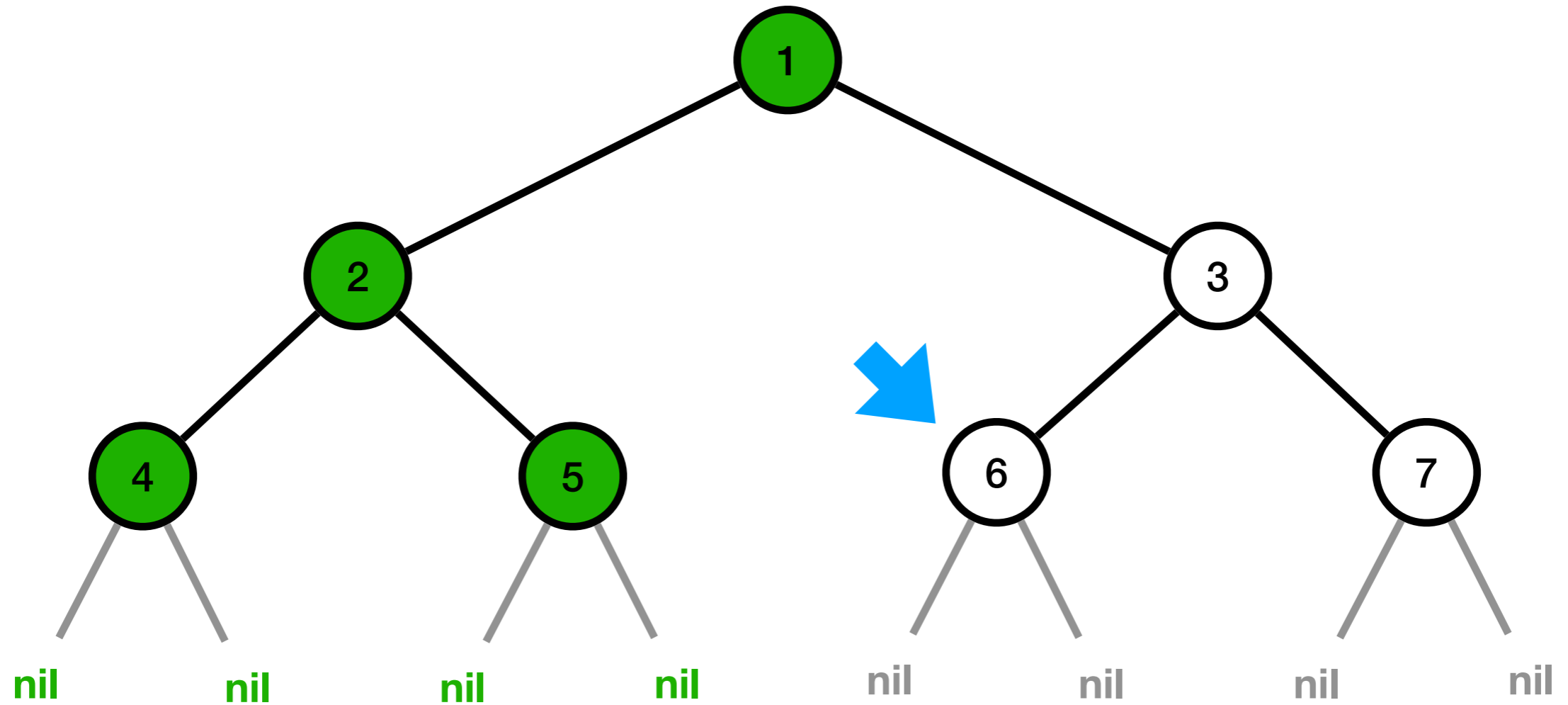
Parcours infixe



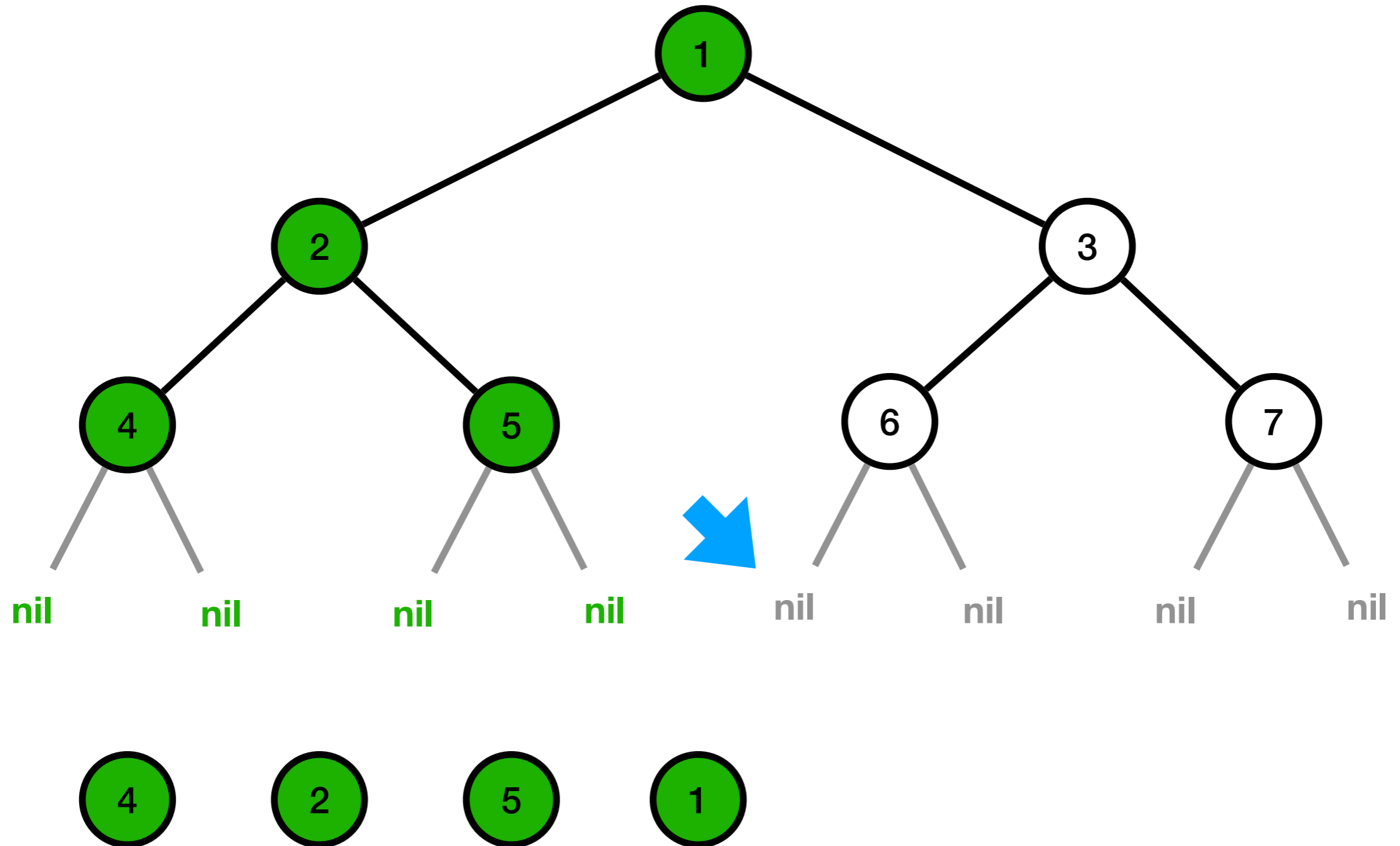
Parcours infixe



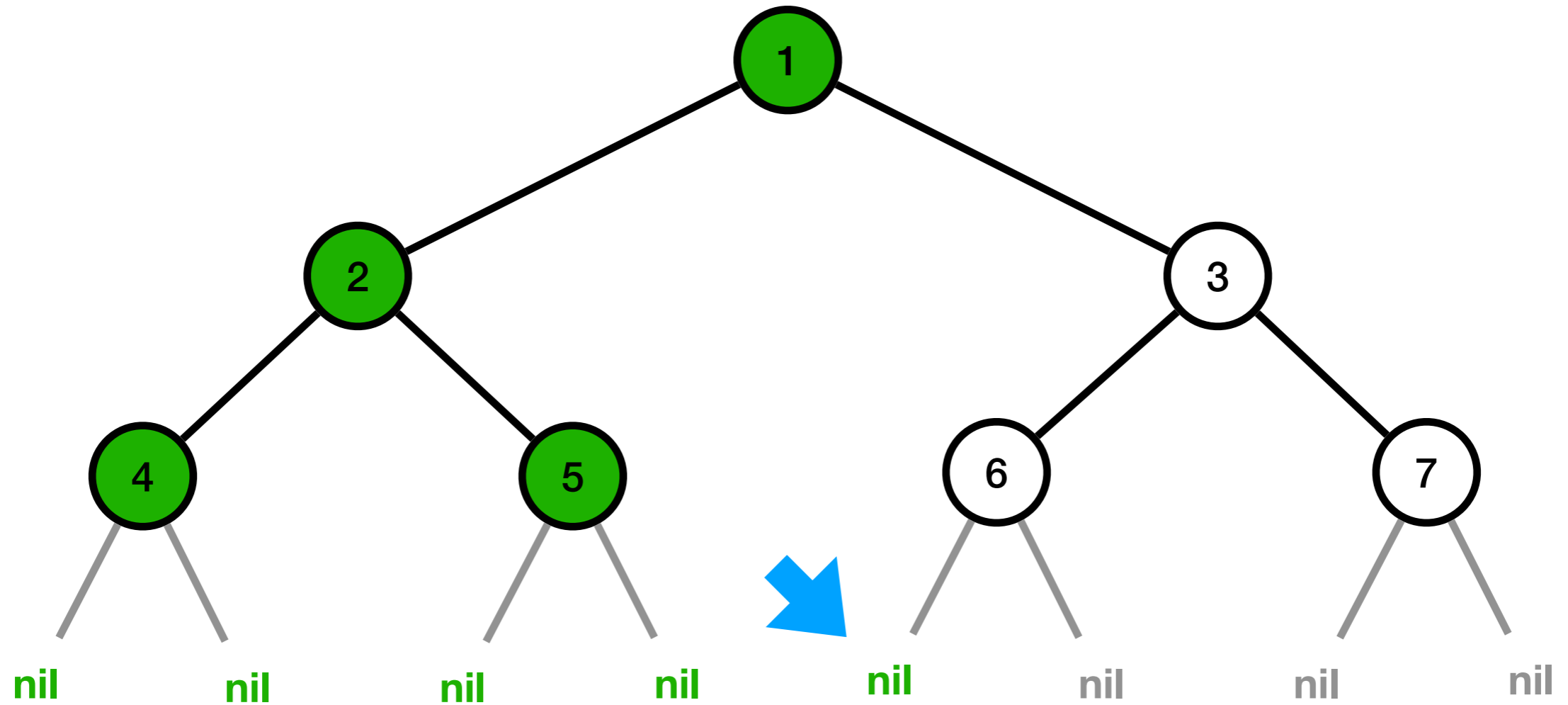
Parcours infixe



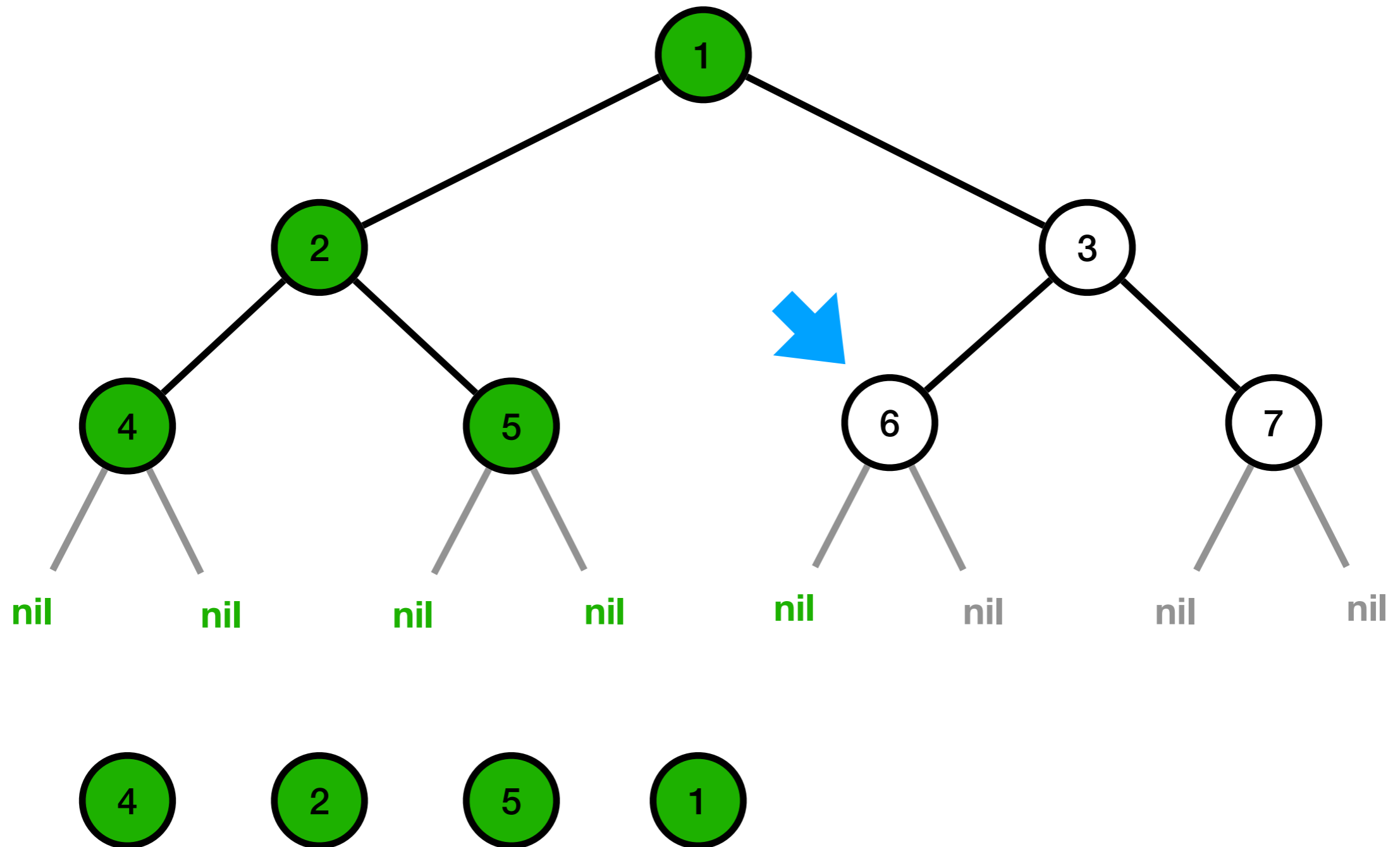
Parcours infixe



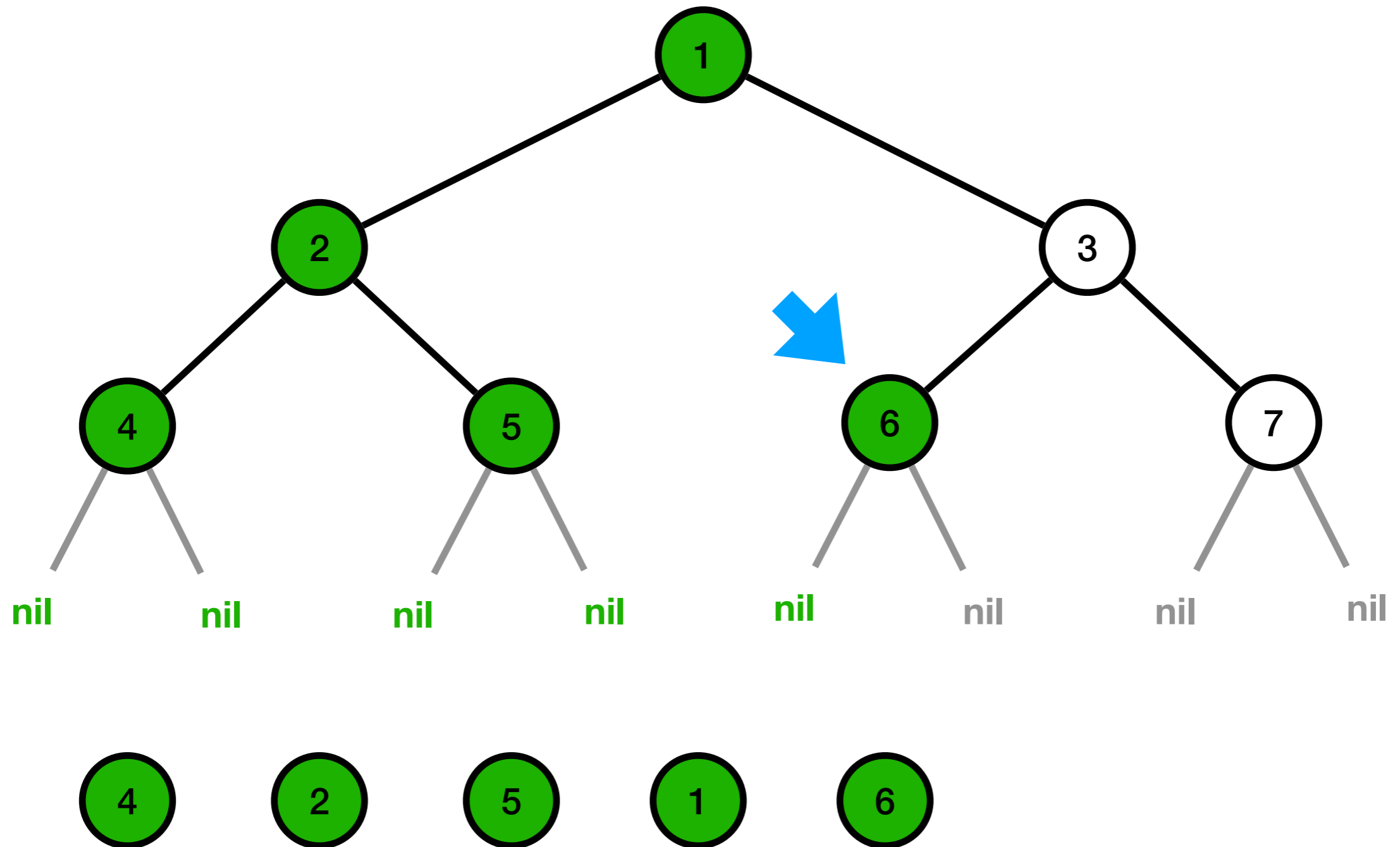
Parcours infixe



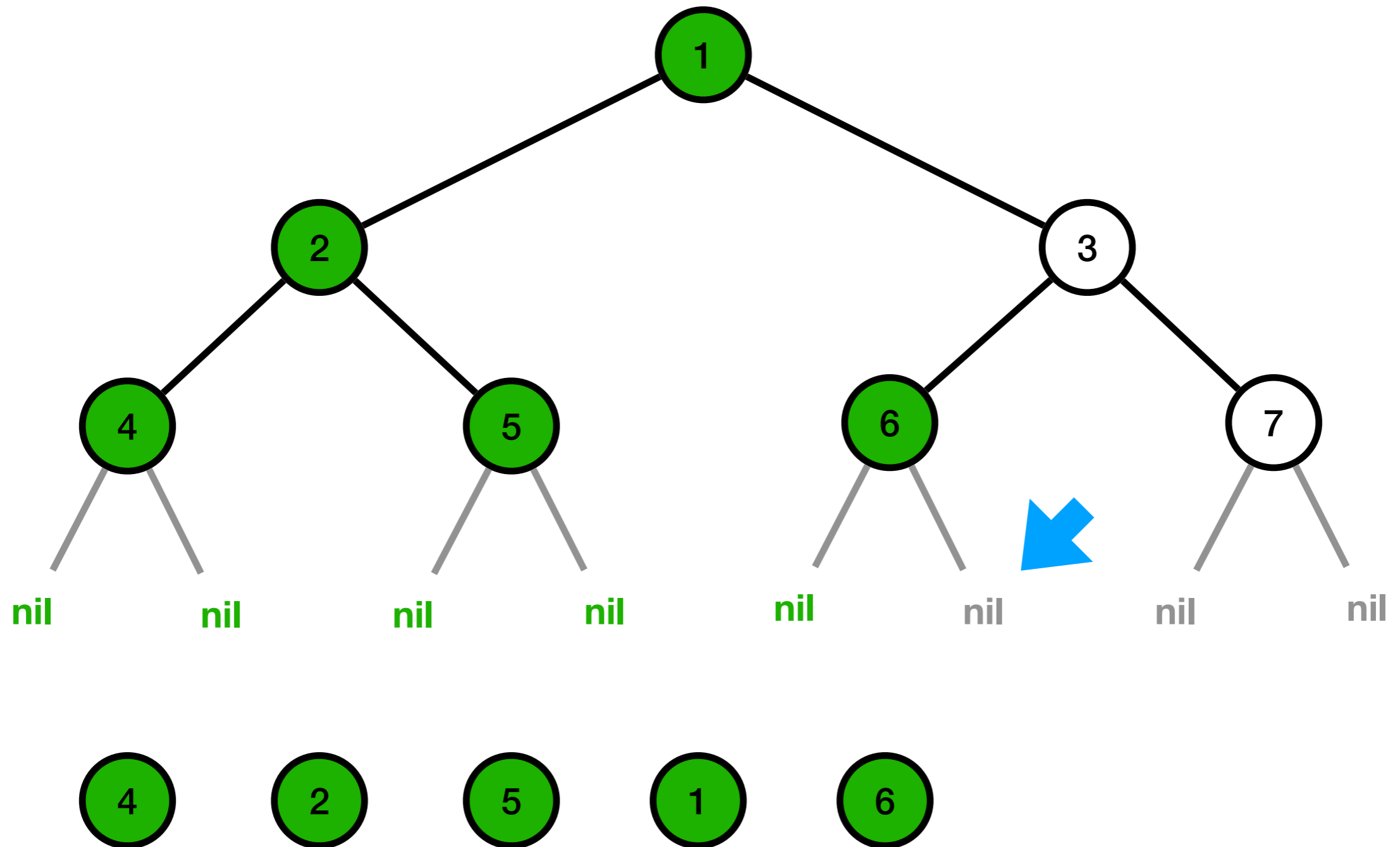
Parcours infixe



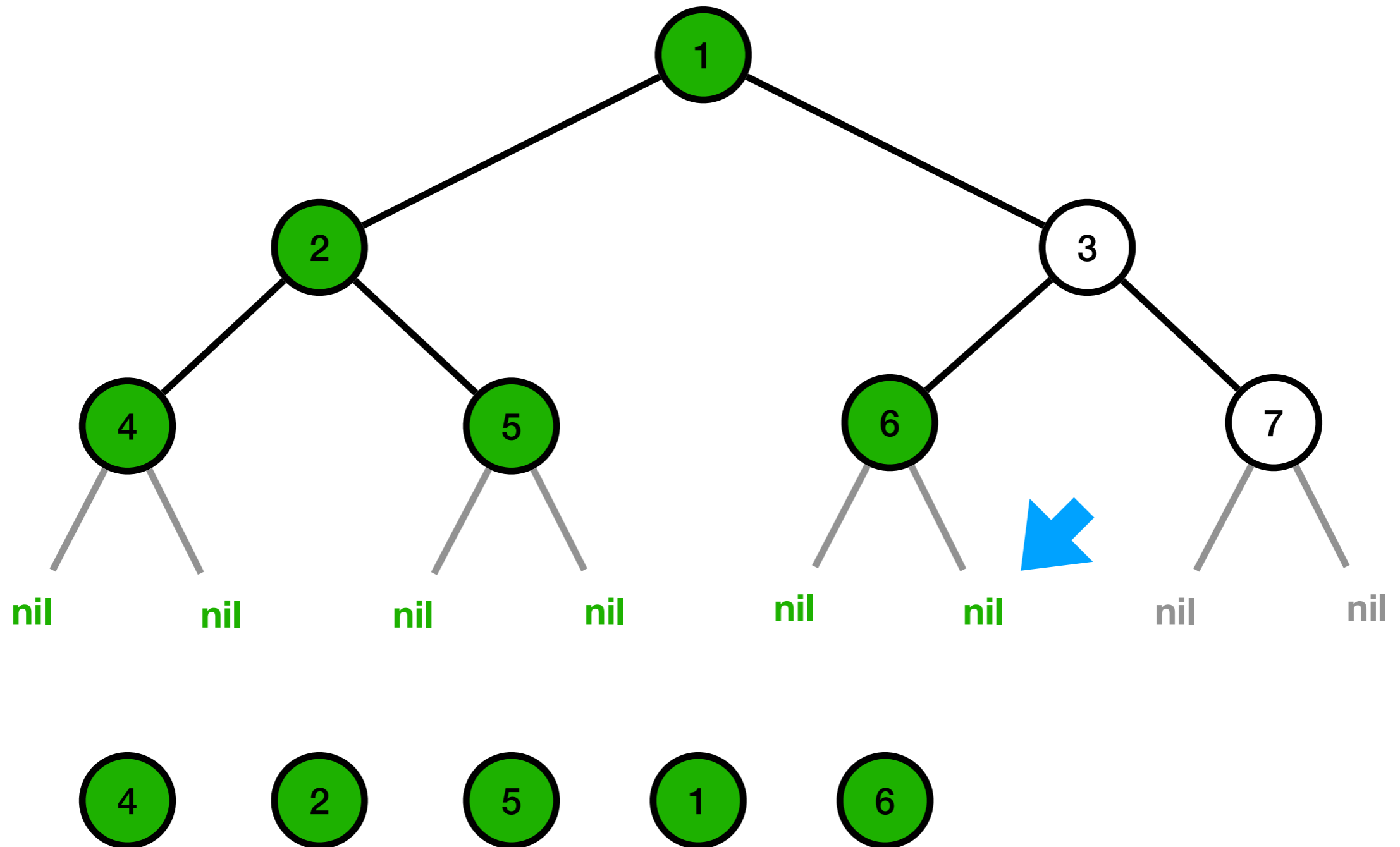
Parcours infixe



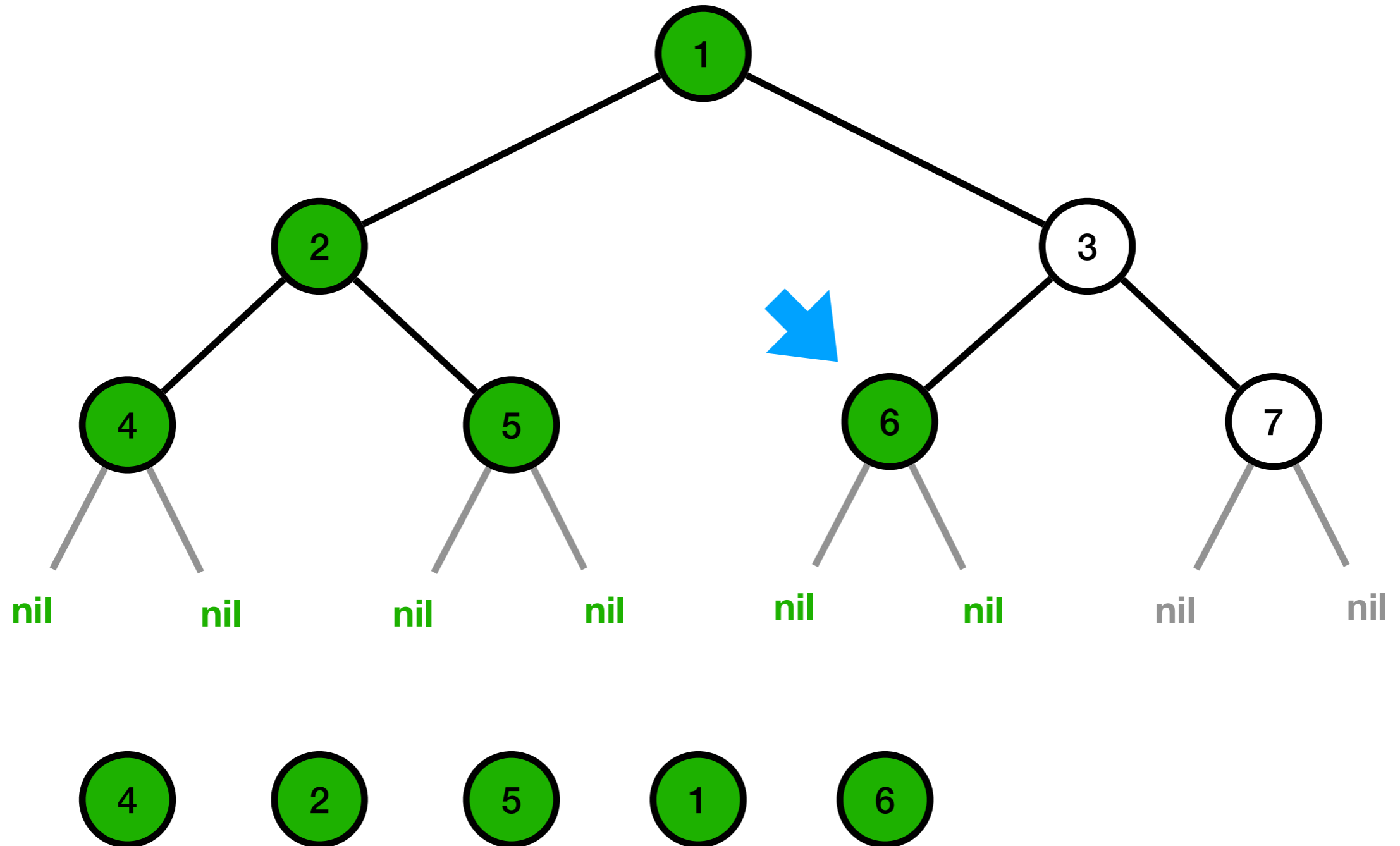
Parcours infixe



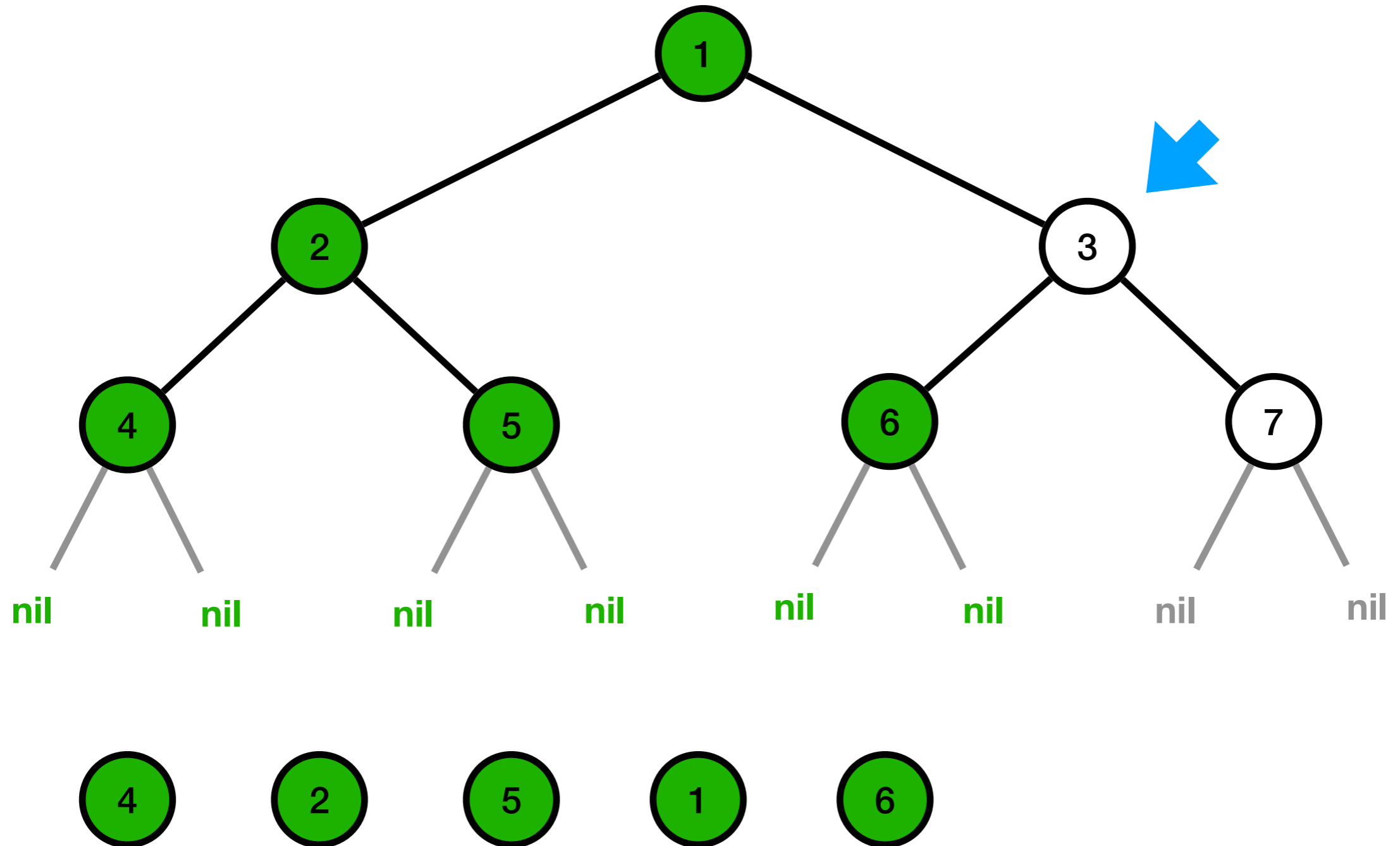
Parcours infixe



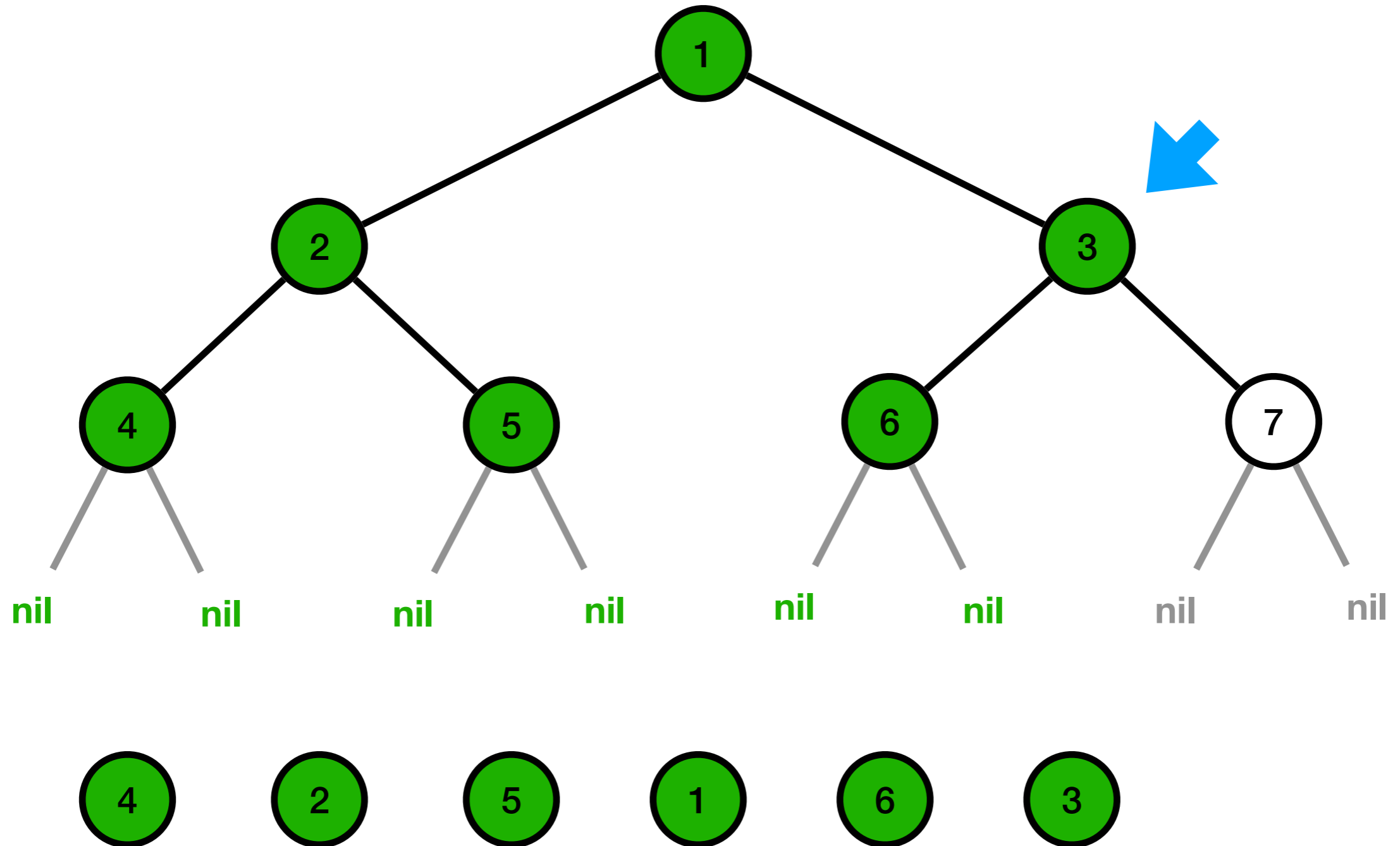
Parcours infixe



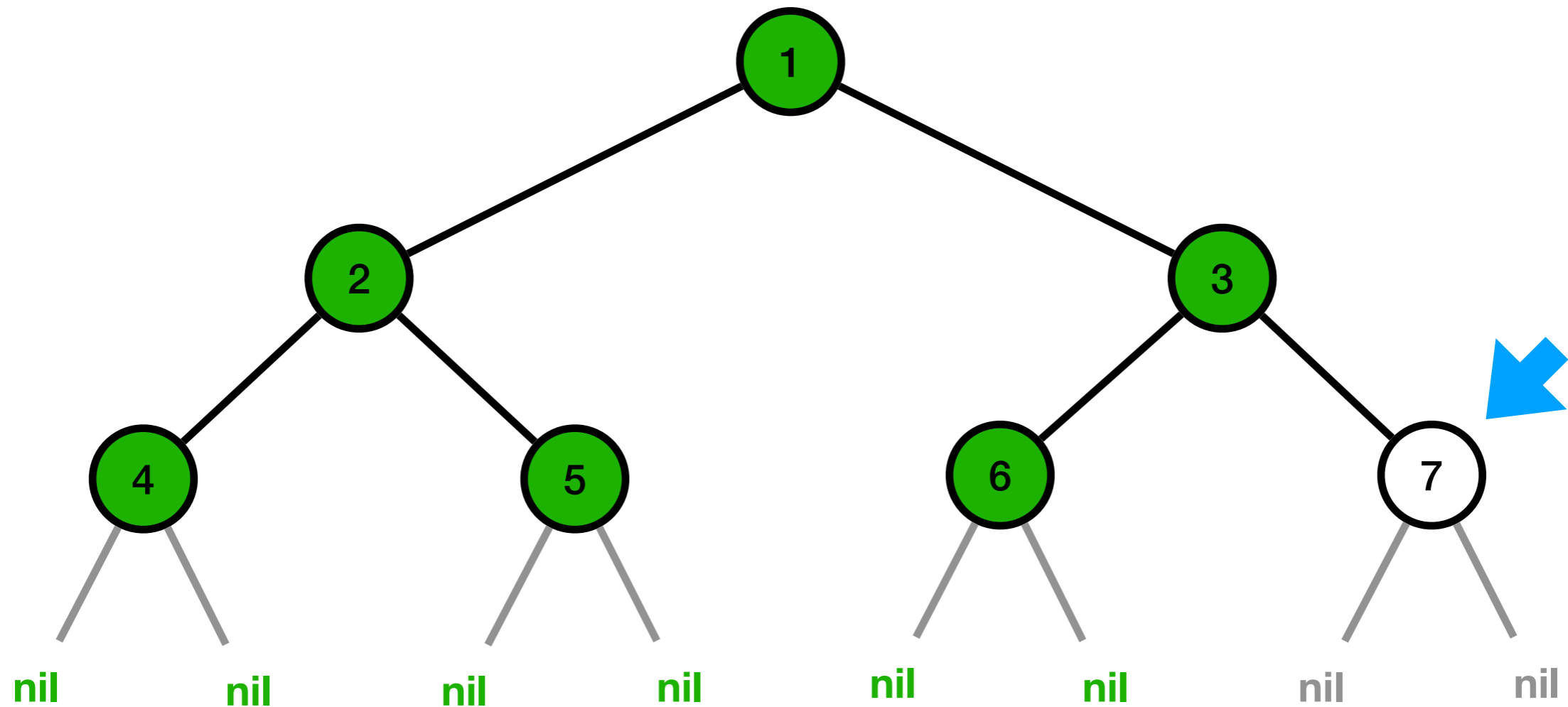
Parcours infixe



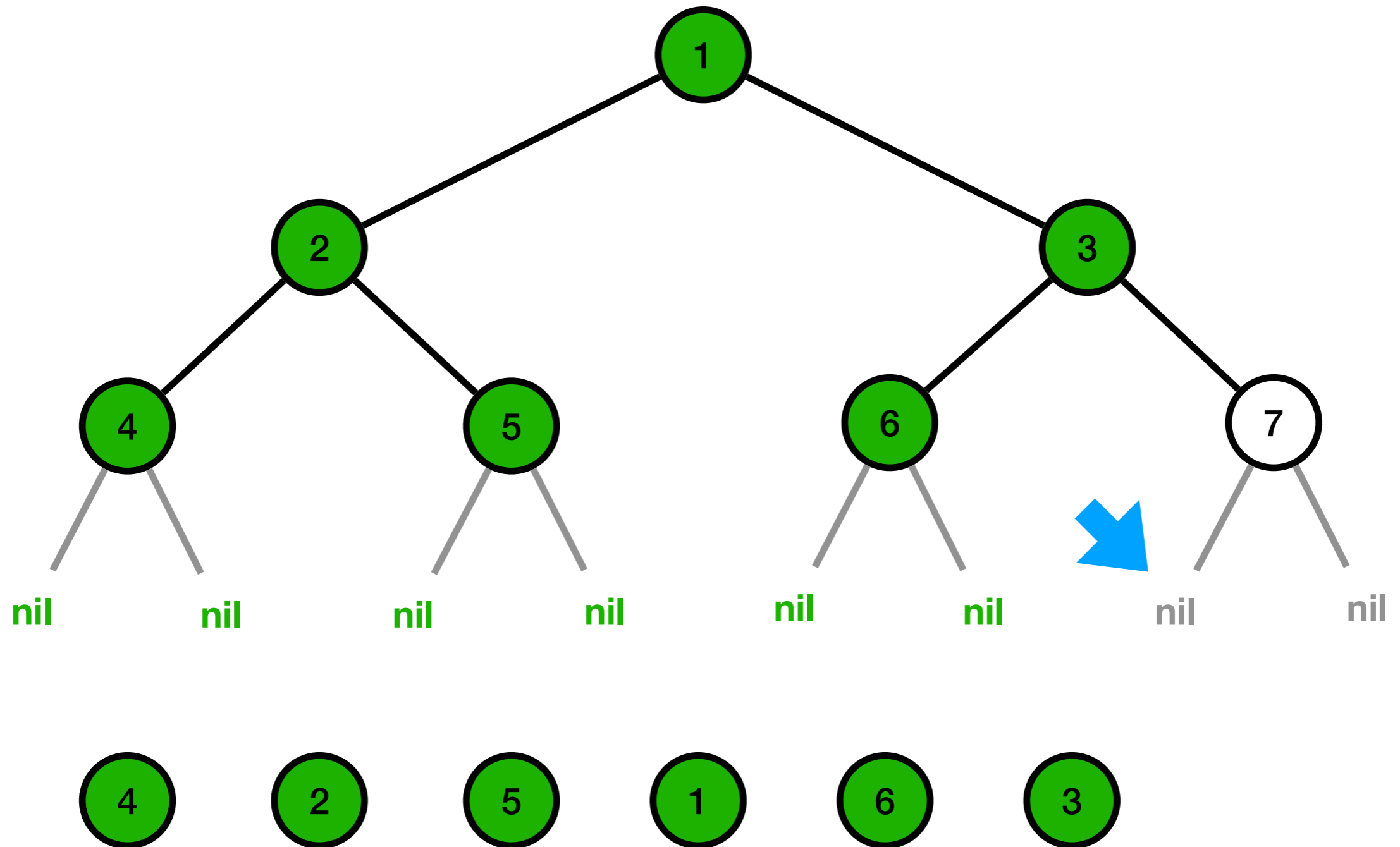
Parcours infixe



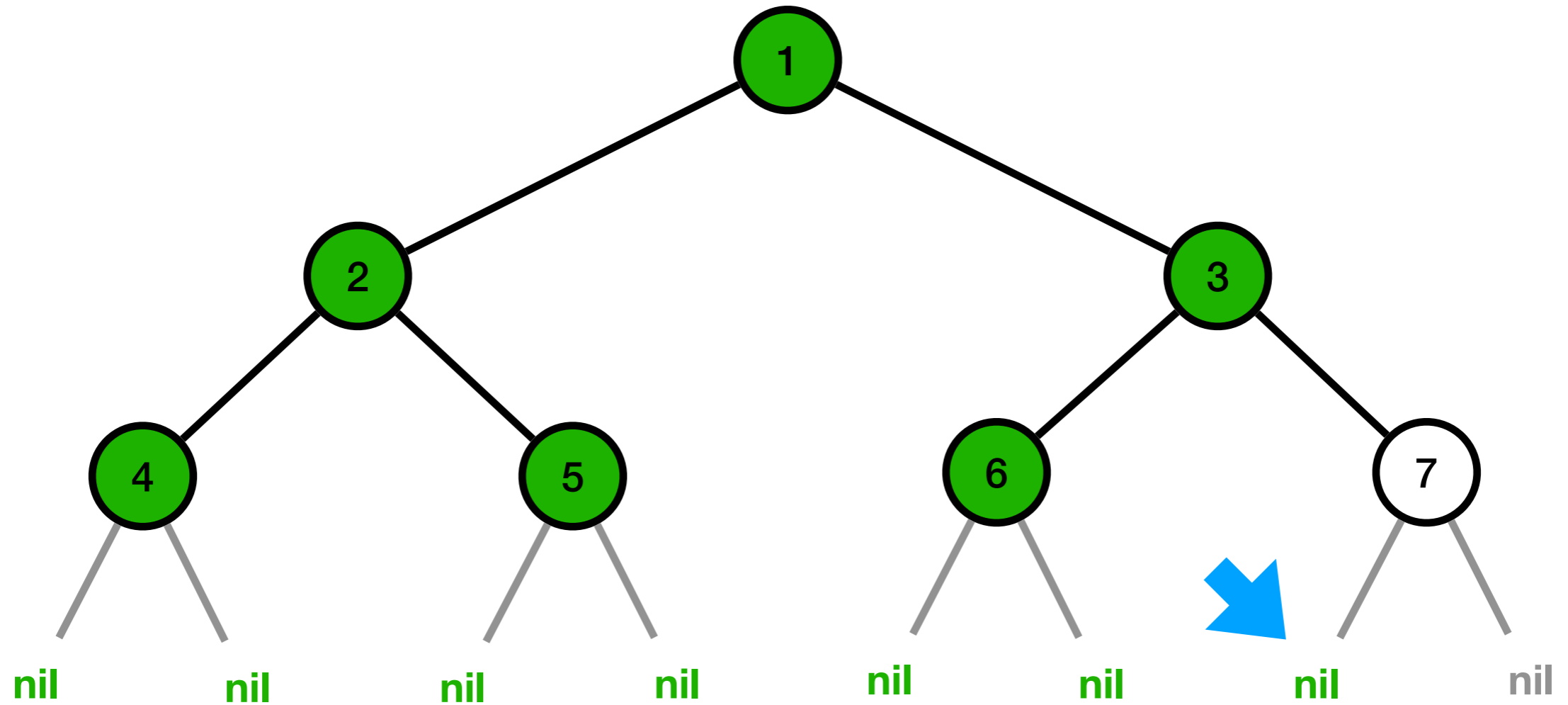
Parcours infixe



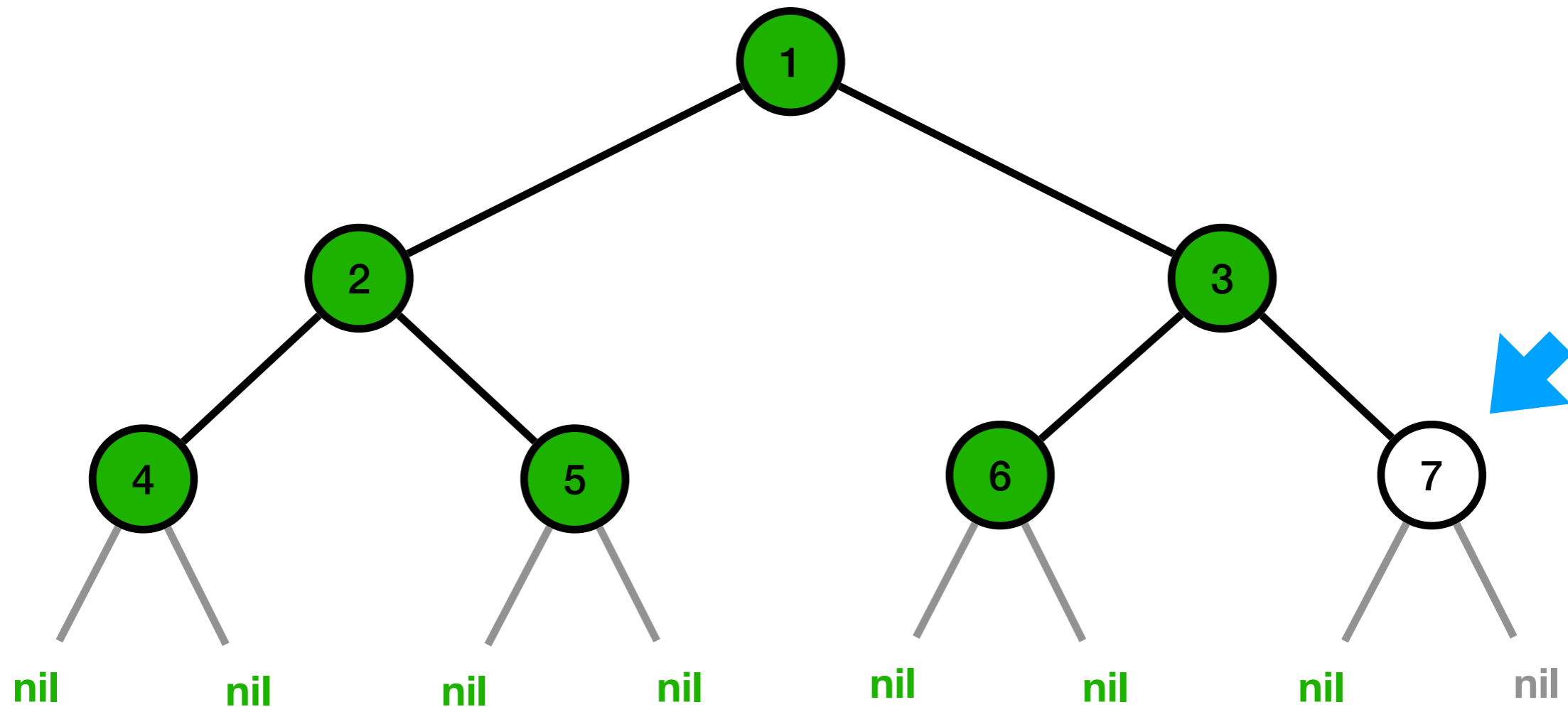
Parcours infixe



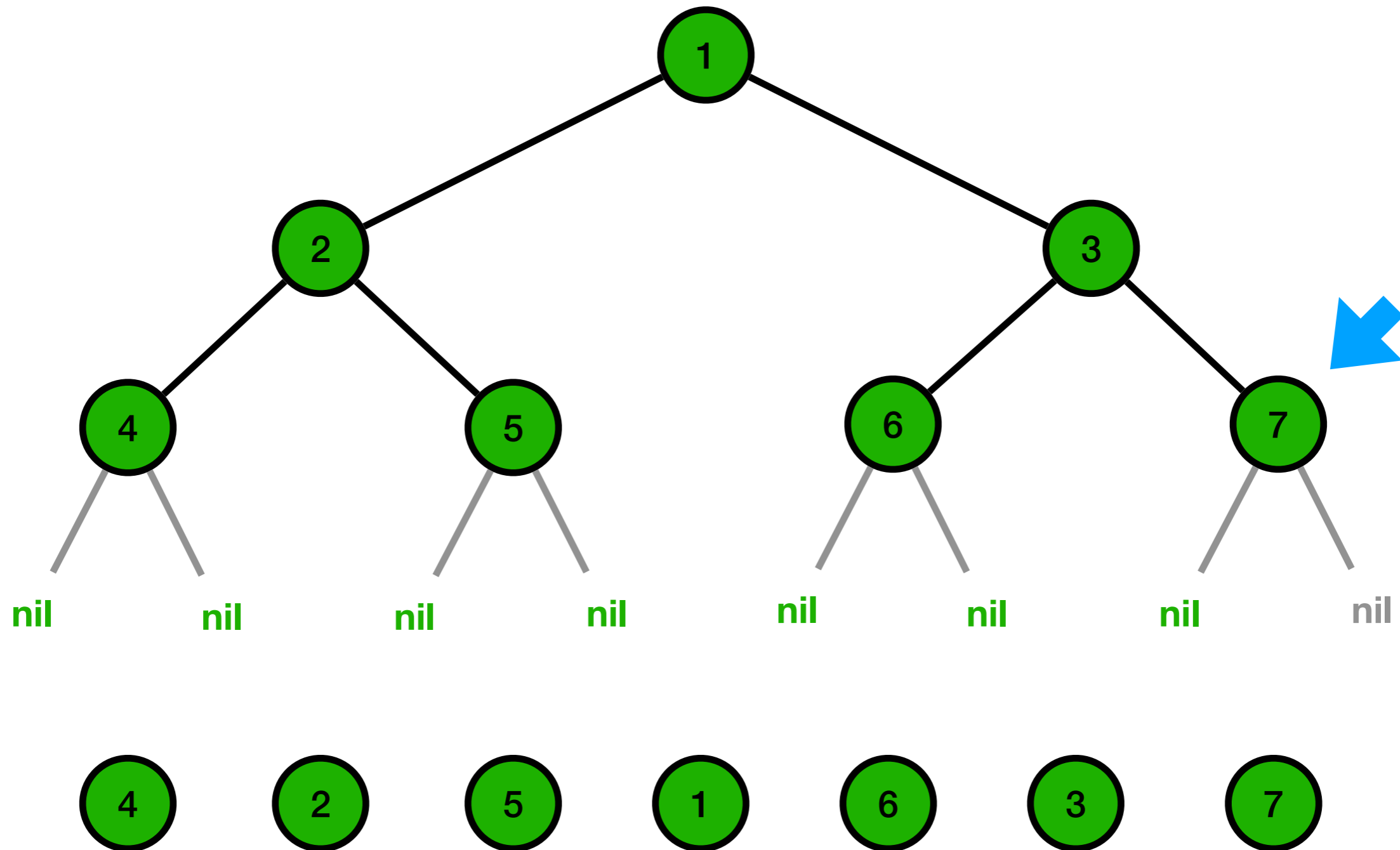
Parcours infixe



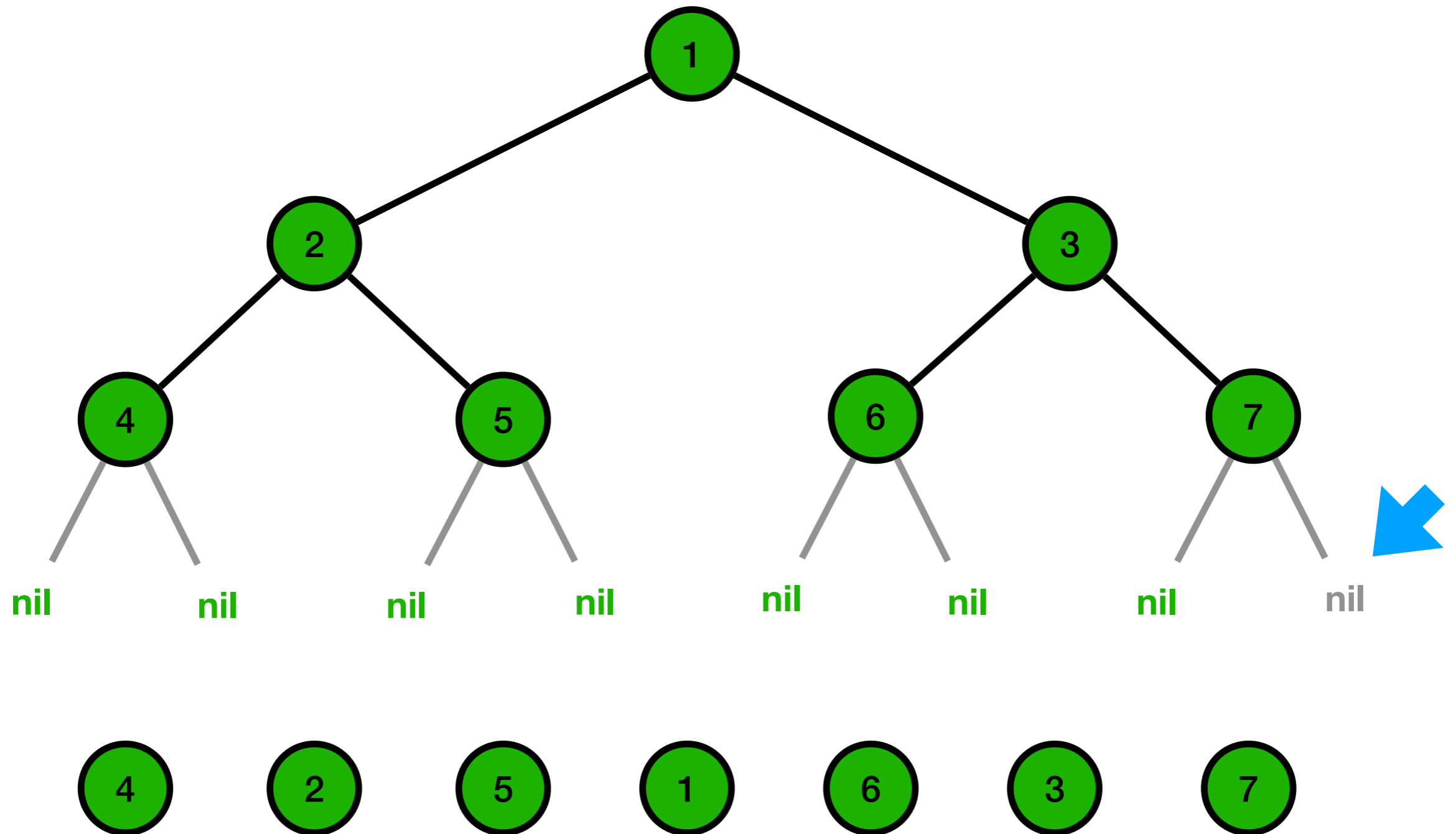
Parcours infixe



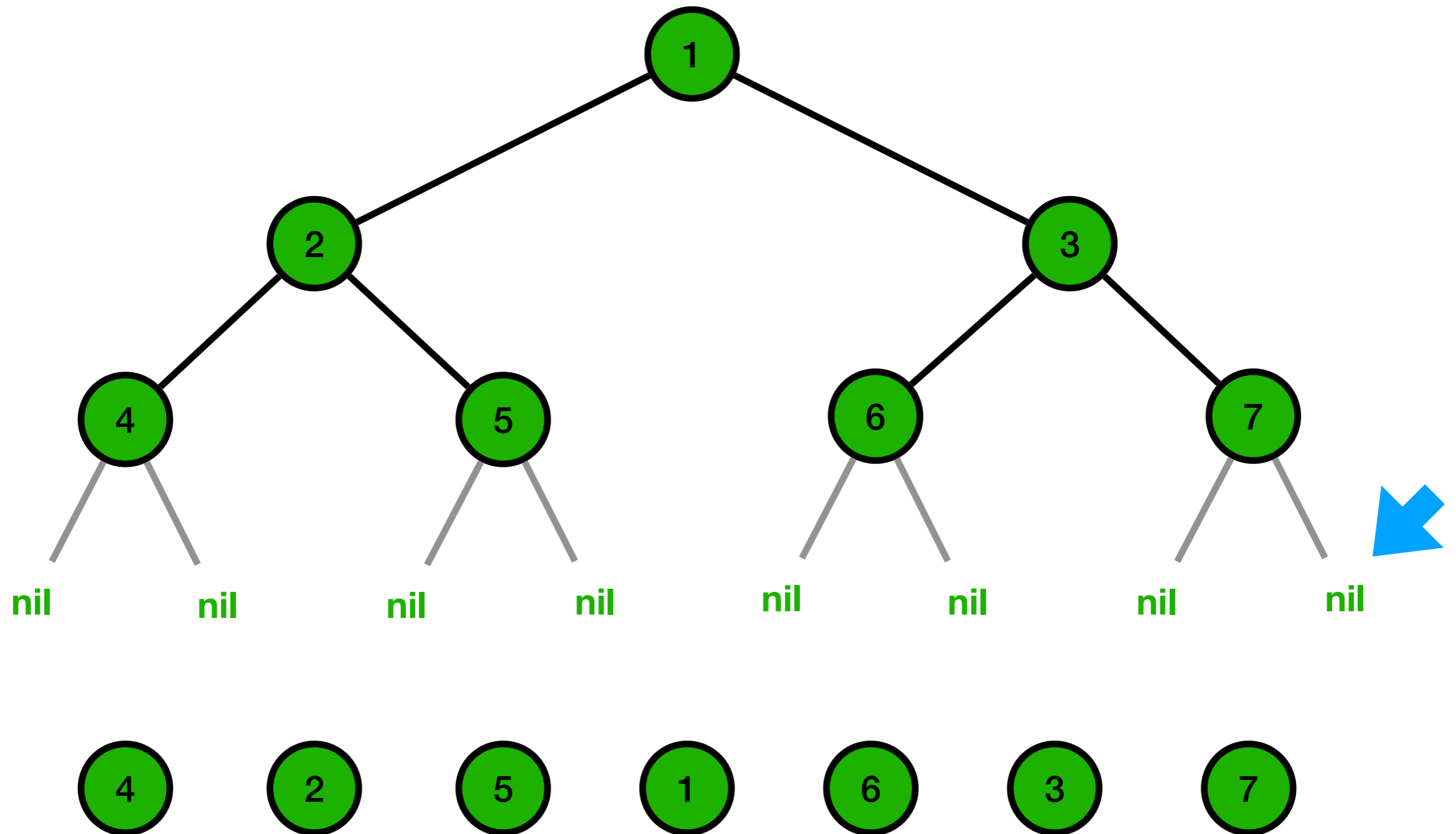
Parcours infixe



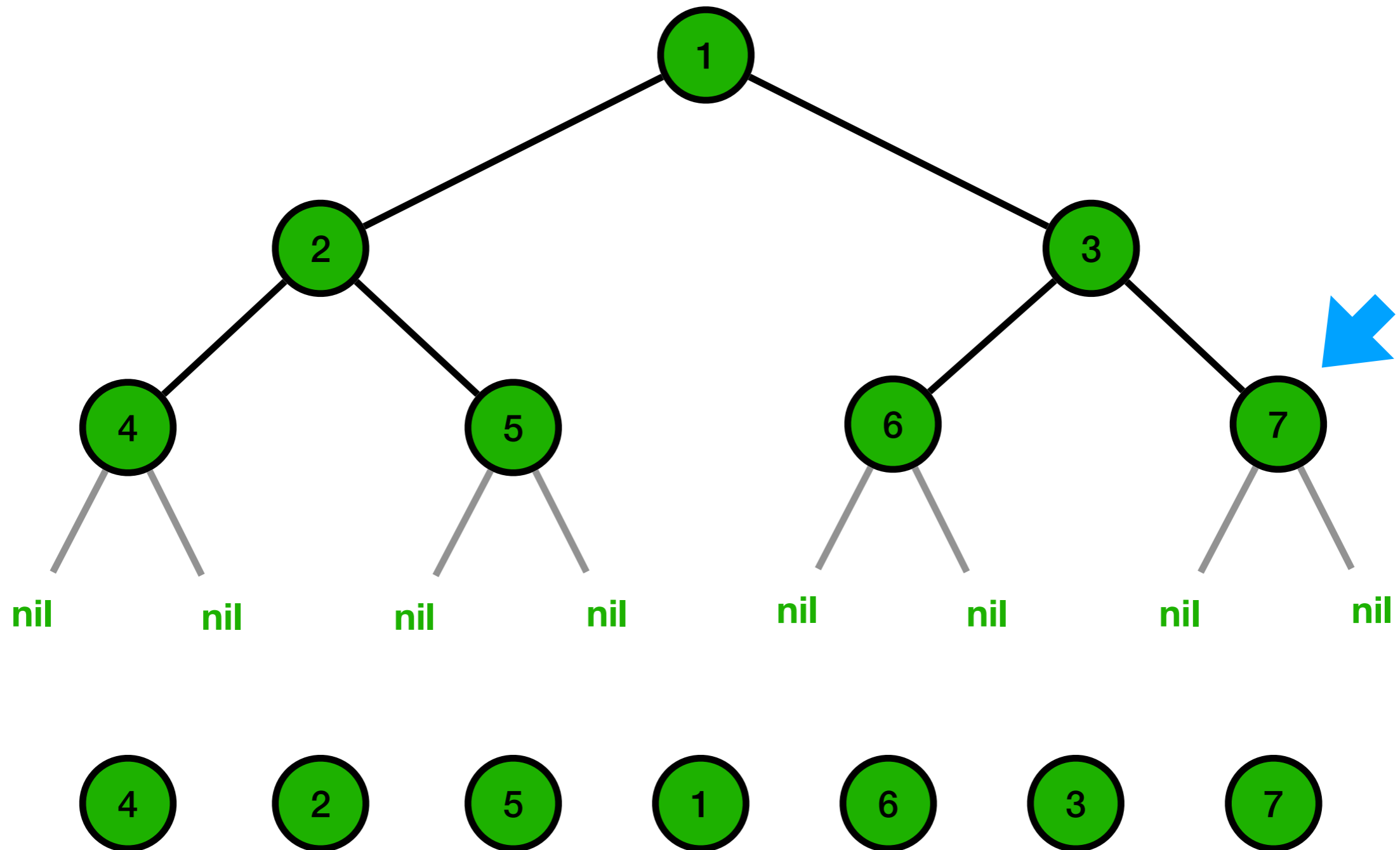
Parcours infixe



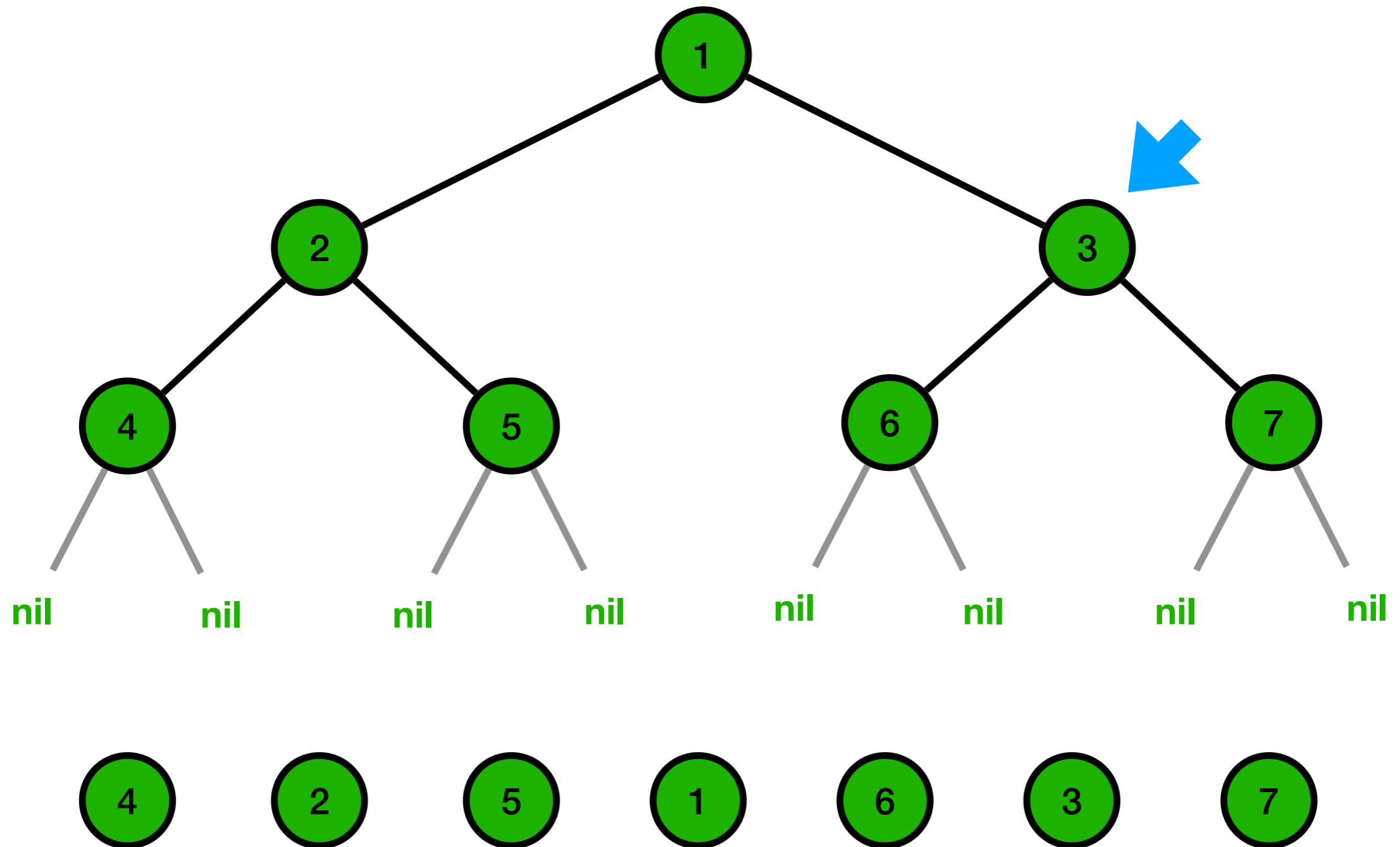
Parcours infixe



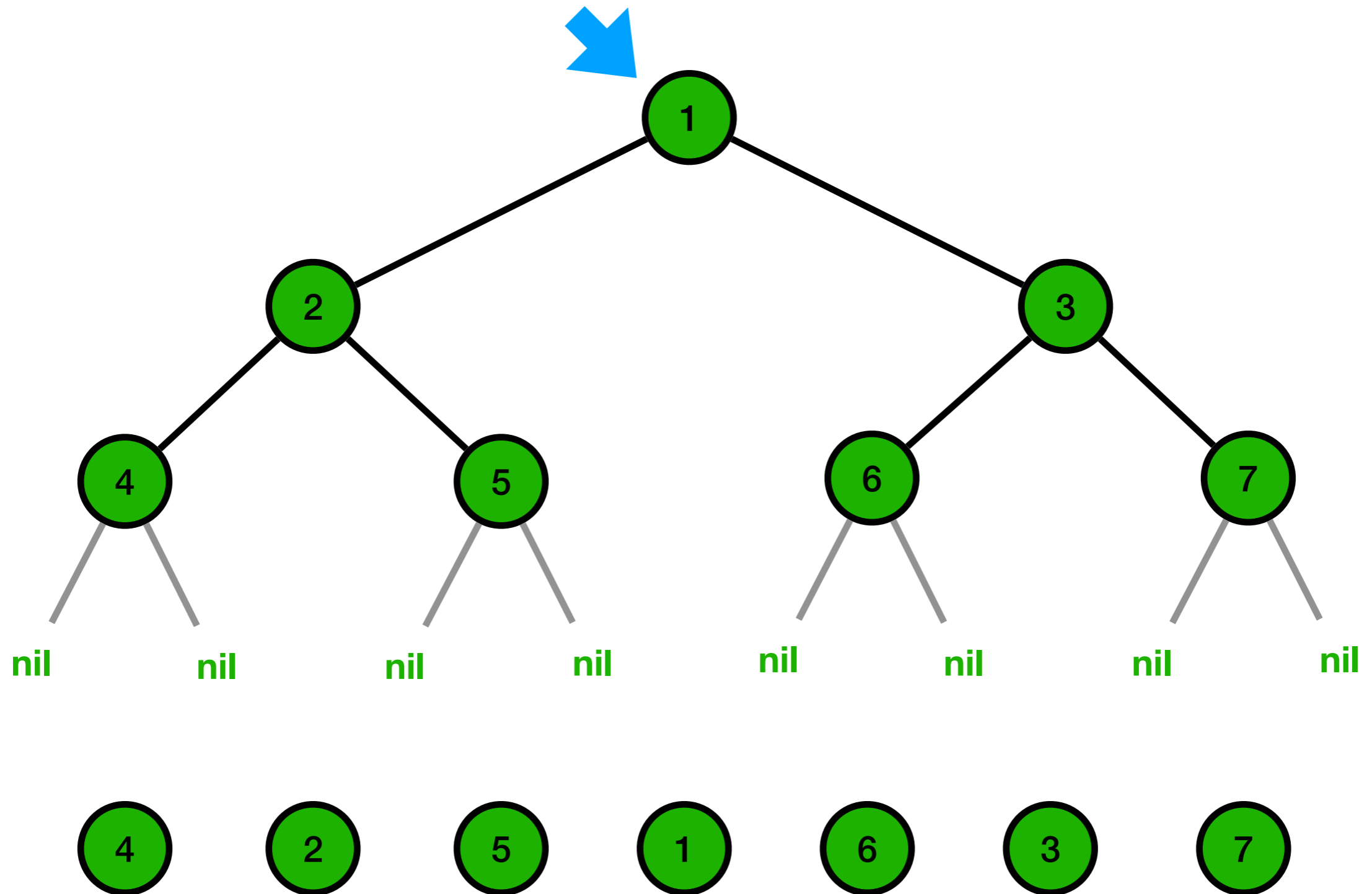
Parcours infixe



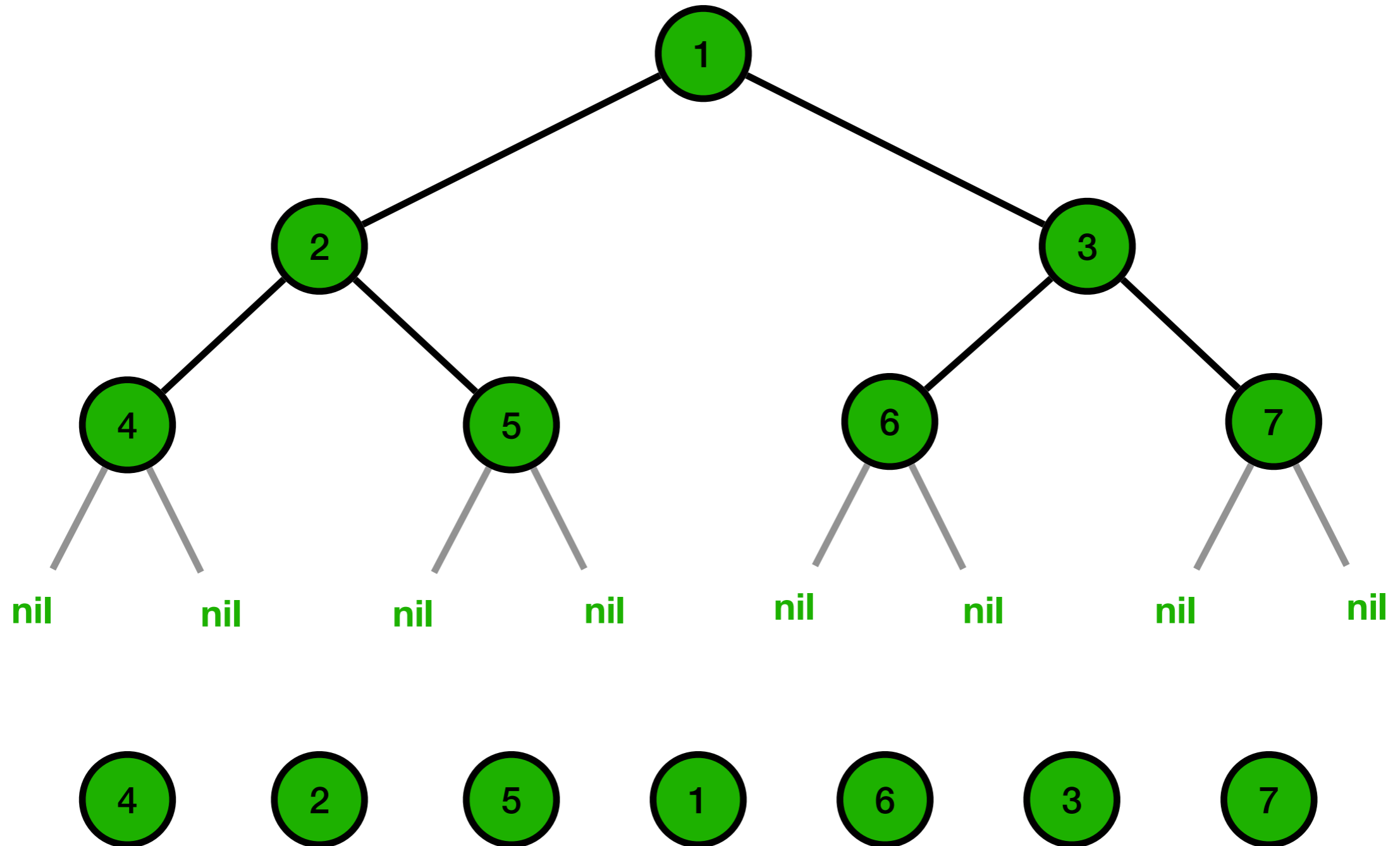
Parcours infixe



Parcours infixe



Parcours infixe



Exercice 2 du TD5