

Introduction à l'informatique CM9

Antonio E. Porreca
aeporreca.org/introinfo

Interface pour les arbres binaire

```
def nouveau_nœud(valeur, gauche, droite):
```

```
    ...
```

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

```
    ...
```

```
def valeur(nœud):
```

```
    ...
```

```
def gauche(nœud):
```

```
    ...
```

```
def droite(nœud):
```

```
    ...
```

Implémentation avec des tableaux

Implémentation avec des tableaux

```
def nouveau_nœud(valeur, gauche, droite):  
    return [valeur, gauche, droite]
```

Implémentation avec des tableaux

```
def nouveau_nœud(valeur, gauche, droite):  
    return [valeur, gauche, droite]  
  
def est_vide(arbre):  
    return arbre is None
```

Implémentation avec des tableaux

```
def nouveau_nœud(valeur, gauche, droite):  
    return [valeur, gauche, droite]  
  
def est_vide(arbre):  
    return arbre is None  
  
def valeur(nœud):  
    return nœud[0]
```

Implémentation avec des tableaux

```
def nouveau_nœud(valeur, gauche, droite):  
    return [valeur, gauche, droite]
```

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

```
def valeur(nœud):  
    return nœud[0]
```

```
def gauche(nœud):  
    return nœud[1]
```

Implémentation avec des tableaux

```
def nouveau_nœud(valeur, gauche, droite):  
    return [valeur, gauche, droite]
```

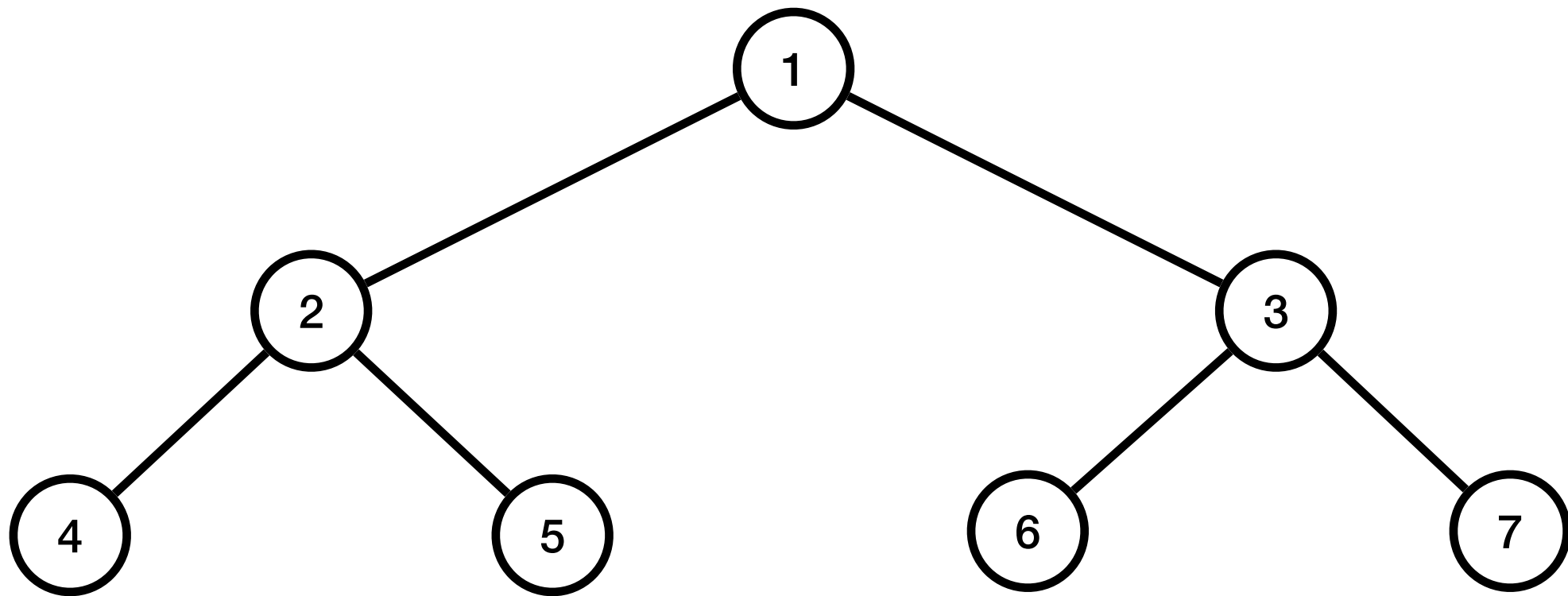
```
def est_vide(arbre):  
    return arbre is None
```

```
def valeur(nœud):  
    return nœud[0]
```

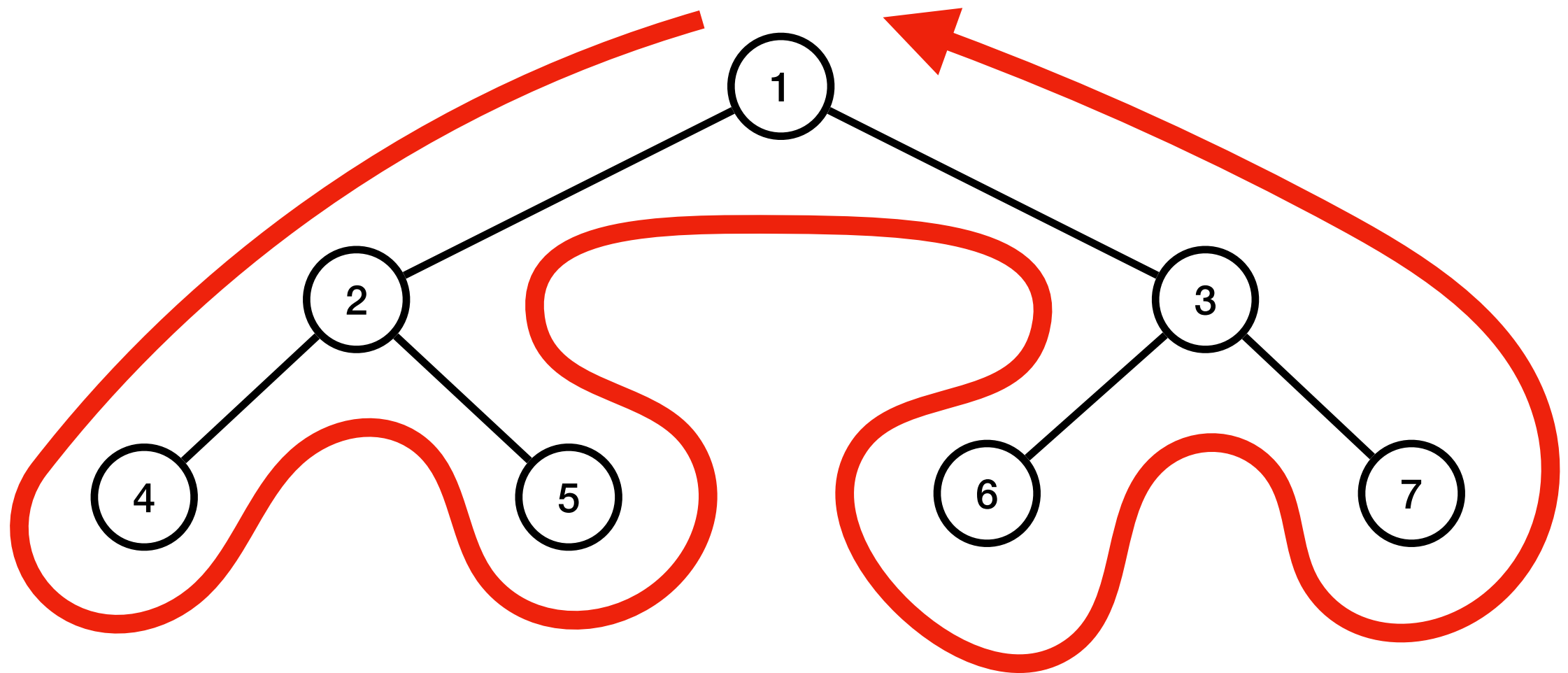
```
def gauche(nœud):  
    return nœud[1]
```

```
def droite(nœud):  
    return nœud[2]
```

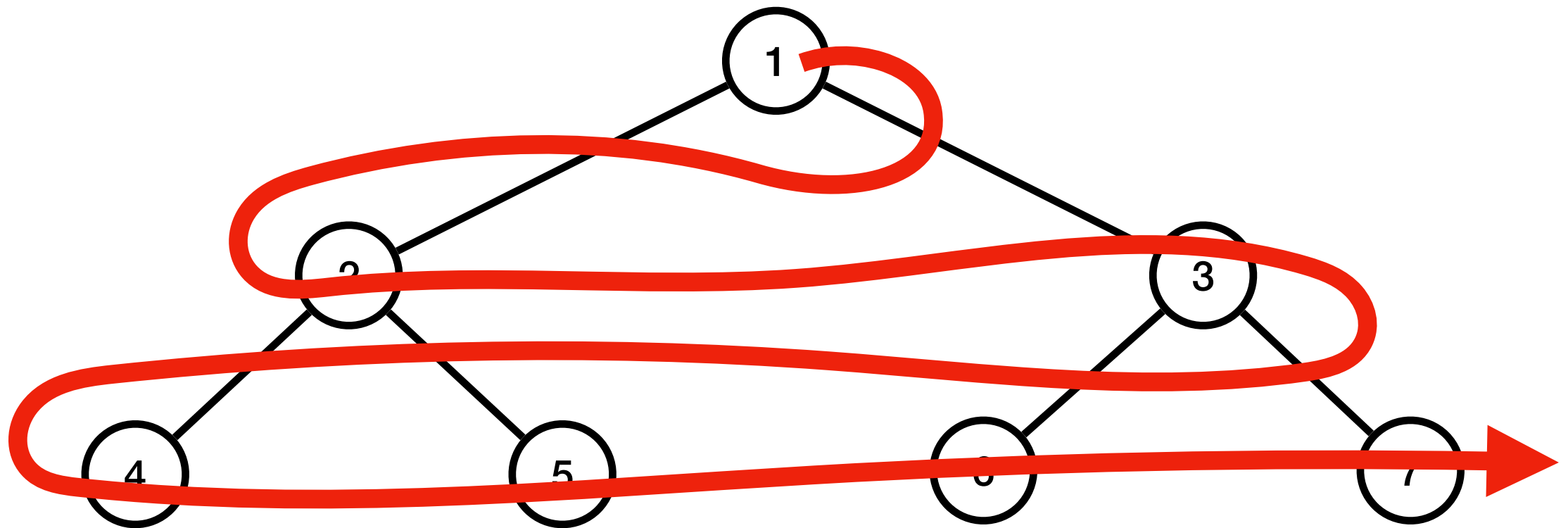

Parcours d'arbres binaires



Parcours d'arbres binaires



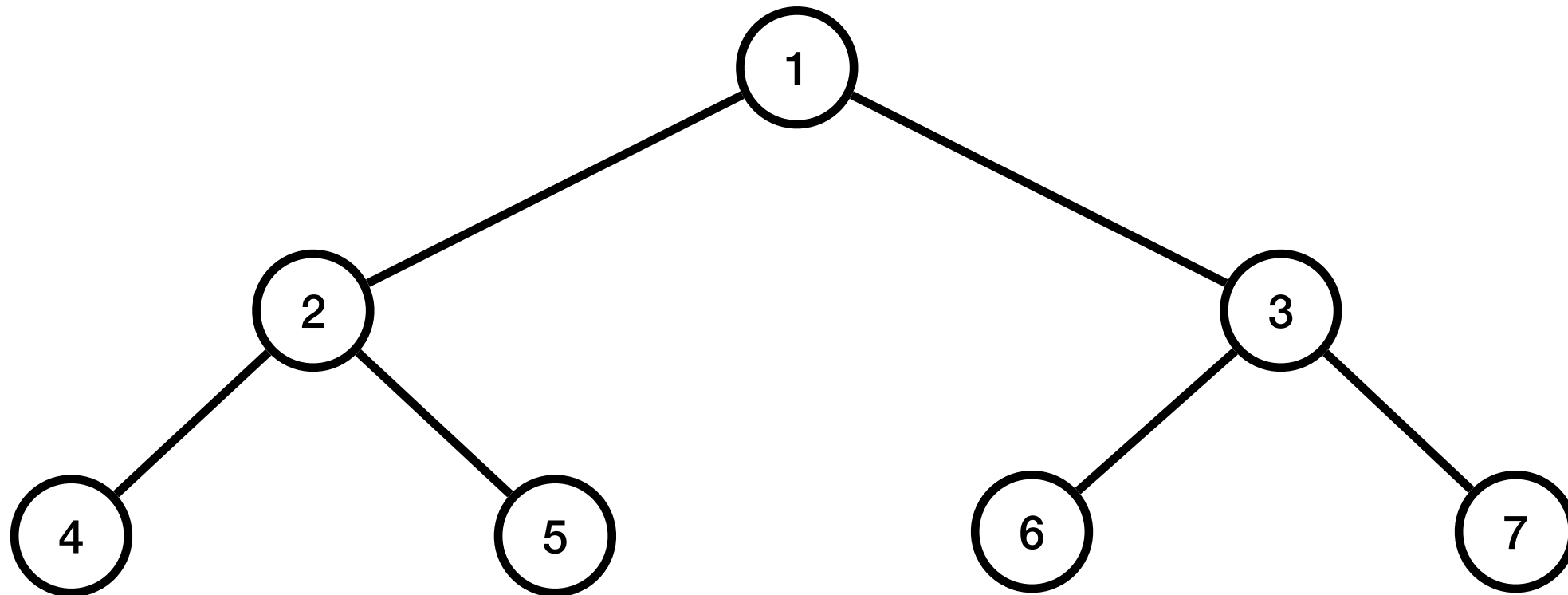
Parcours d'arbres binaires



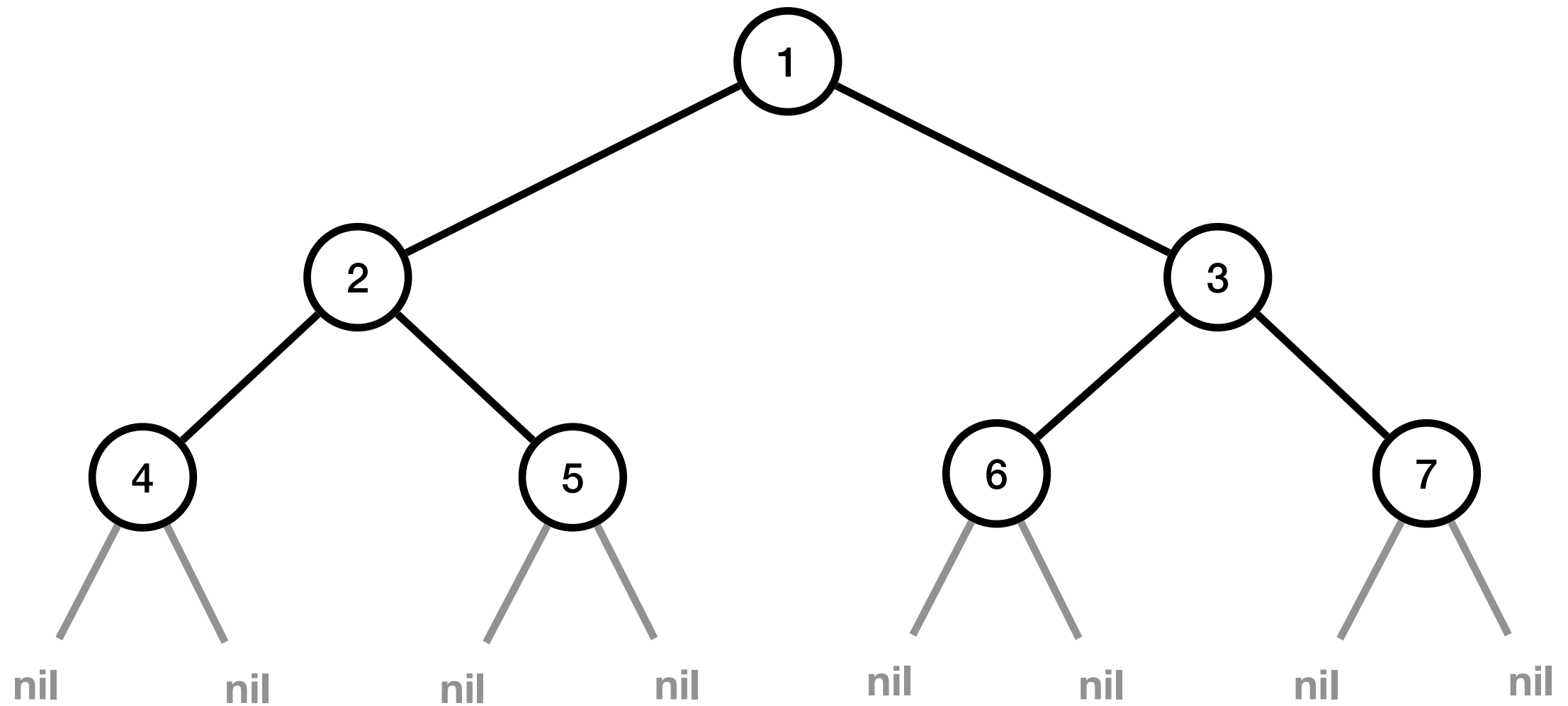
Parcours préfixe

```
def parcours_préfixe(arbre):  
    if not est_vide(arbre):  
        print(valeur(arbre))  
        parcours_préfixe(gauche(arbre))  
        parcours_préfixe(droite(arbre))
```

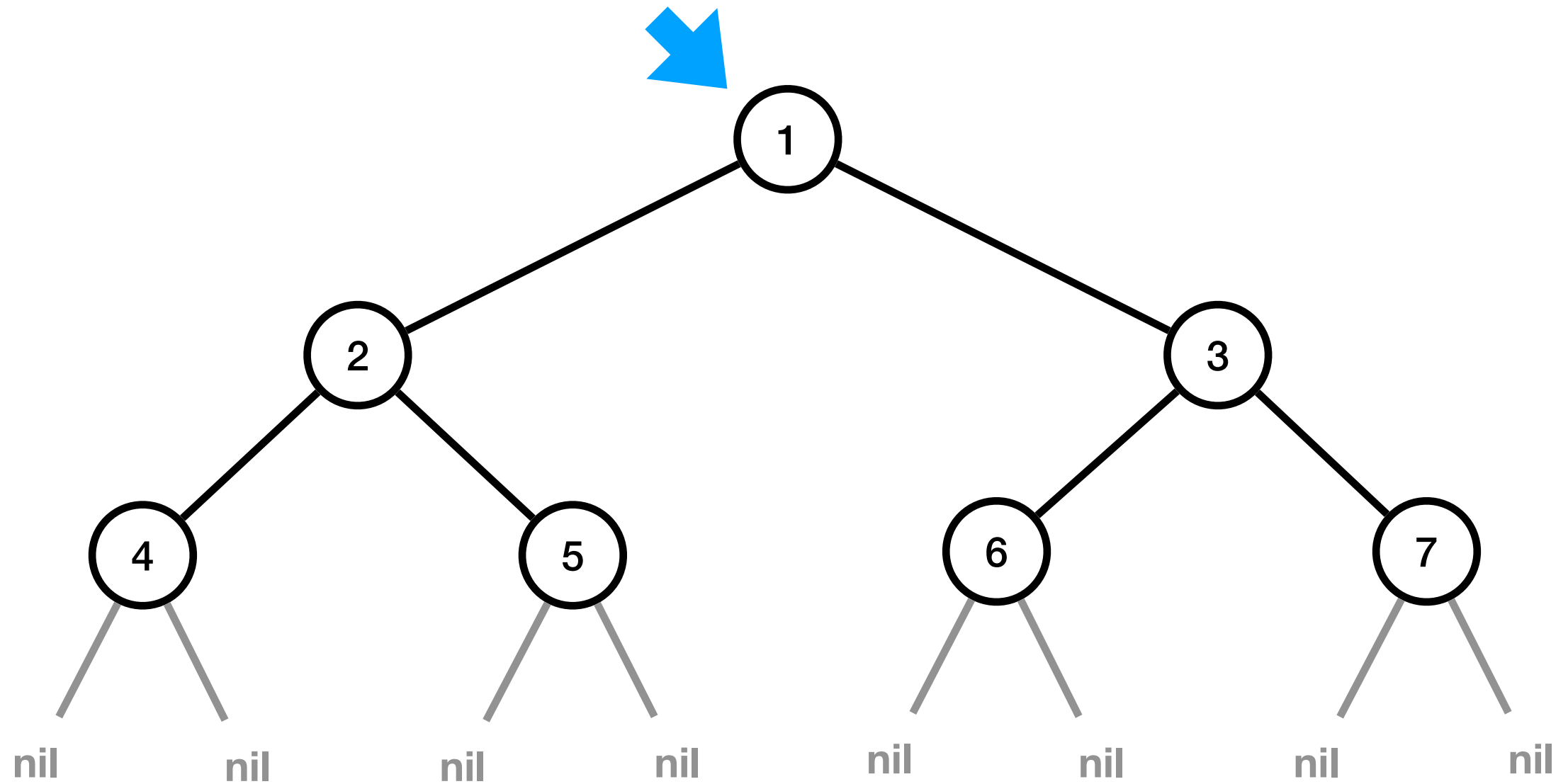
Parcours préfixe



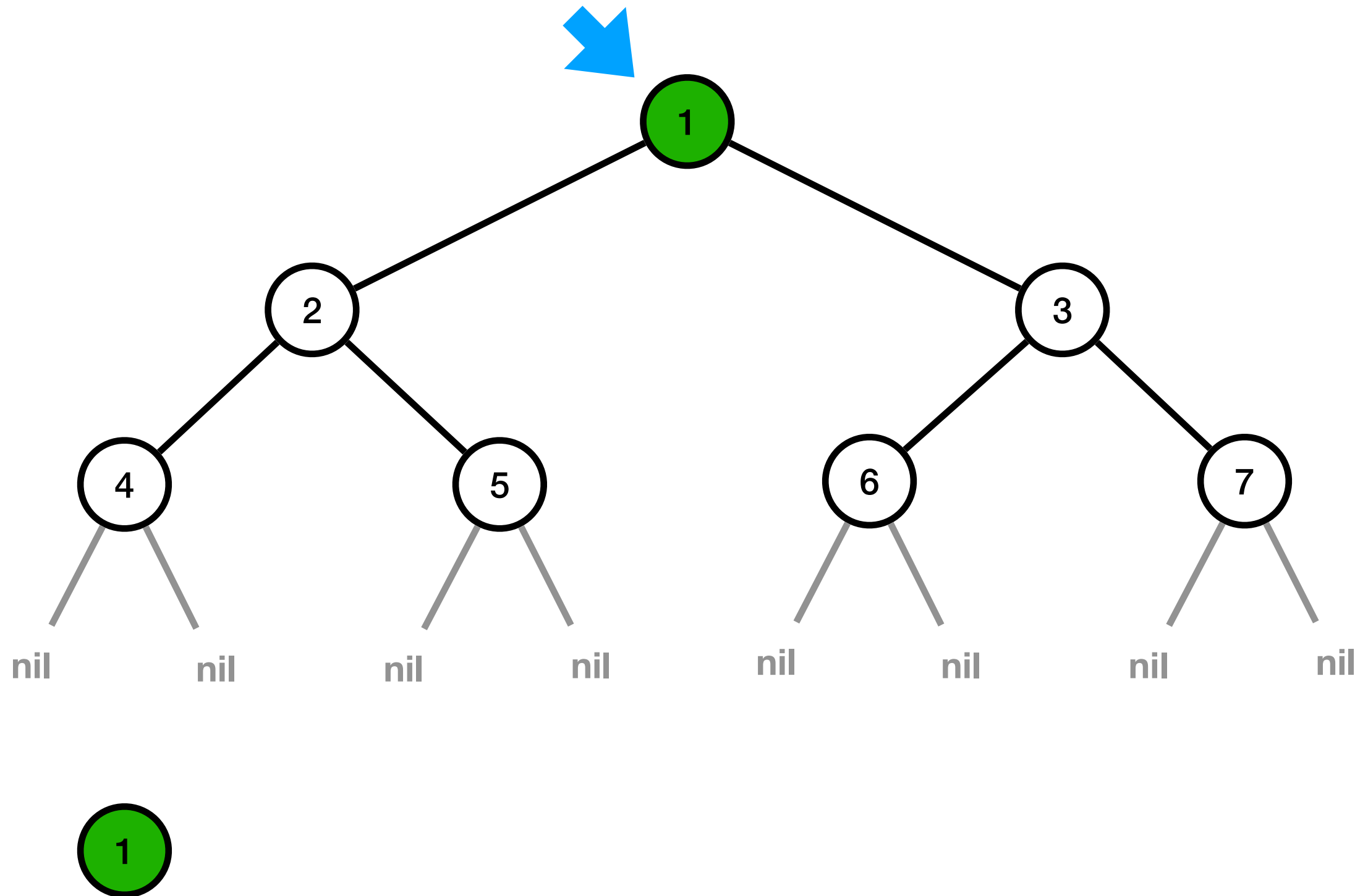
Parcours préfixe



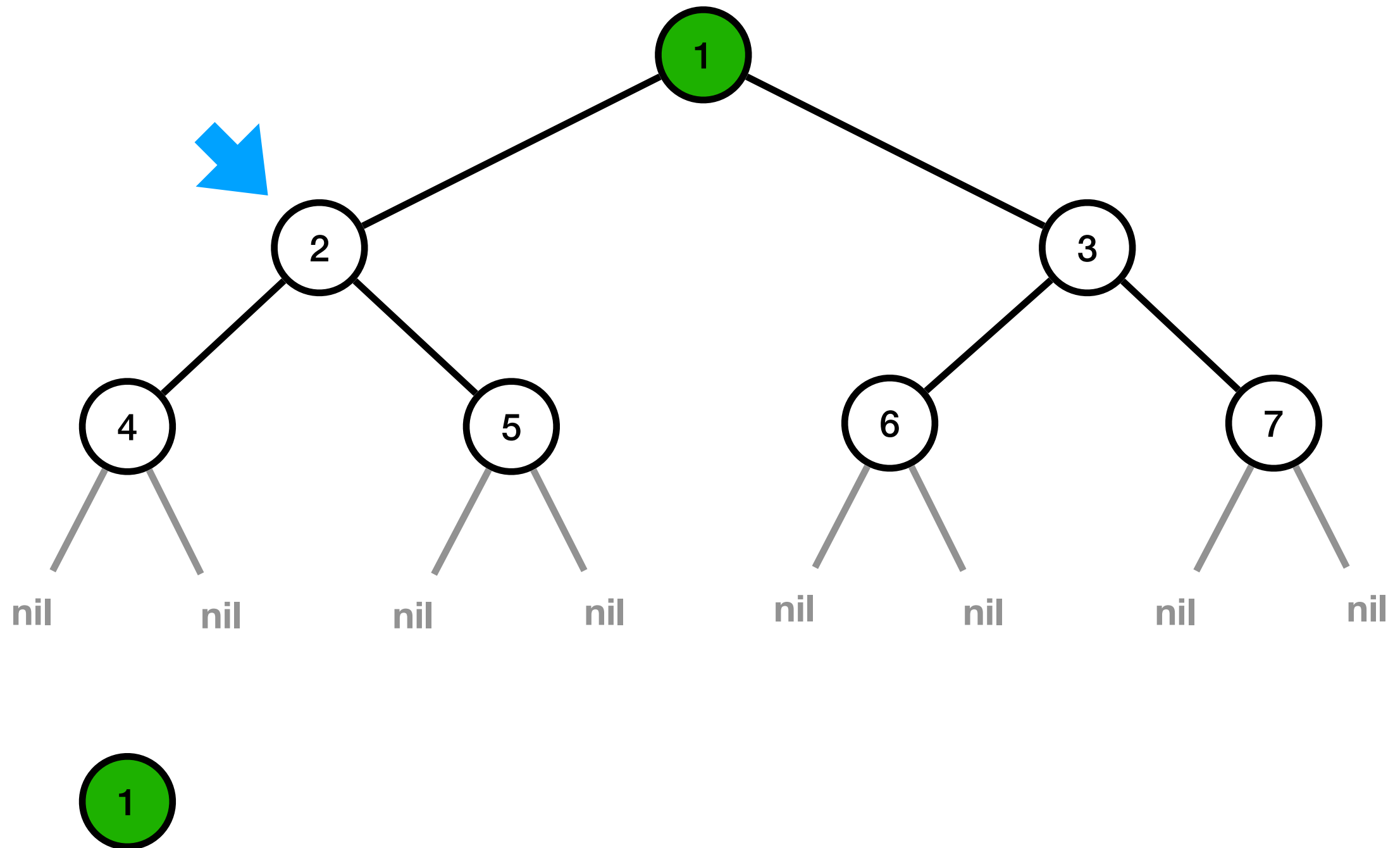
Parcours préfixe



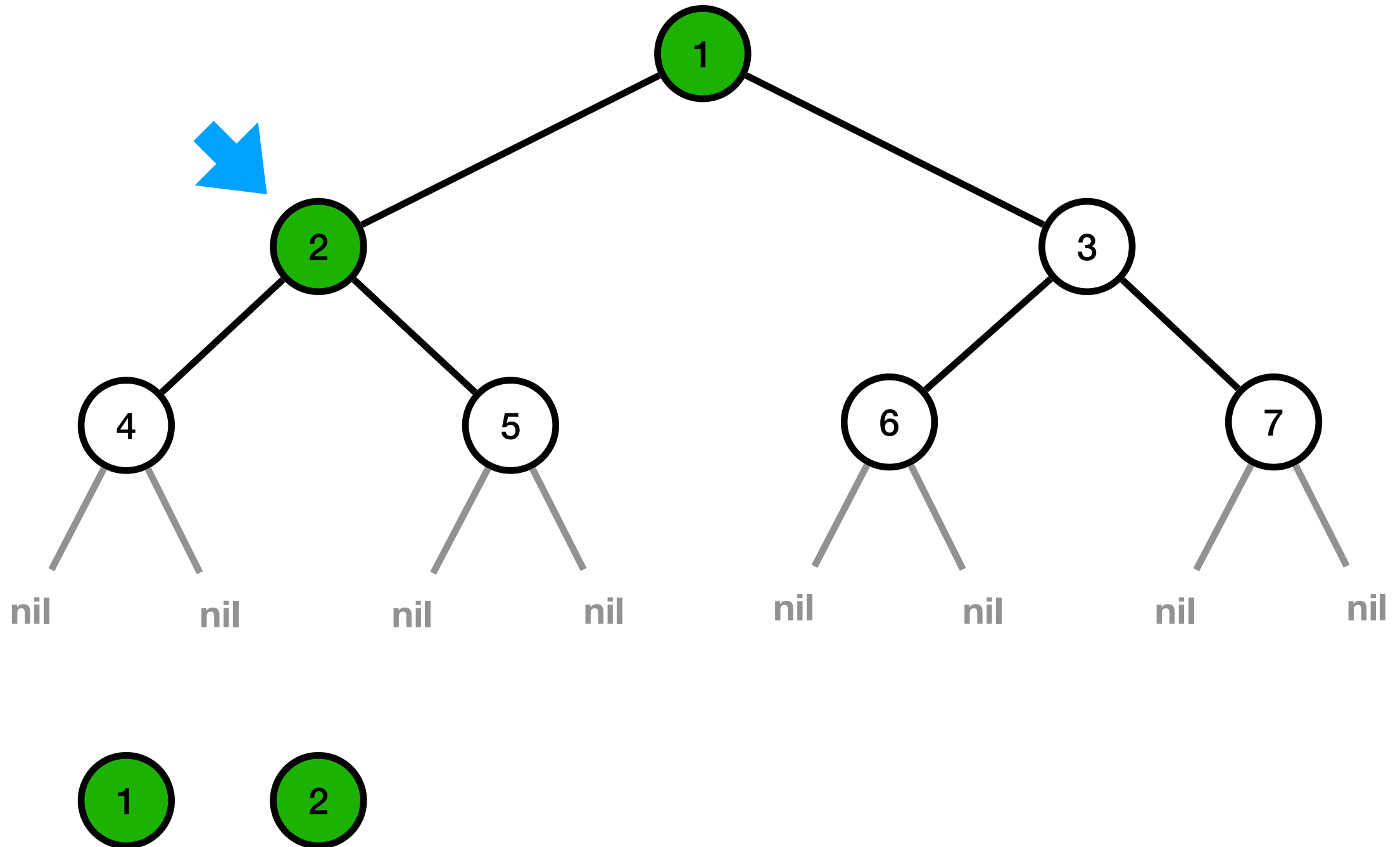
Parcours préfixe



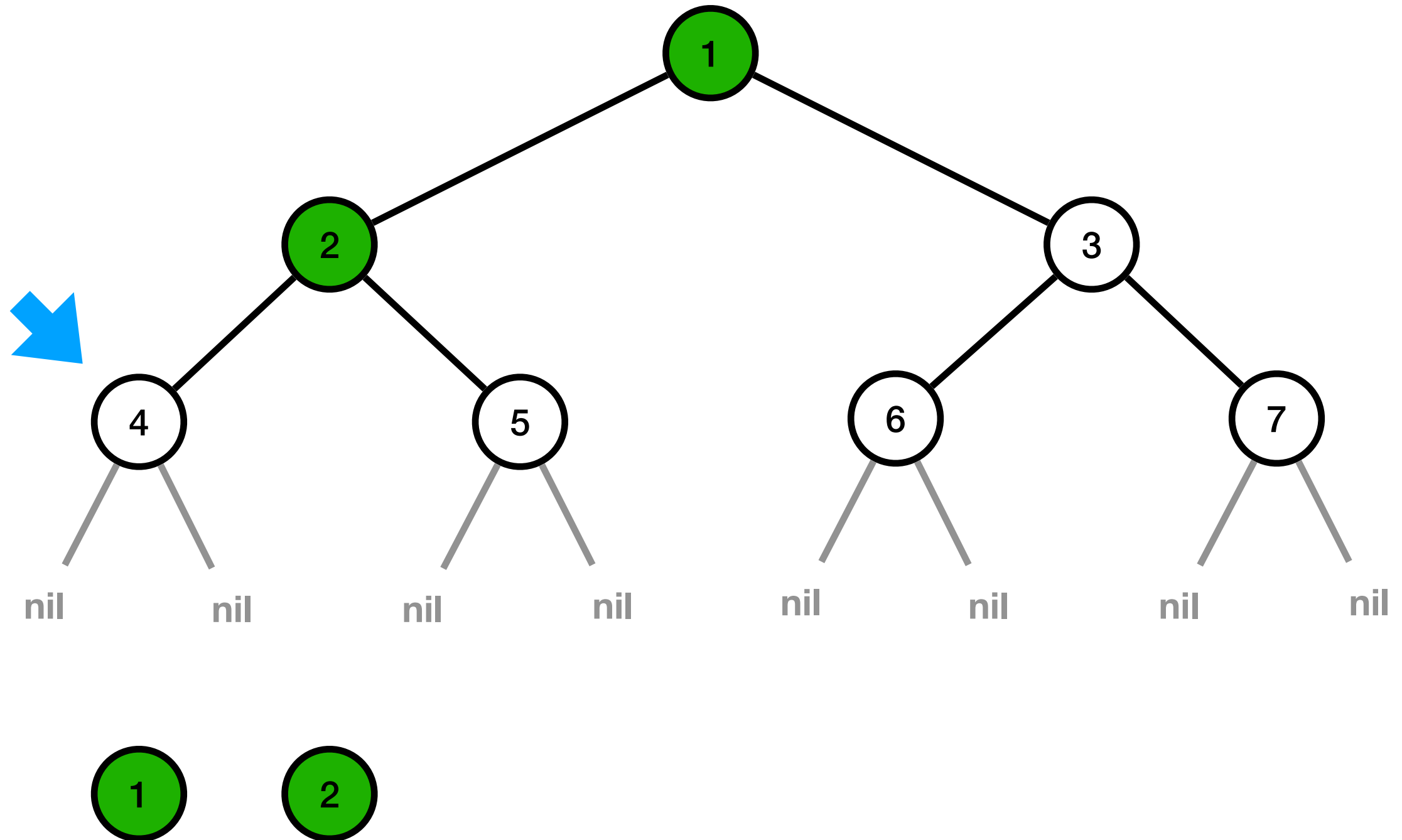
Parcours préfixe



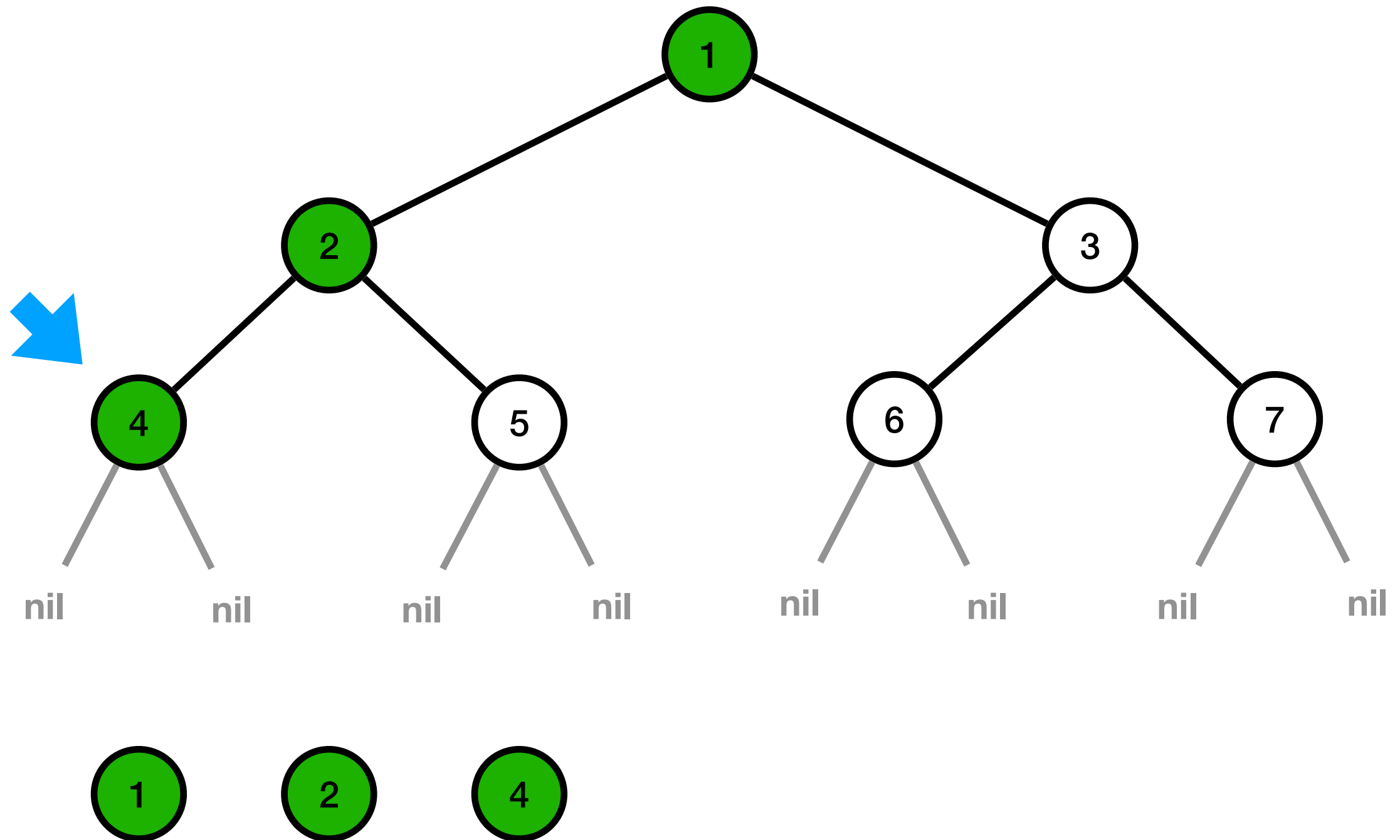
Parcours préfixe



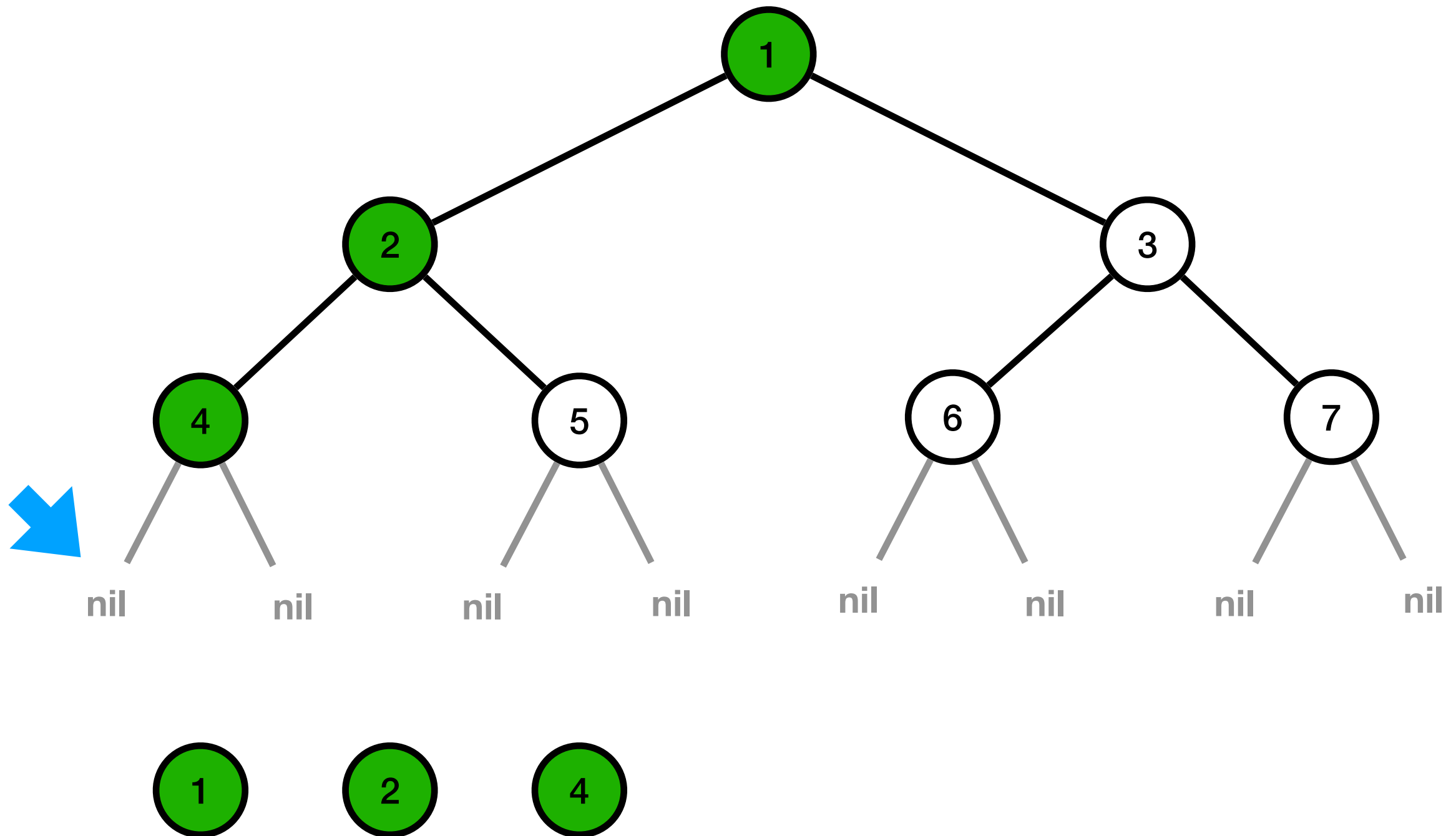
Parcours préfixe



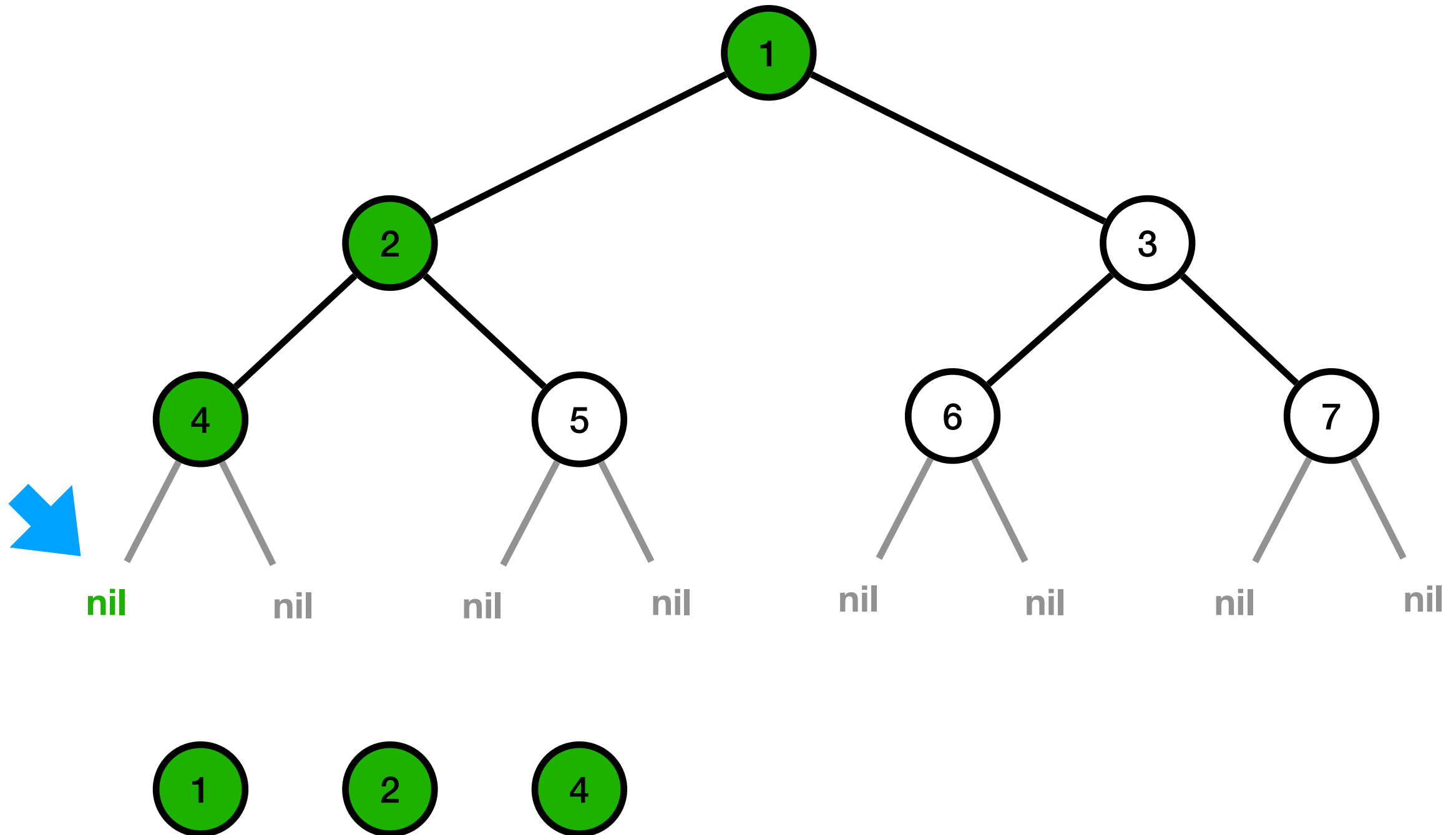
Parcours préfixe



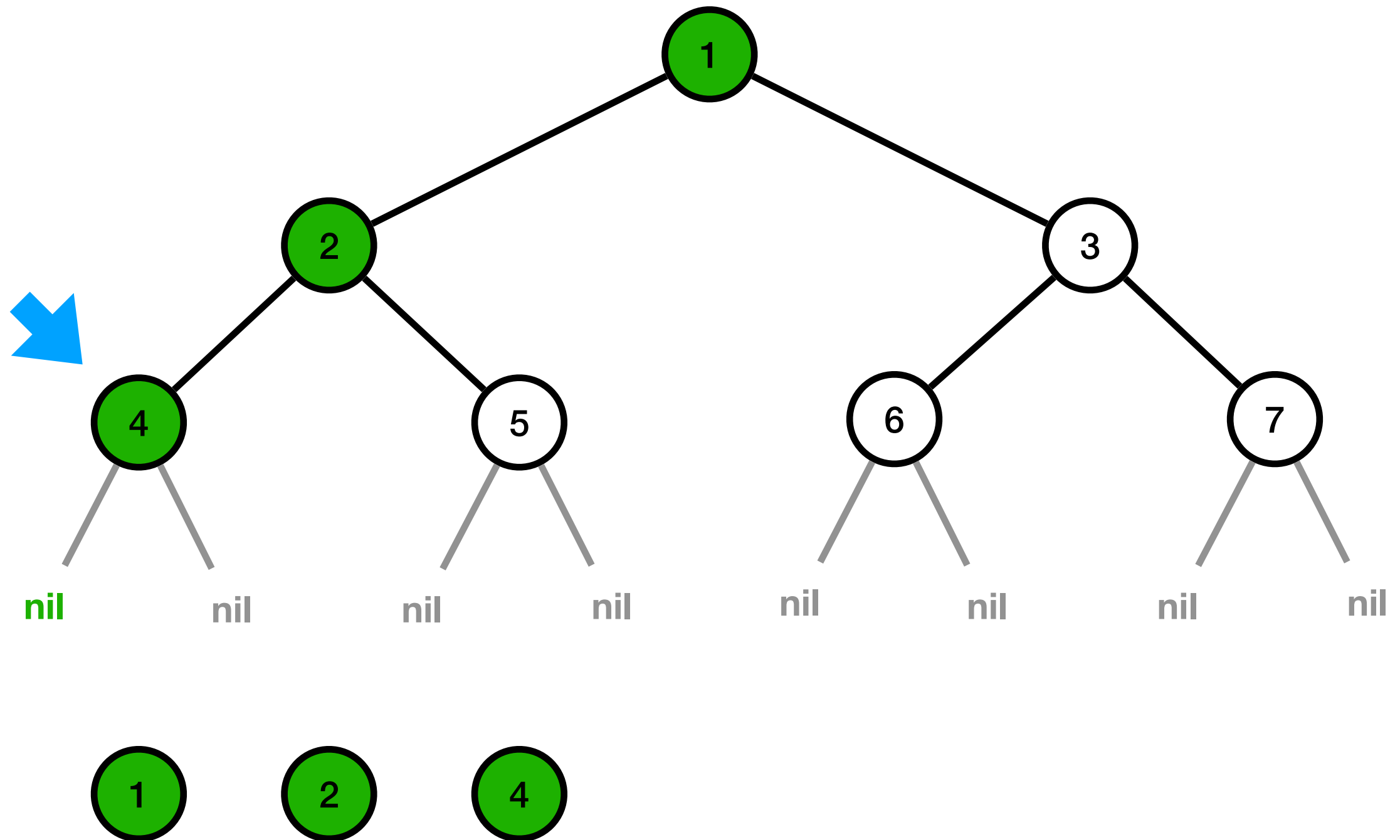
Parcours préfixe



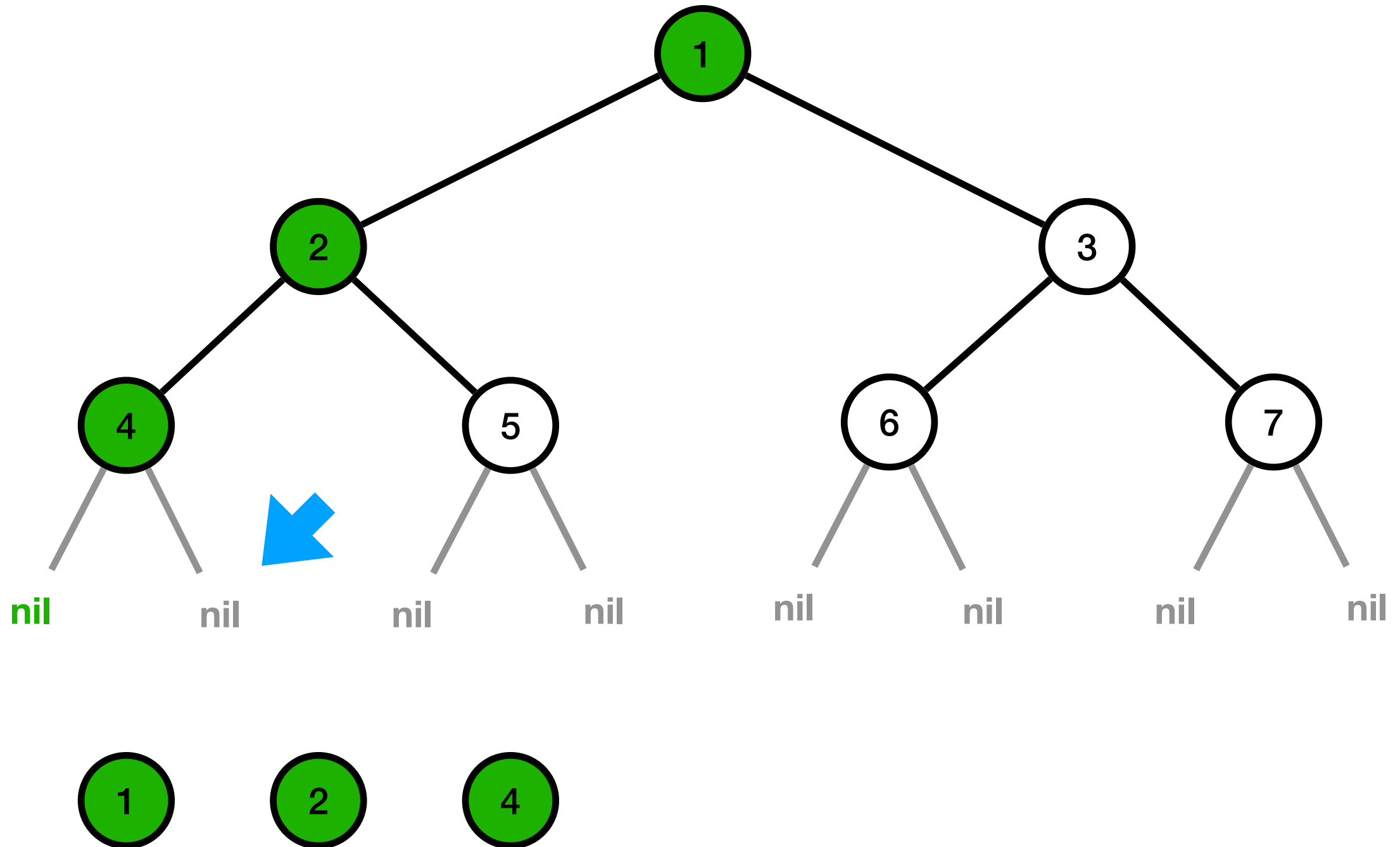
Parcours préfixe



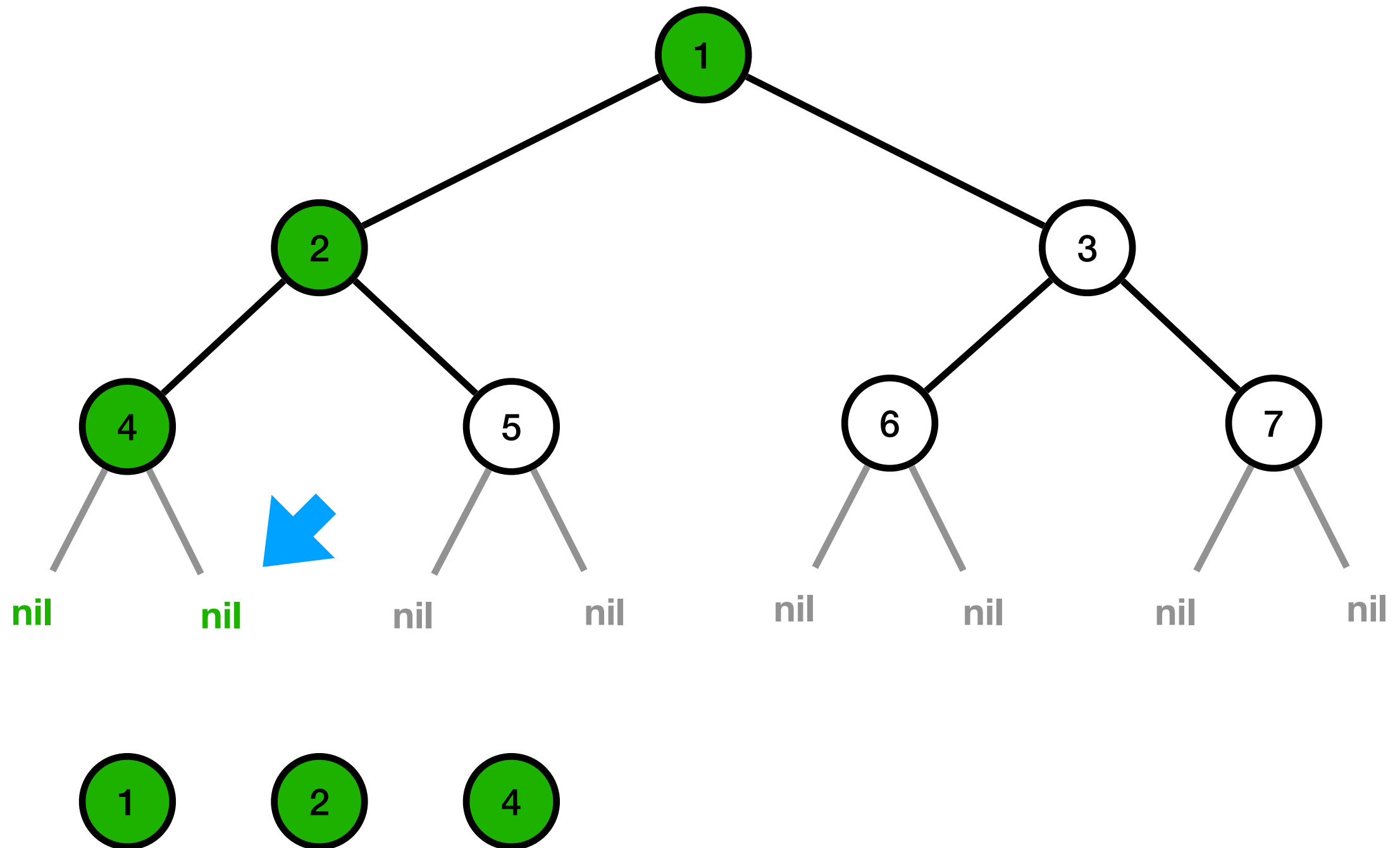
Parcours préfixe



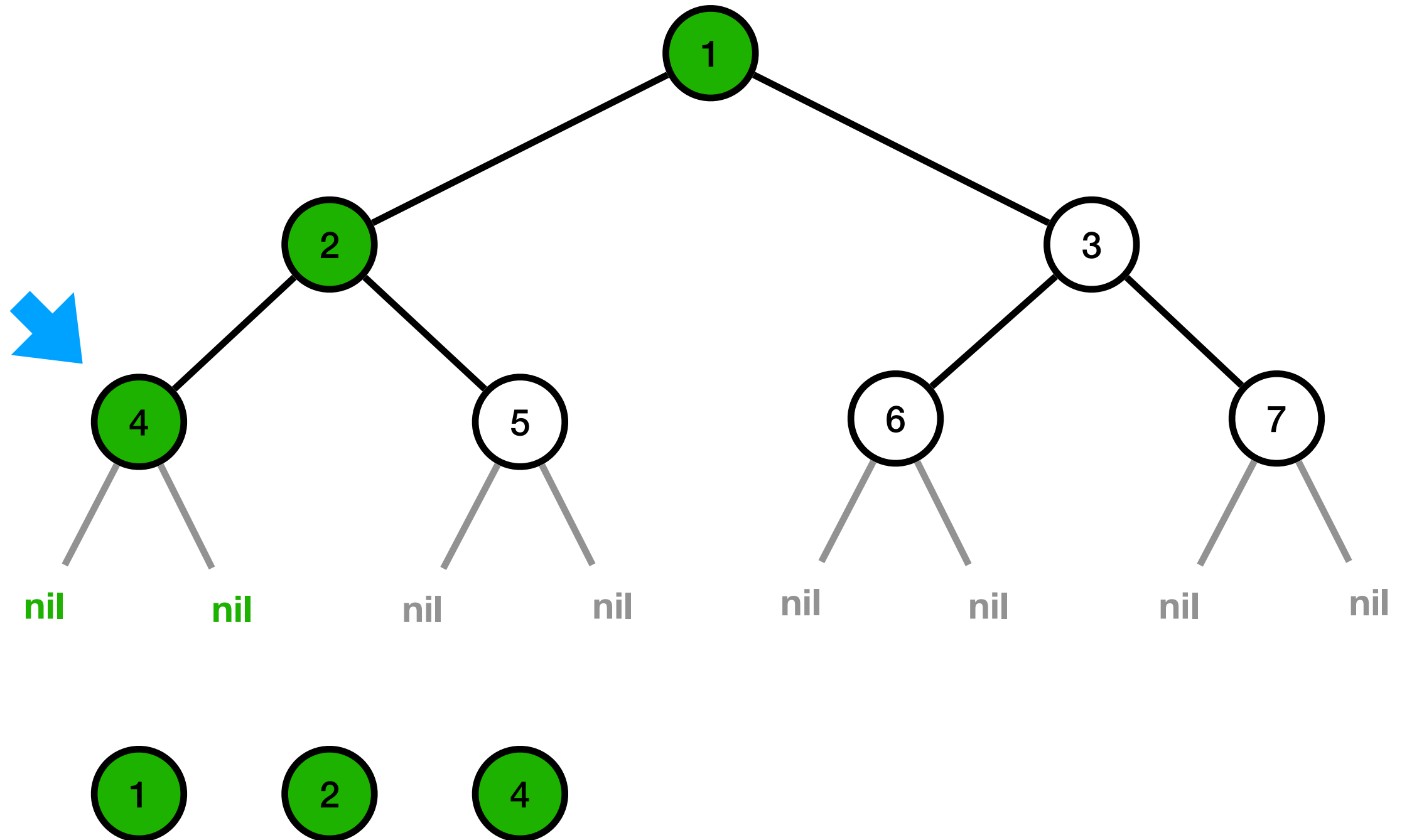
Parcours préfixe



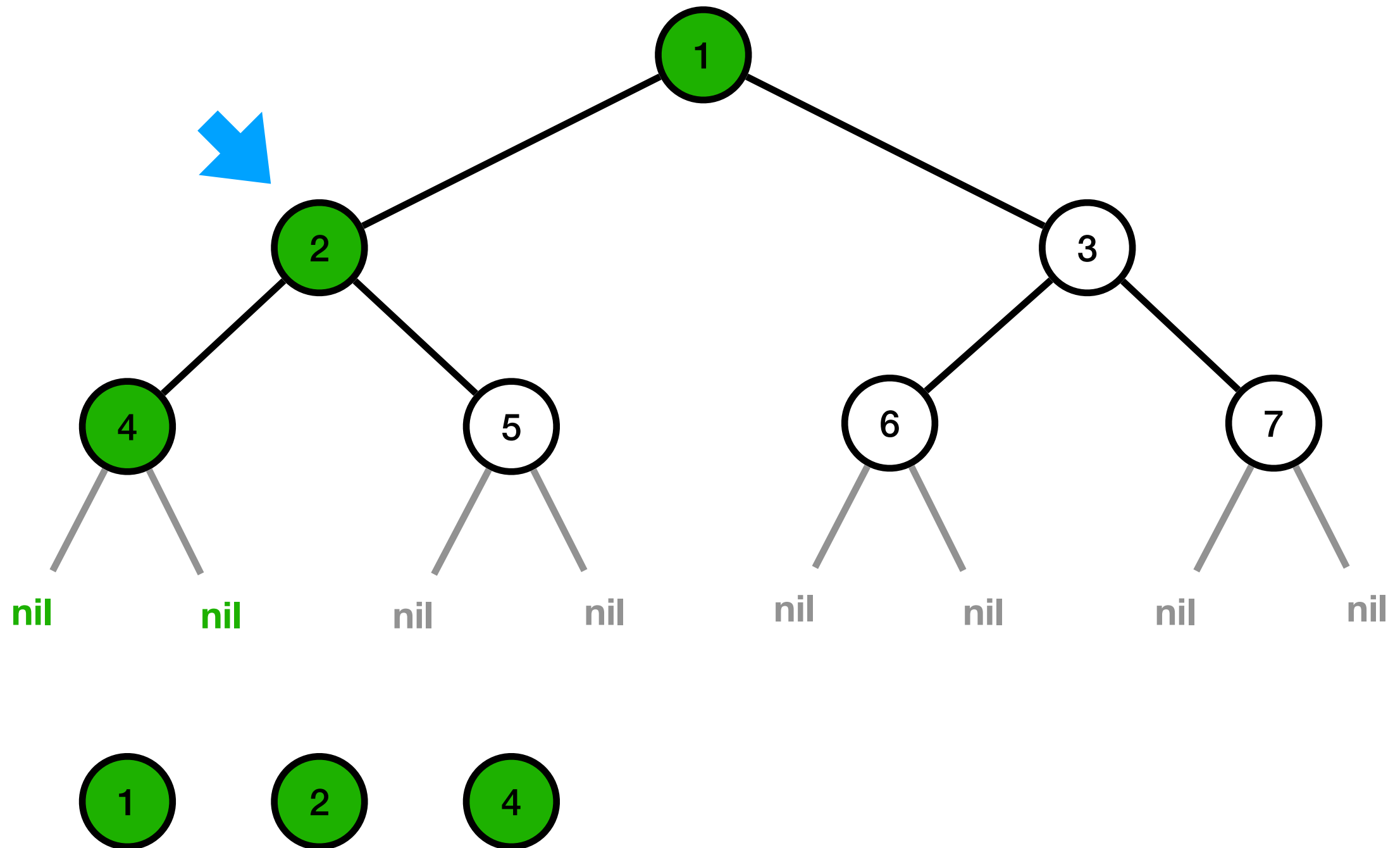
Parcours préfixe



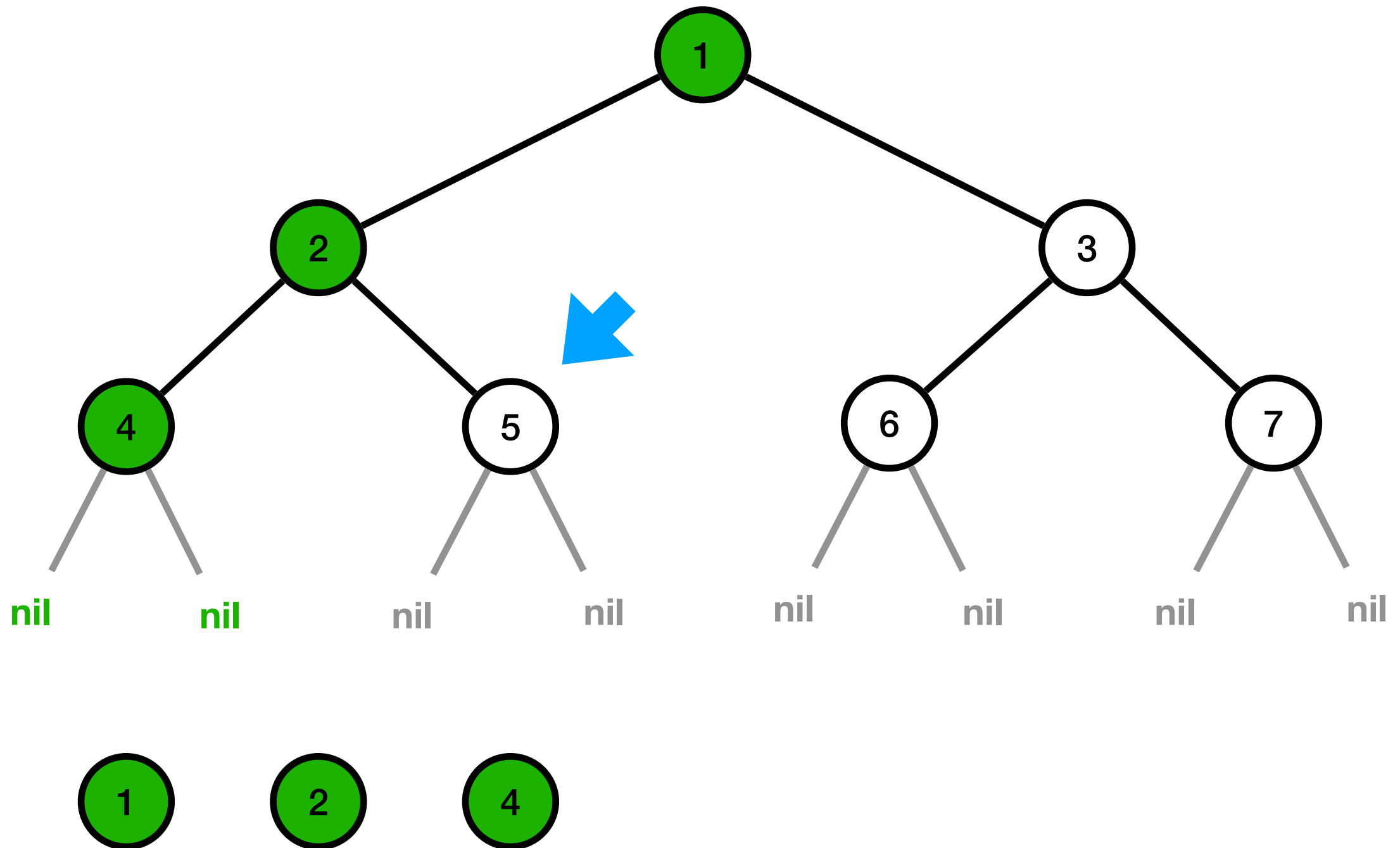
Parcours préfixe



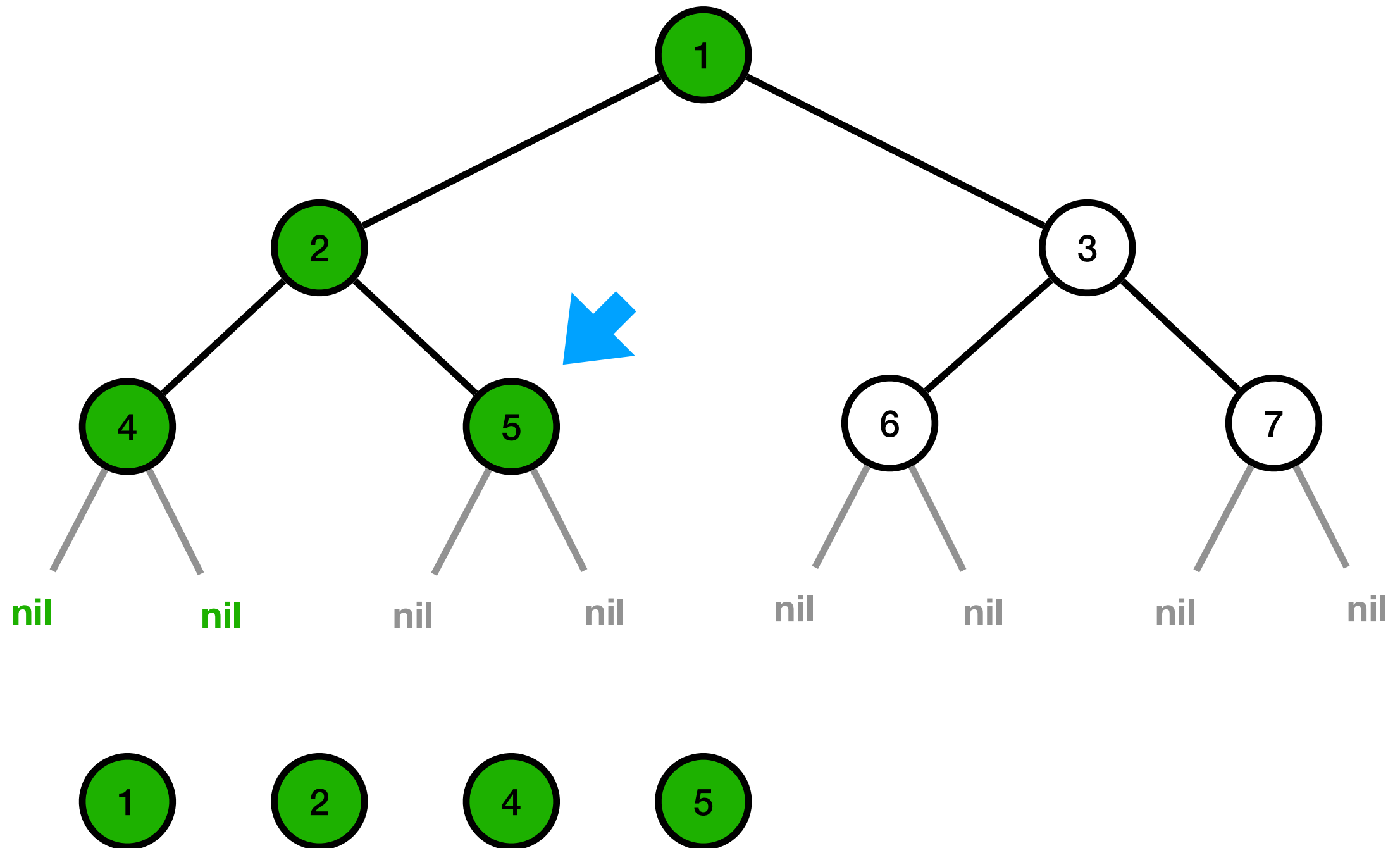
Parcours préfixe



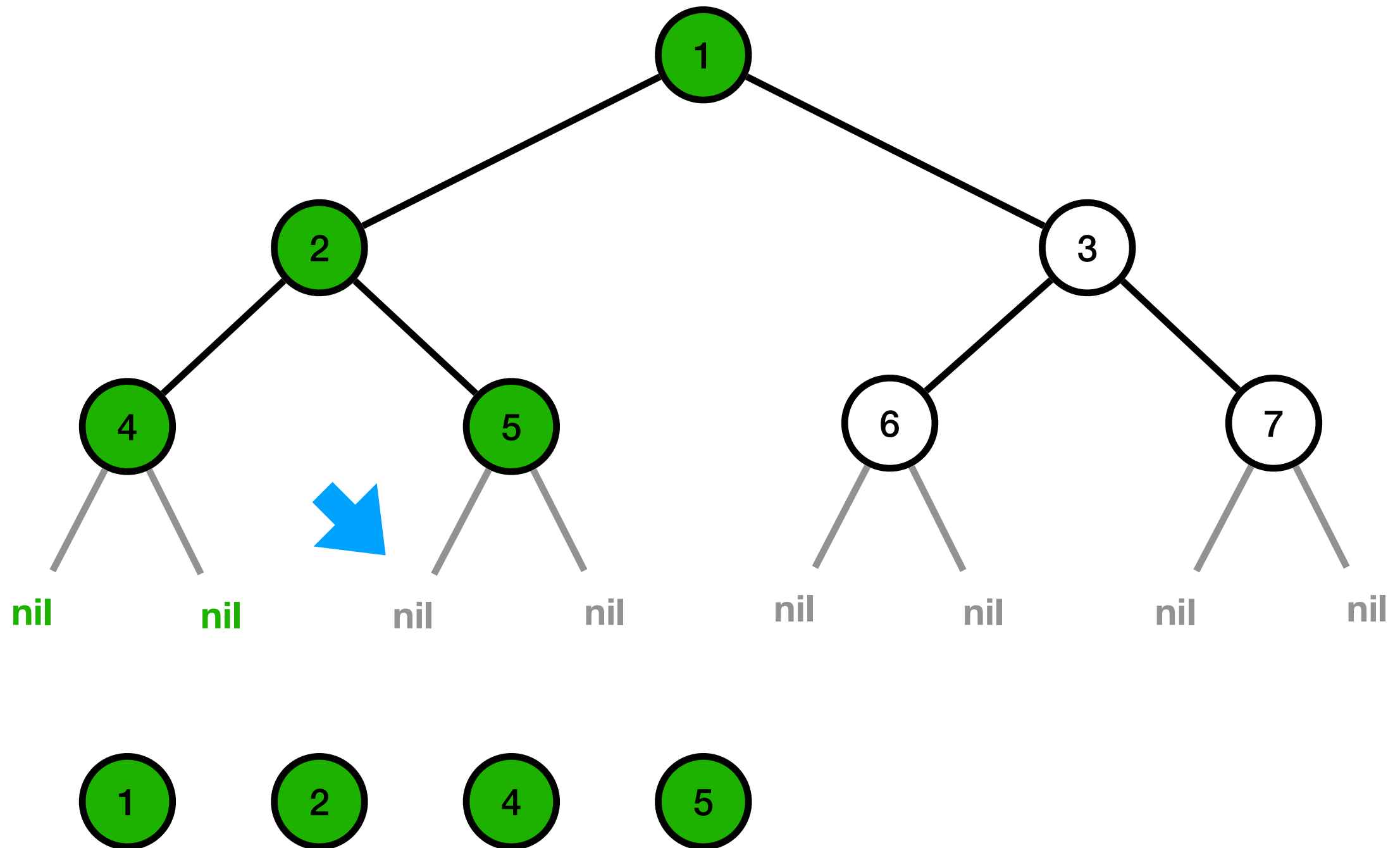
Parcours préfixe



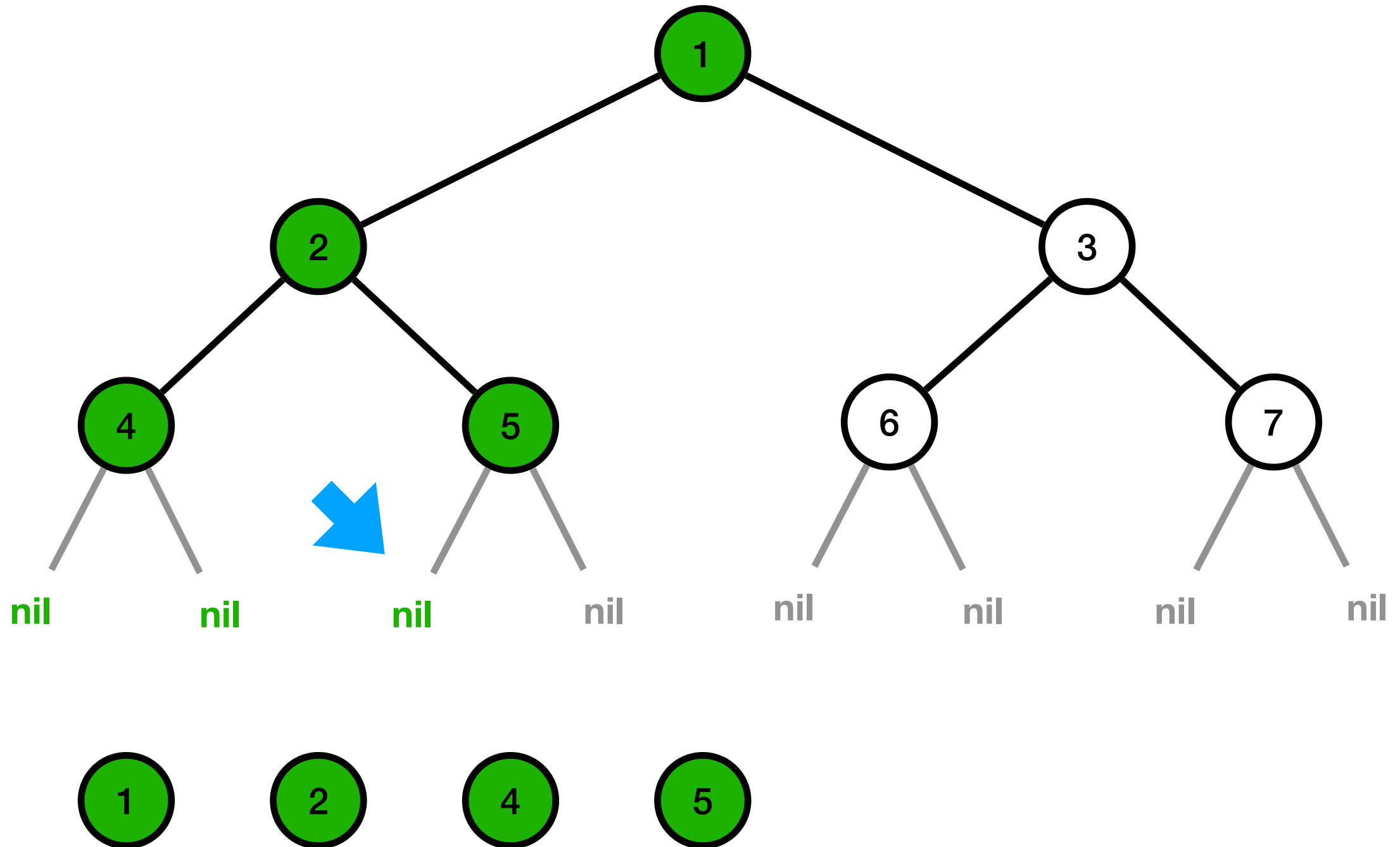
Parcours préfixe



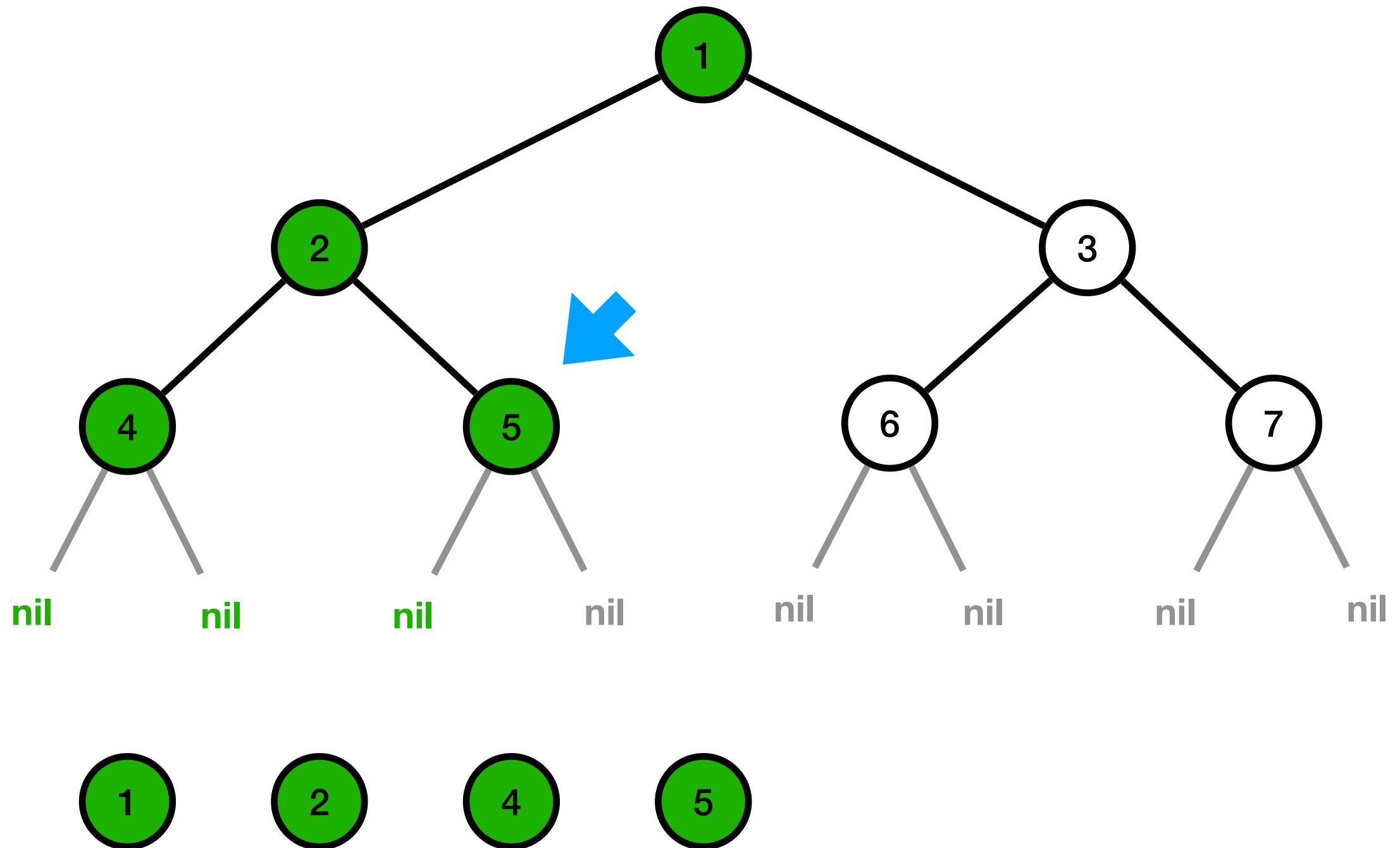
Parcours préfixe



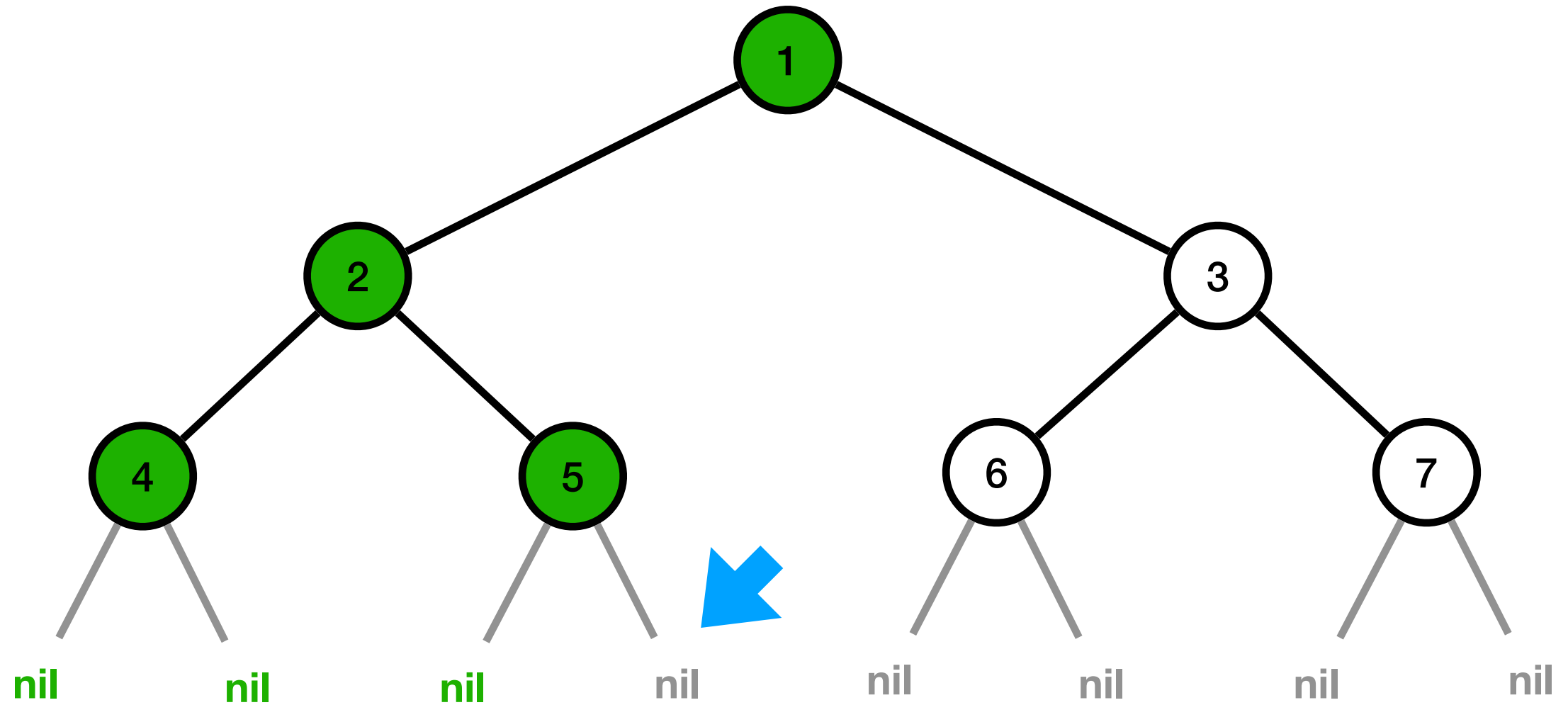
Parcours préfixe



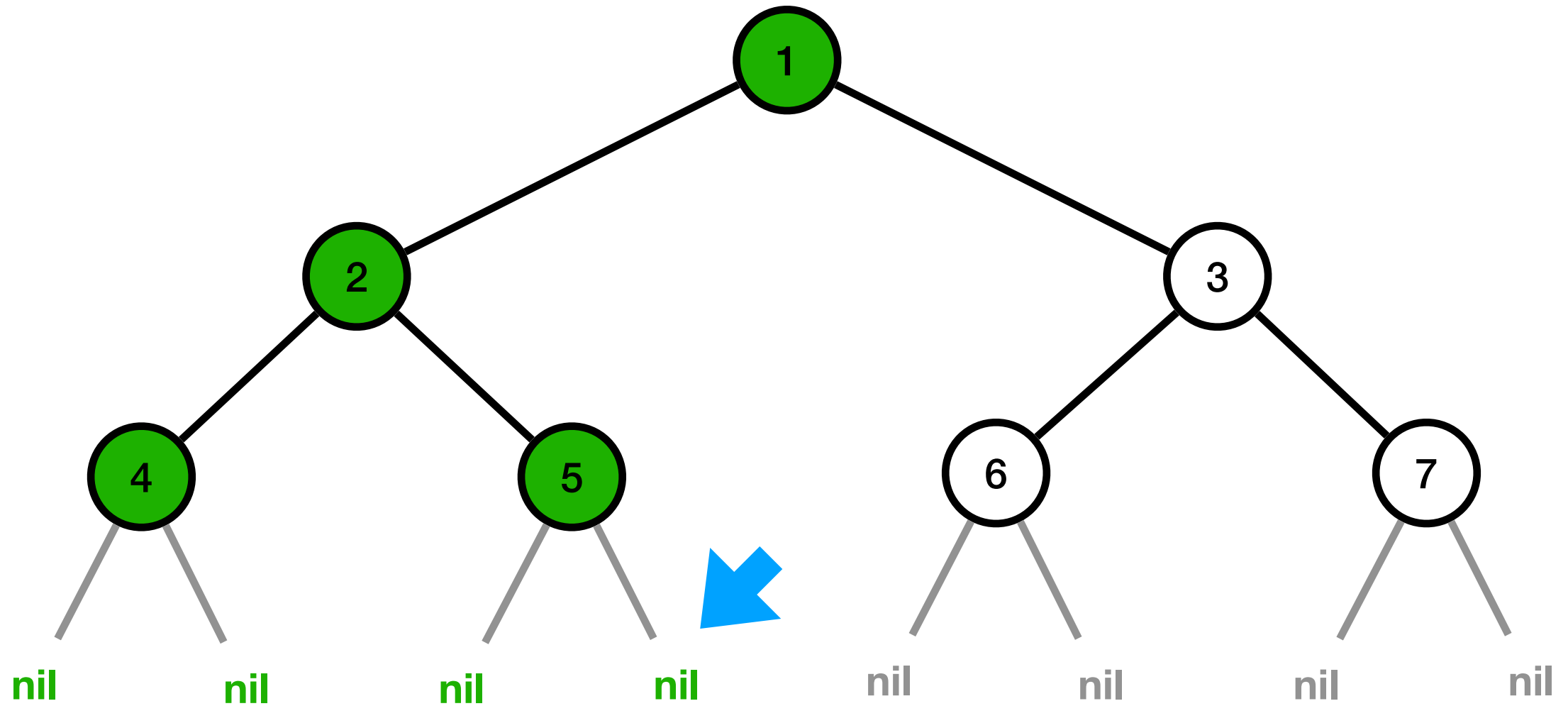
Parcours préfixe



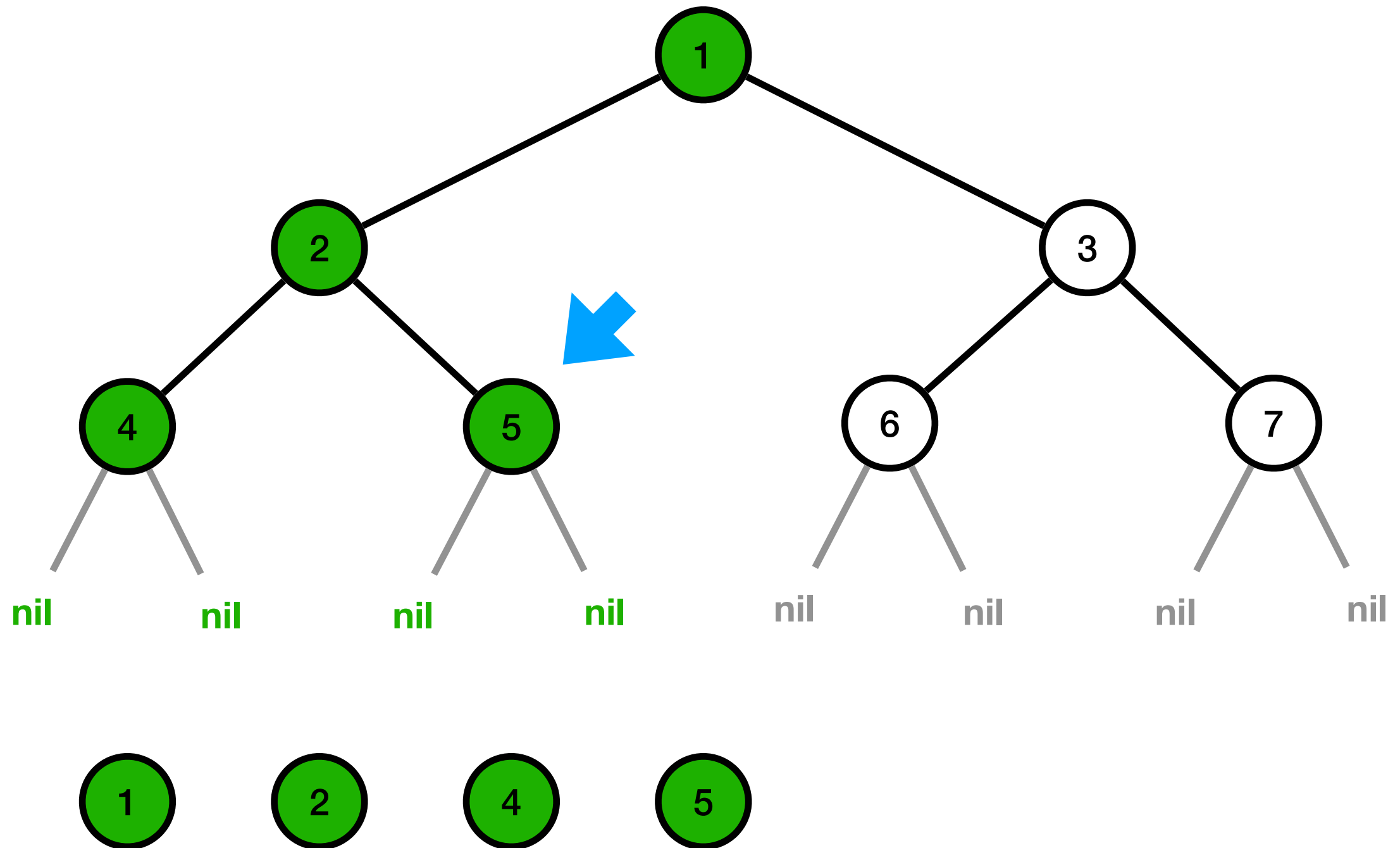
Parcours préfixe



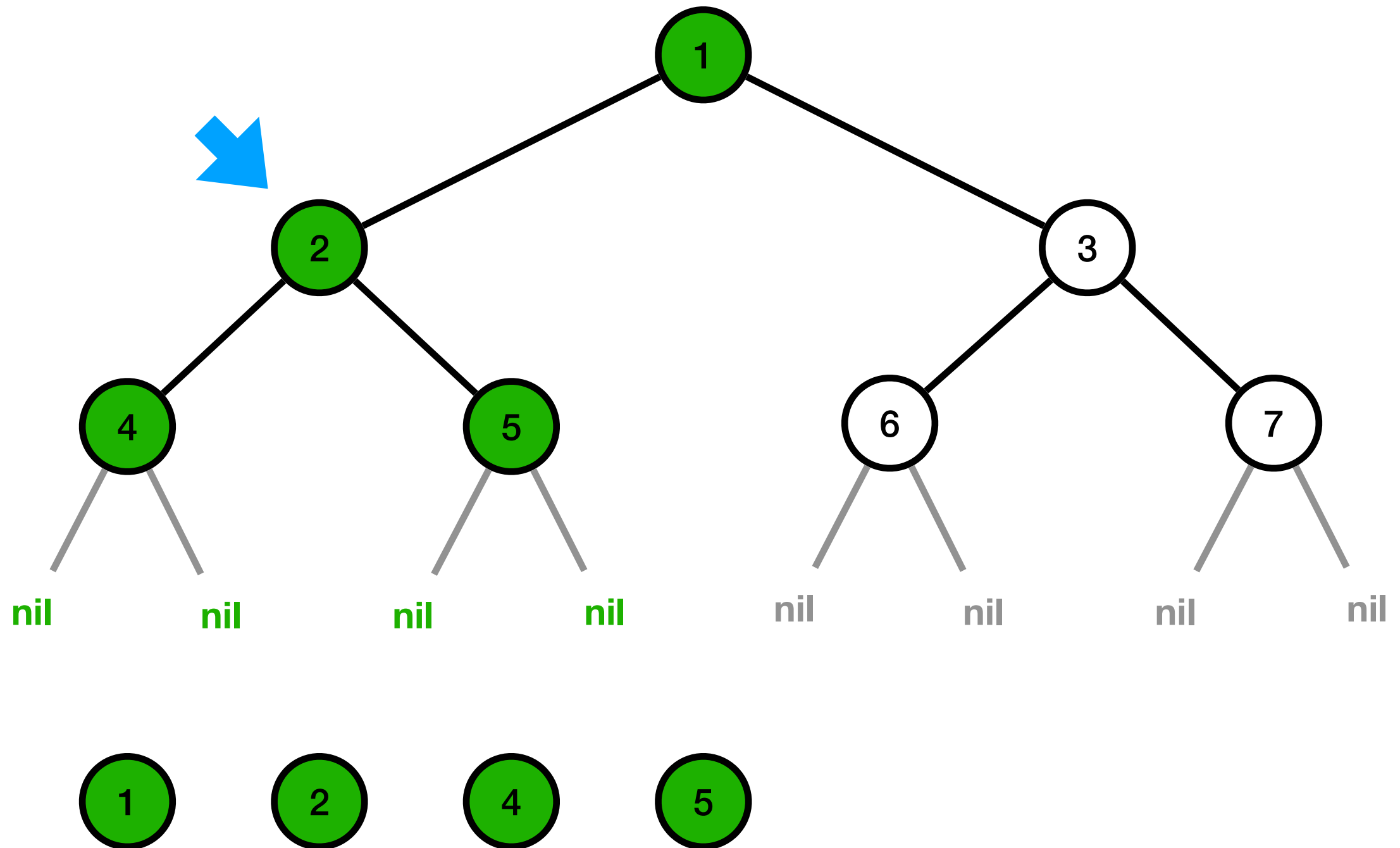
Parcours préfixe



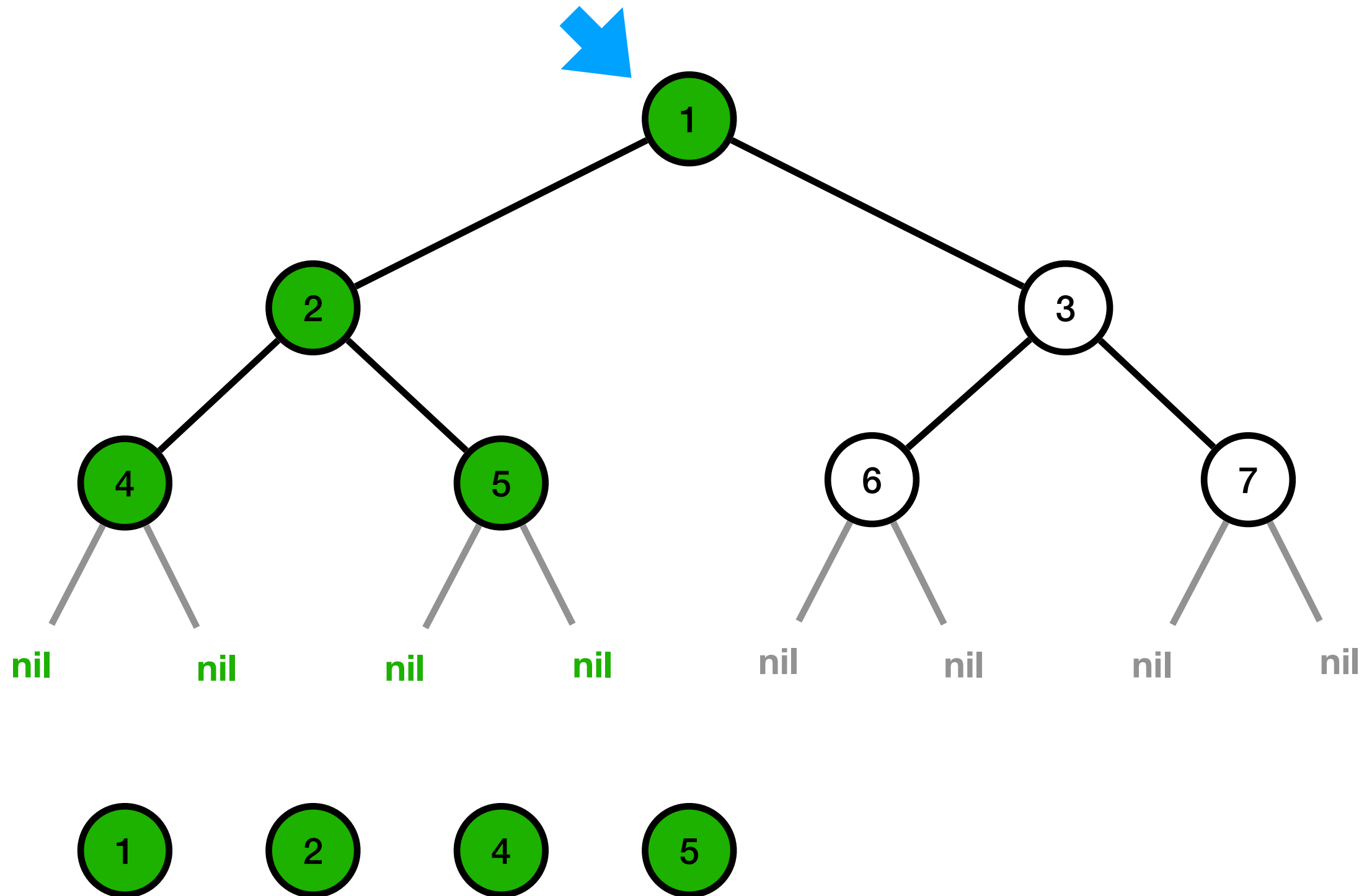
Parcours préfixe



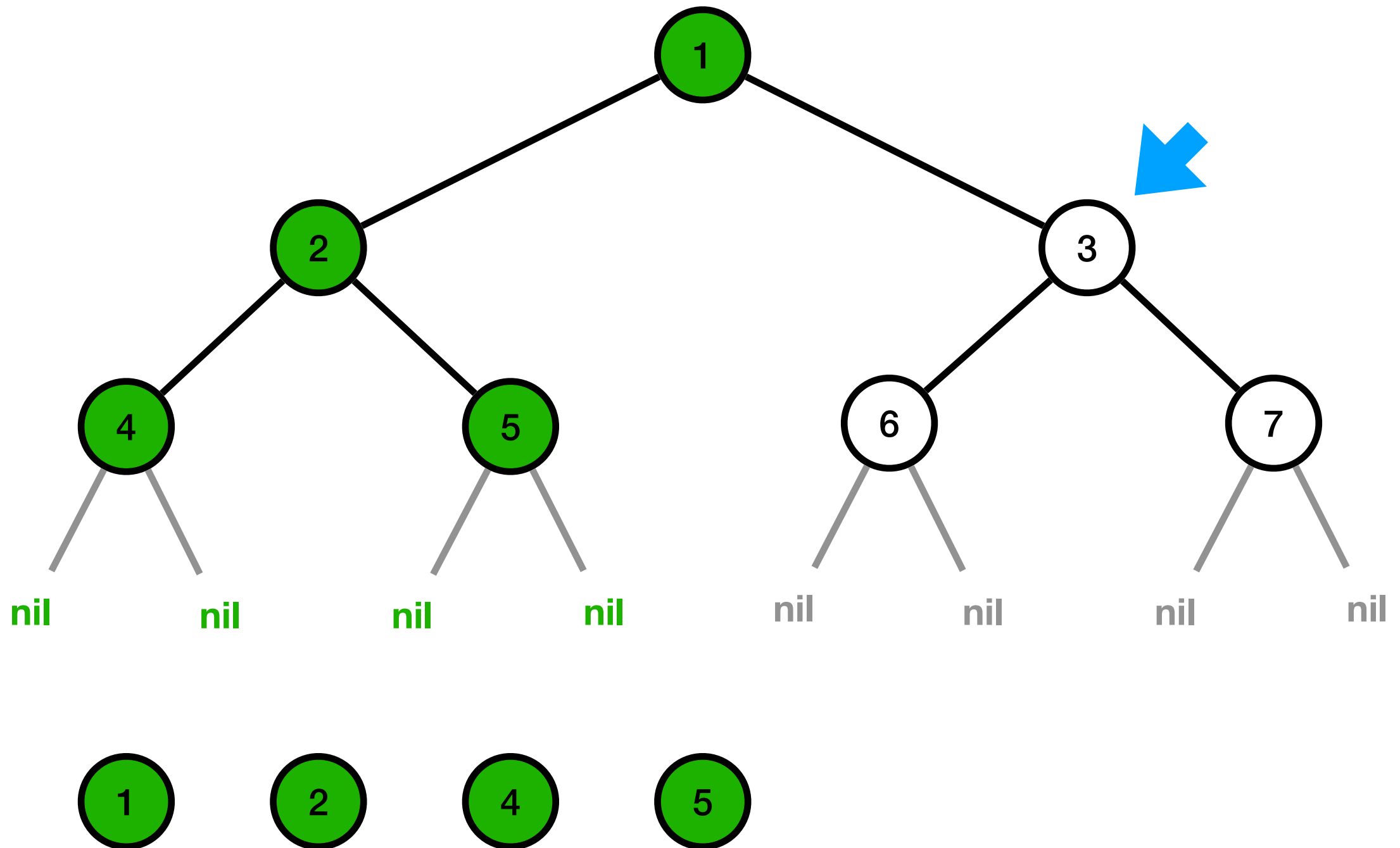
Parcours préfixe



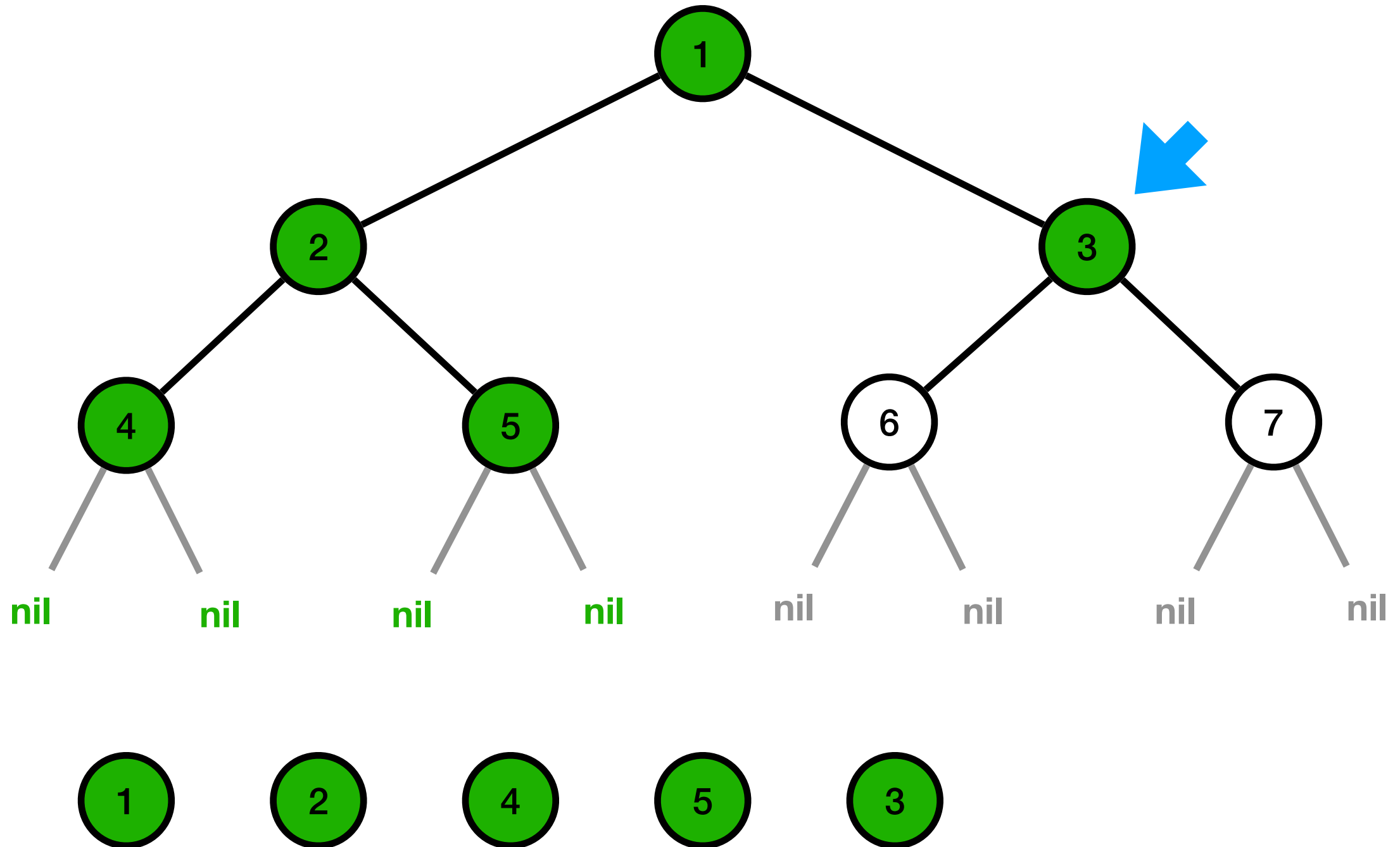
Parcours préfixe



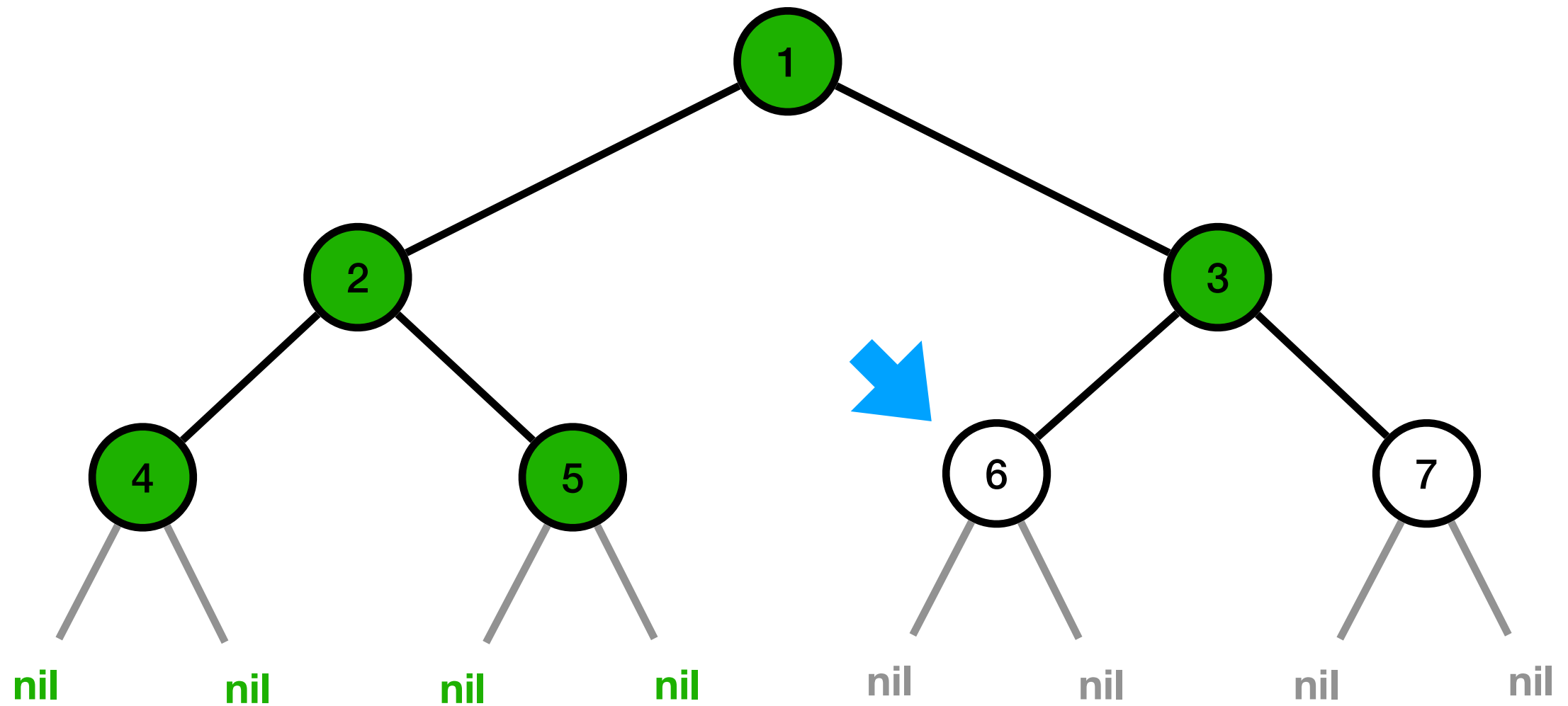
Parcours préfixe



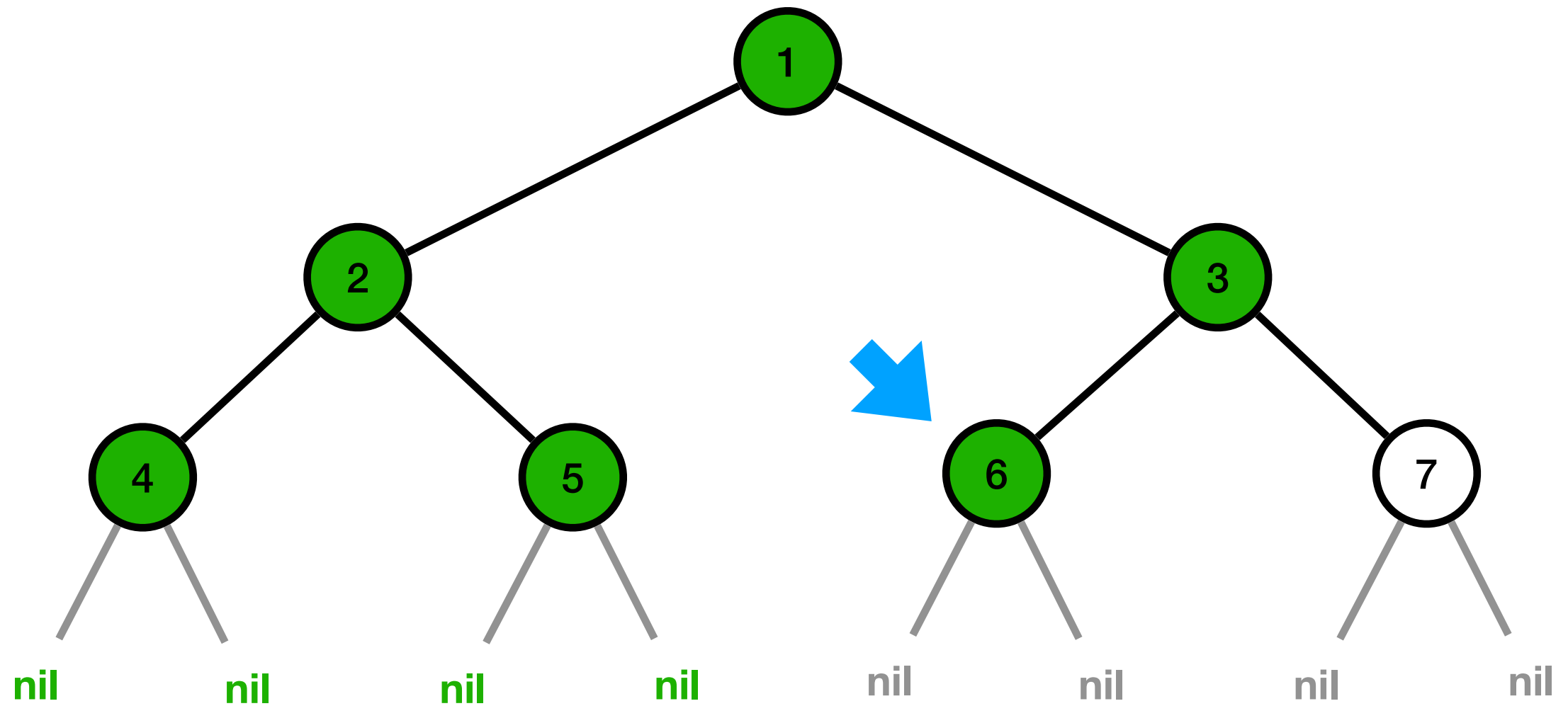
Parcours préfixe



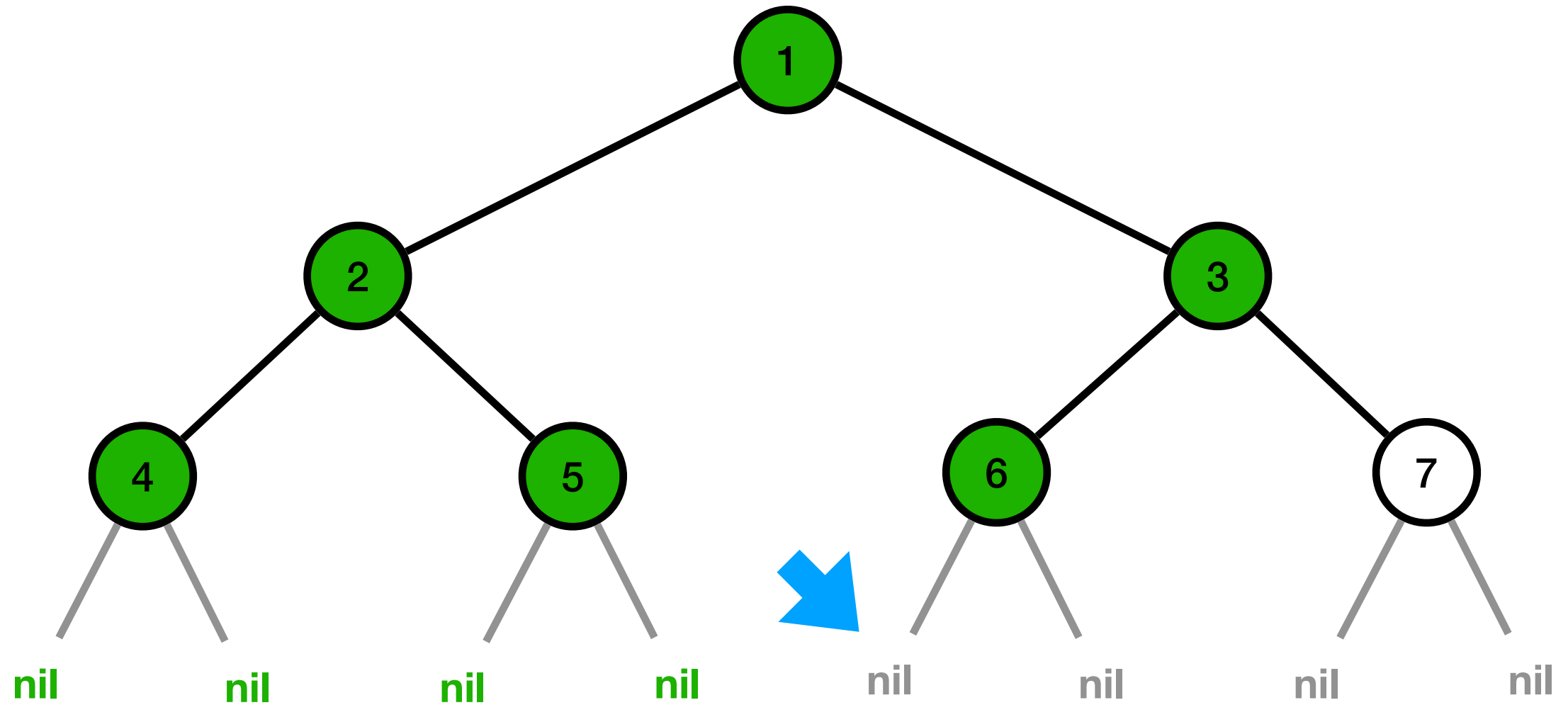
Parcours préfixe



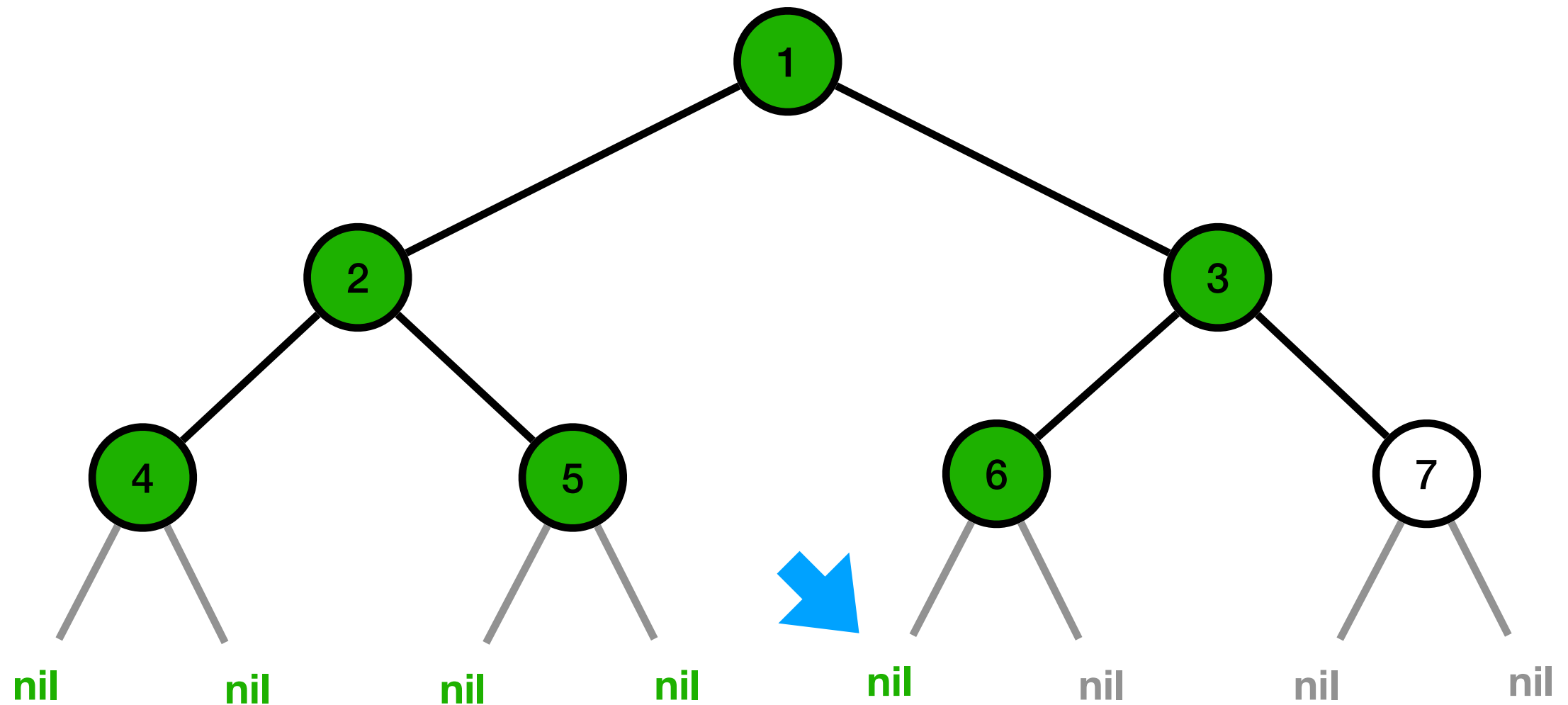
Parcours préfixe



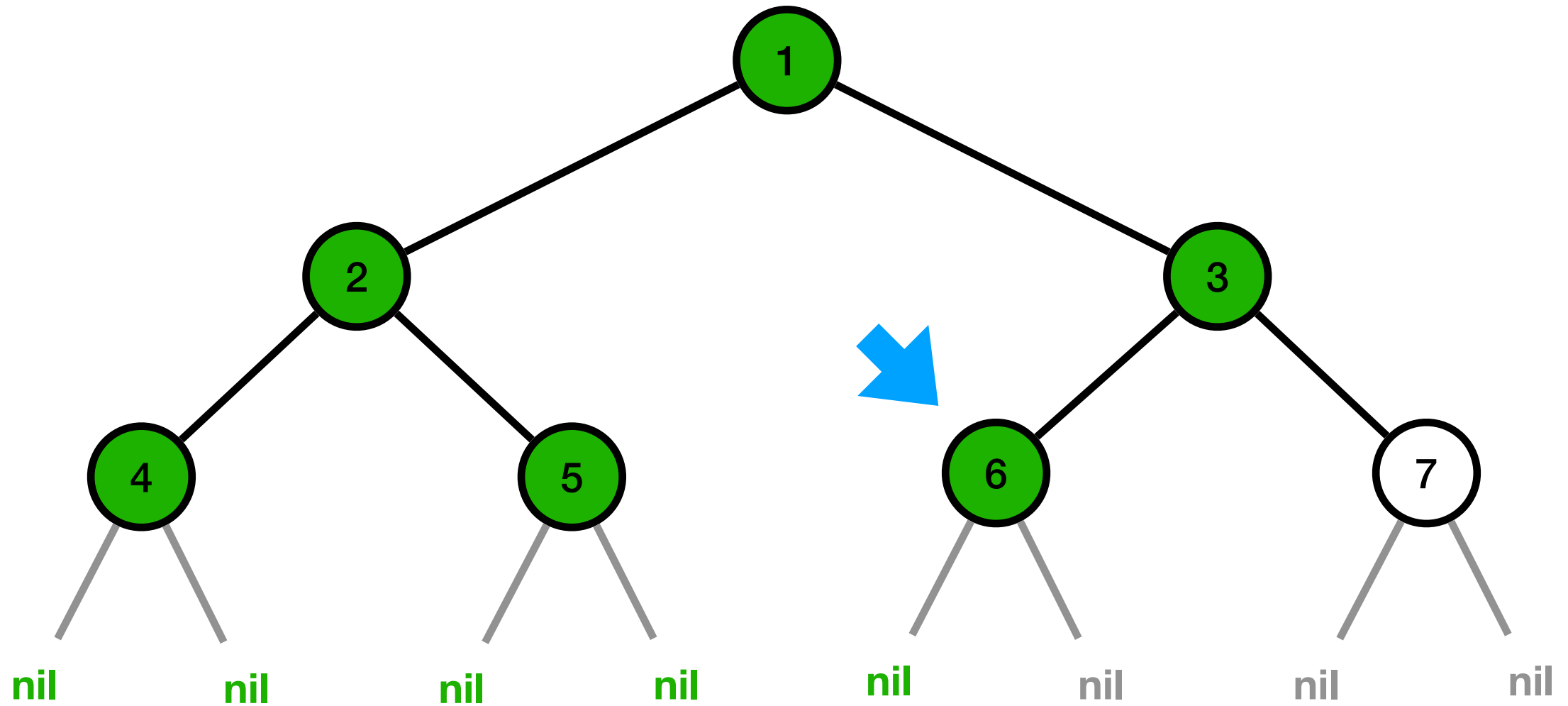
Parcours préfixe



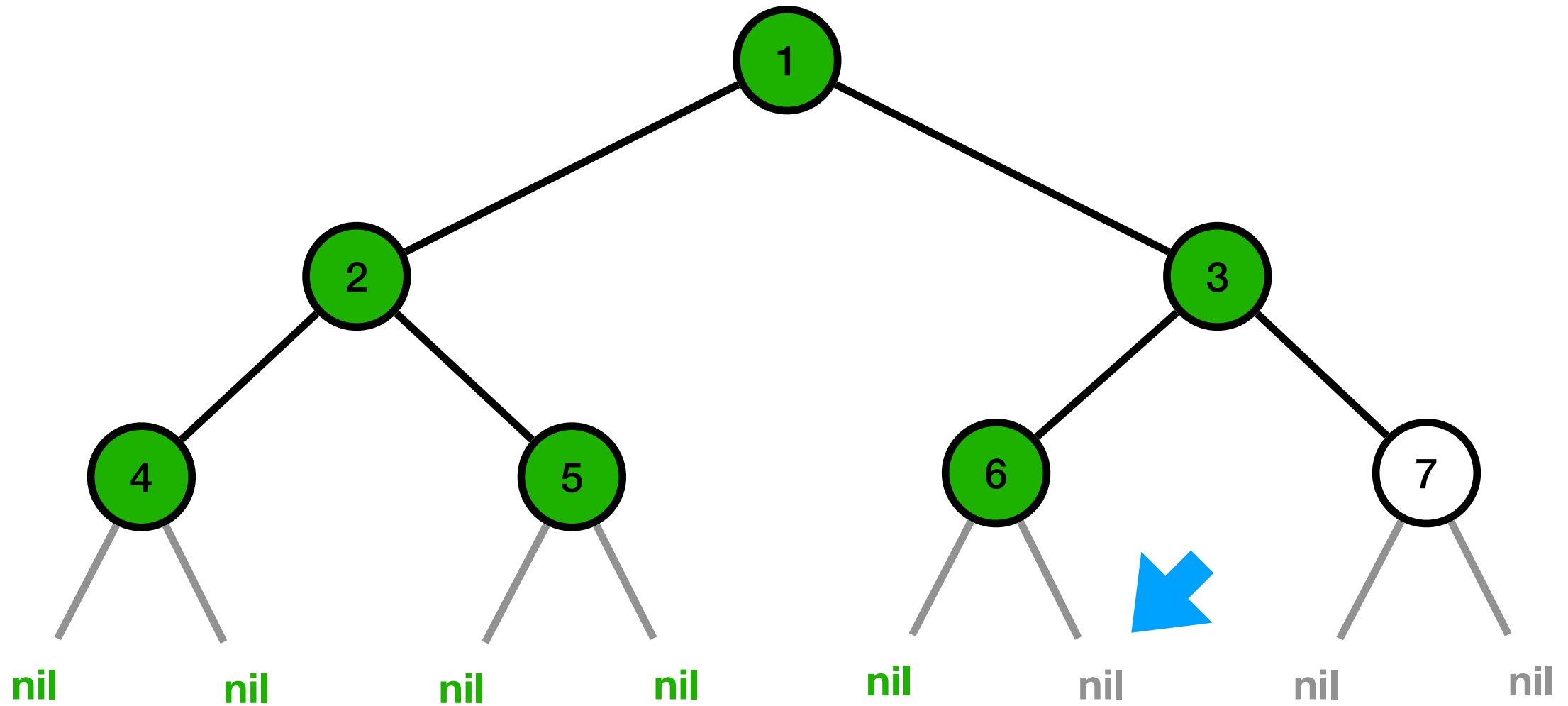
Parcours préfixe



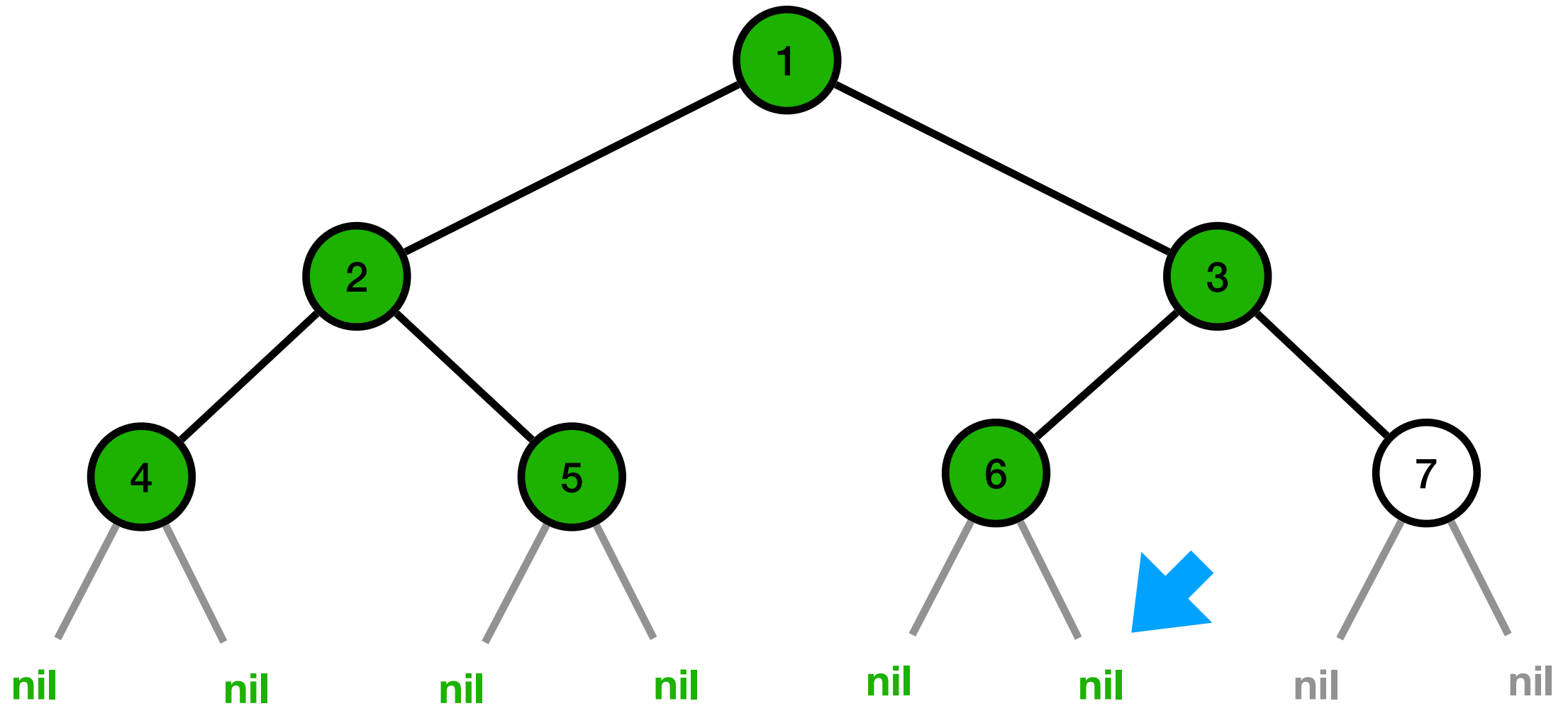
Parcours préfixe



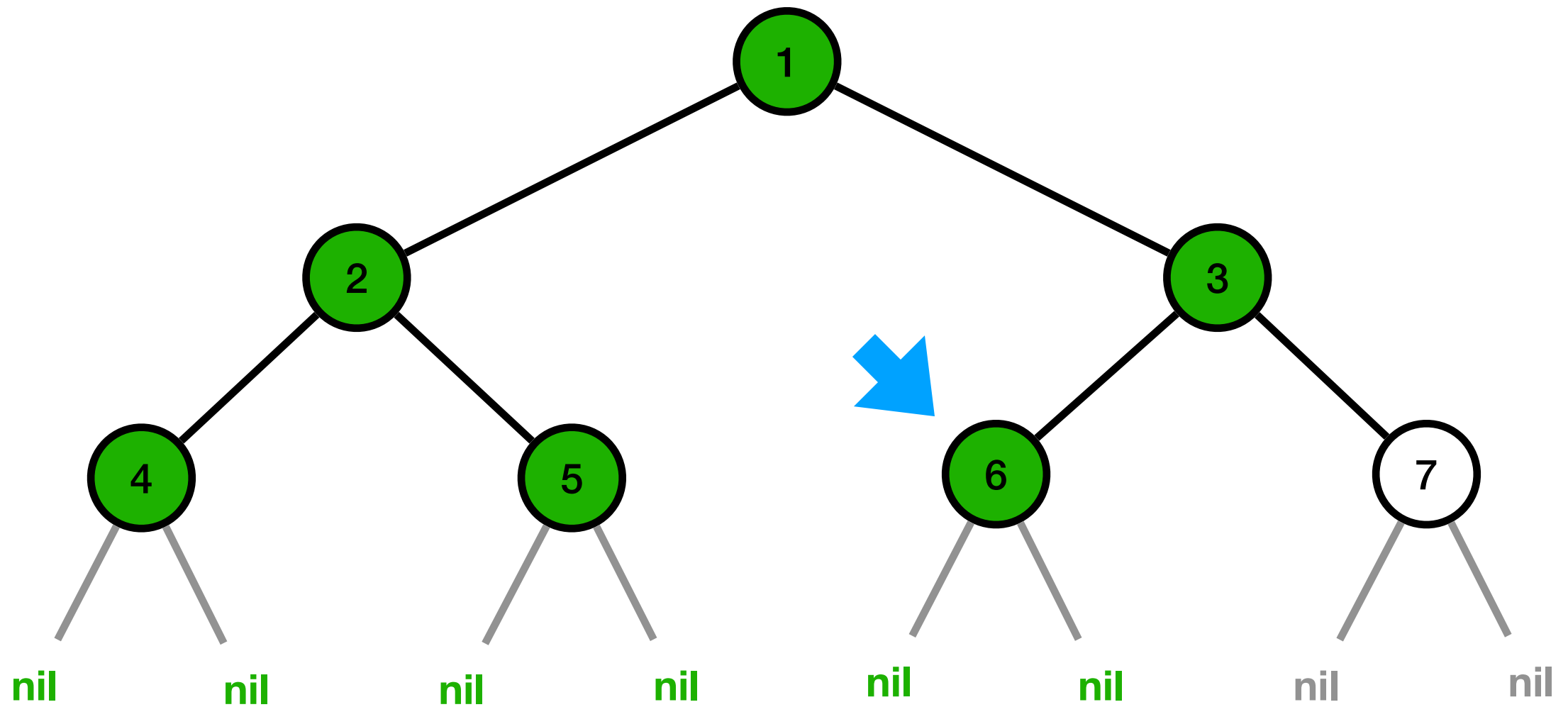
Parcours préfixe



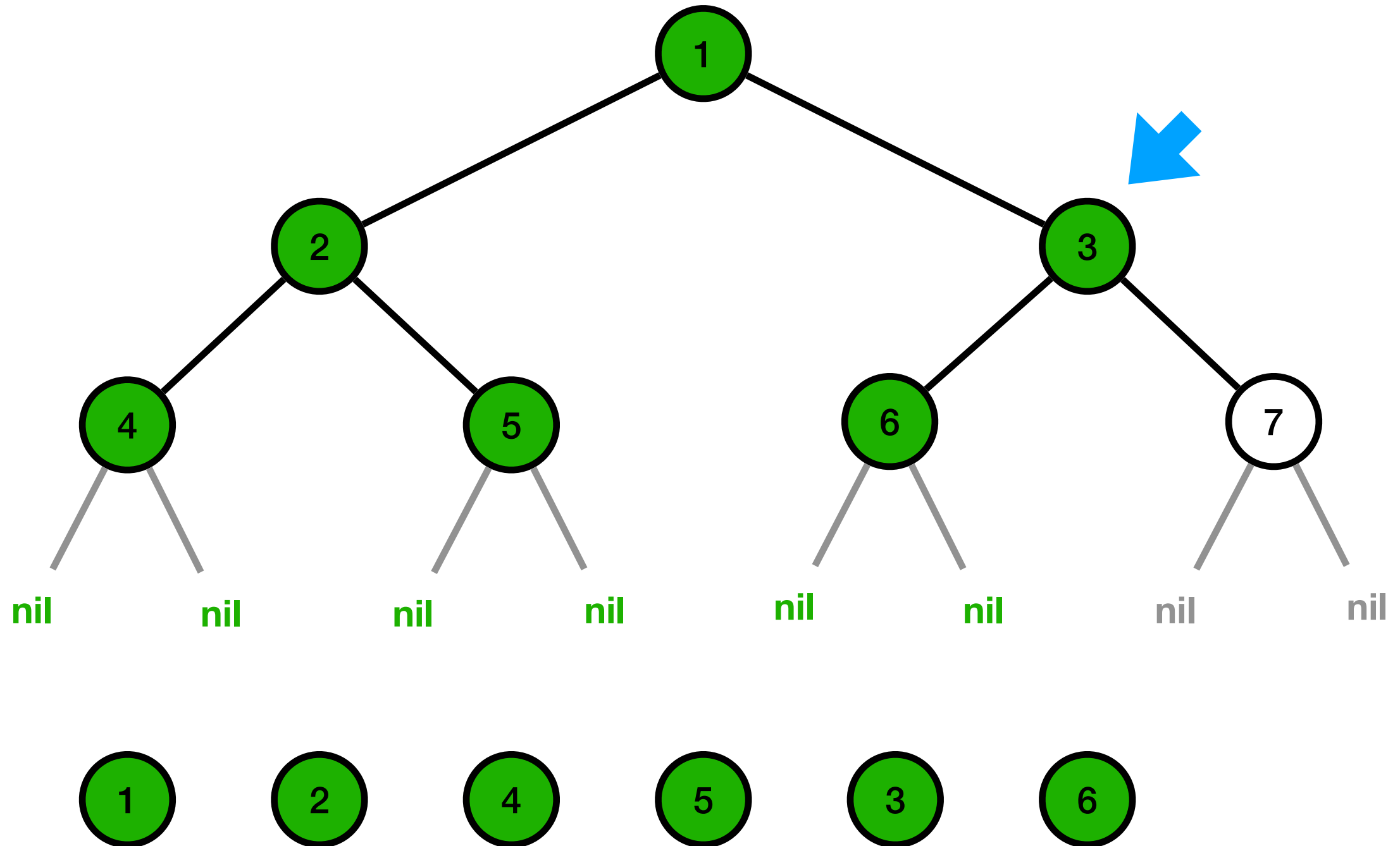
Parcours préfixe



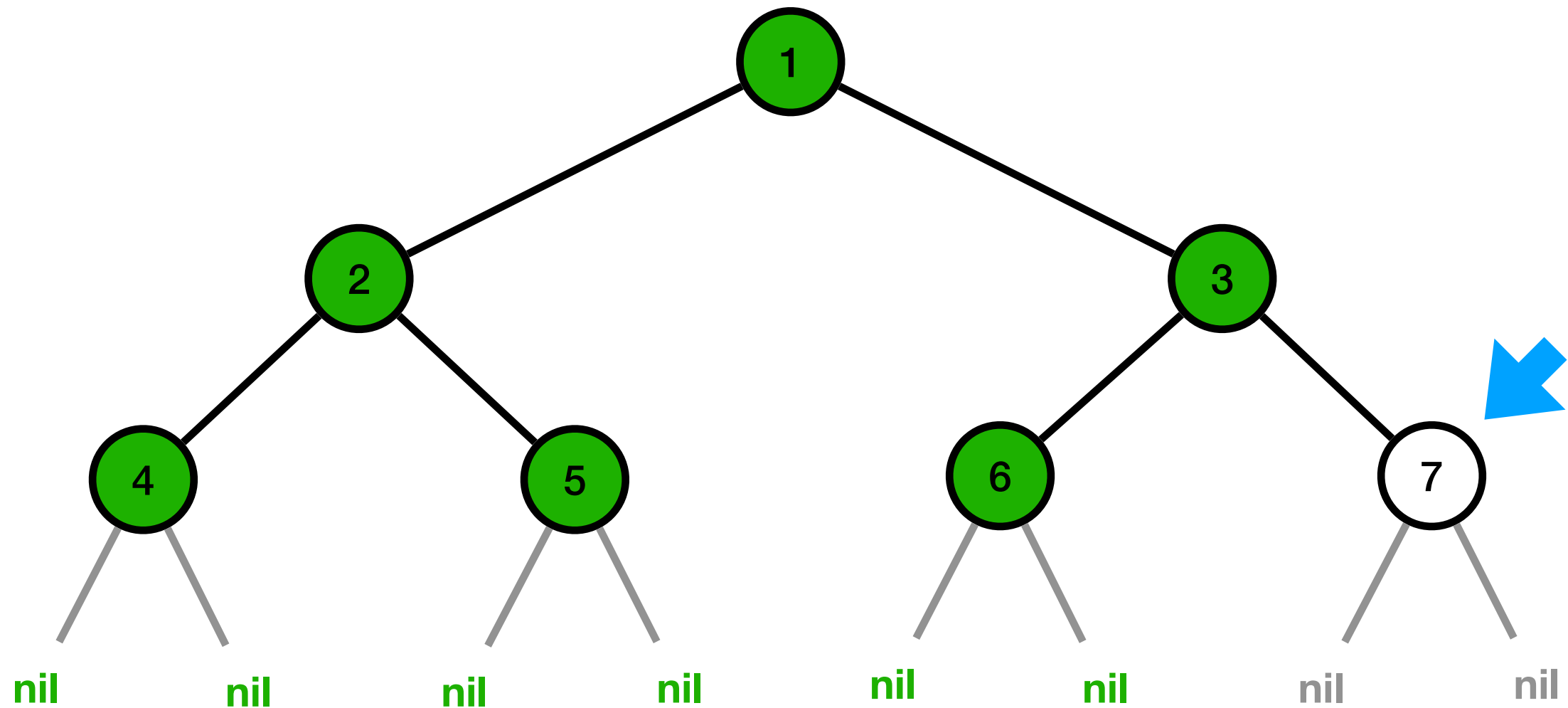
Parcours préfixe



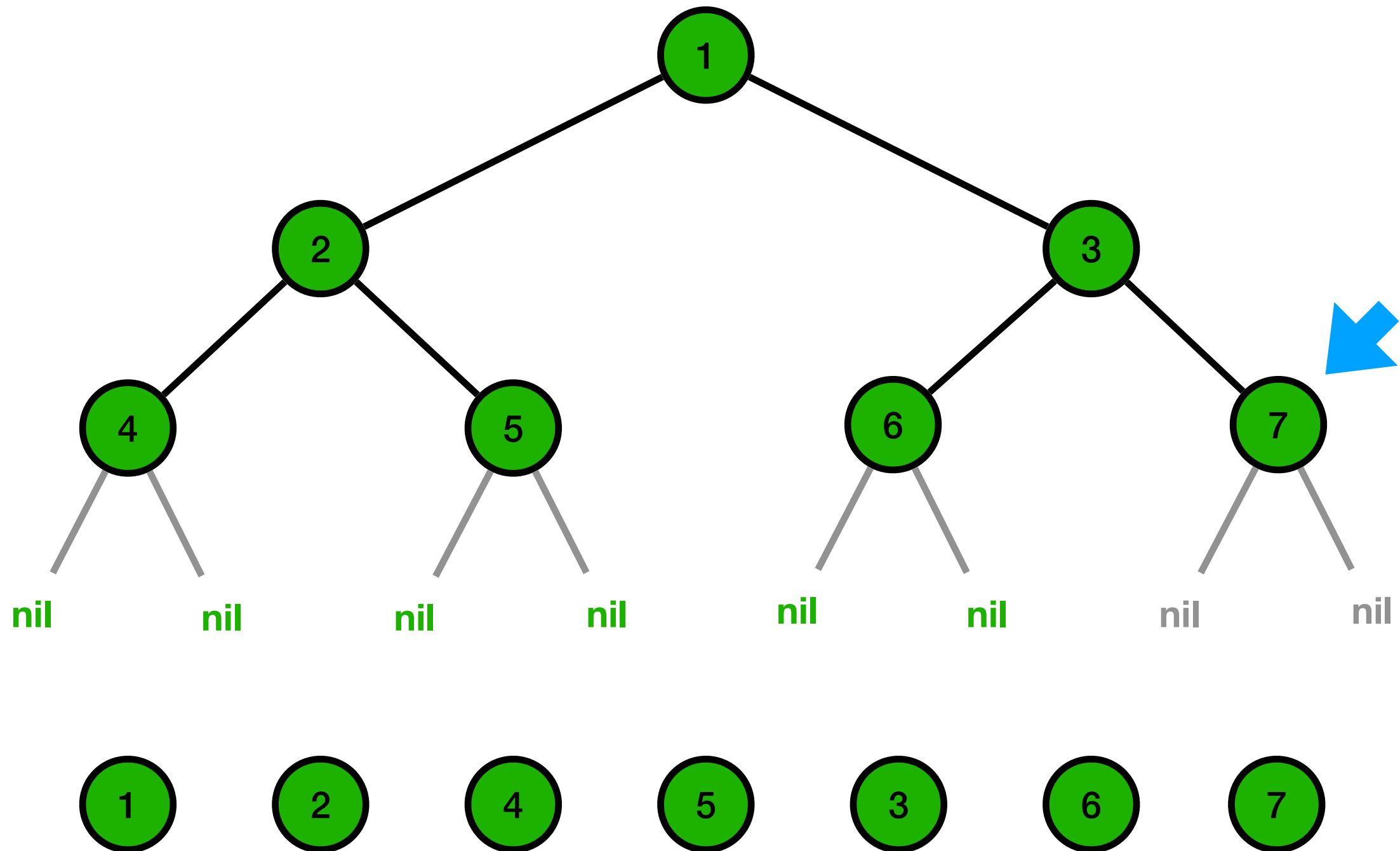
Parcours préfixe



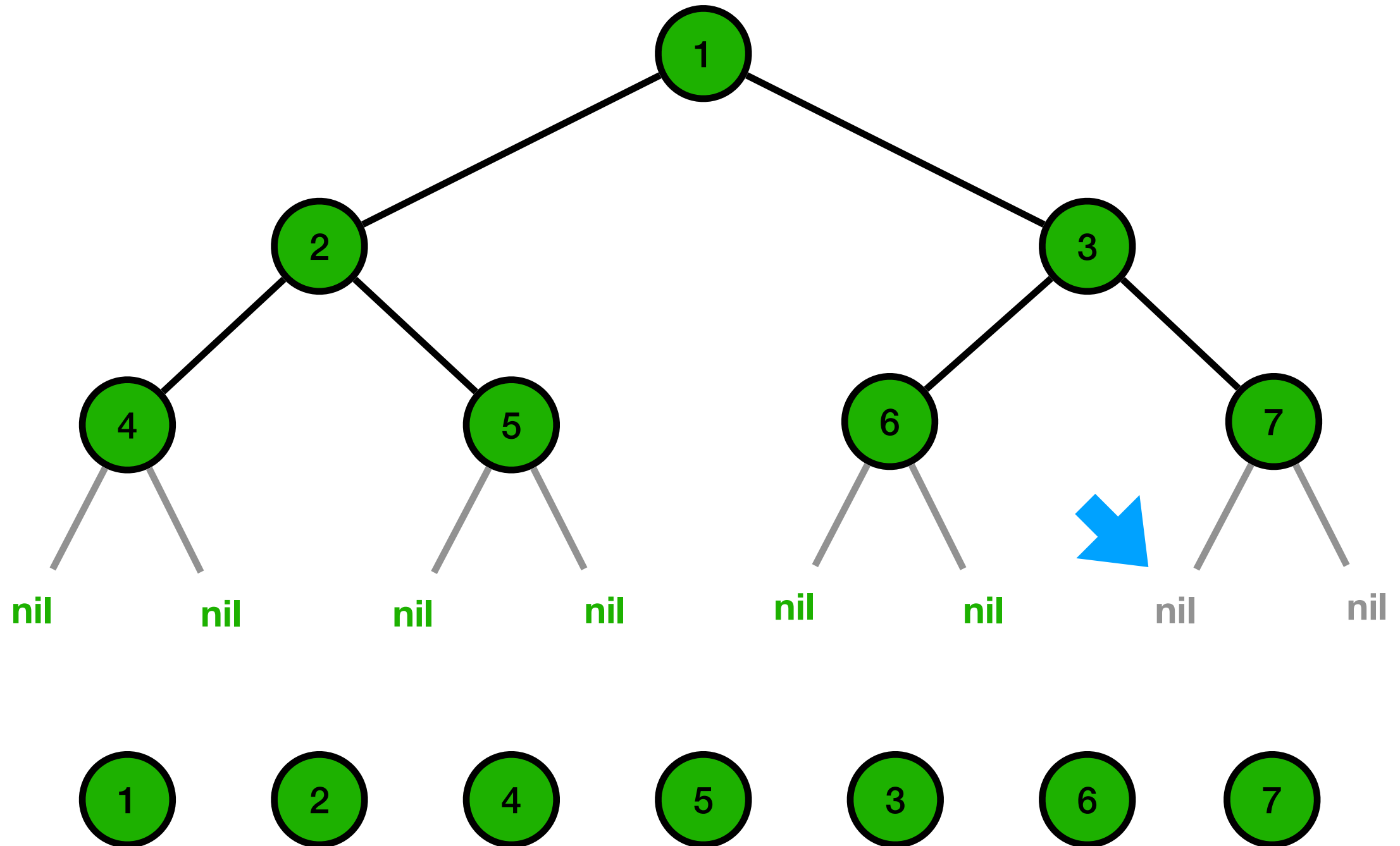
Parcours préfixe



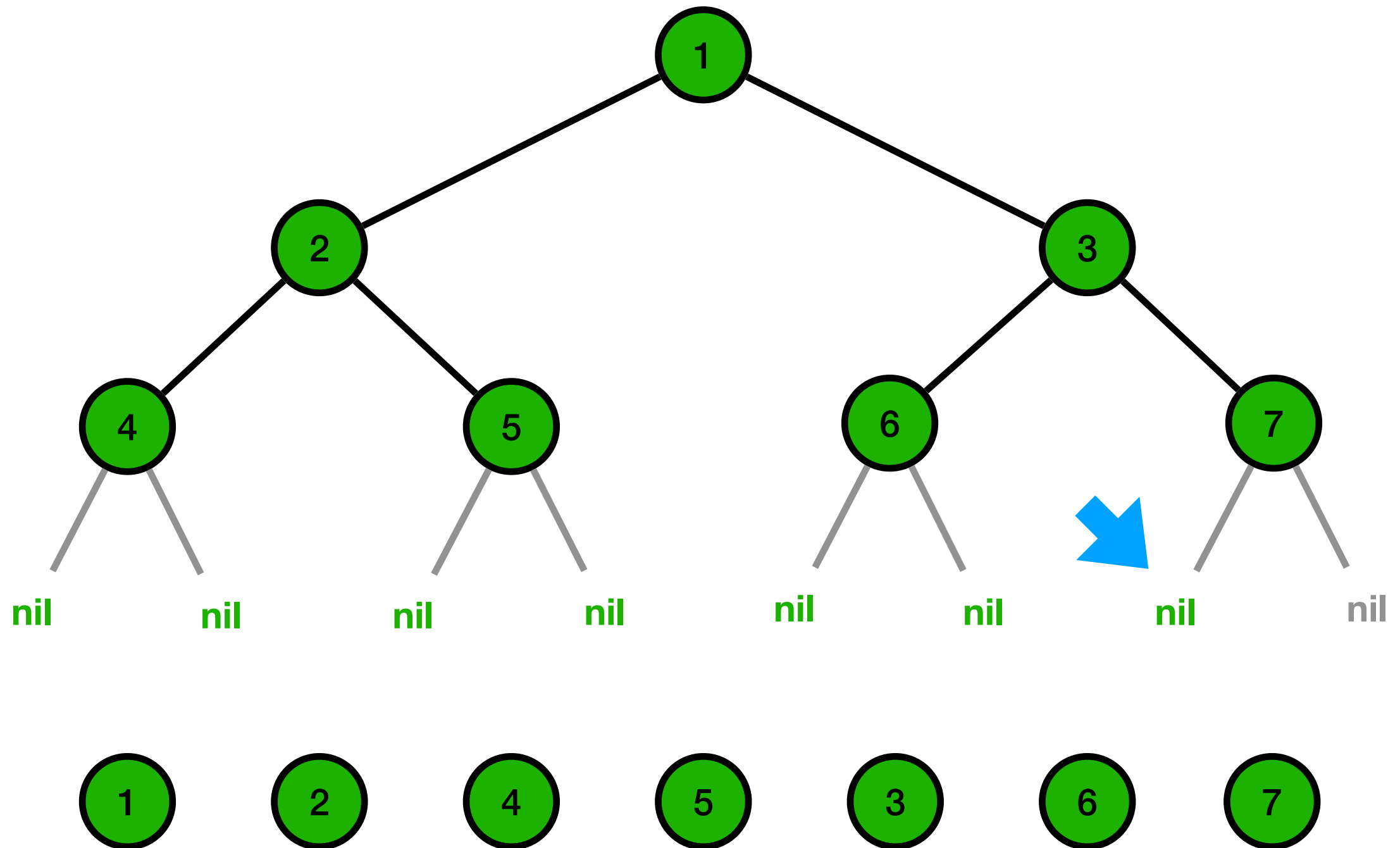
Parcours préfixe



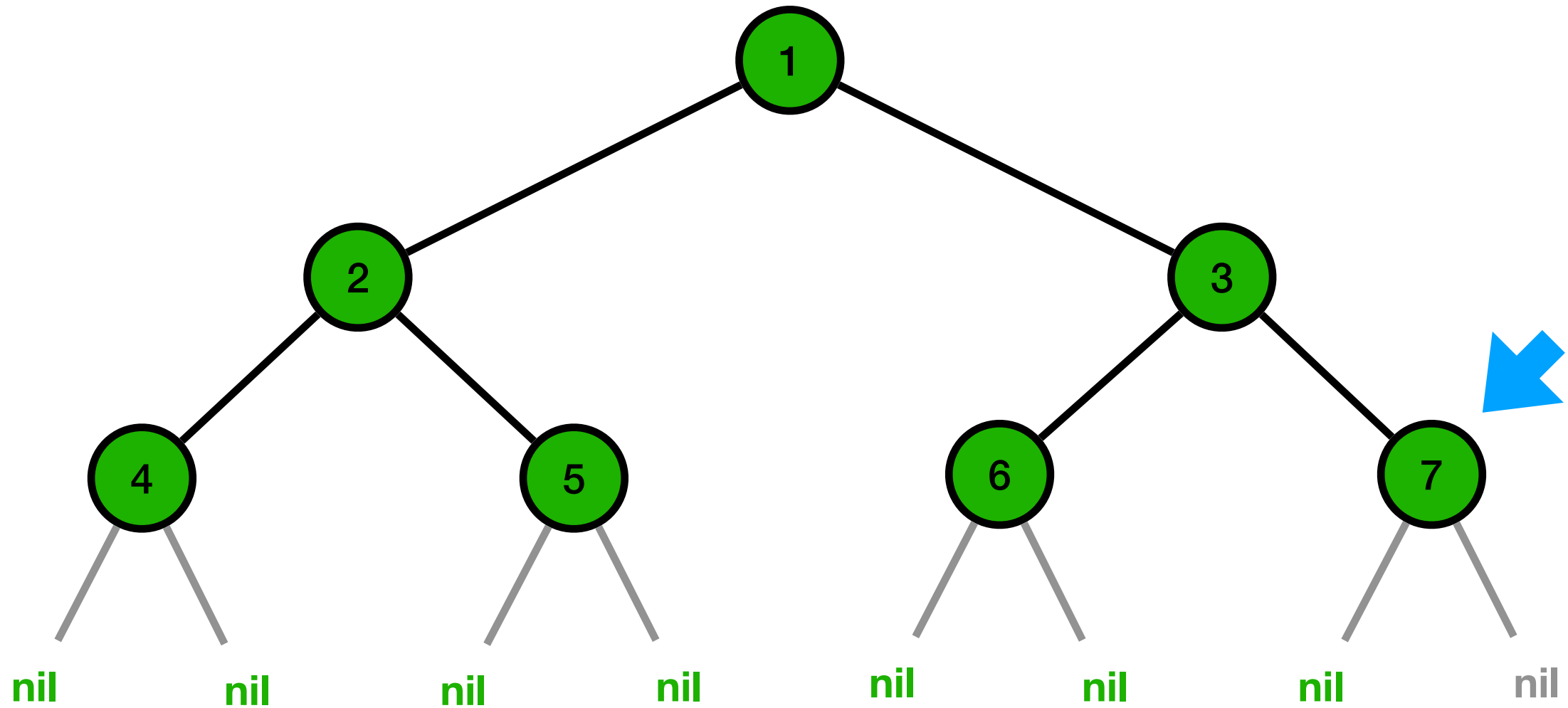
Parcours préfixe



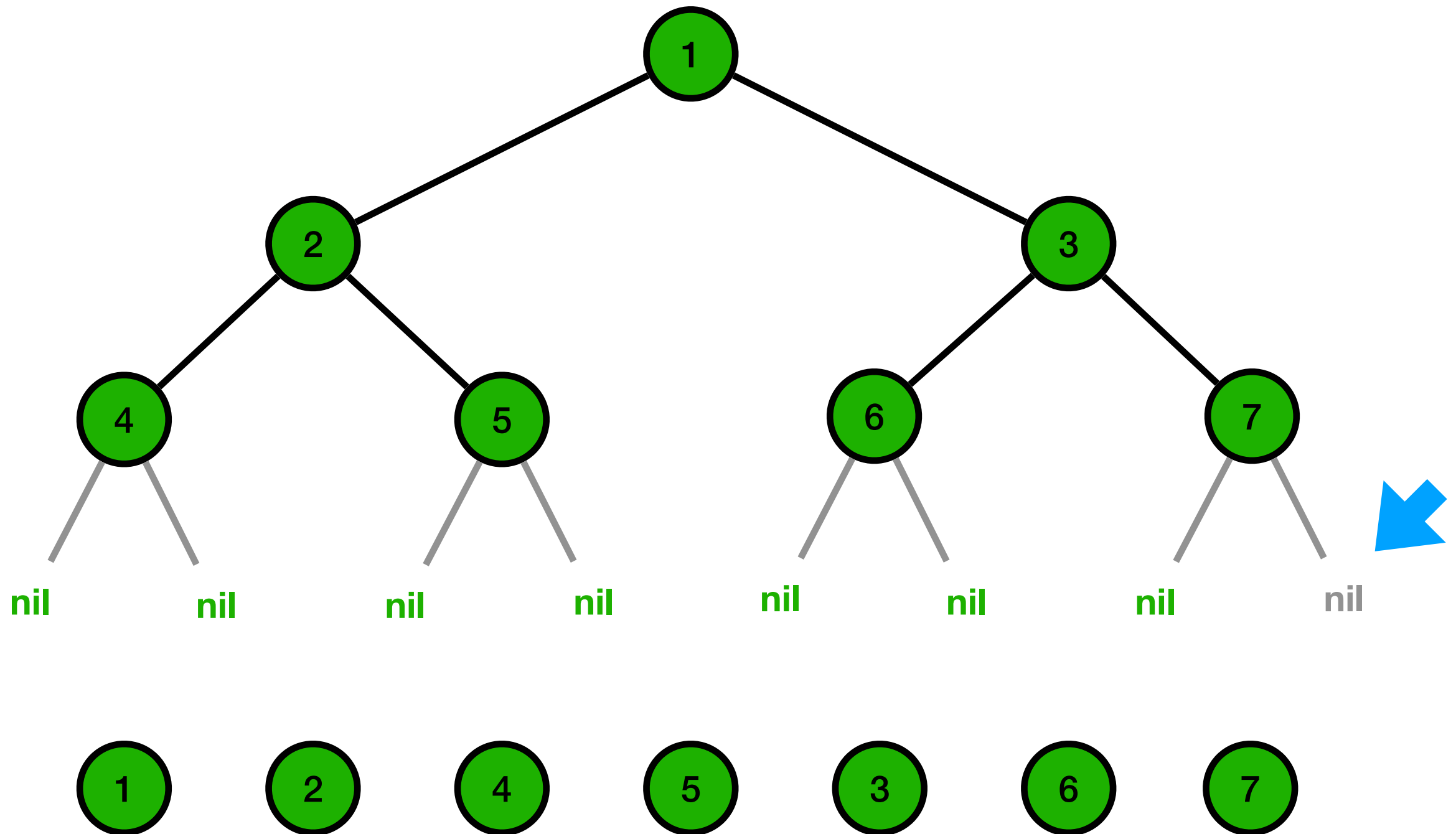
Parcours préfixe



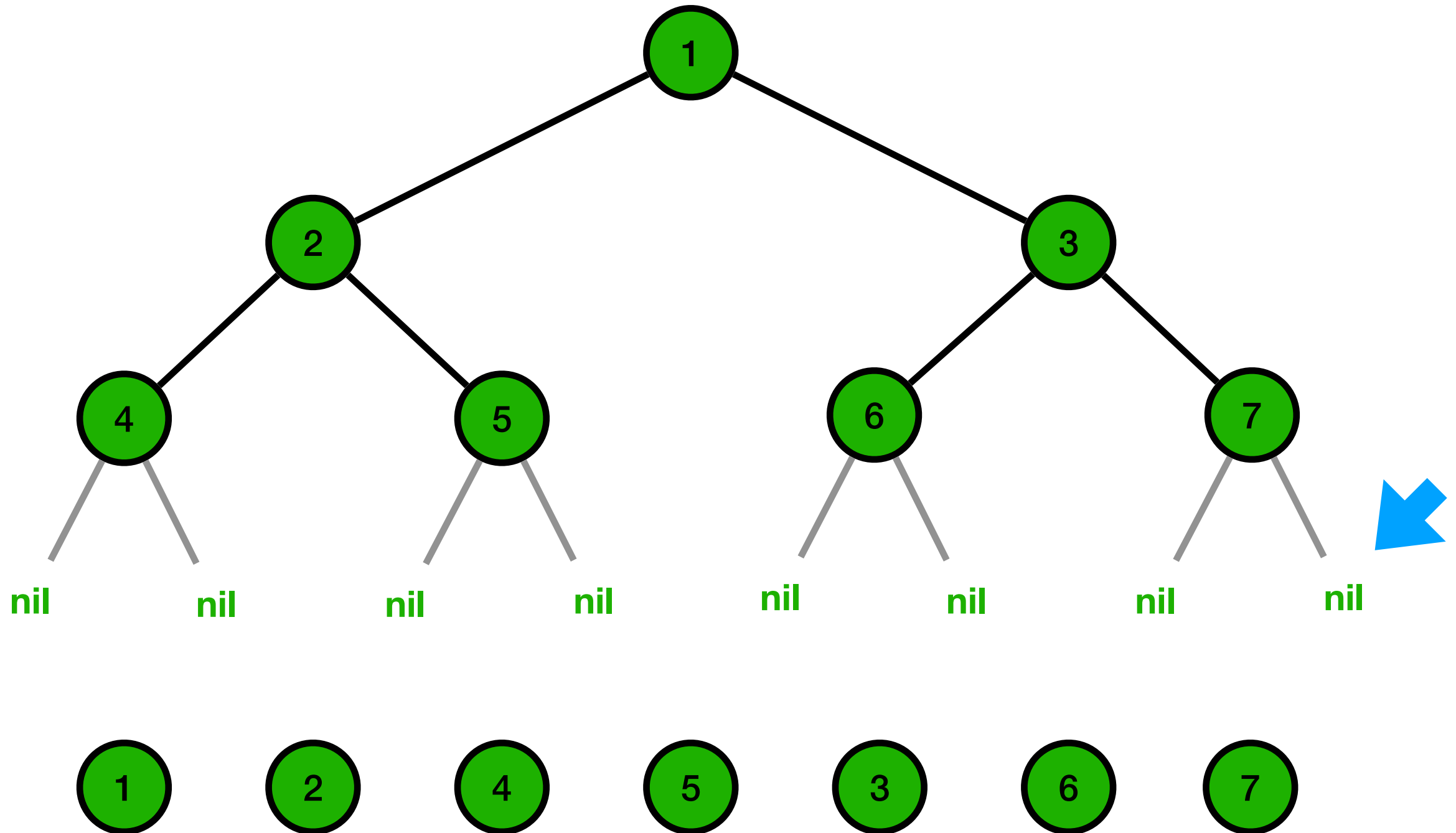
Parcours préfixe



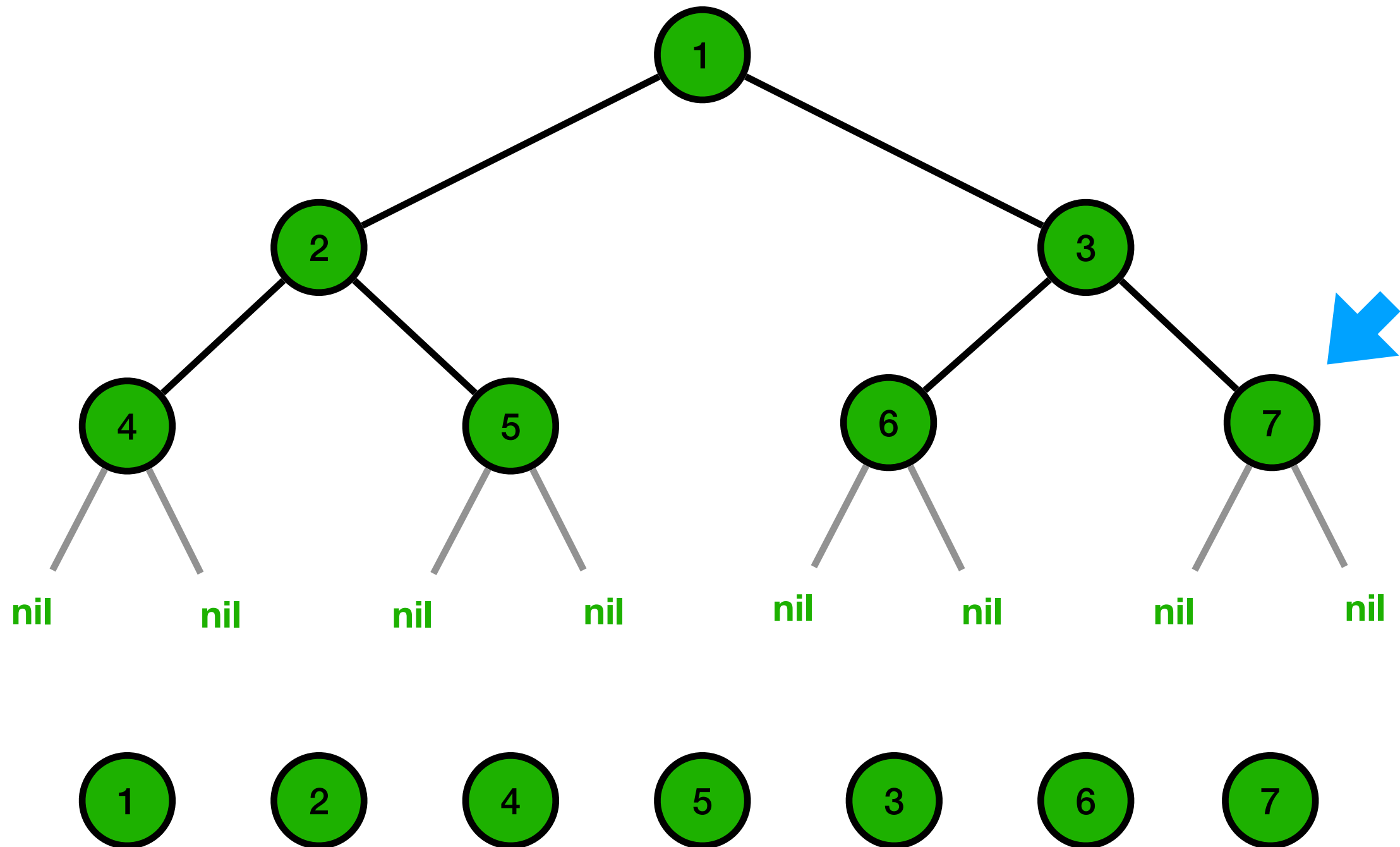
Parcours préfixe



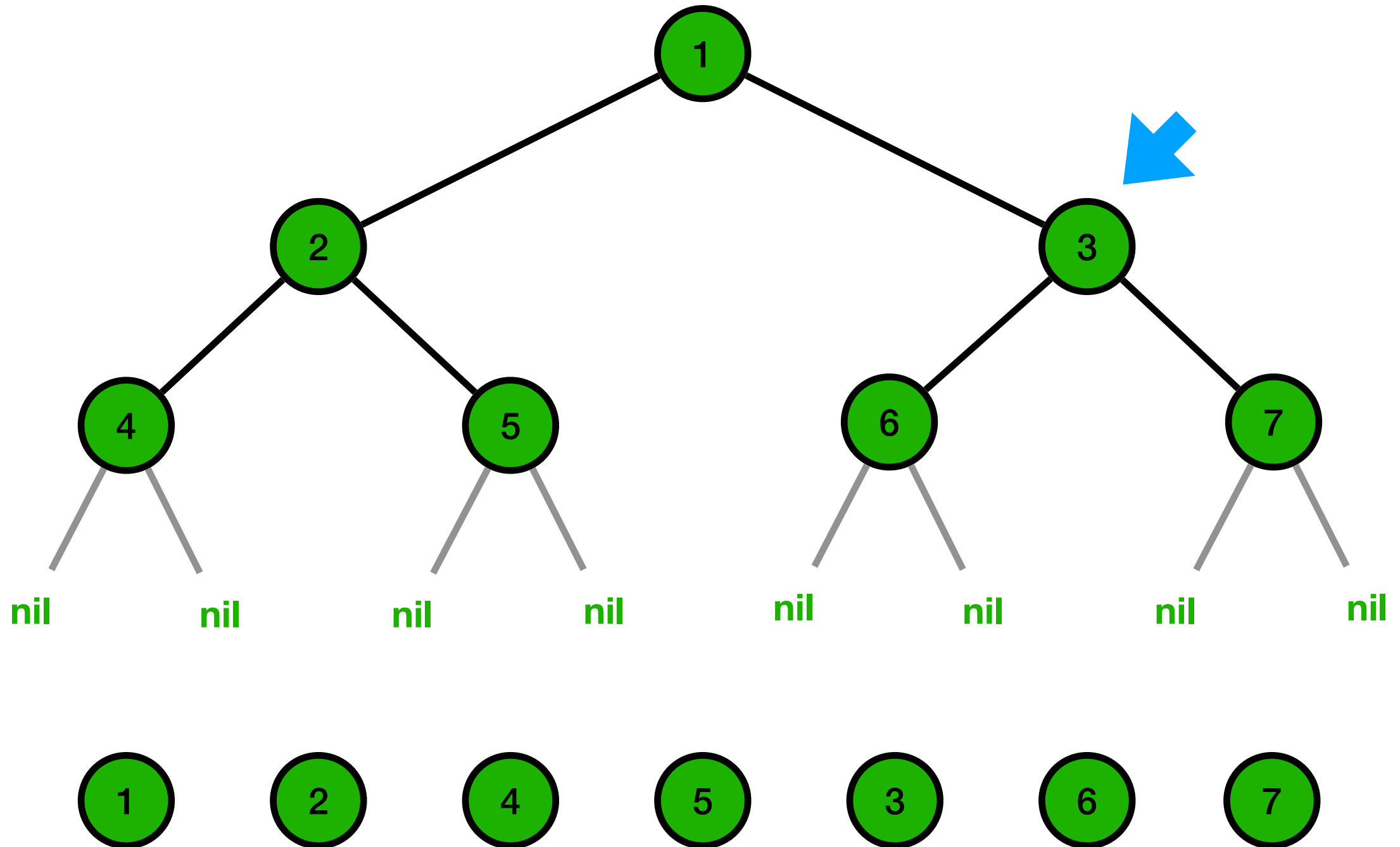
Parcours préfixe



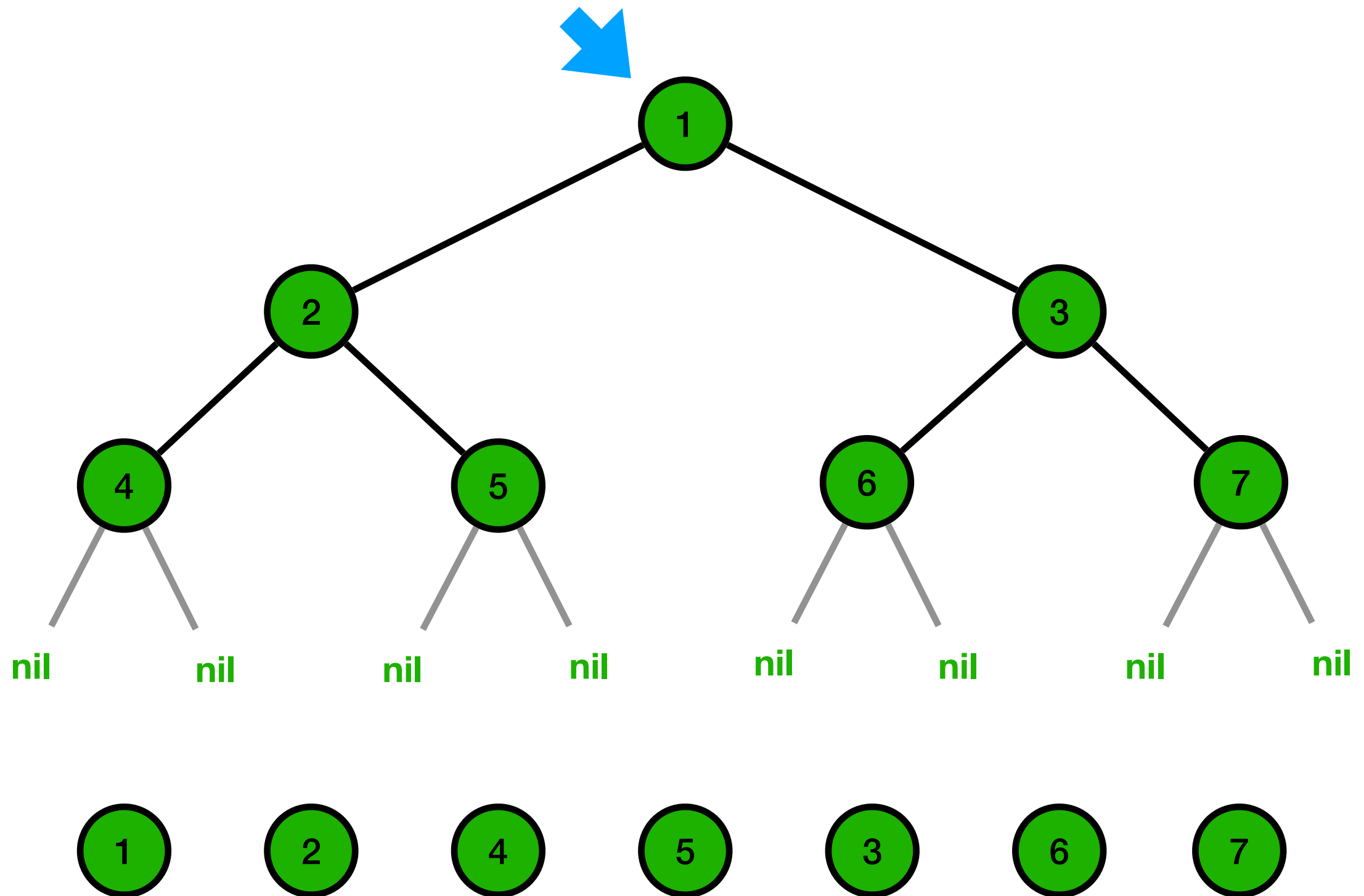
Parcours préfixe



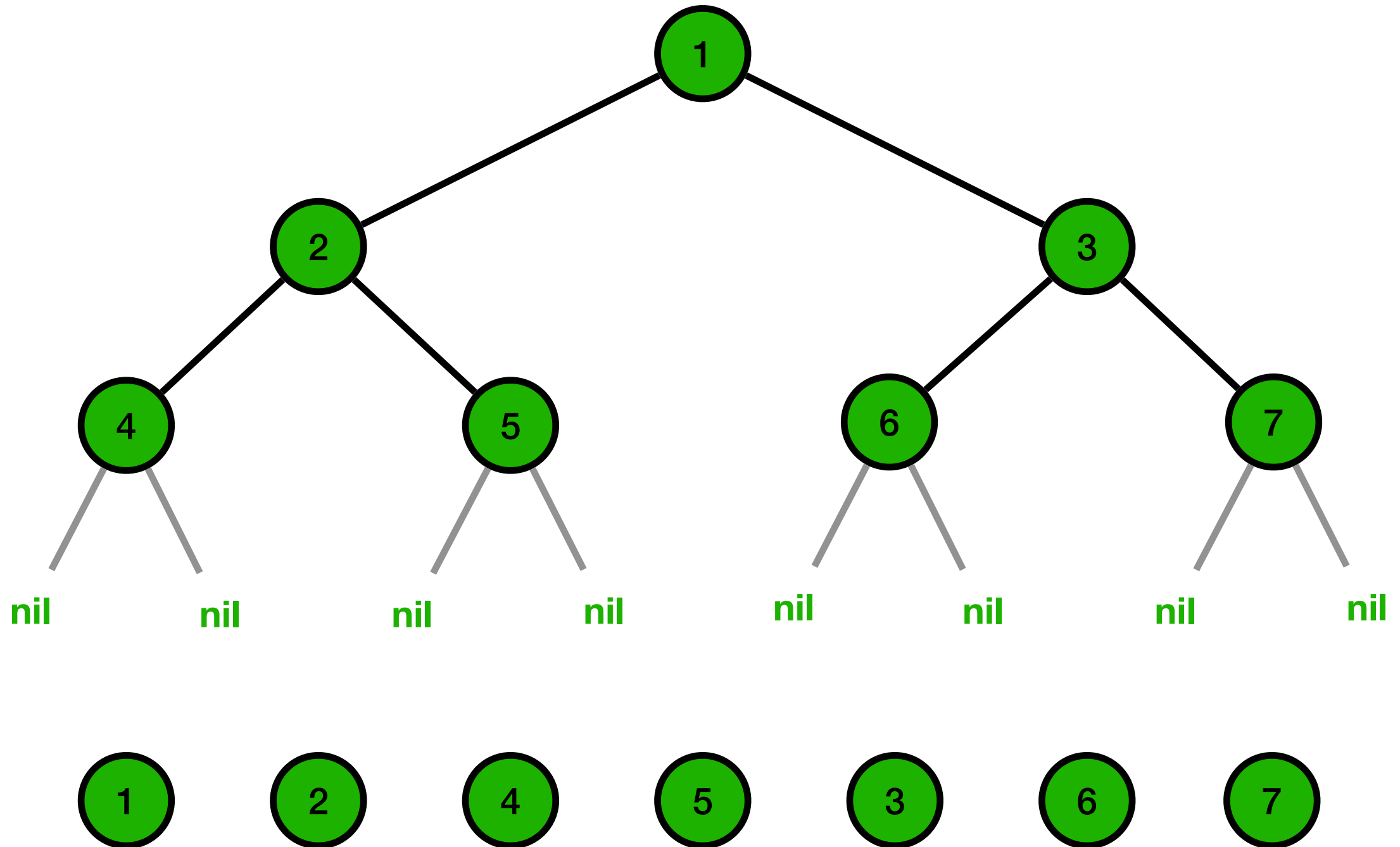
Parcours préfixe



Parcours préfixe



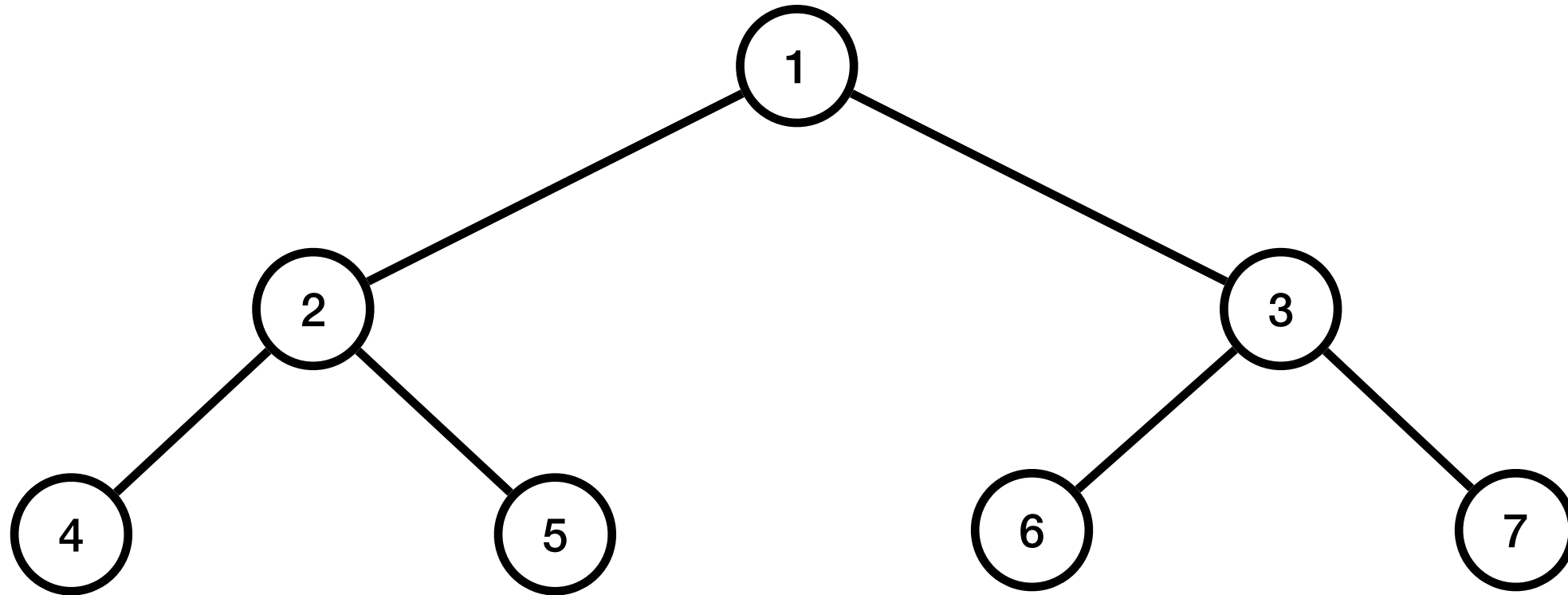
Parcours préfixe



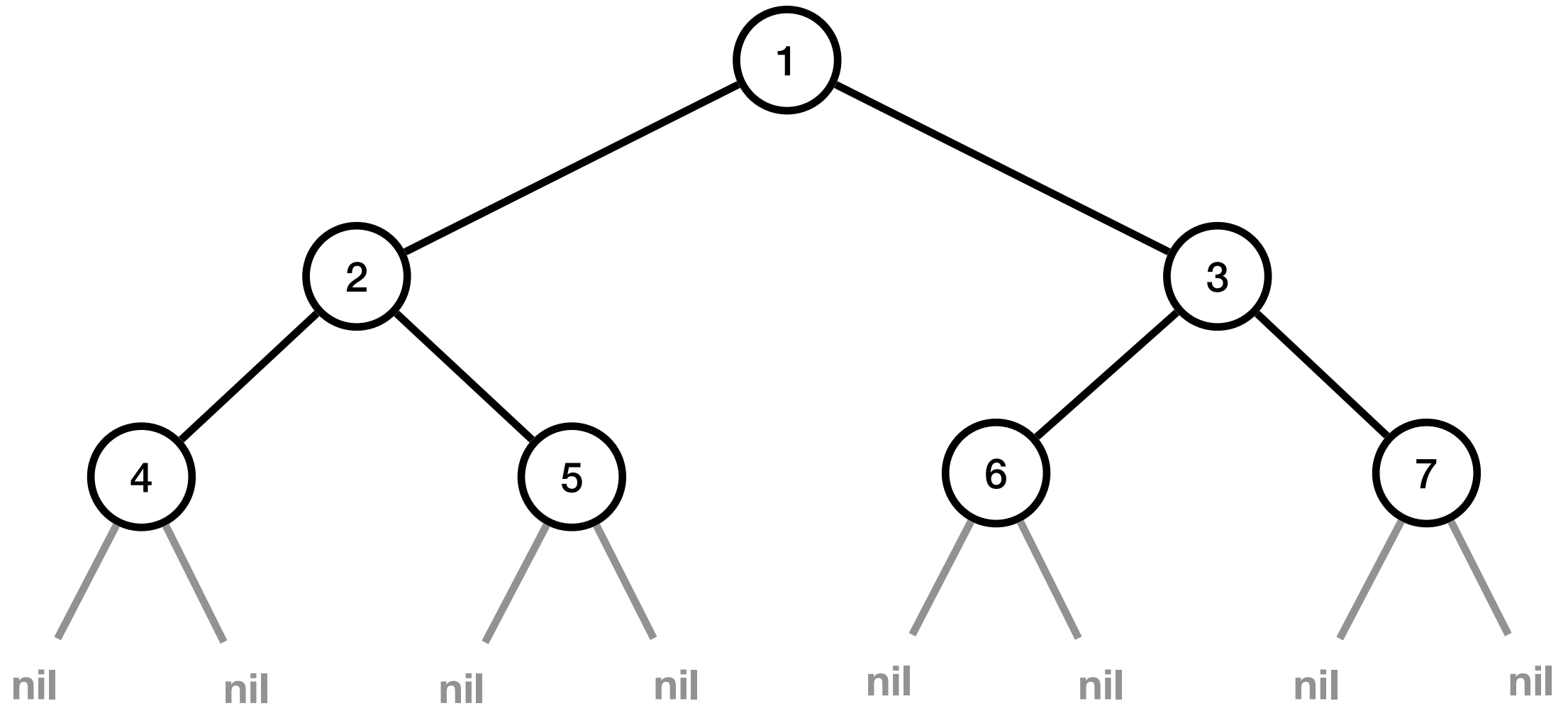
Parcours postfixe (ou suffixe)

```
def parcours_postfixe(arbre):  
    if not est_vide(arbre):  
        parcours_postfixe(gauche(arbre))  
        parcours_postfixe(droite(arbre))  
        print(valeur(arbre))
```

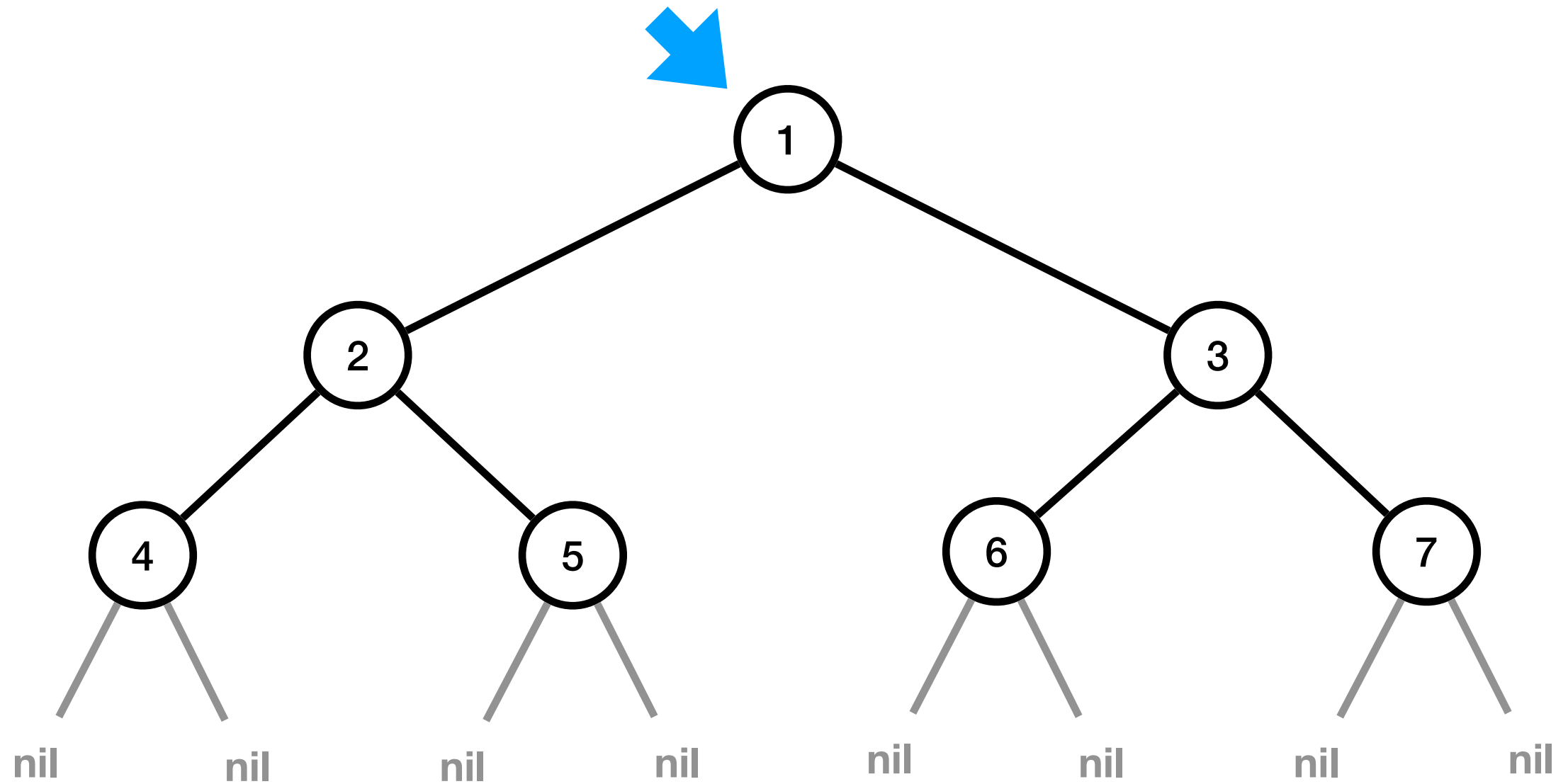
Parcours postfixe



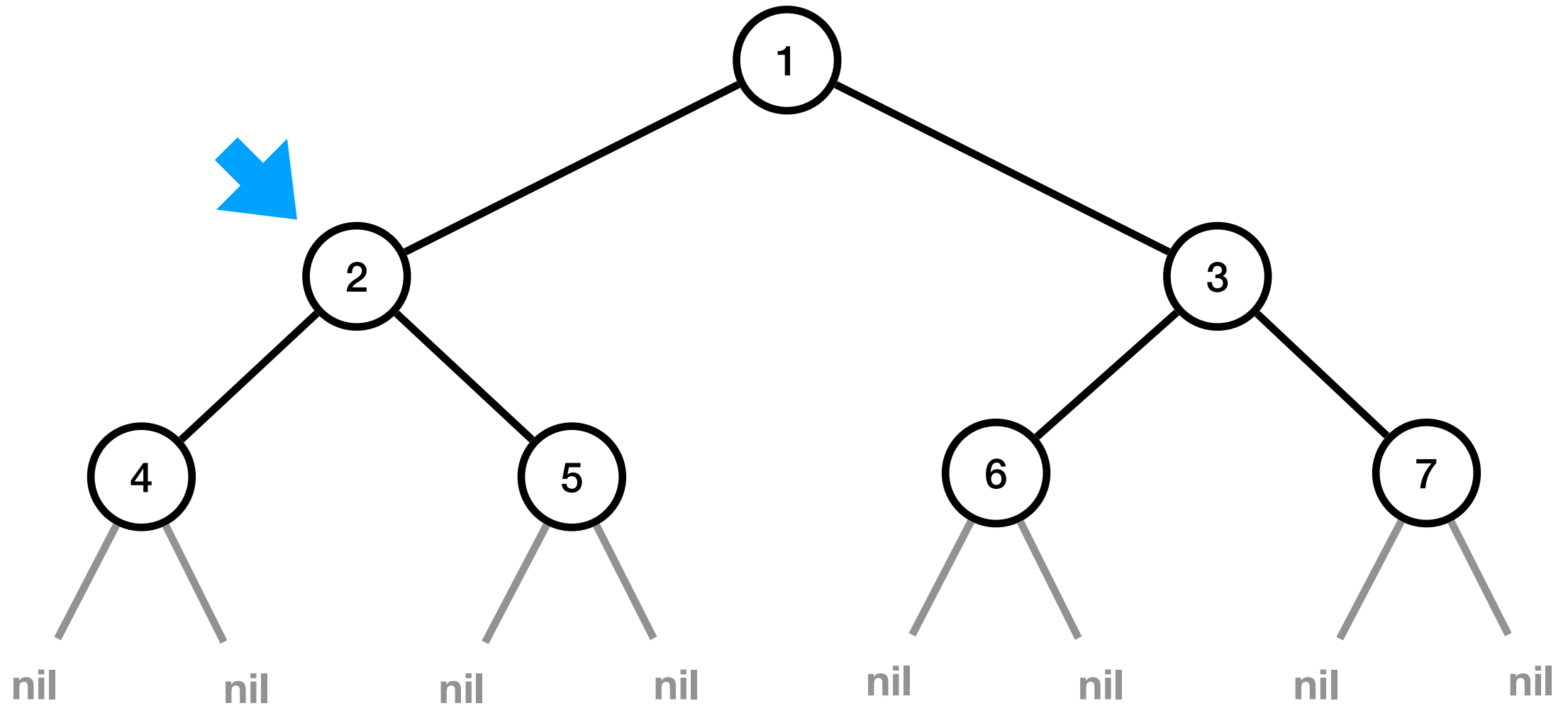
Parcours postfixe



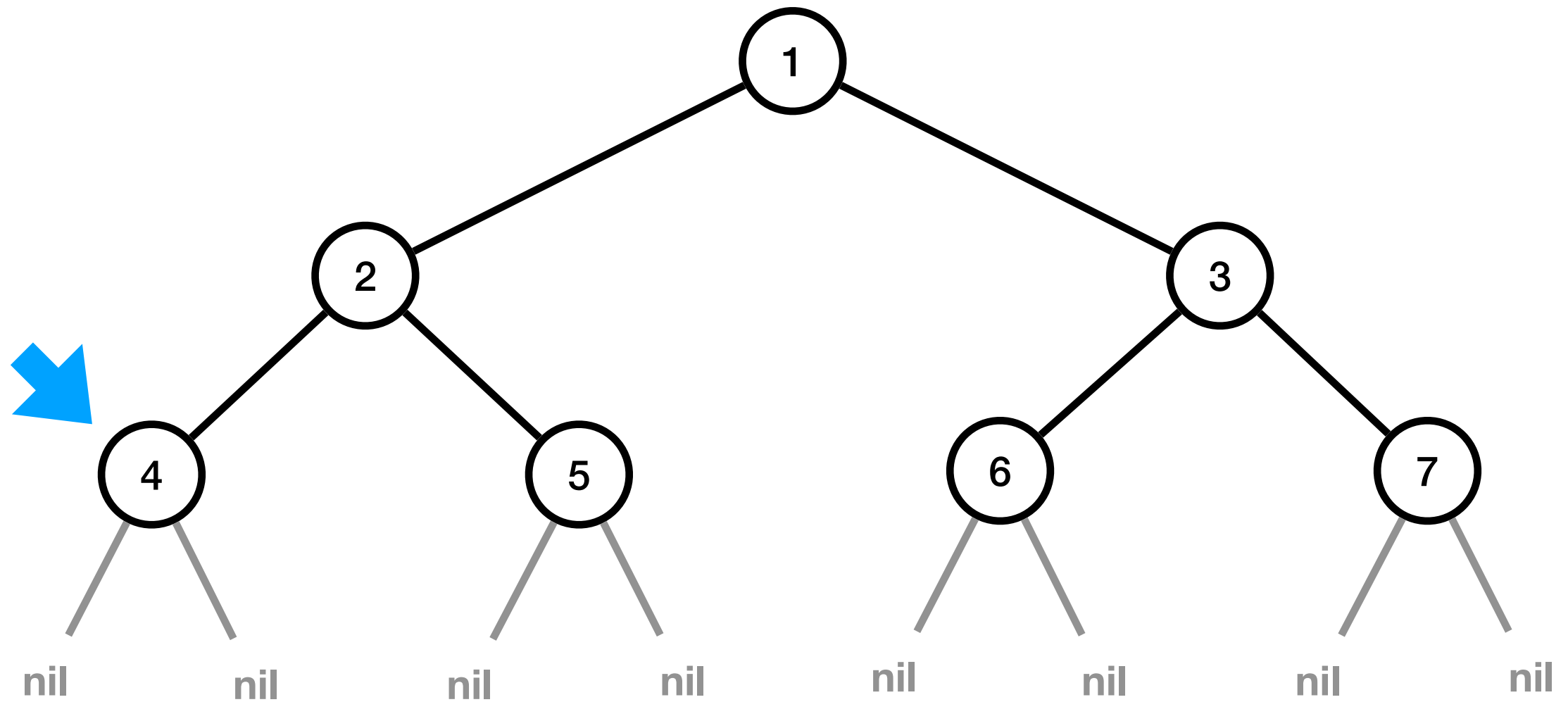
Parcours postfixe



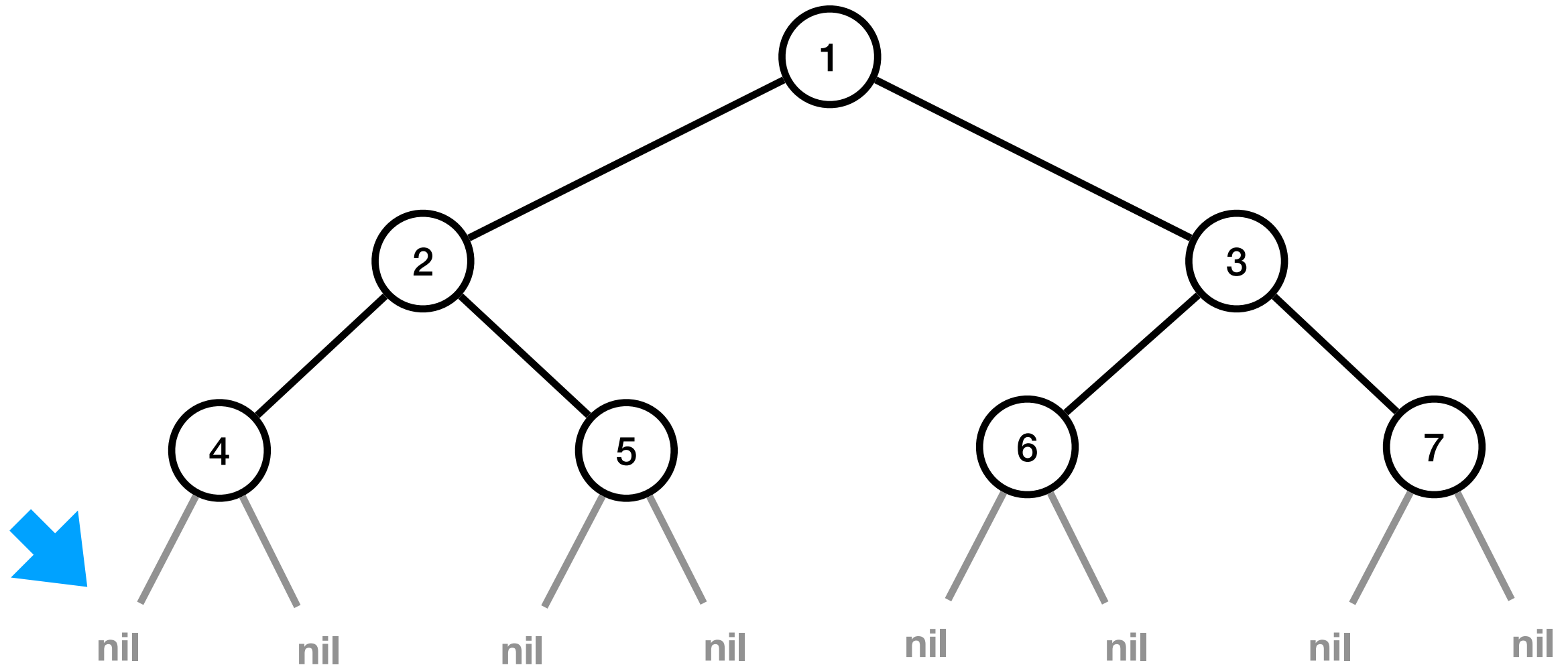
Parcours postfixe



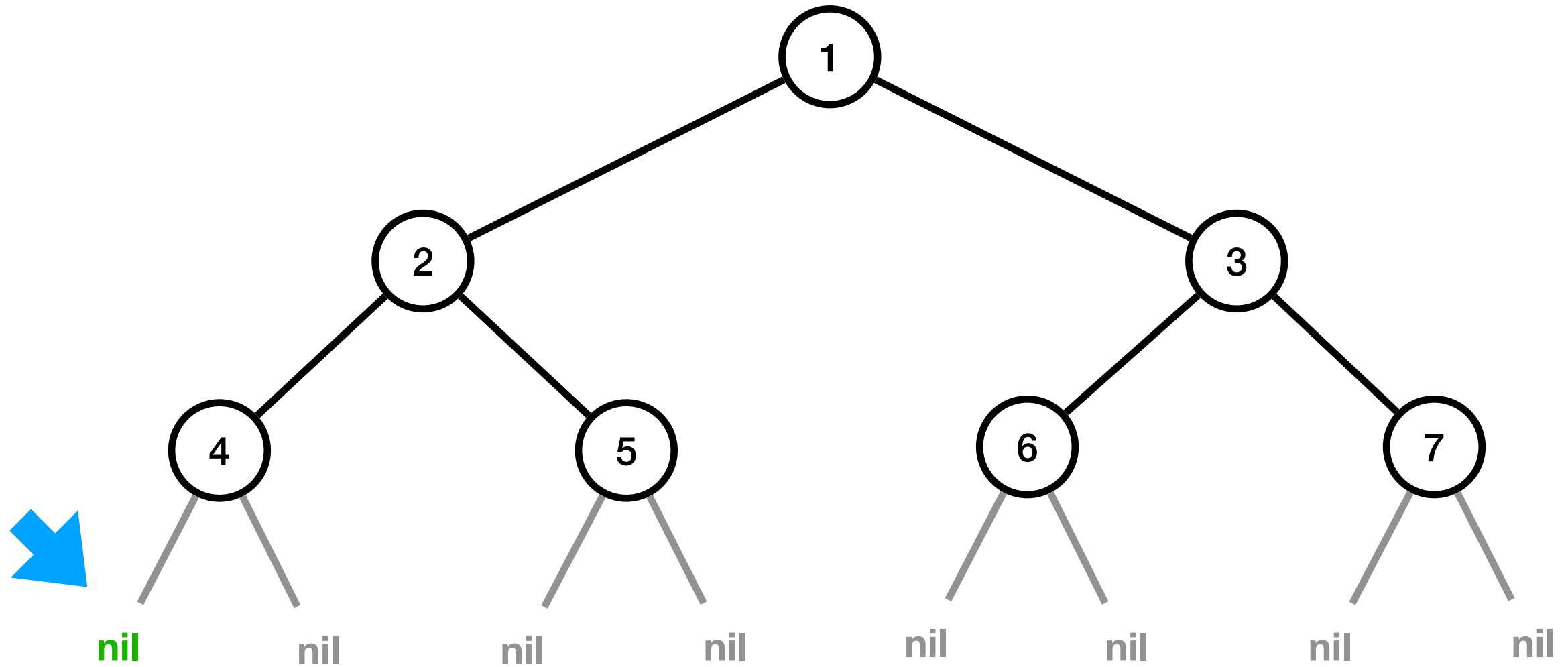
Parcours postfixe



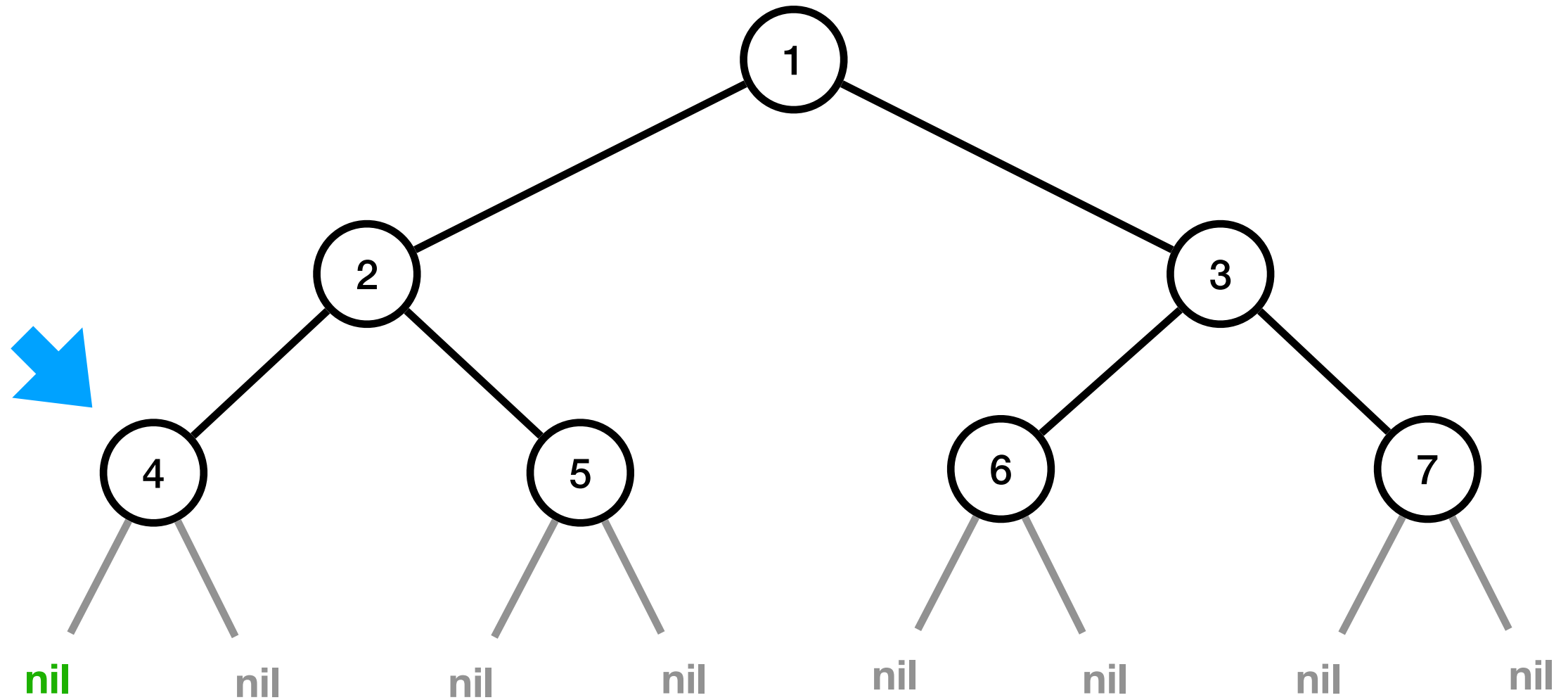
Parcours postfixe



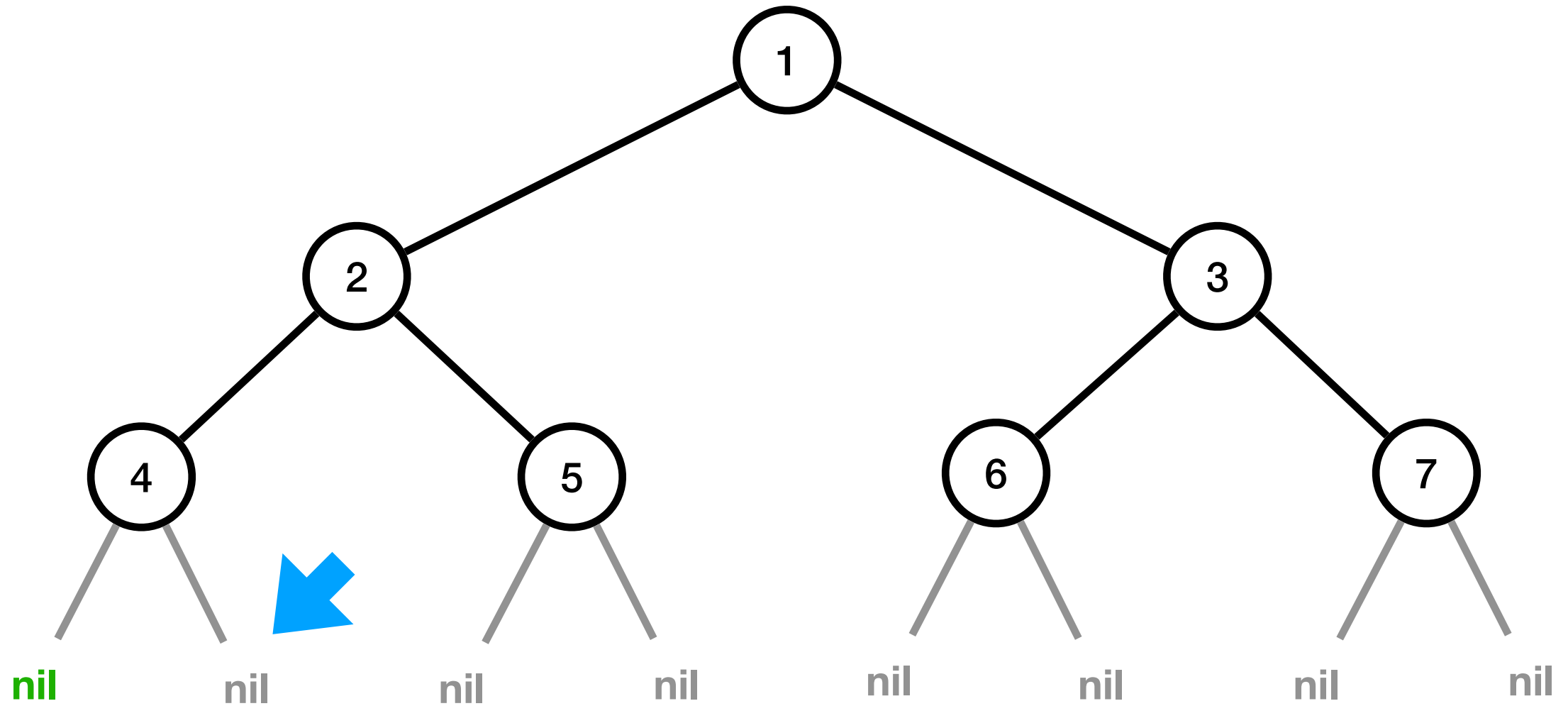
Parcours postfixe



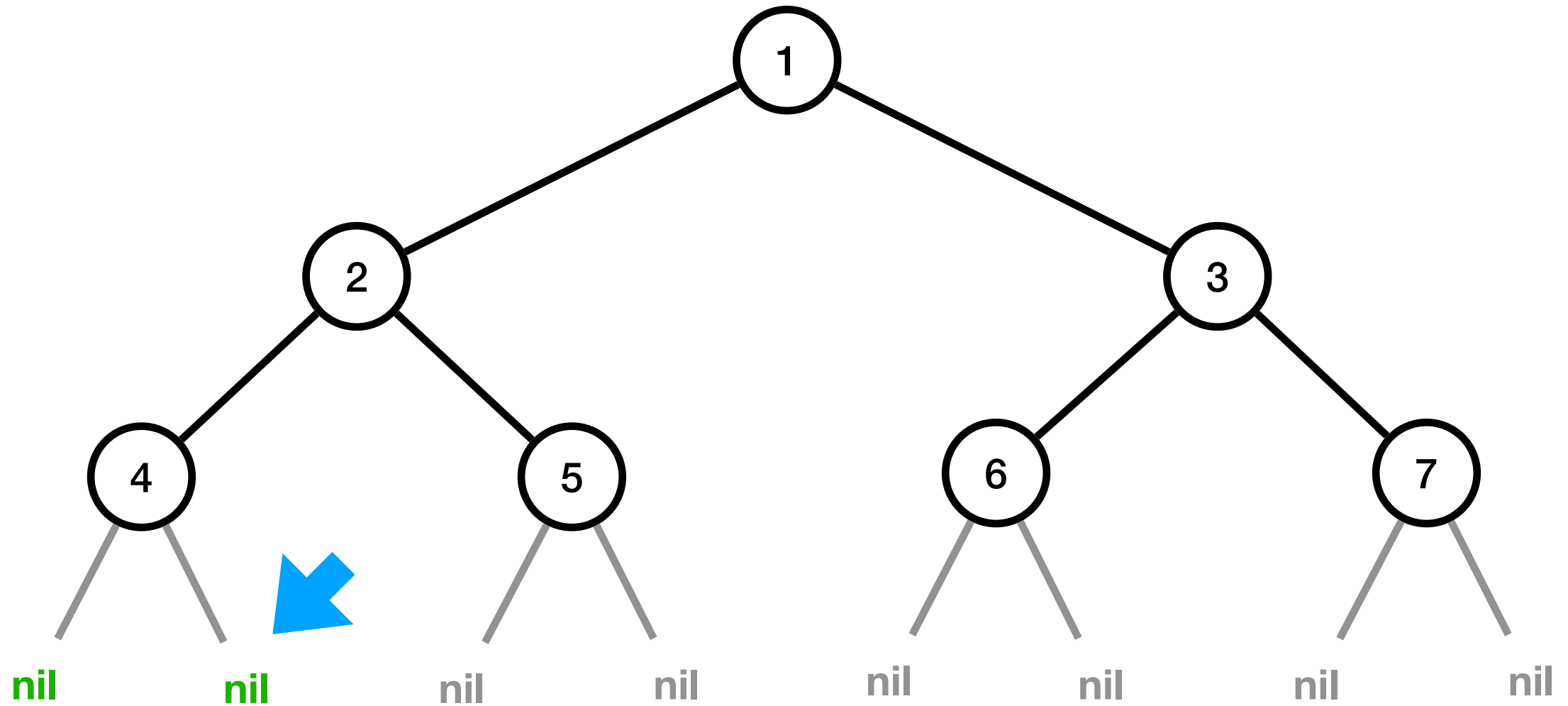
Parcours postfixe



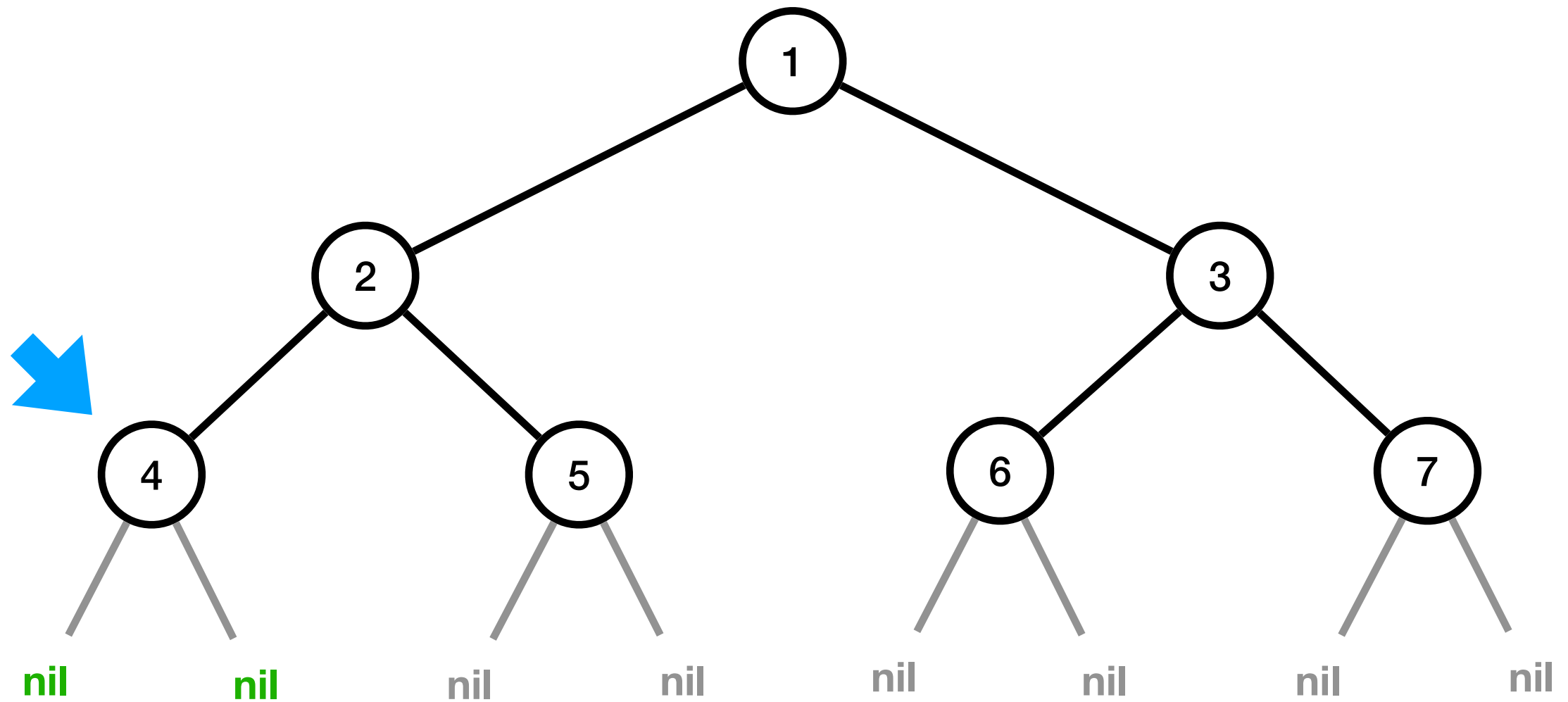
Parcours postfixe



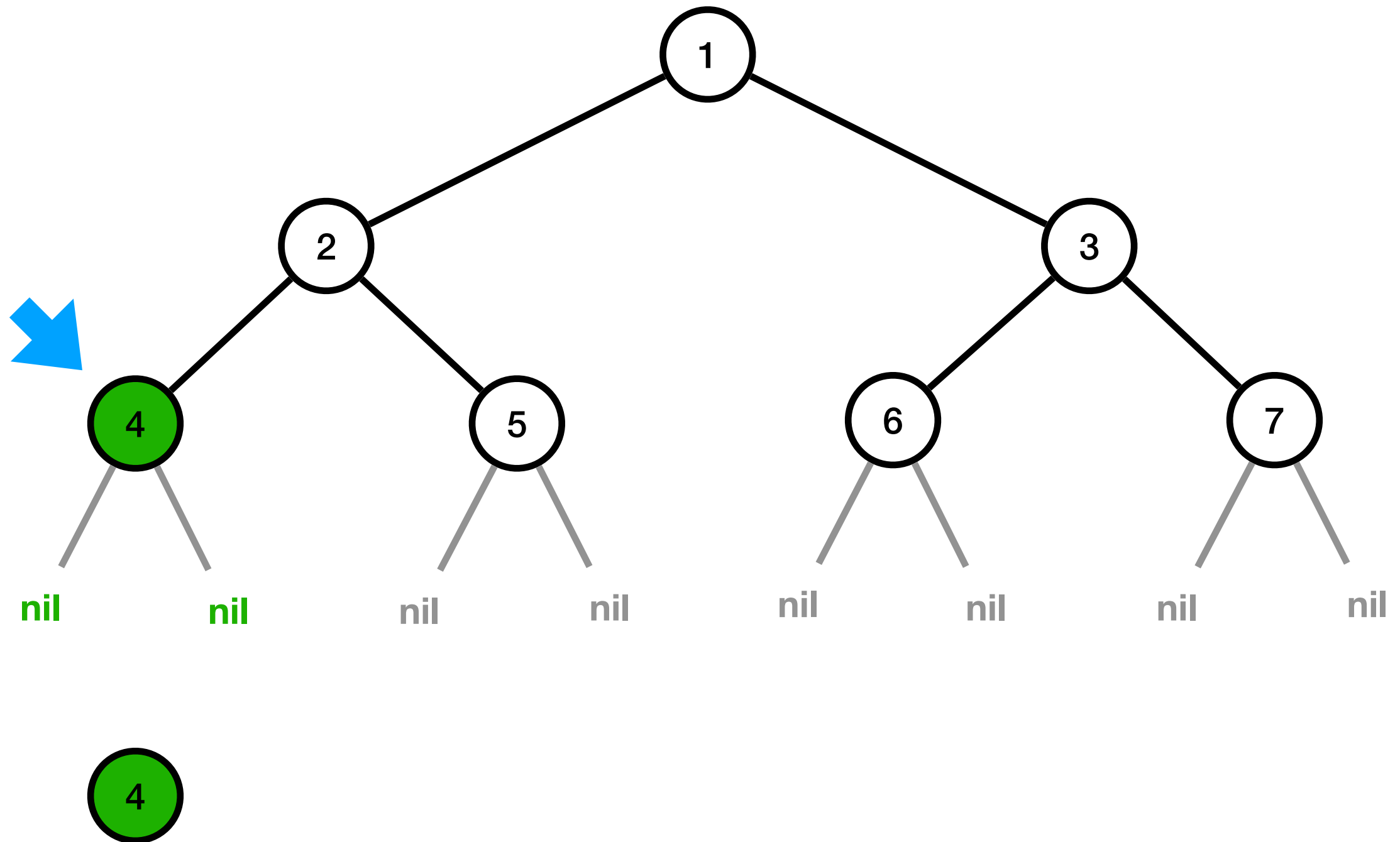
Parcours postfixe



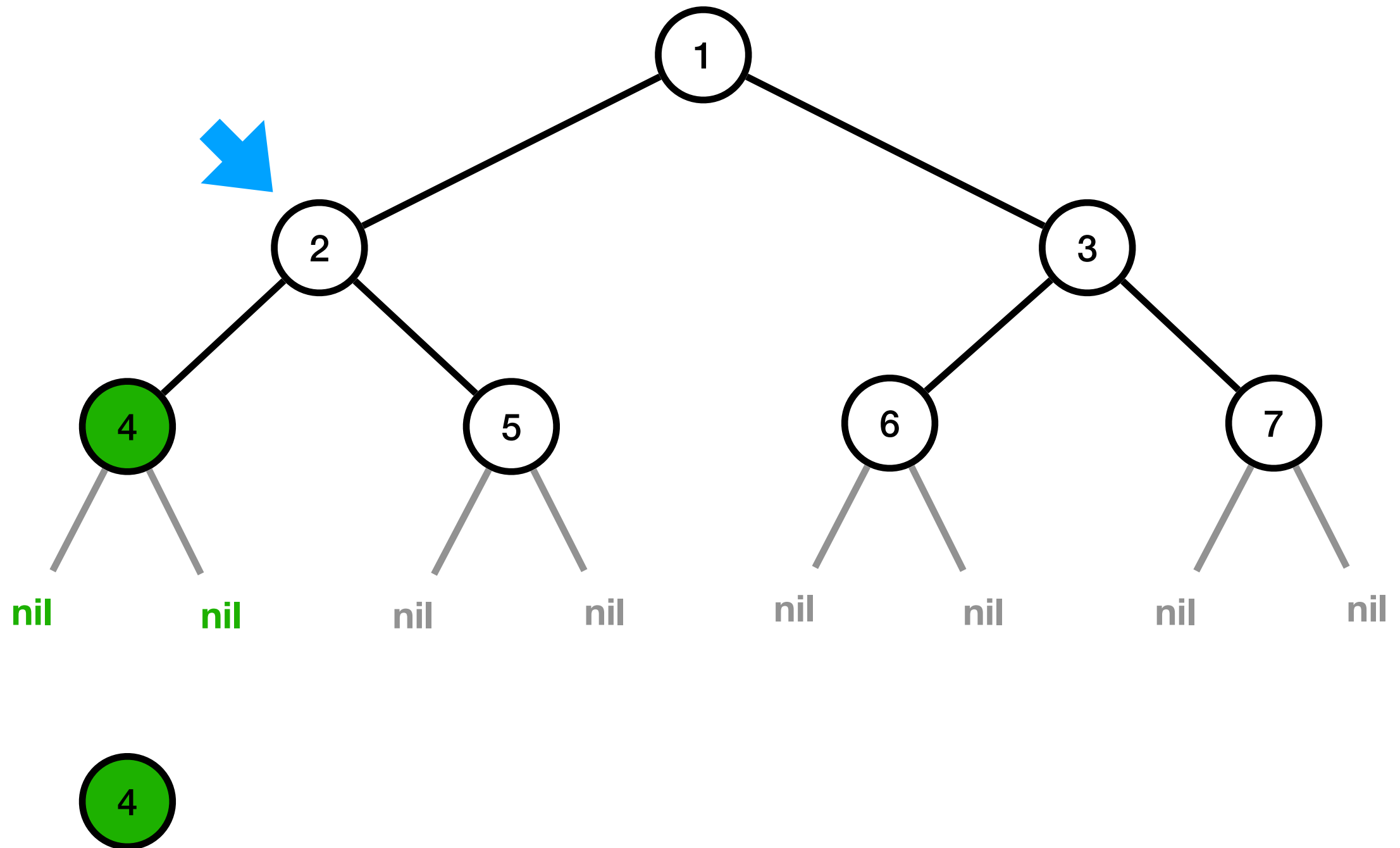
Parcours postfixe



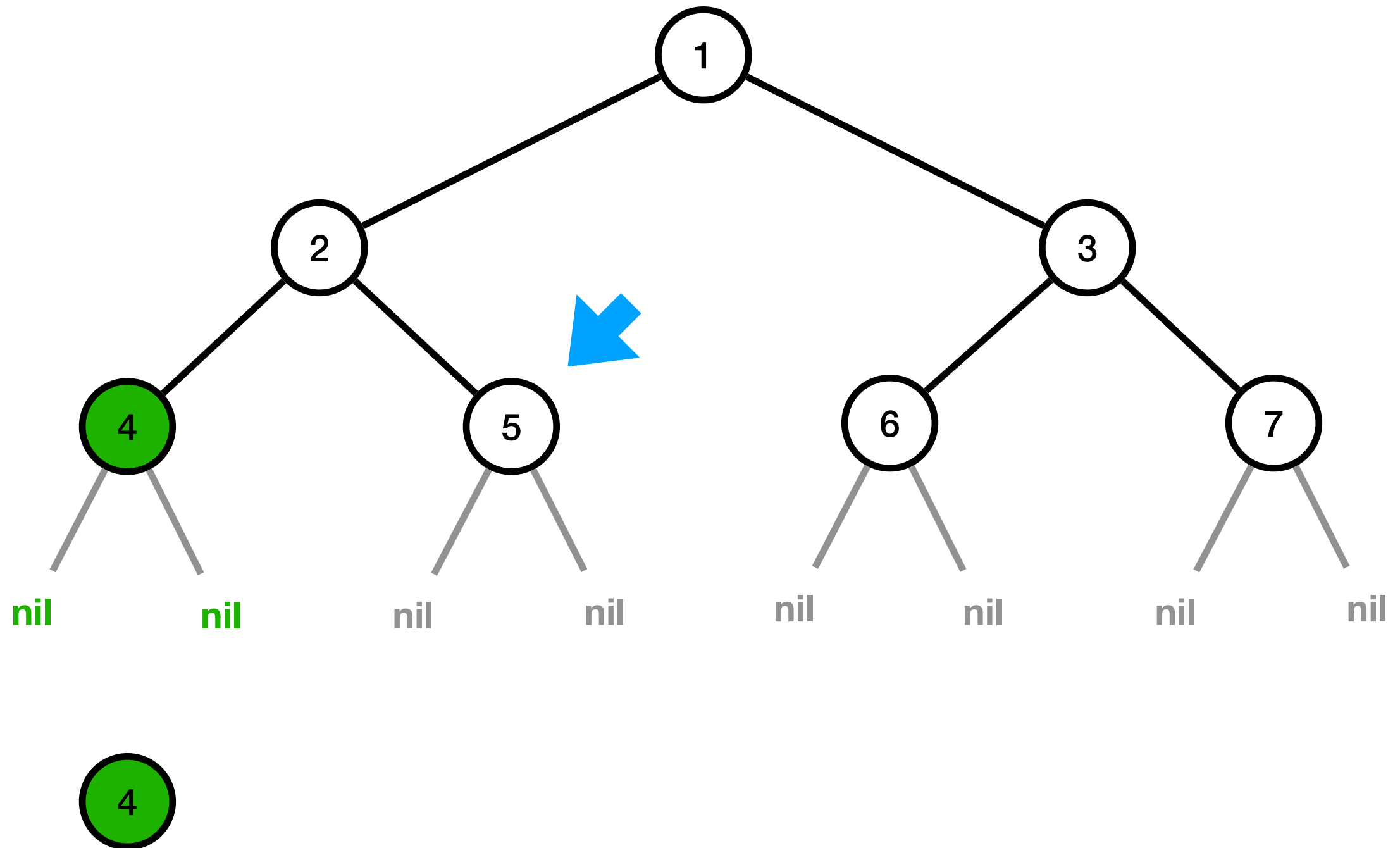
Parcours postfixe



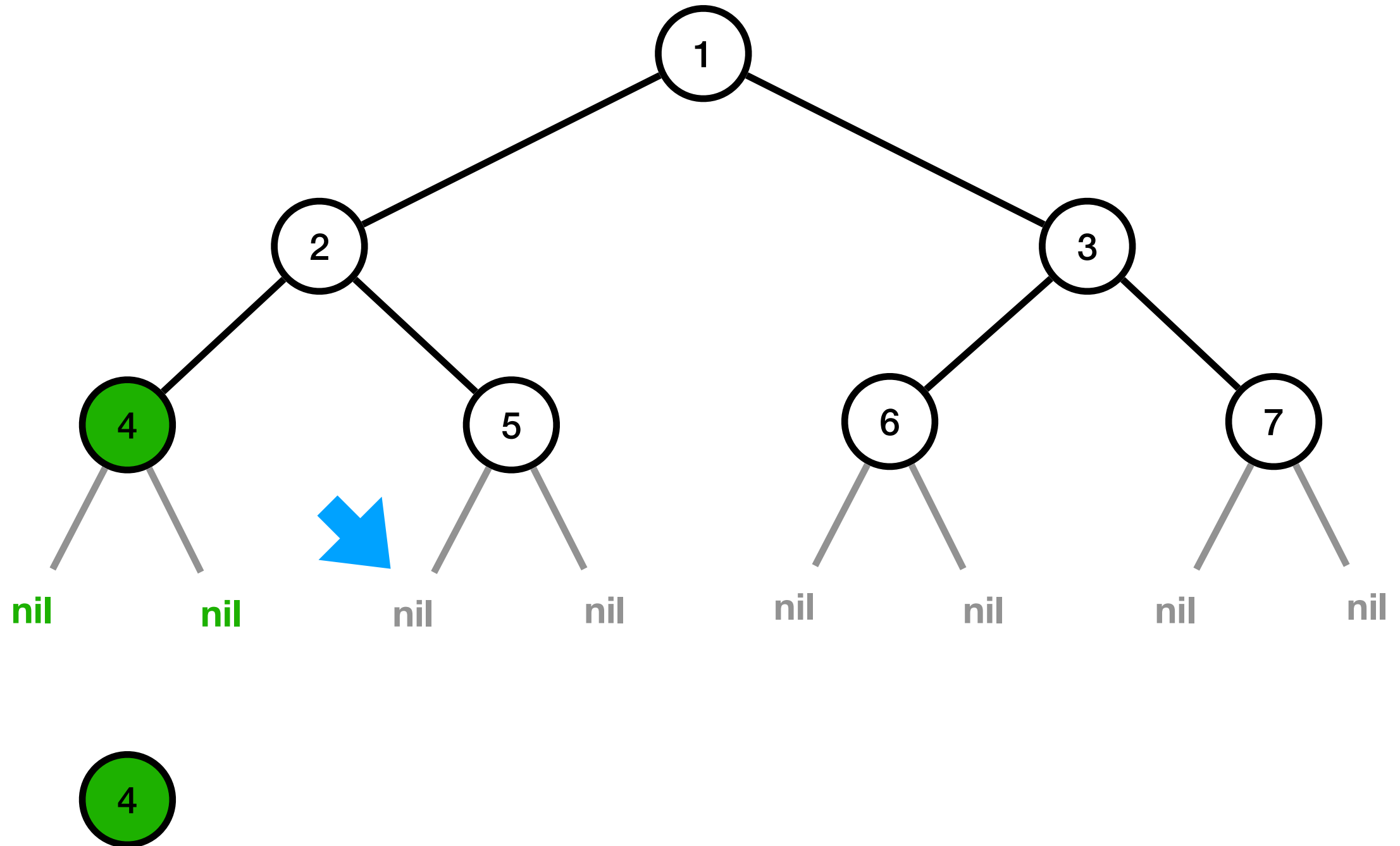
Parcours postfixe



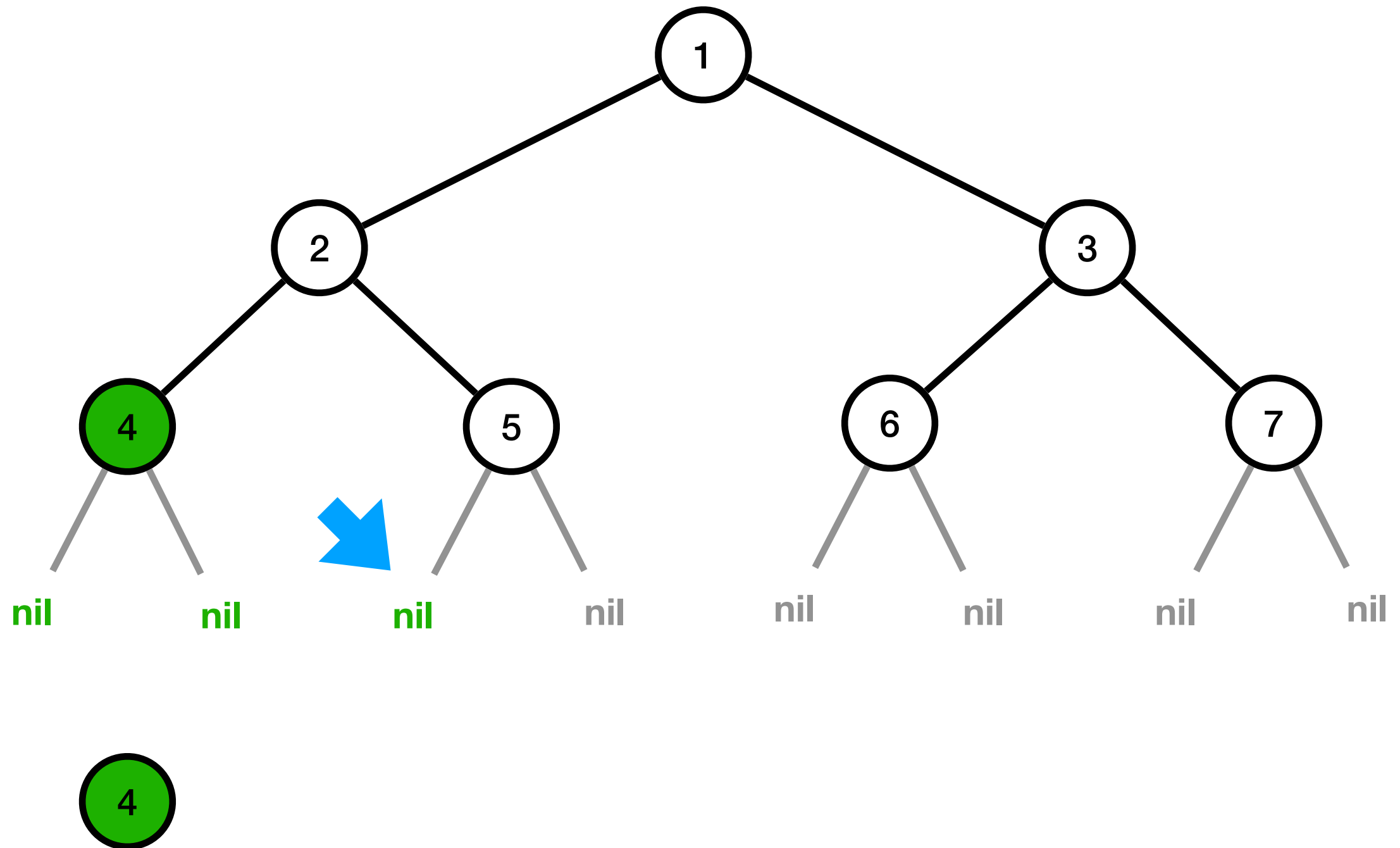
Parcours postfixe



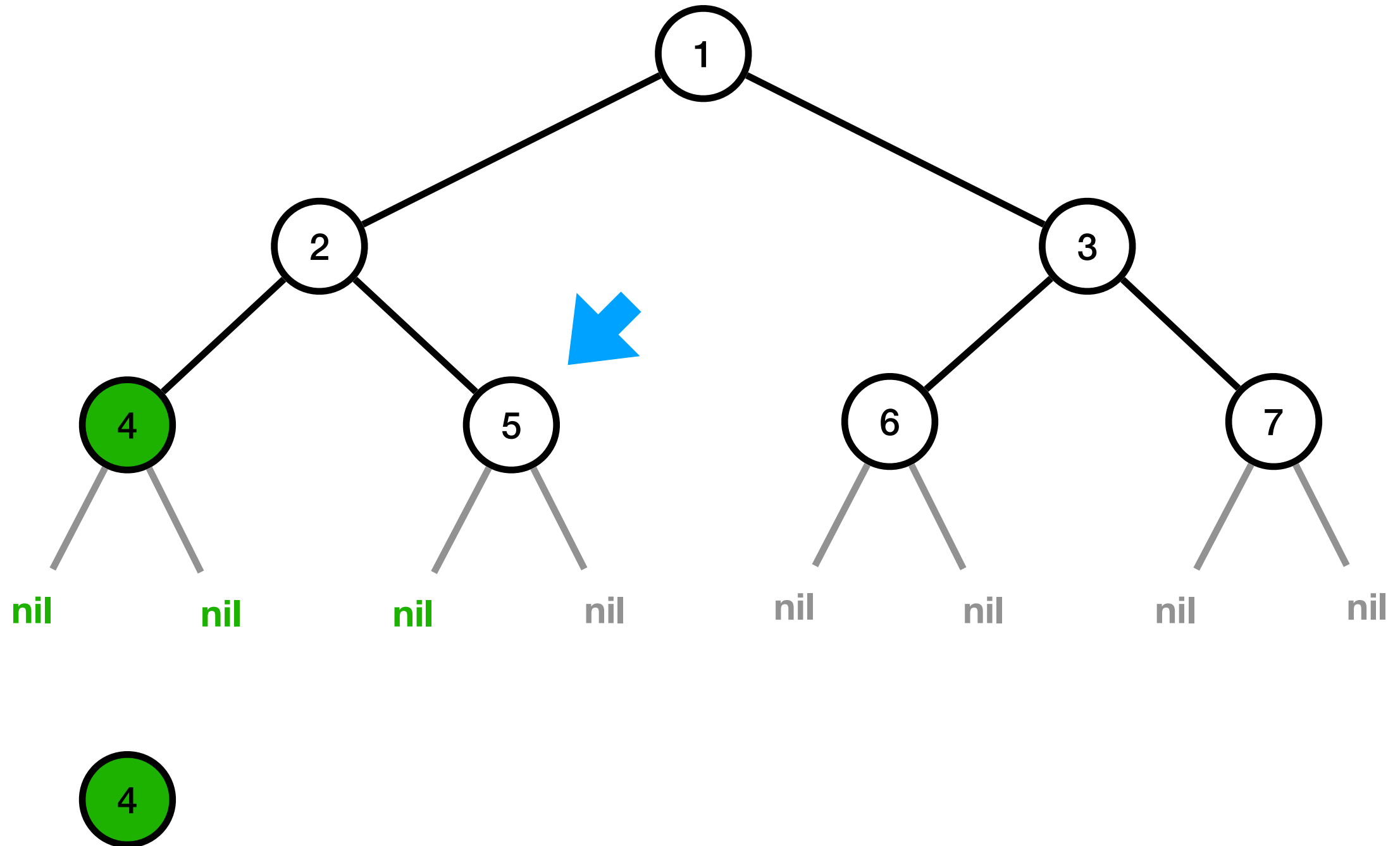
Parcours postfixe



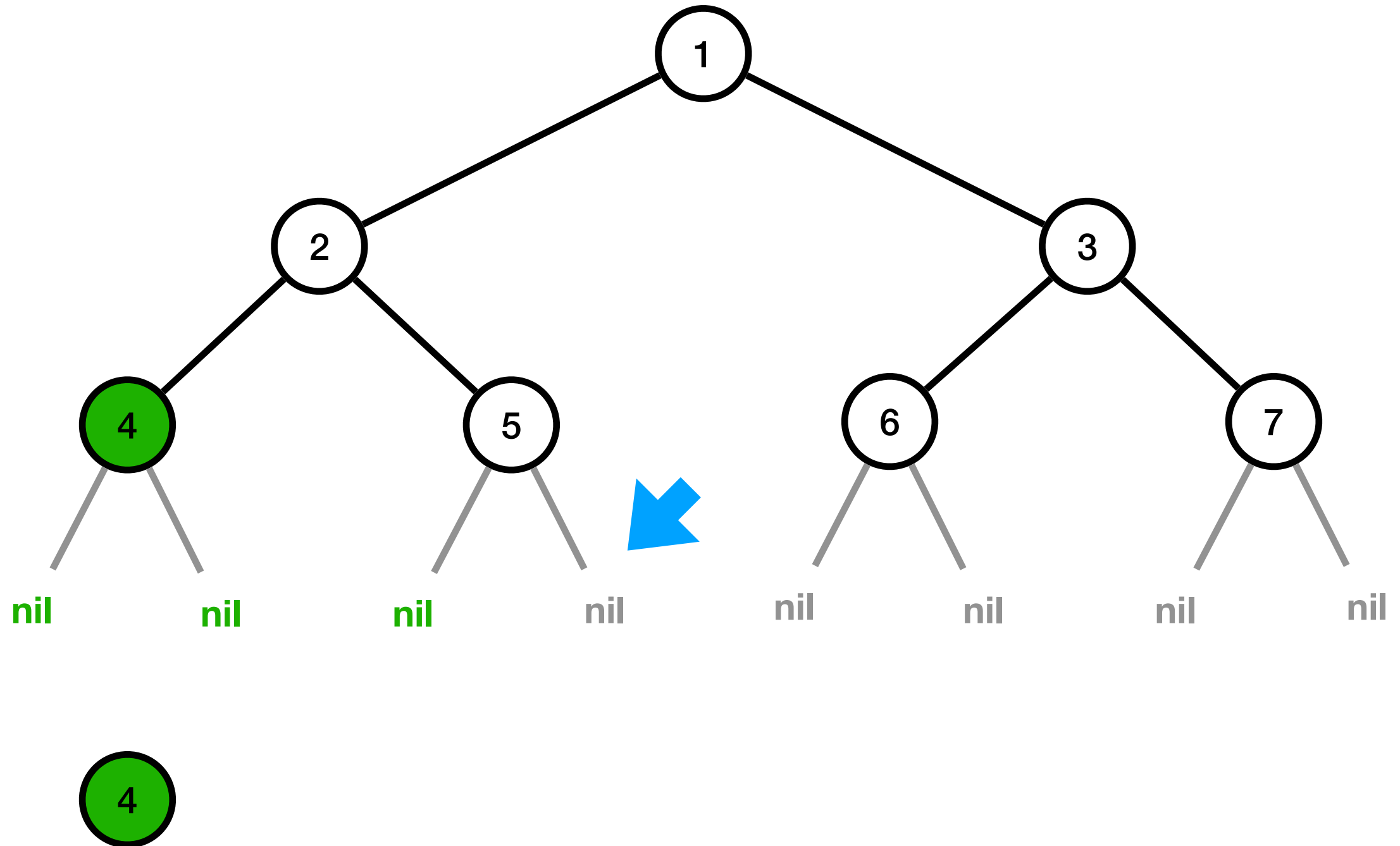
Parcours postfixe



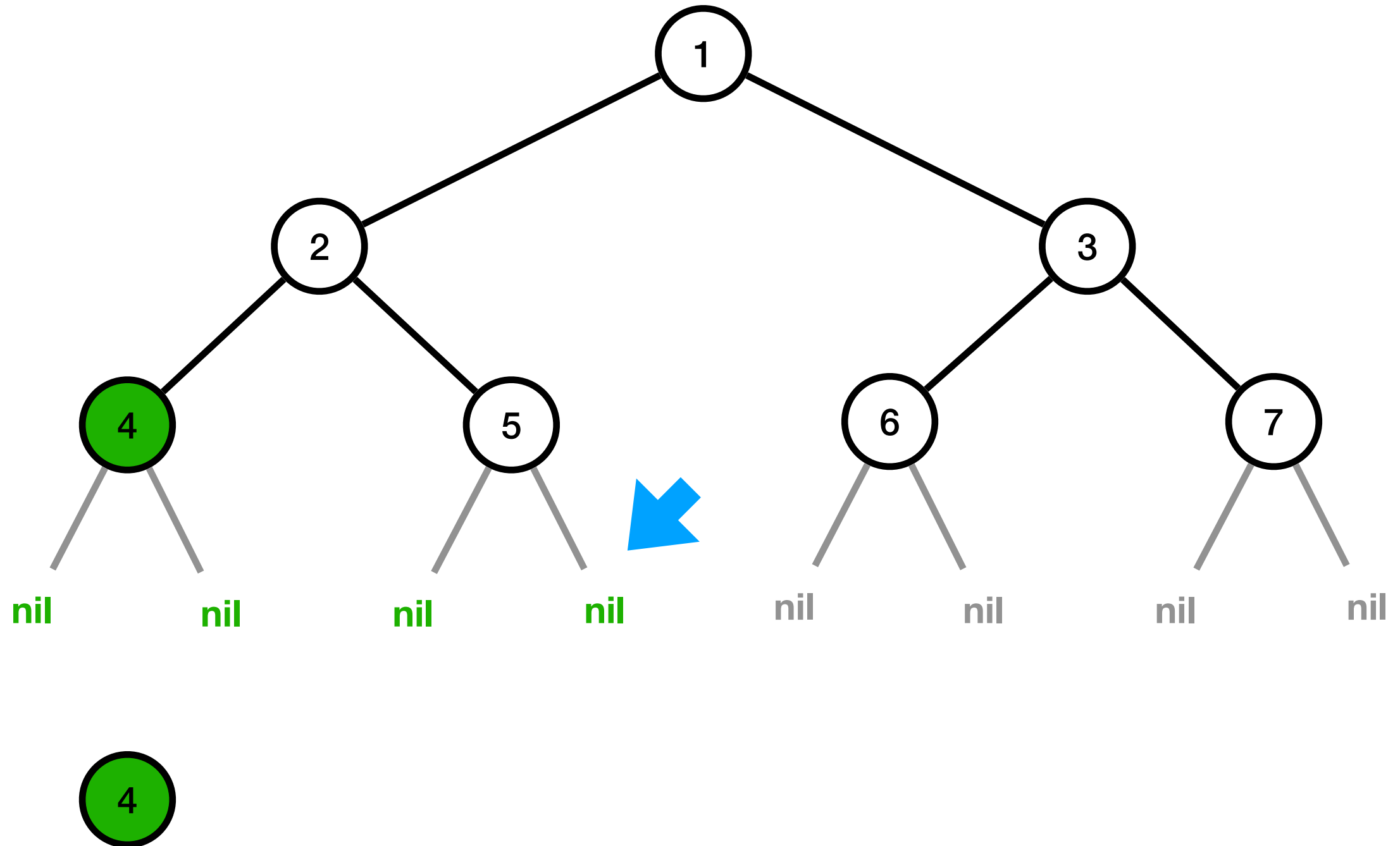
Parcours postfixe



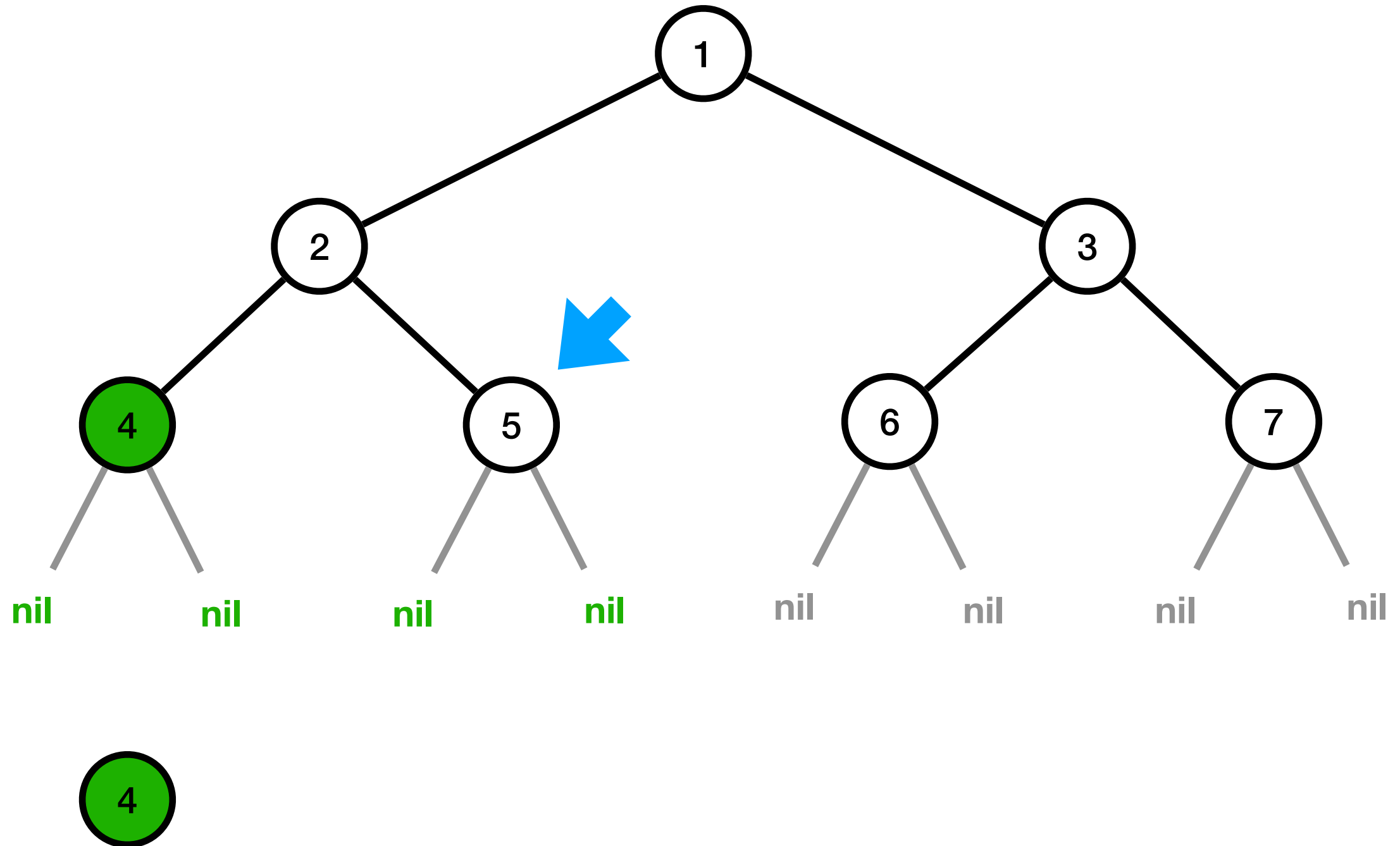
Parcours postfixe



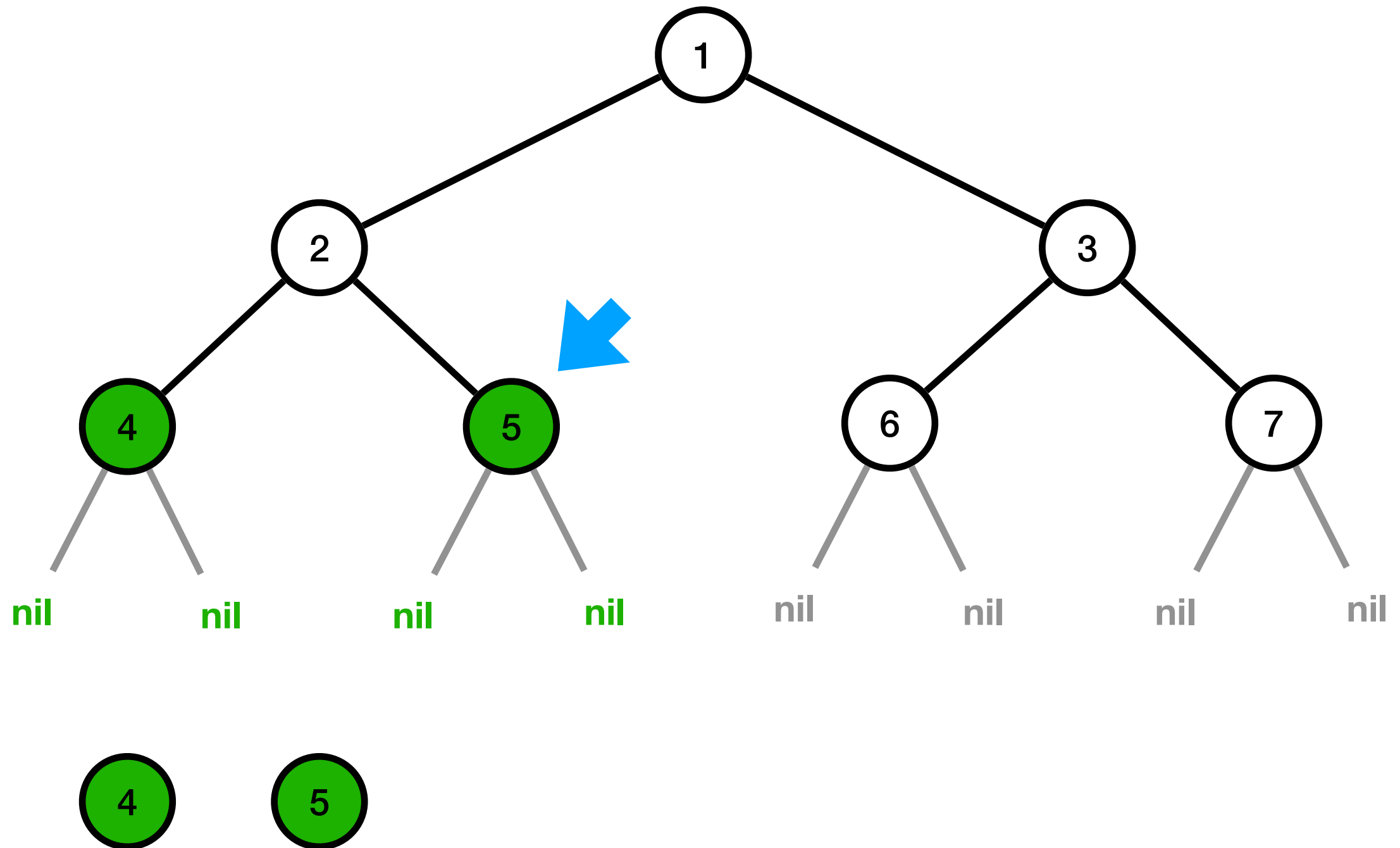
Parcours postfixe



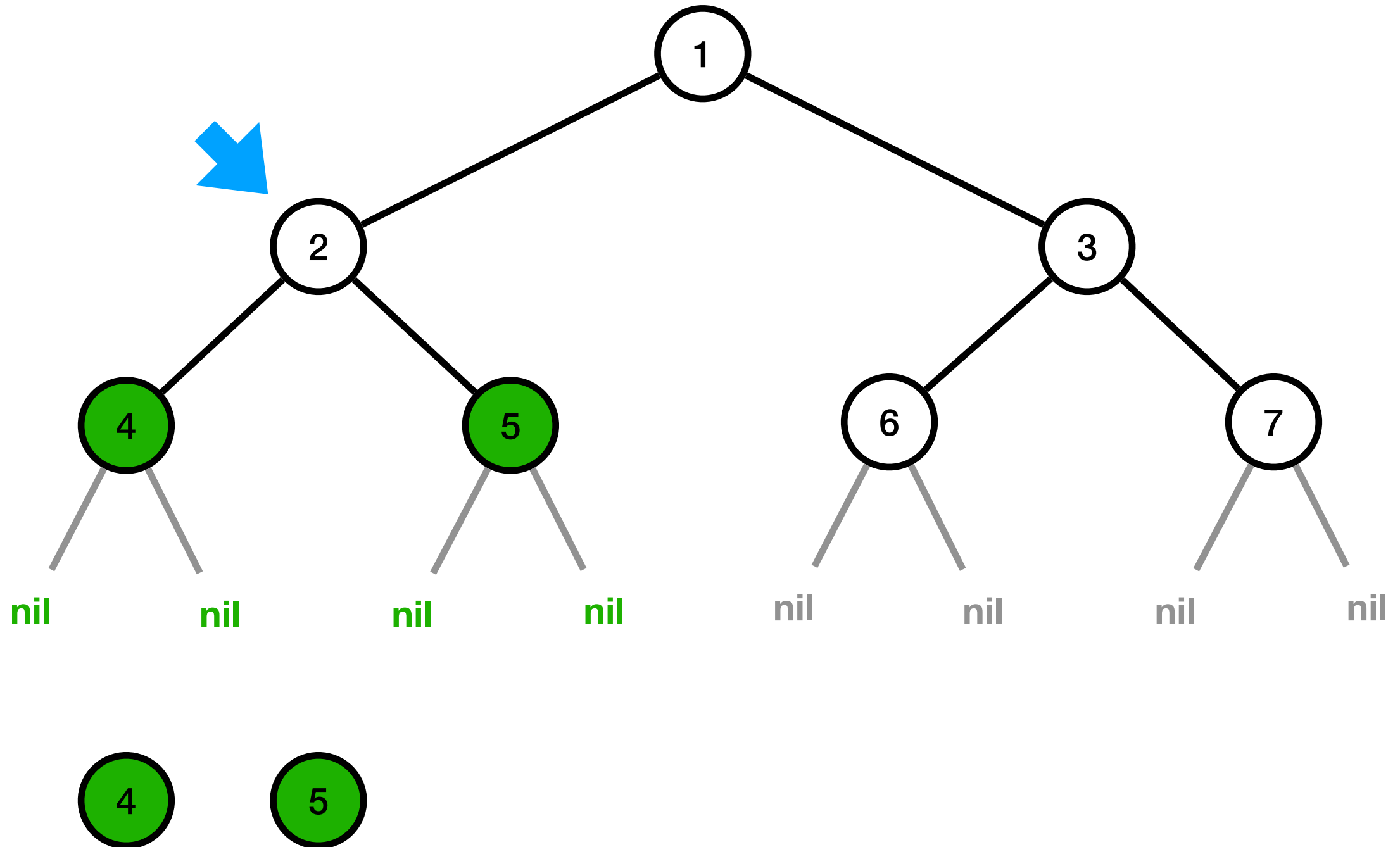
Parcours postfixe



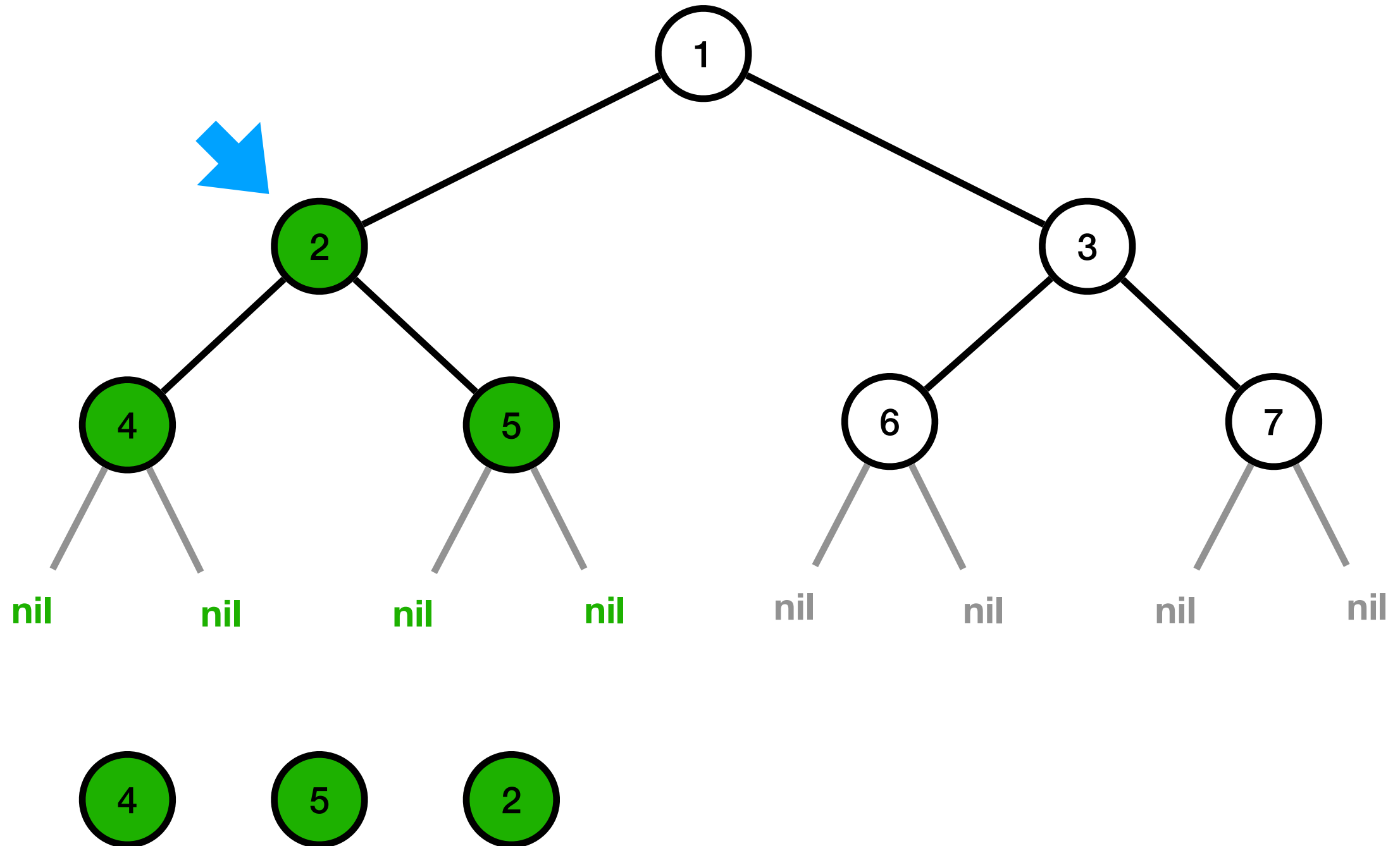
Parcours postfixe



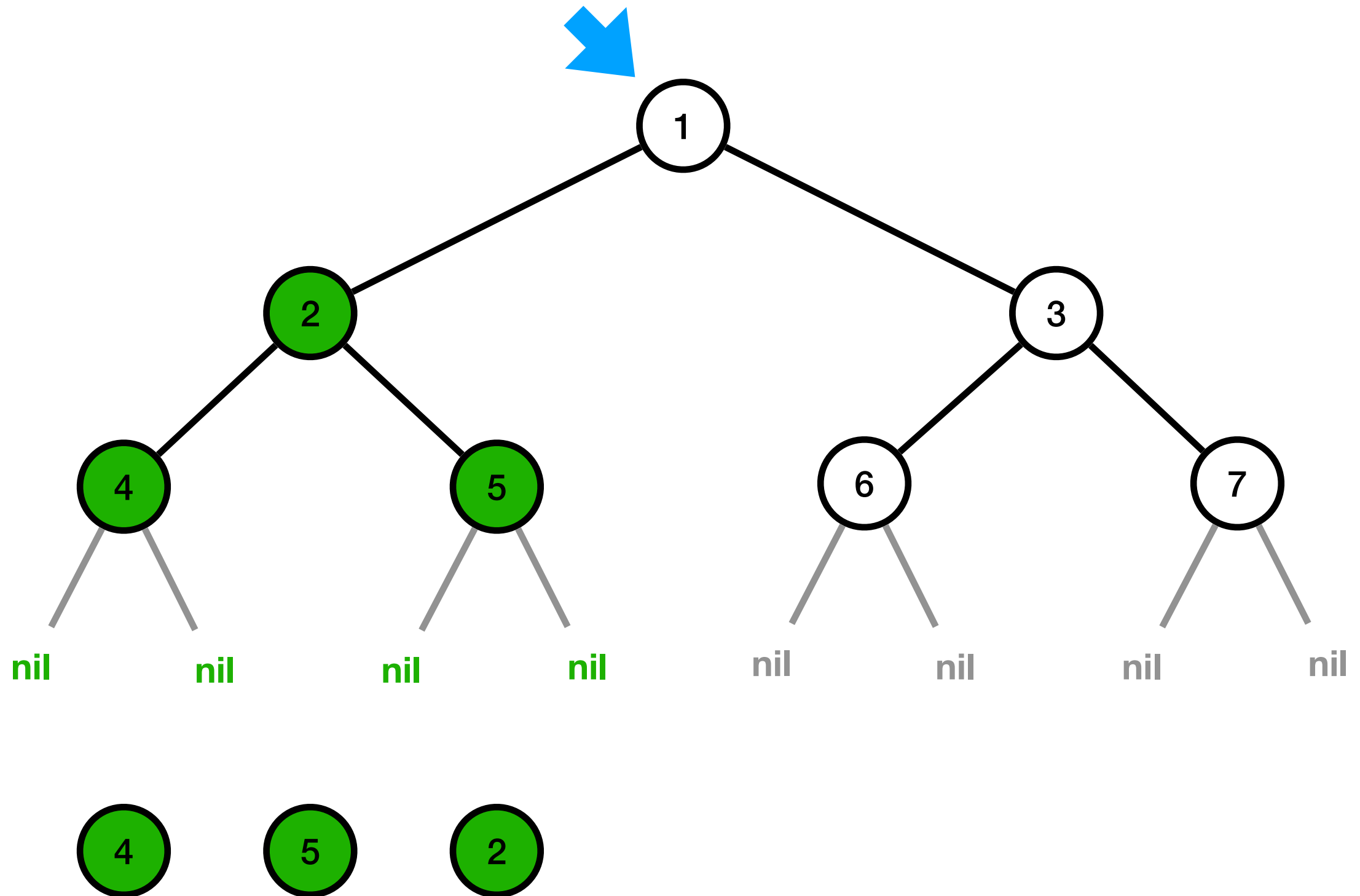
Parcours postfixe



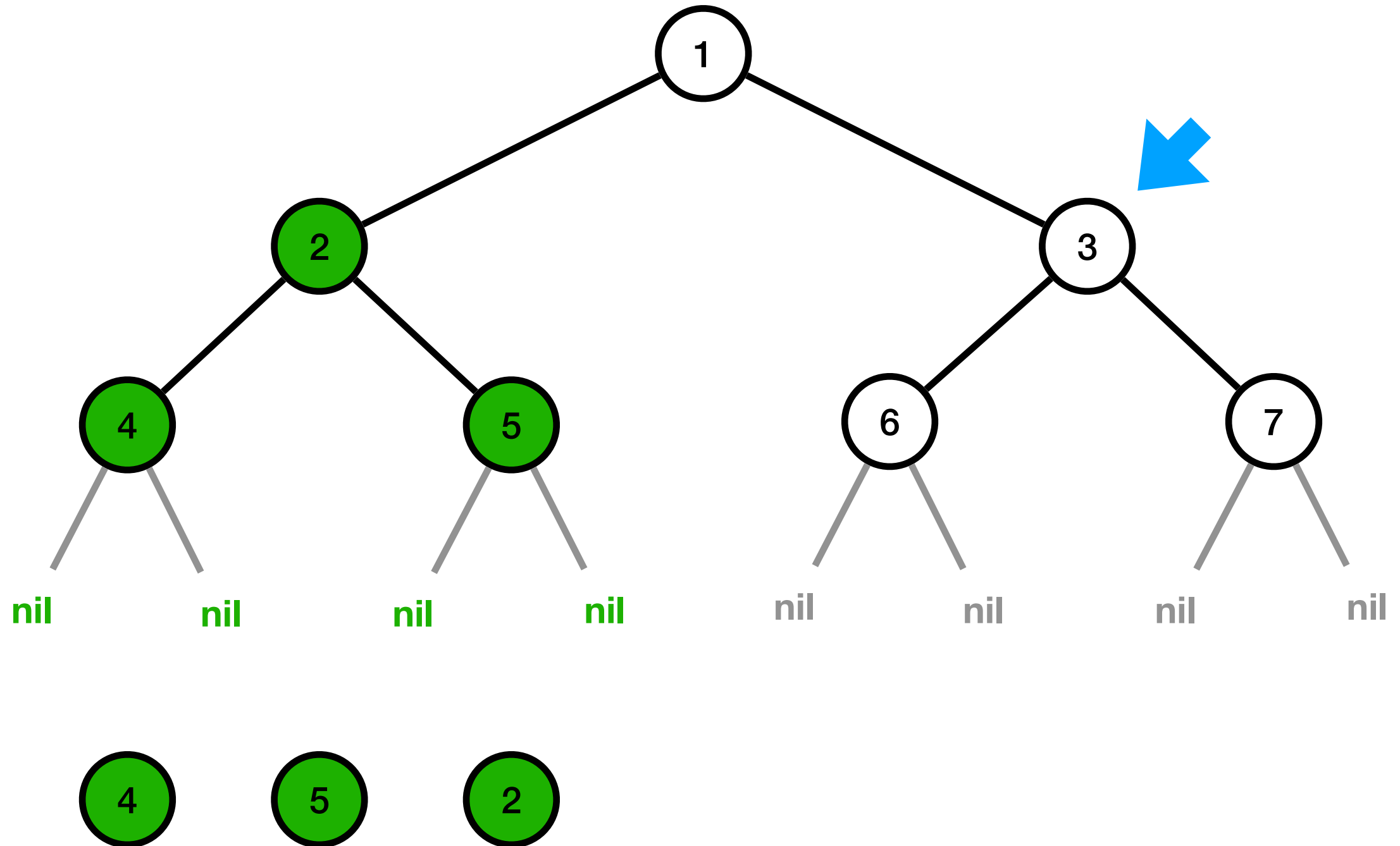
Parcours postfixe



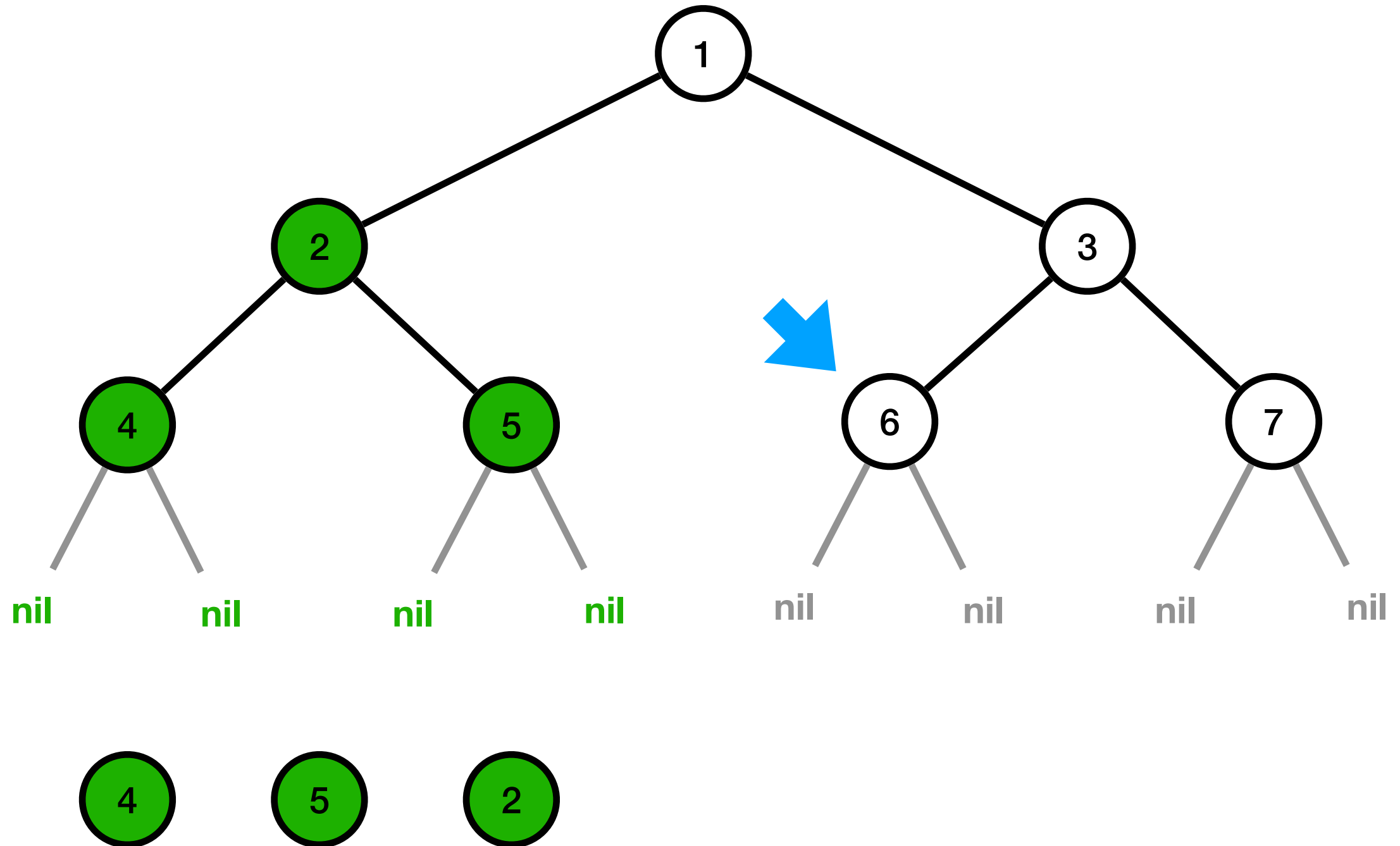
Parcours postfixe



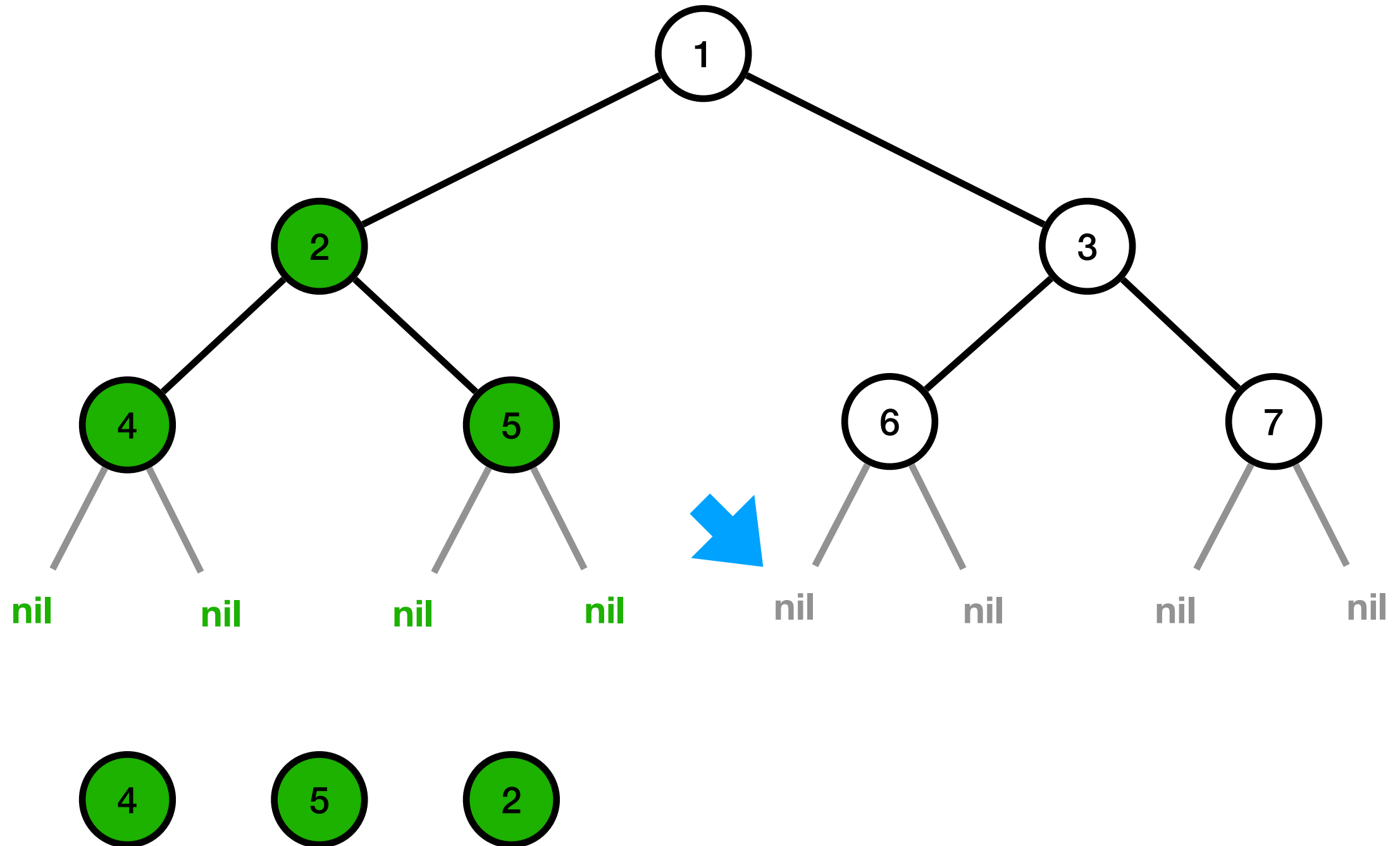
Parcours postfixe



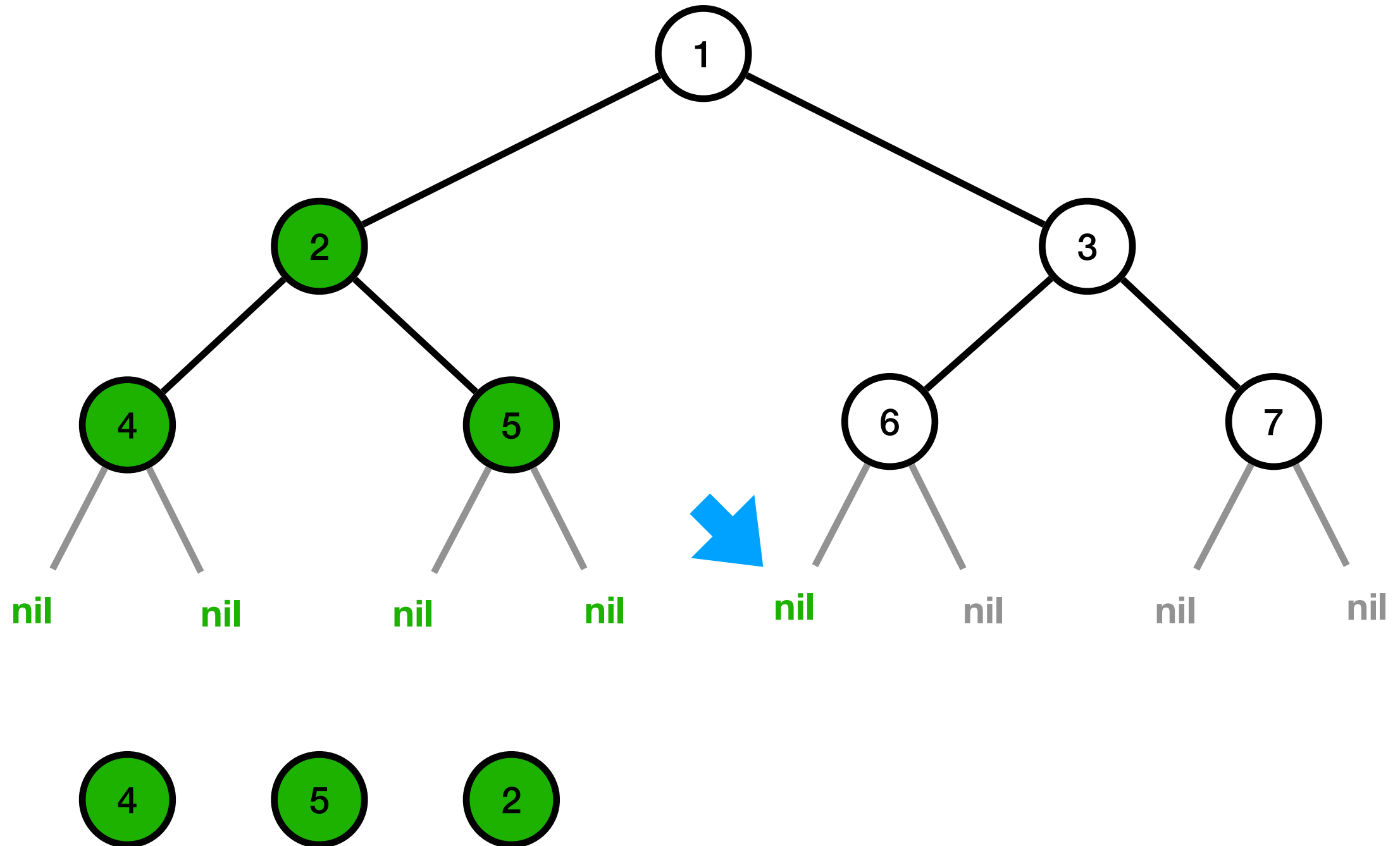
Parcours postfixe



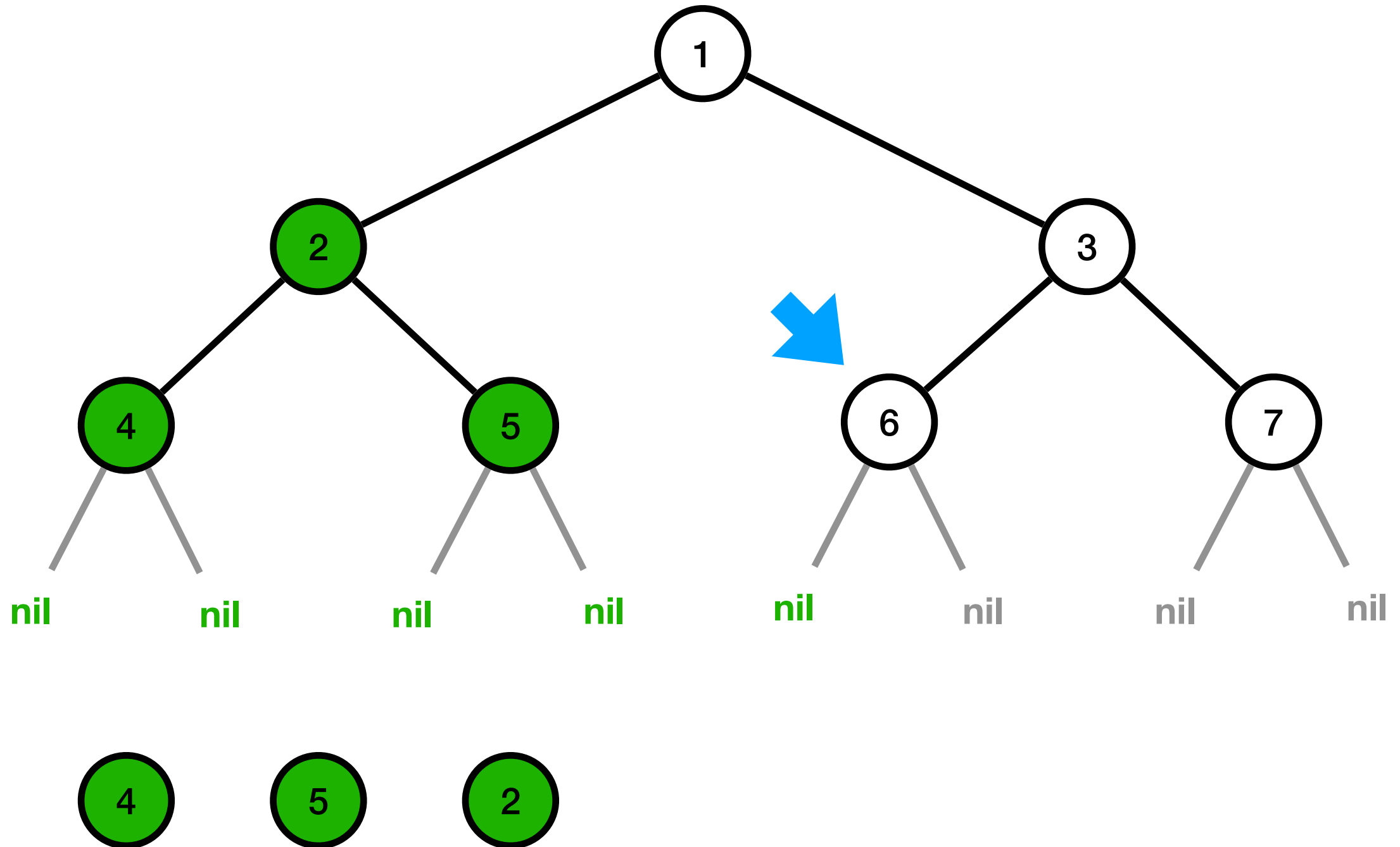
Parcours postfixe



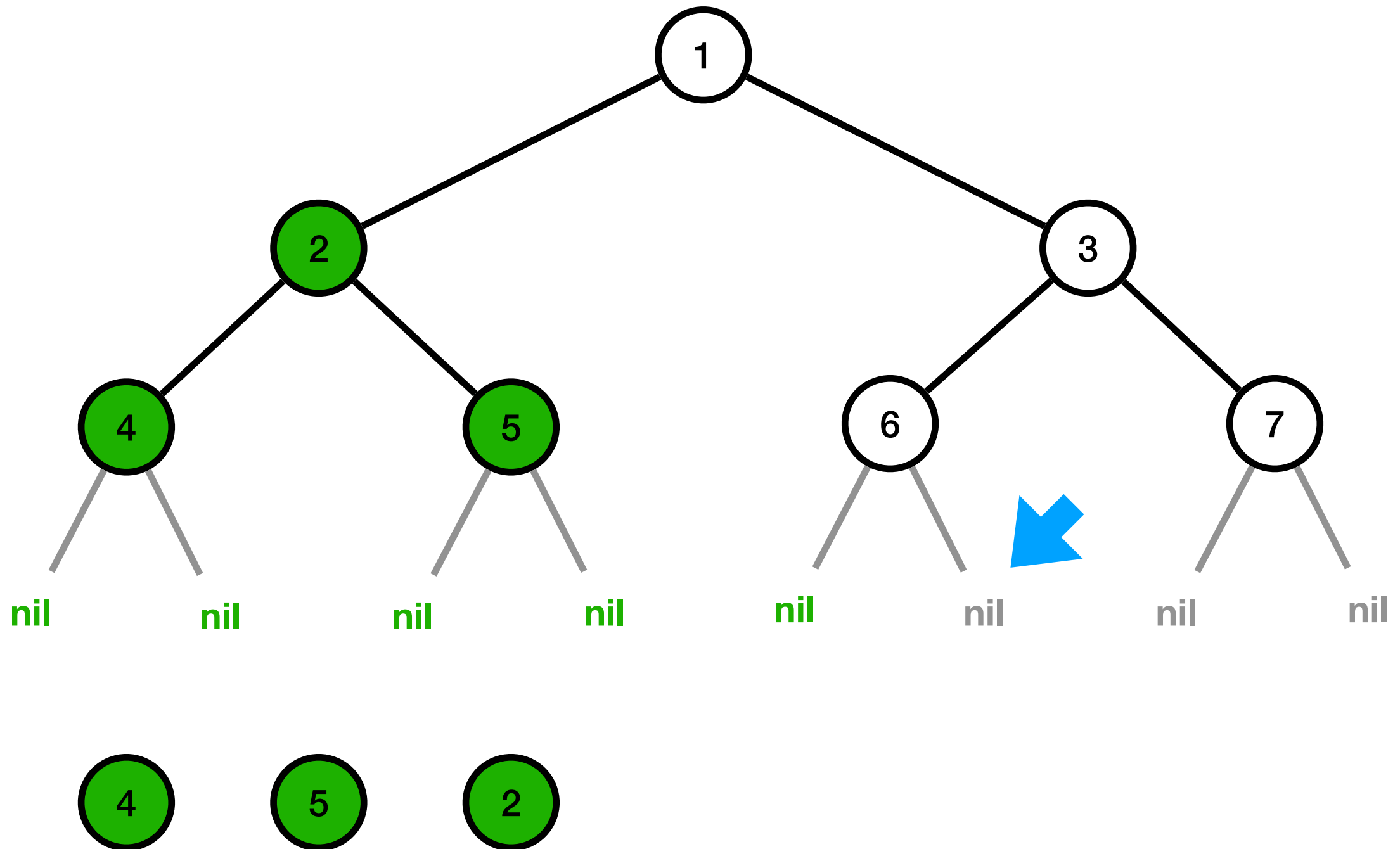
Parcours postfixe



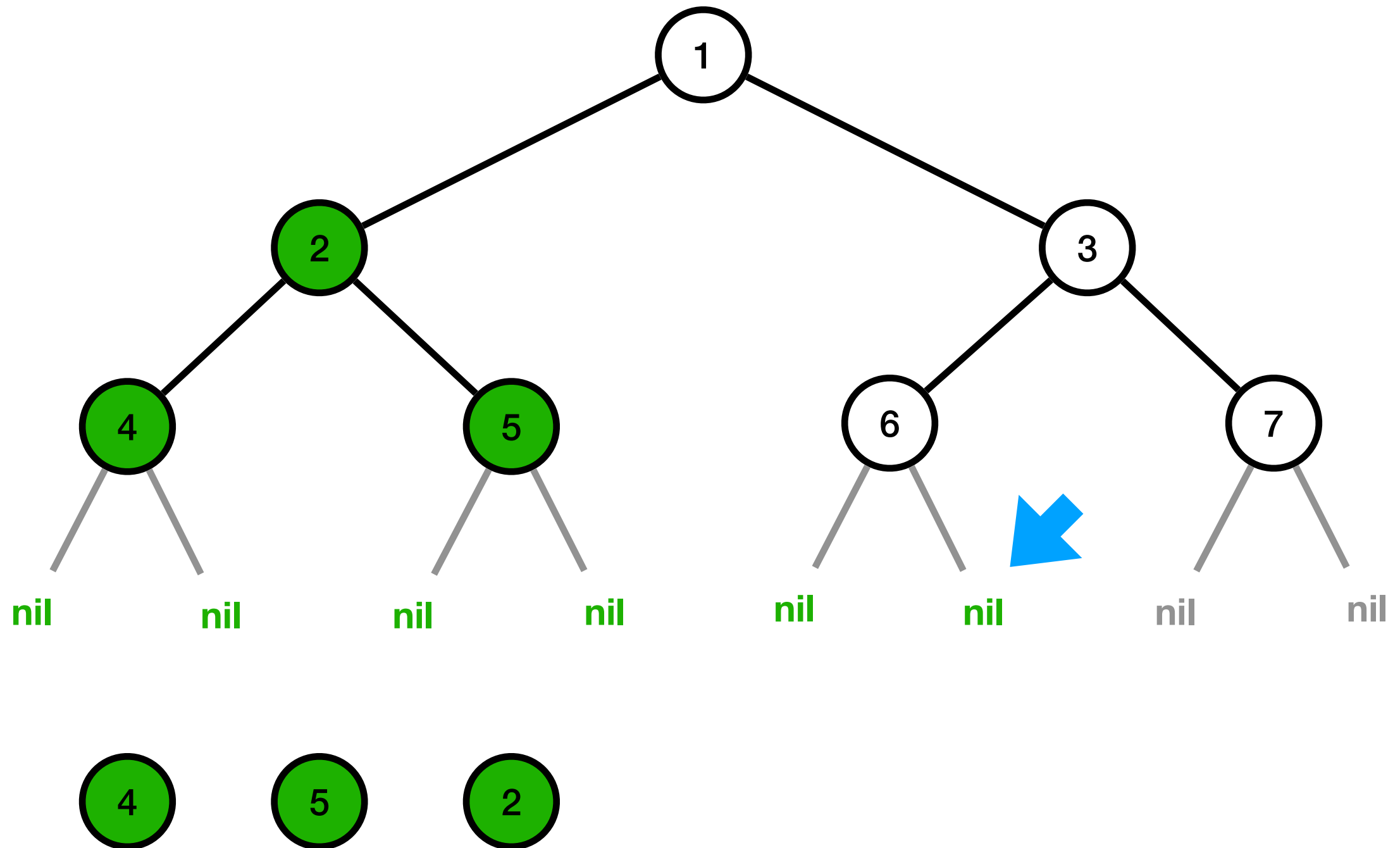
Parcours postfixe



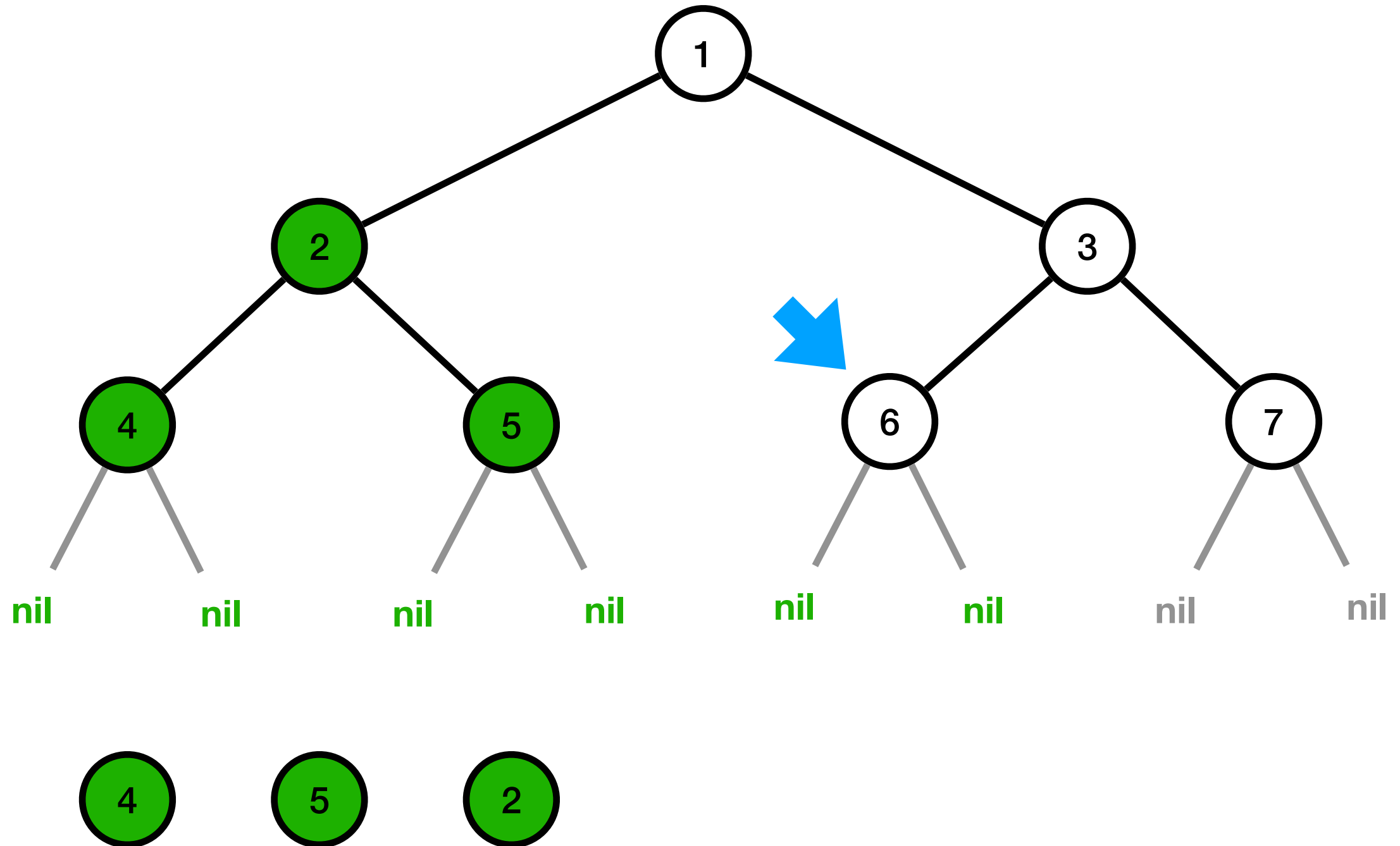
Parcours postfixe



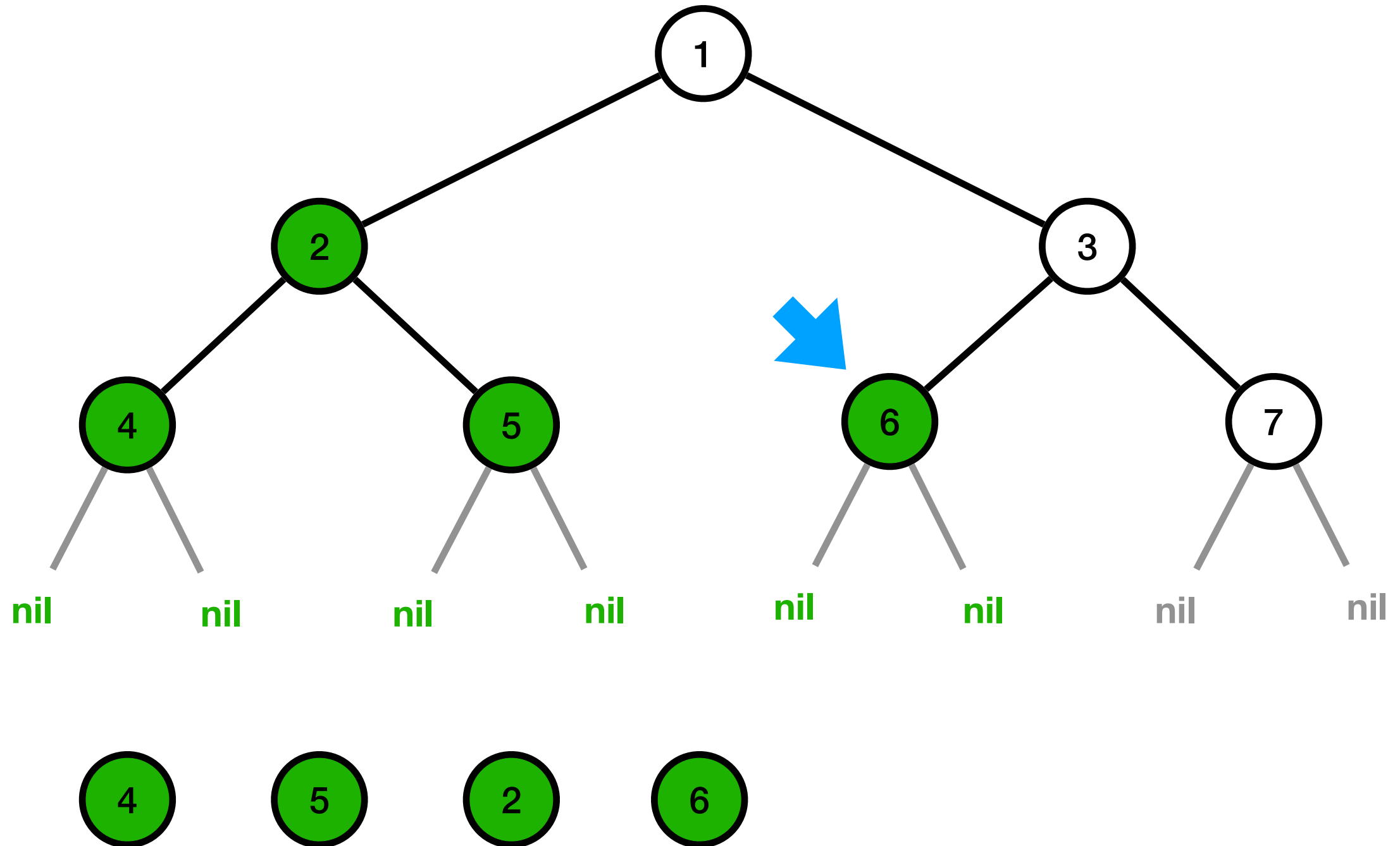
Parcours postfixe



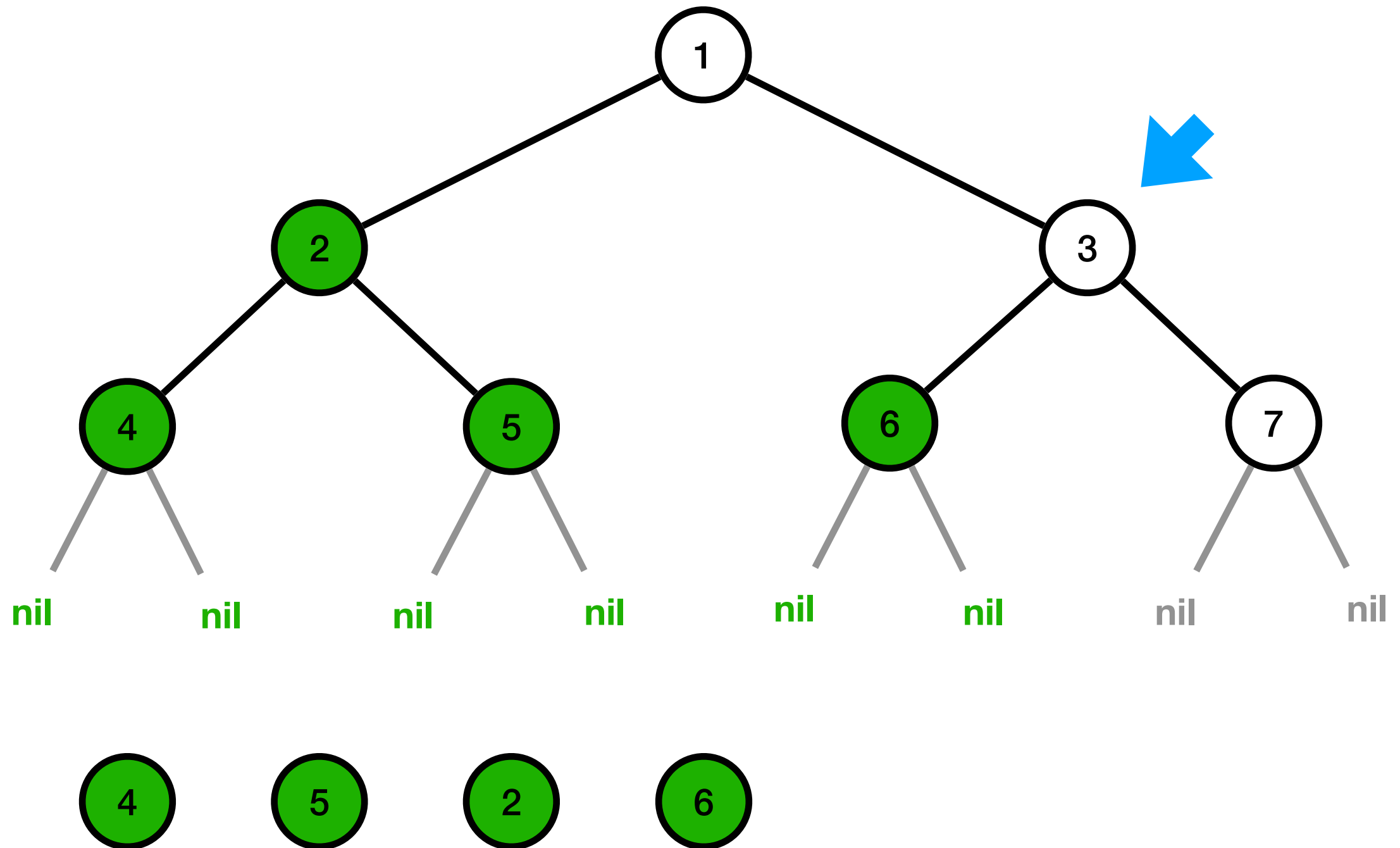
Parcours postfixe



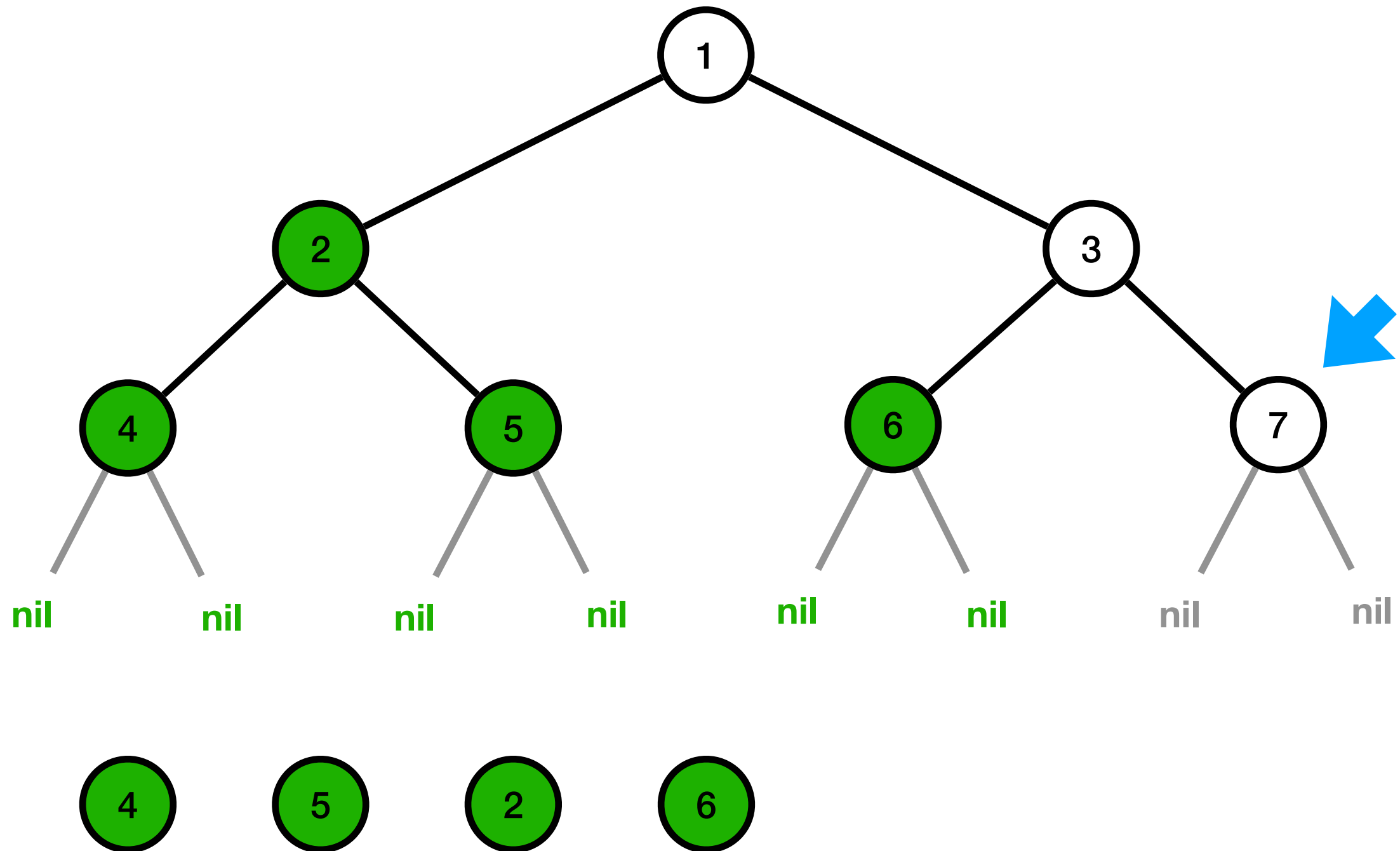
Parcours postfixe



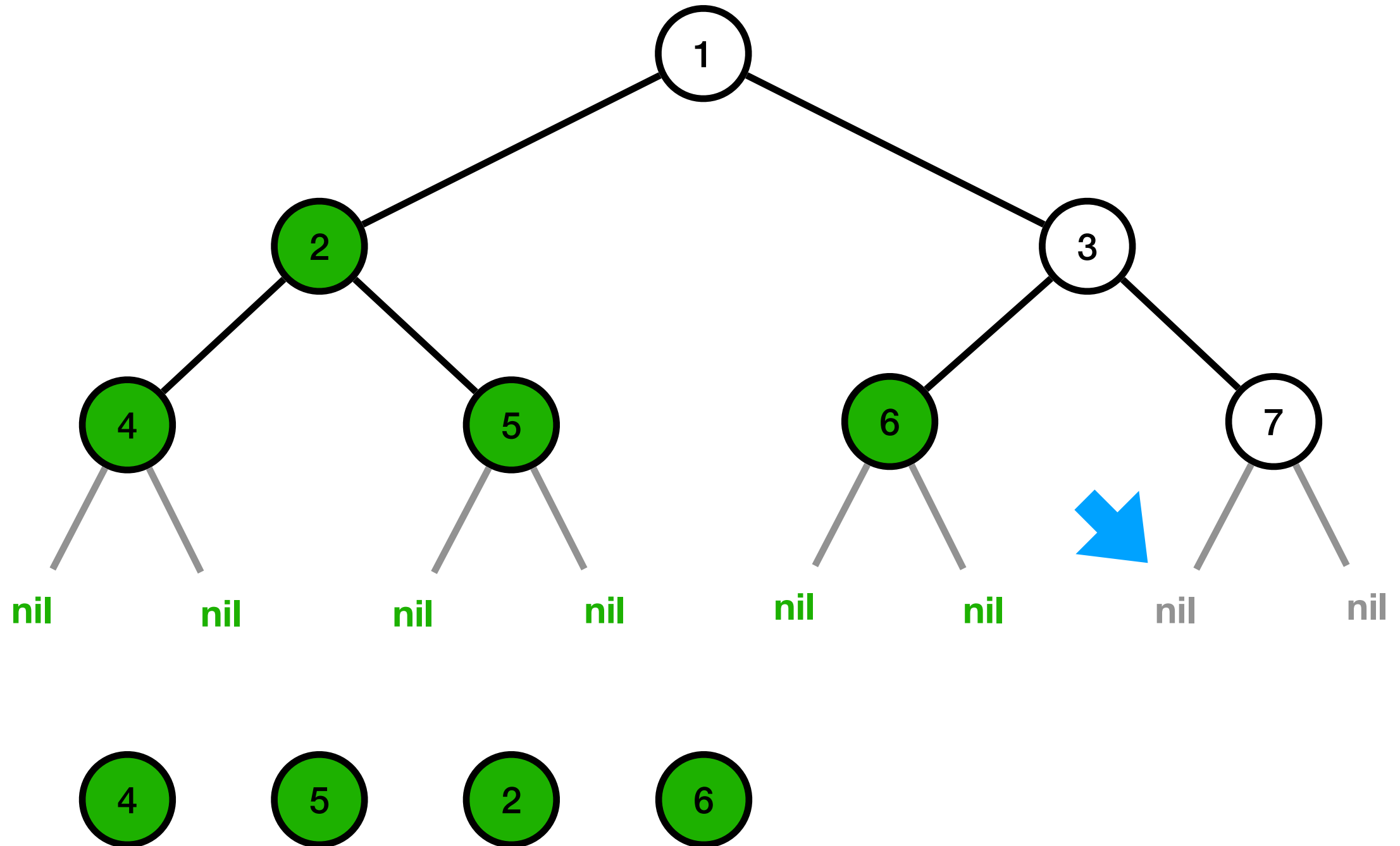
Parcours postfixe



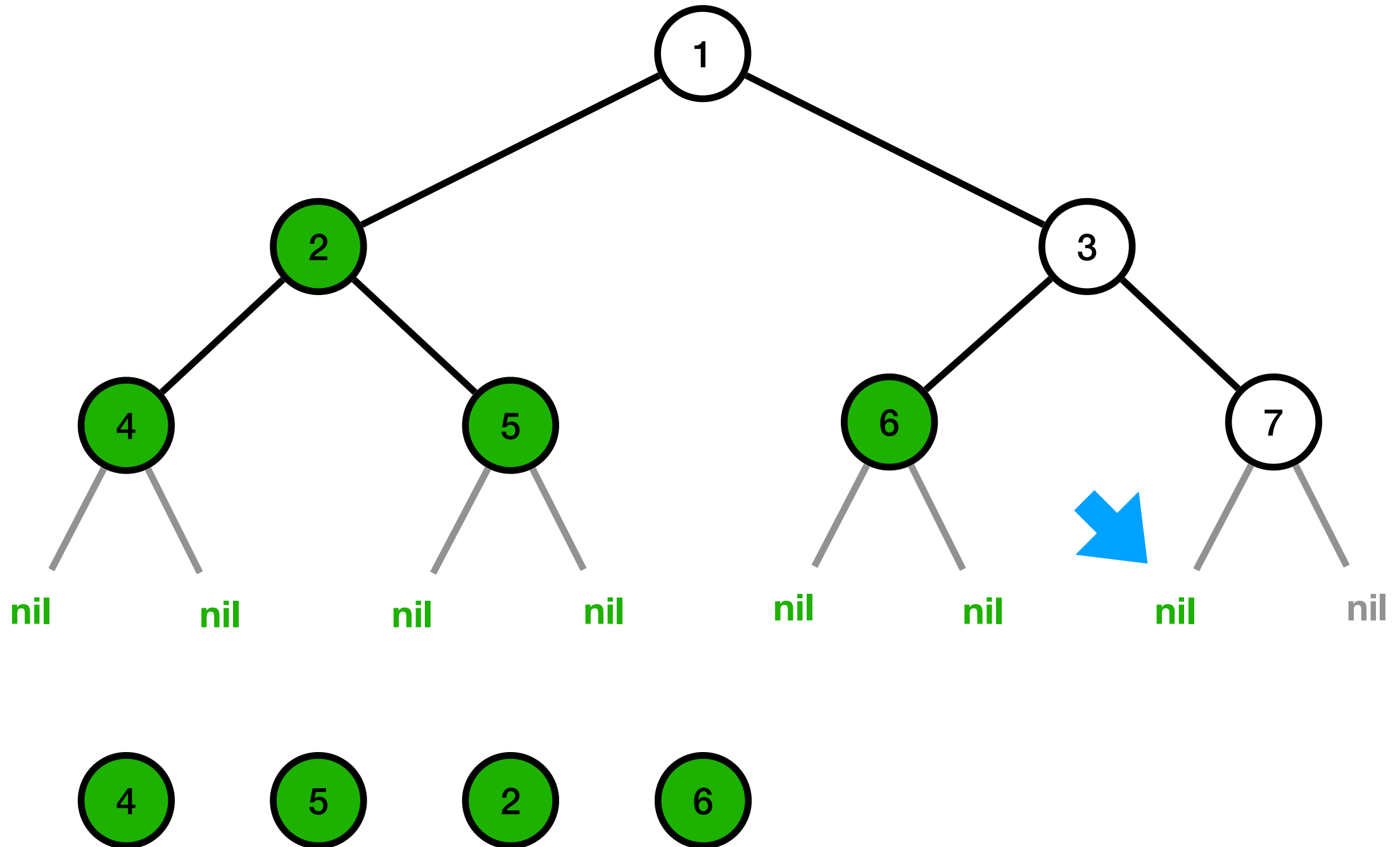
Parcours postfixe



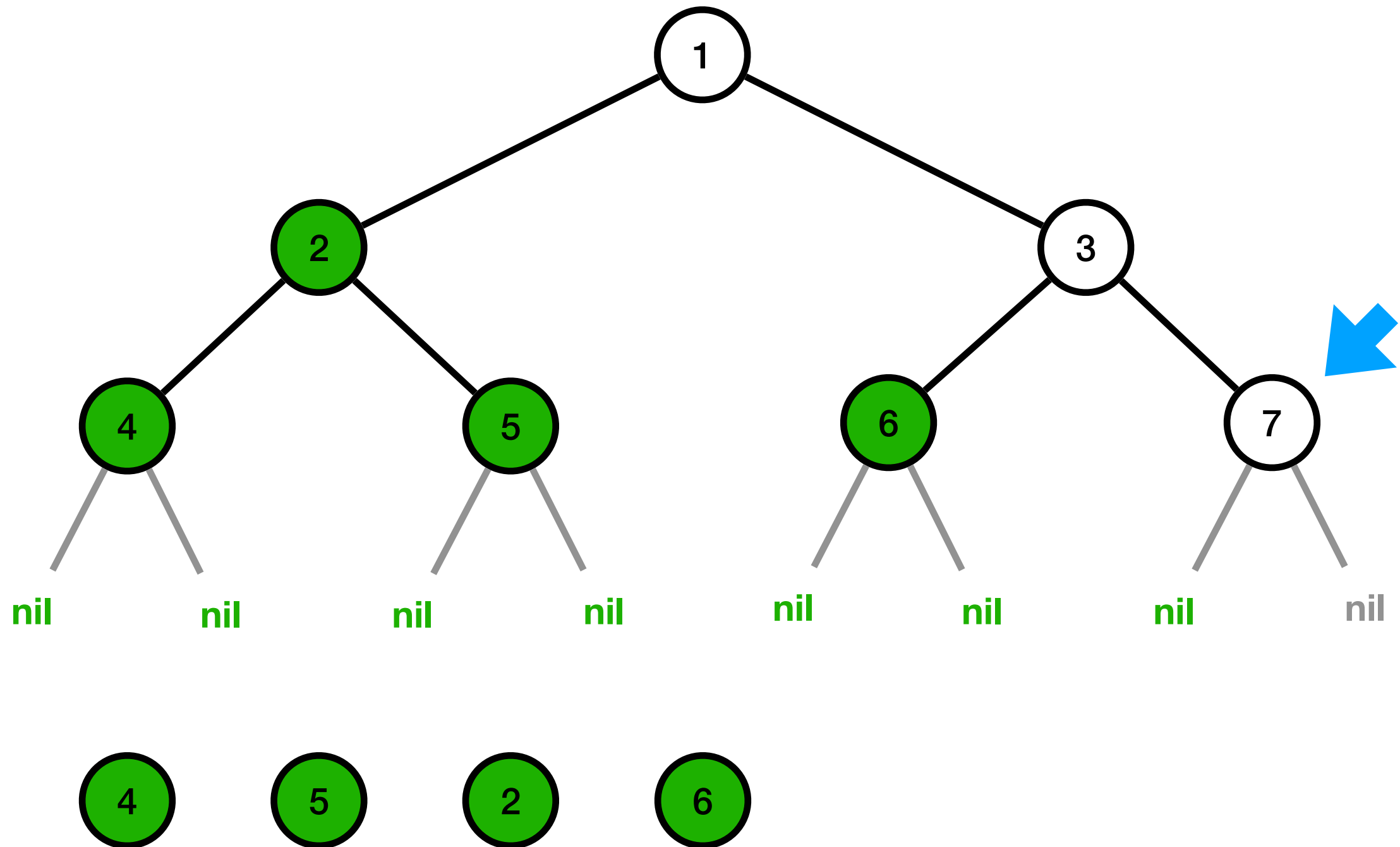
Parcours postfixe



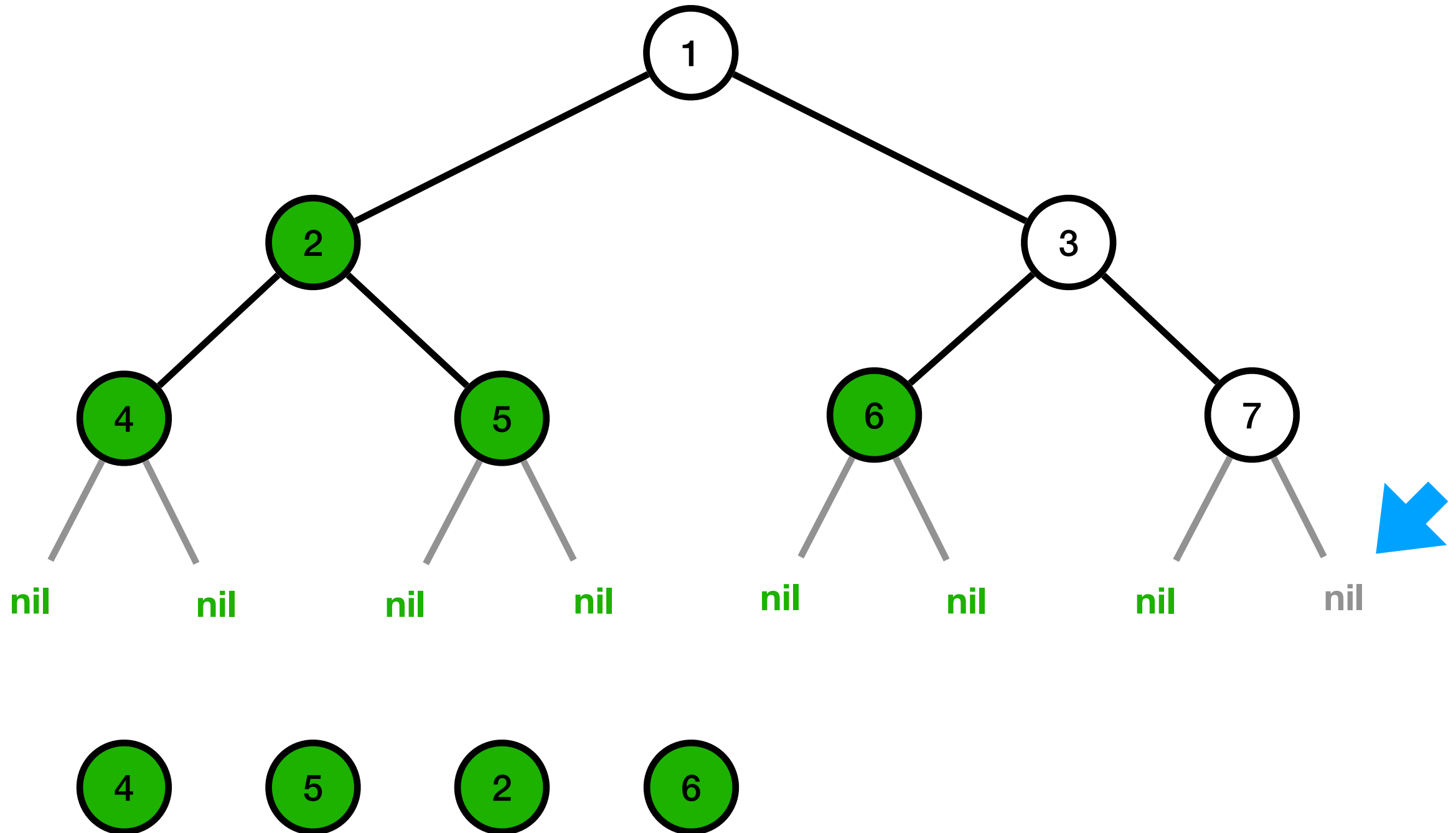
Parcours postfixe



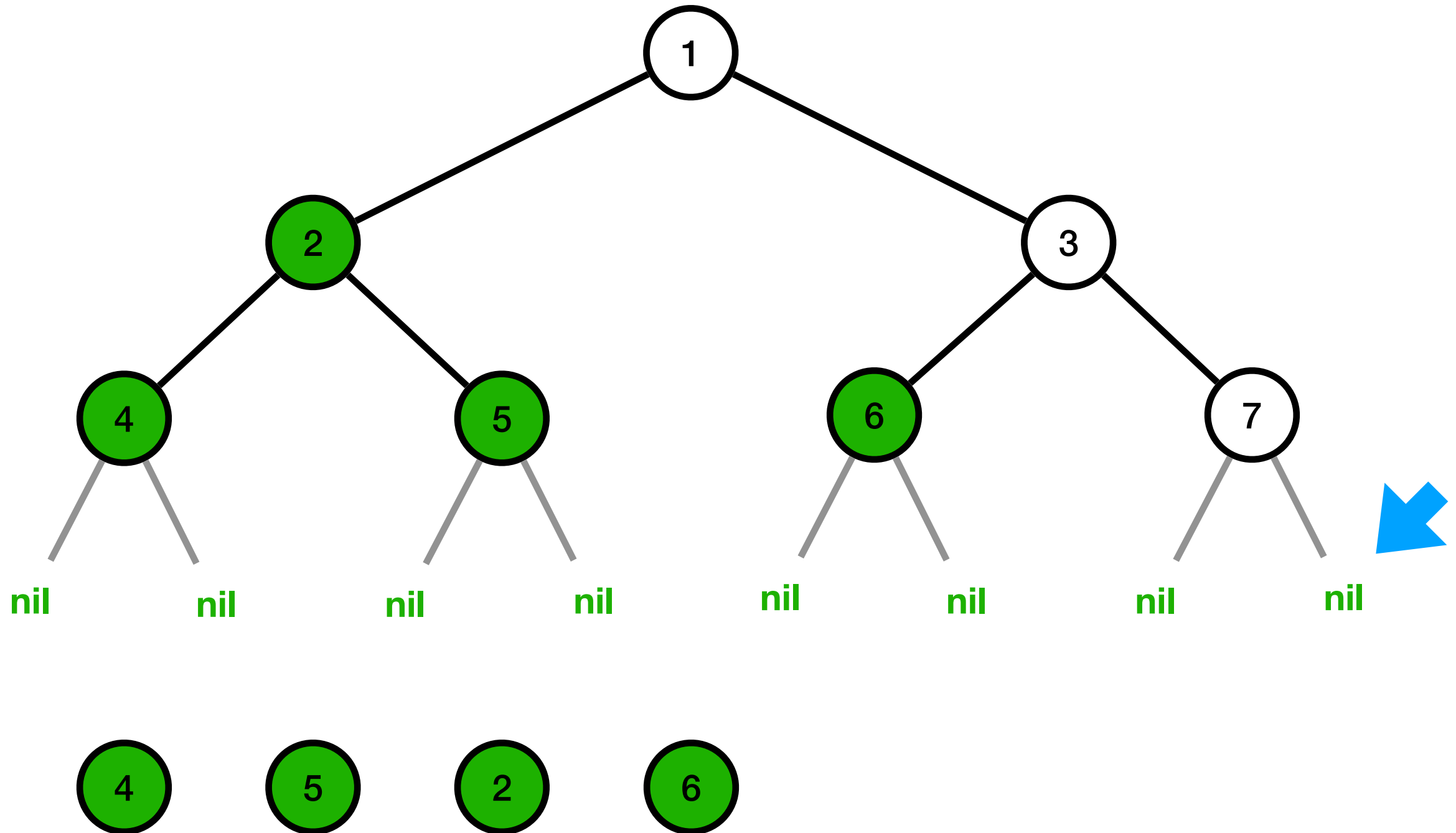
Parcours postfixe



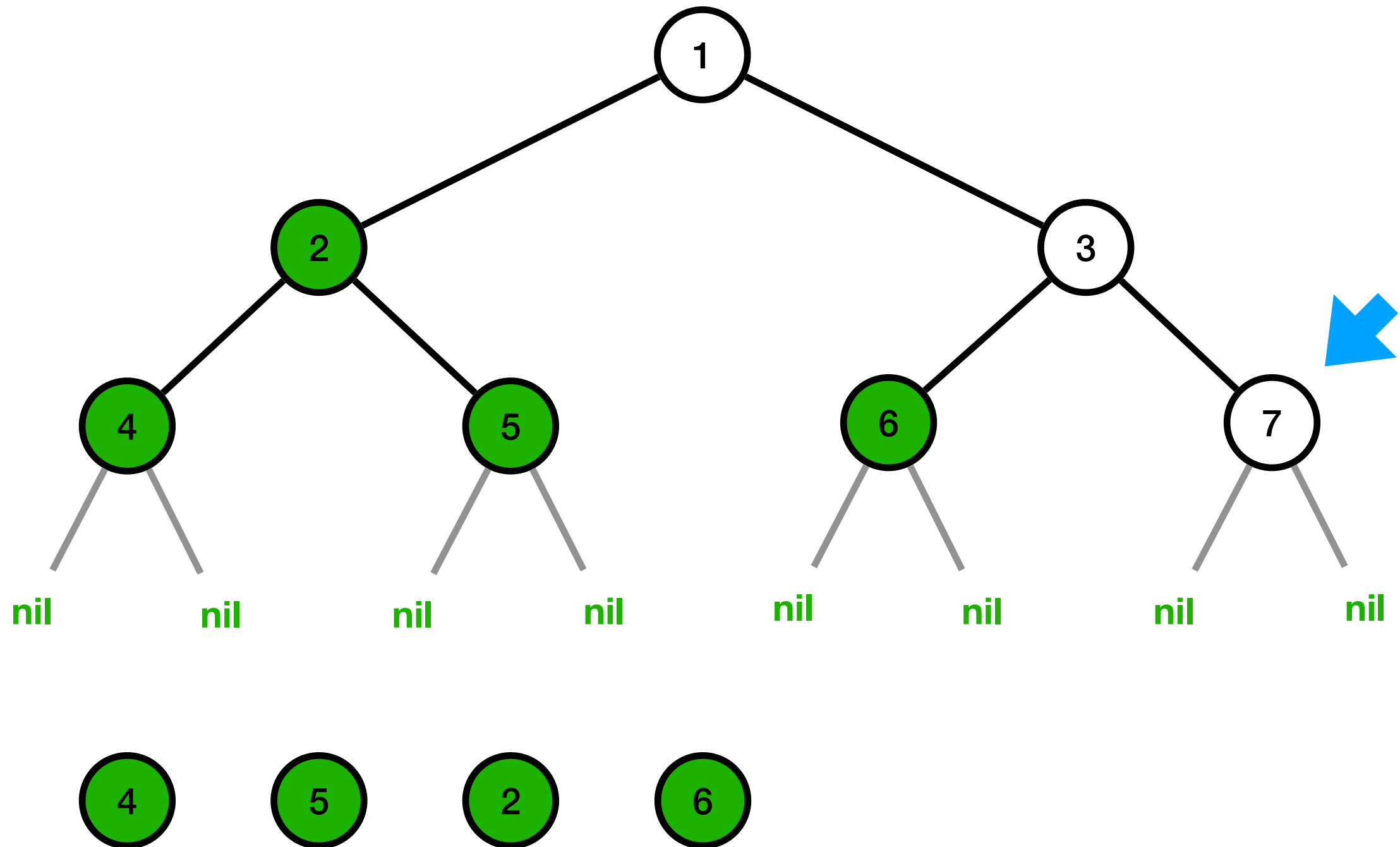
Parcours postfixe



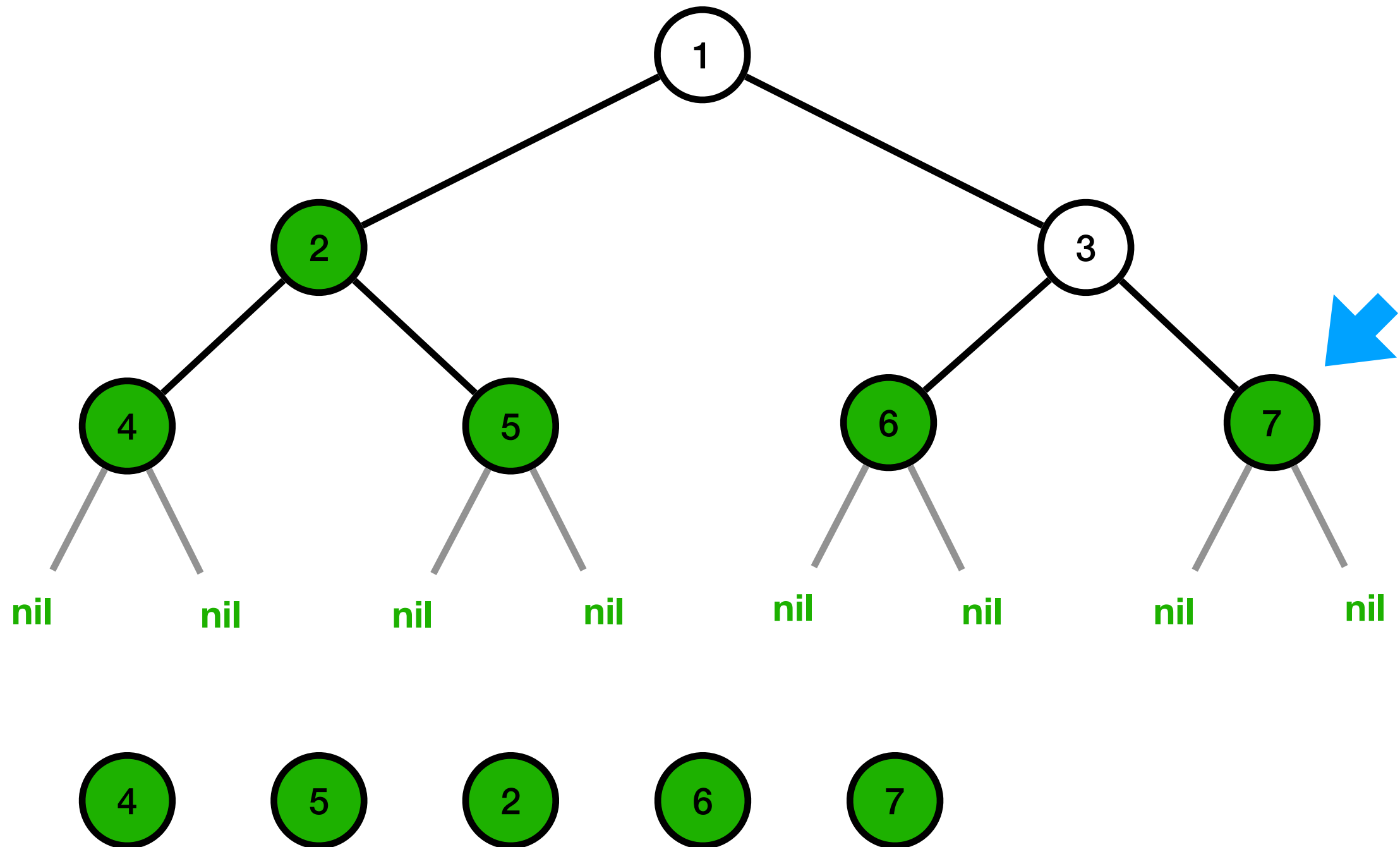
Parcours postfixe



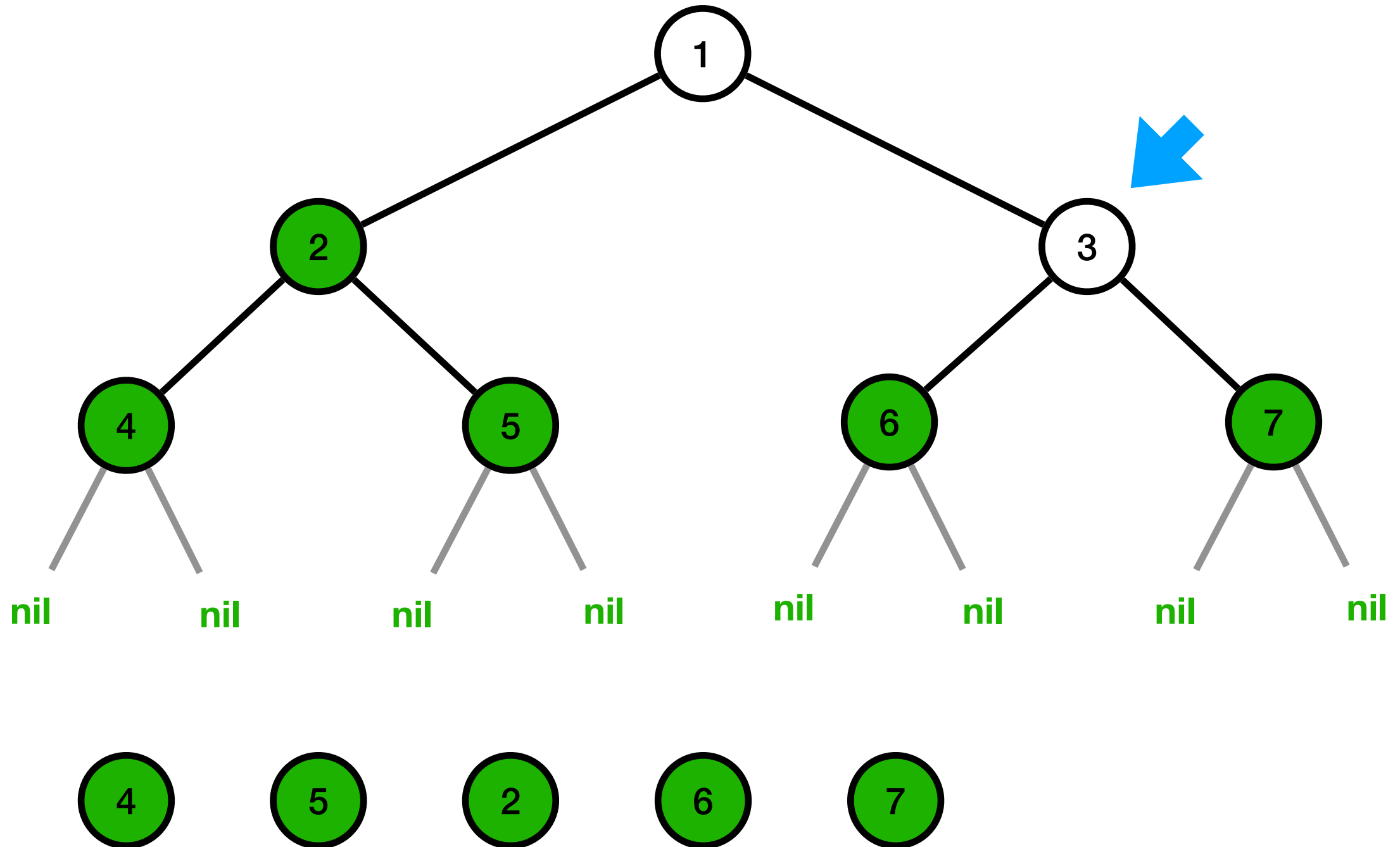
Parcours postfixe



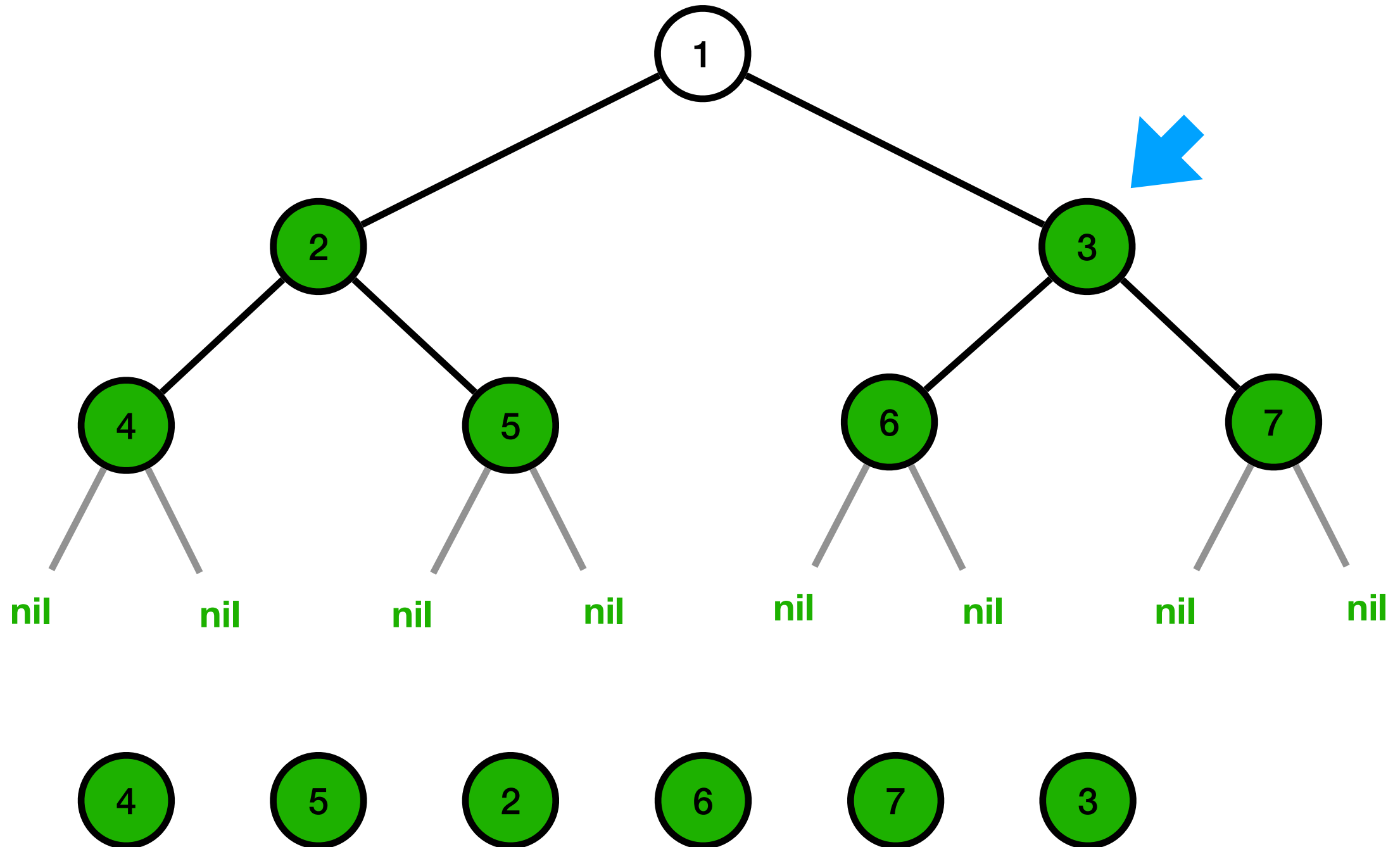
Parcours postfixe



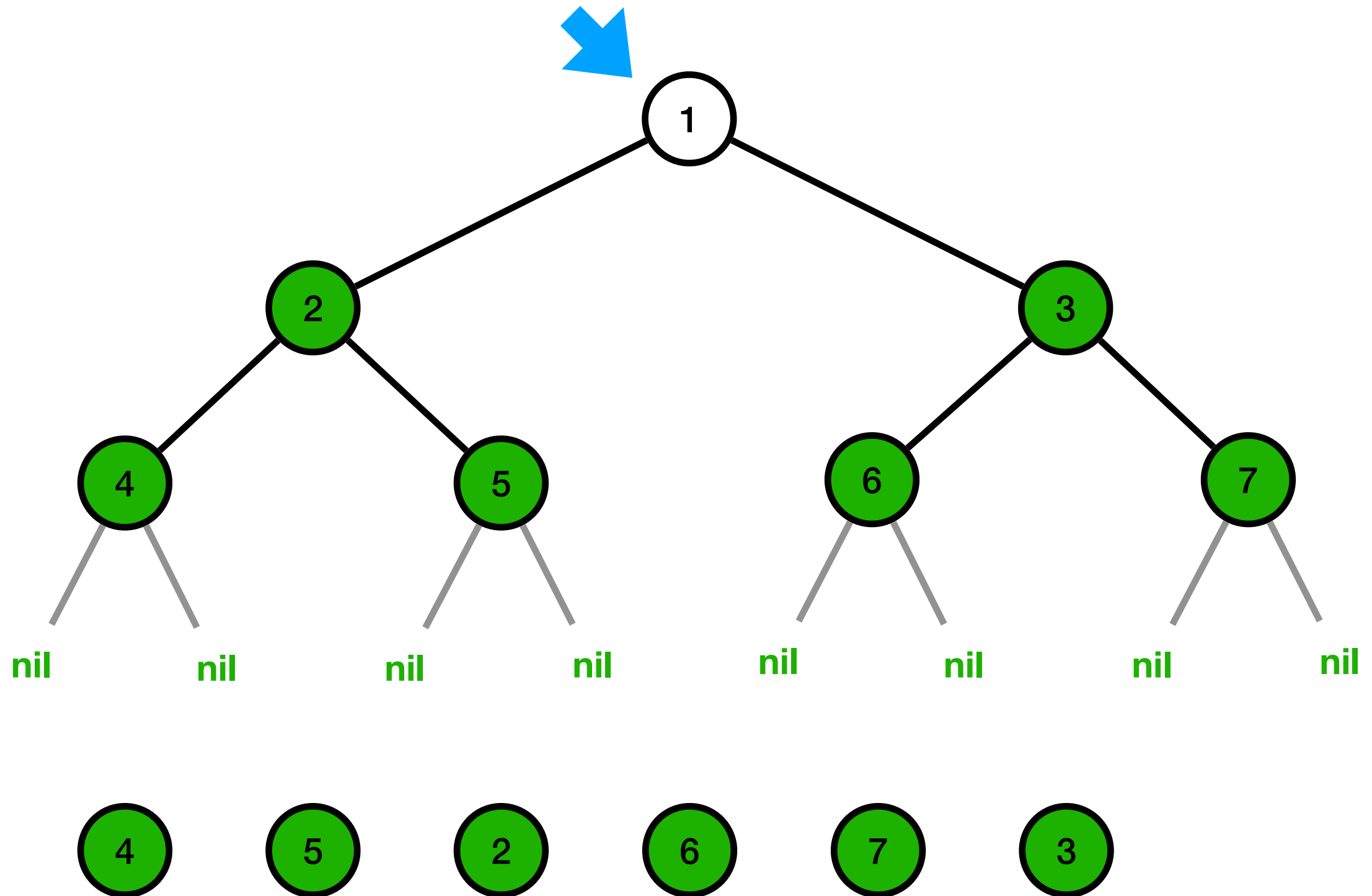
Parcours postfixe



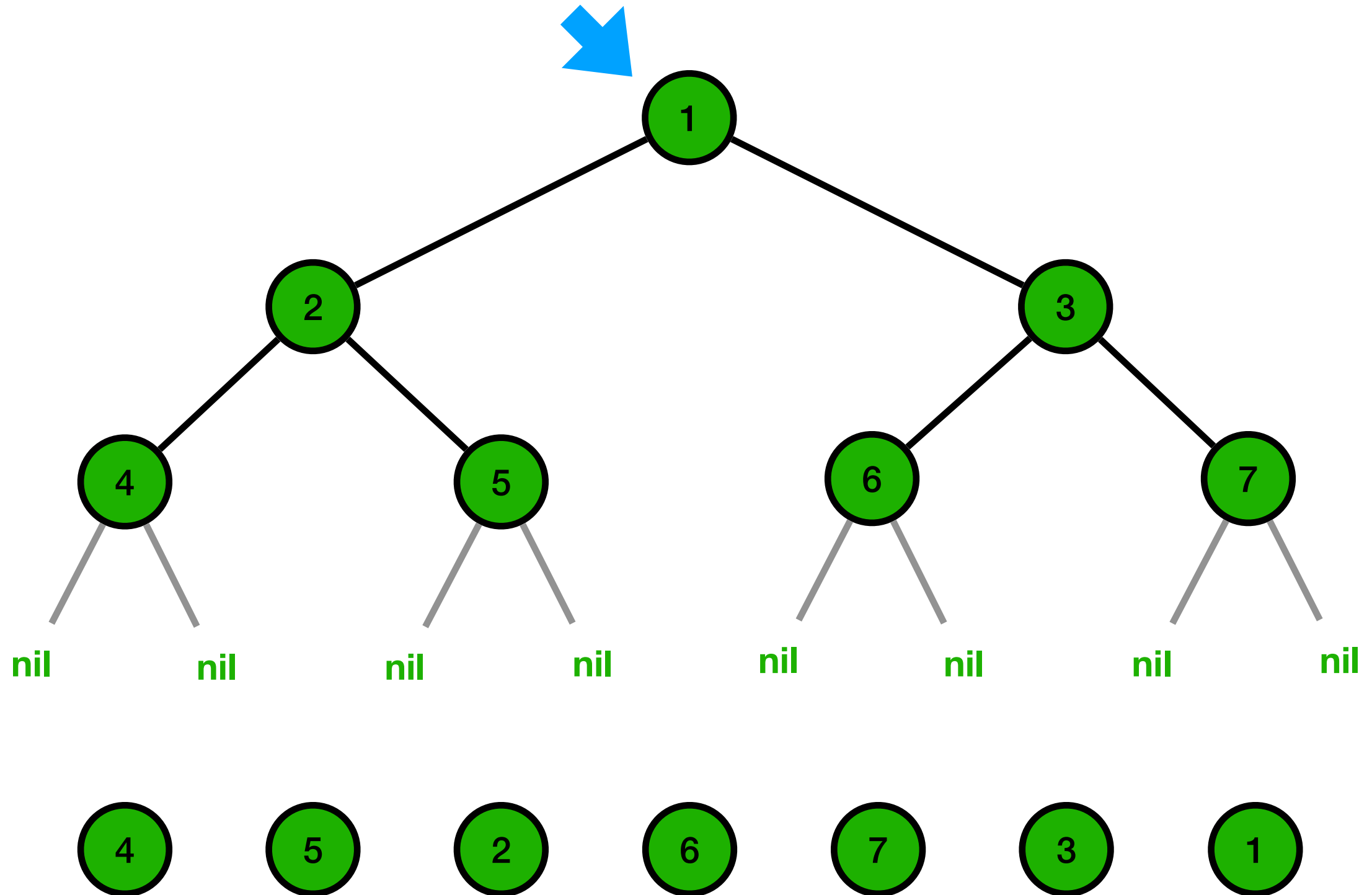
Parcours postfixe



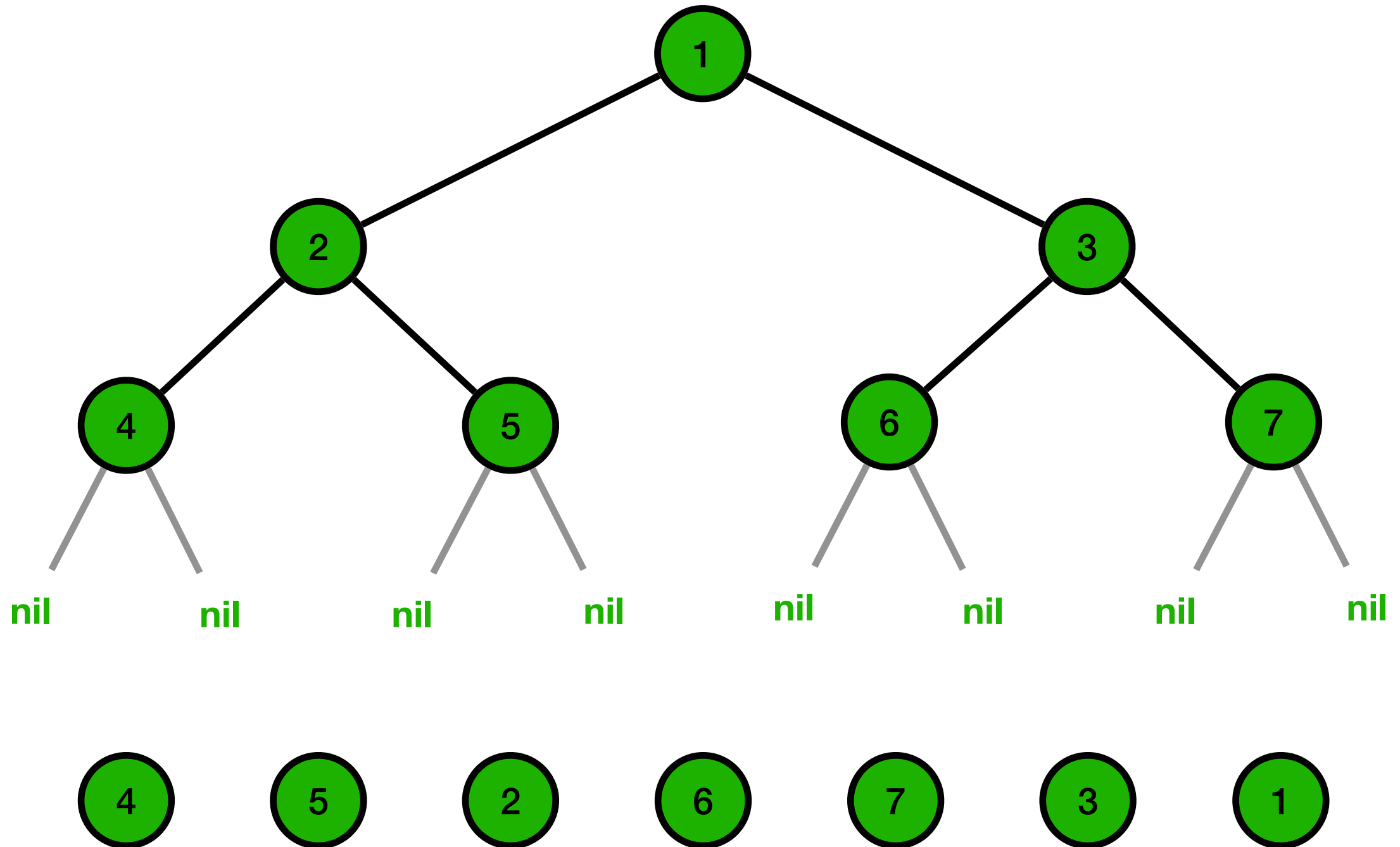
Parcours postfixe



Parcours postfixe



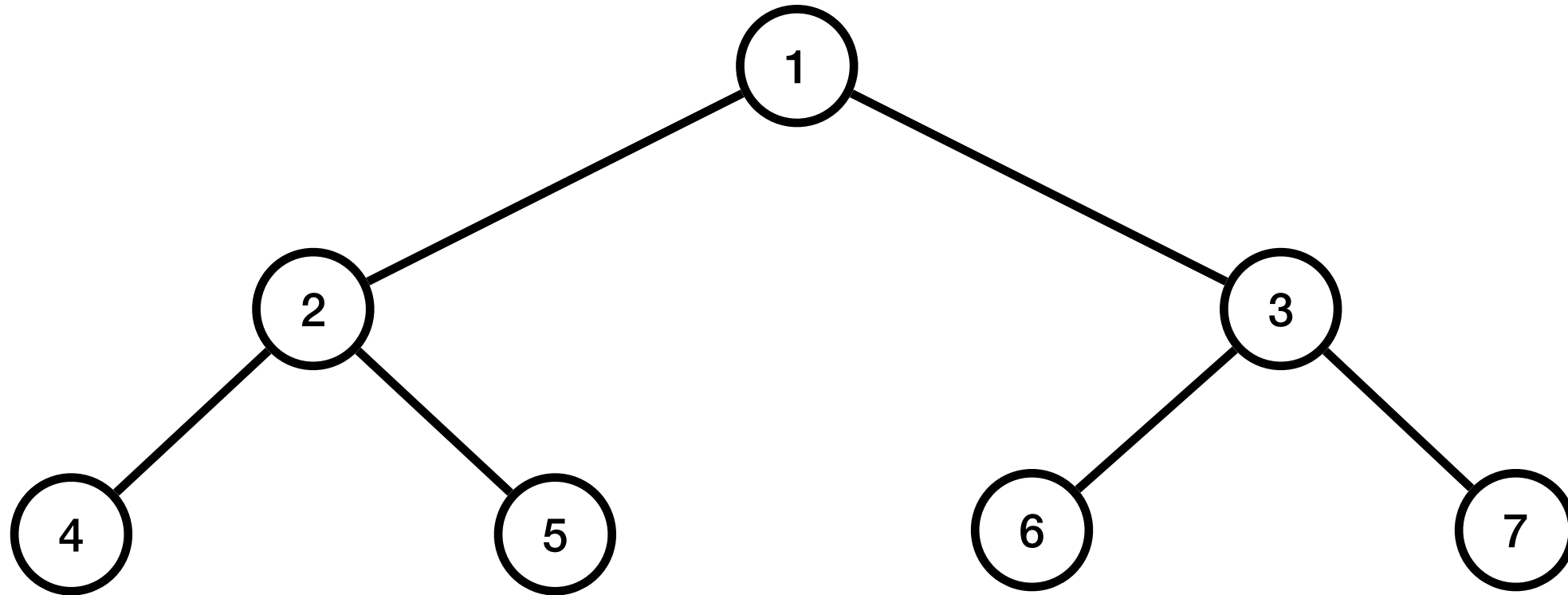
Parcours postfixe



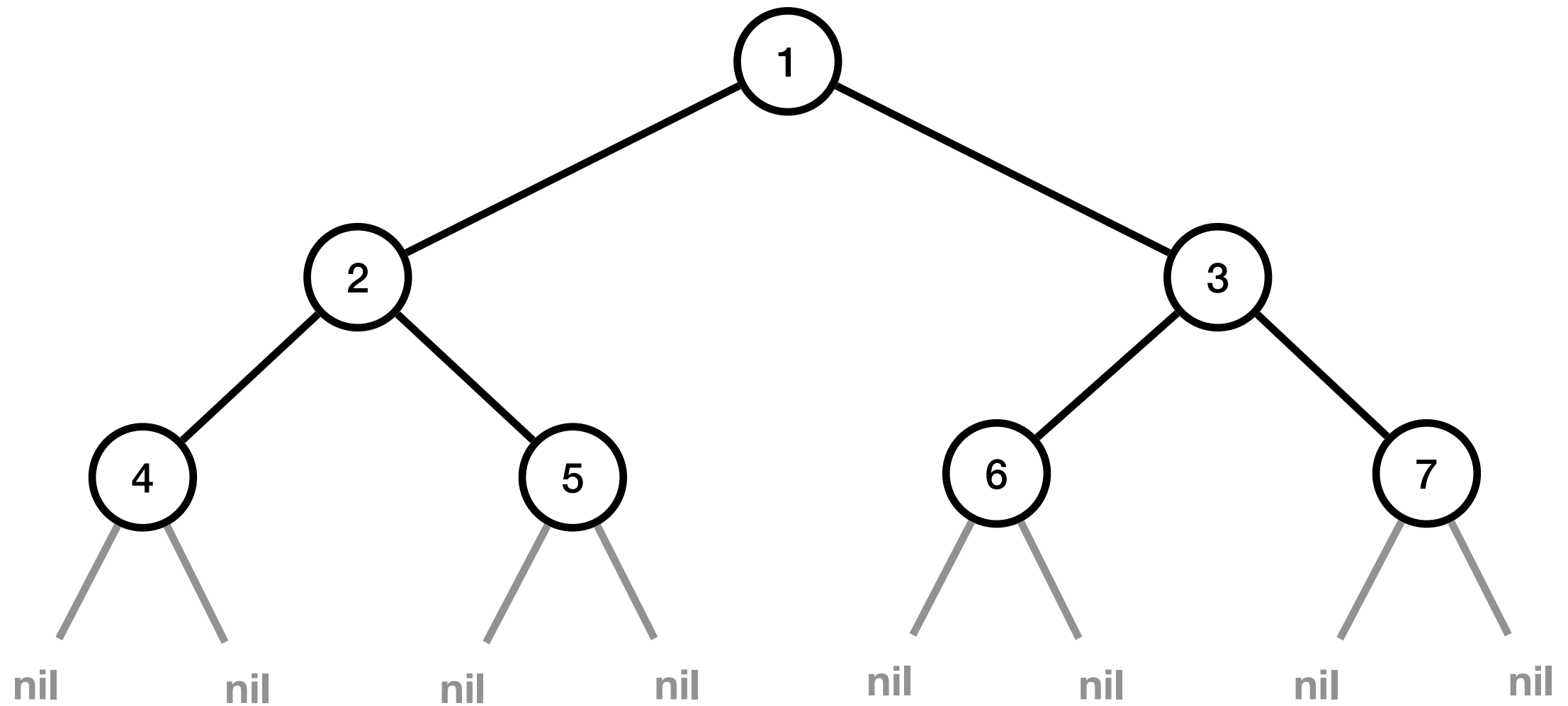
Parcours infixe

```
def parcours_infixe(arbre):  
    if not est_vide(arbre):  
        parcours_infixe(gauche(arbre))  
        print(valeur(arbre))  
        parcours_infixe(droite(arbre))
```

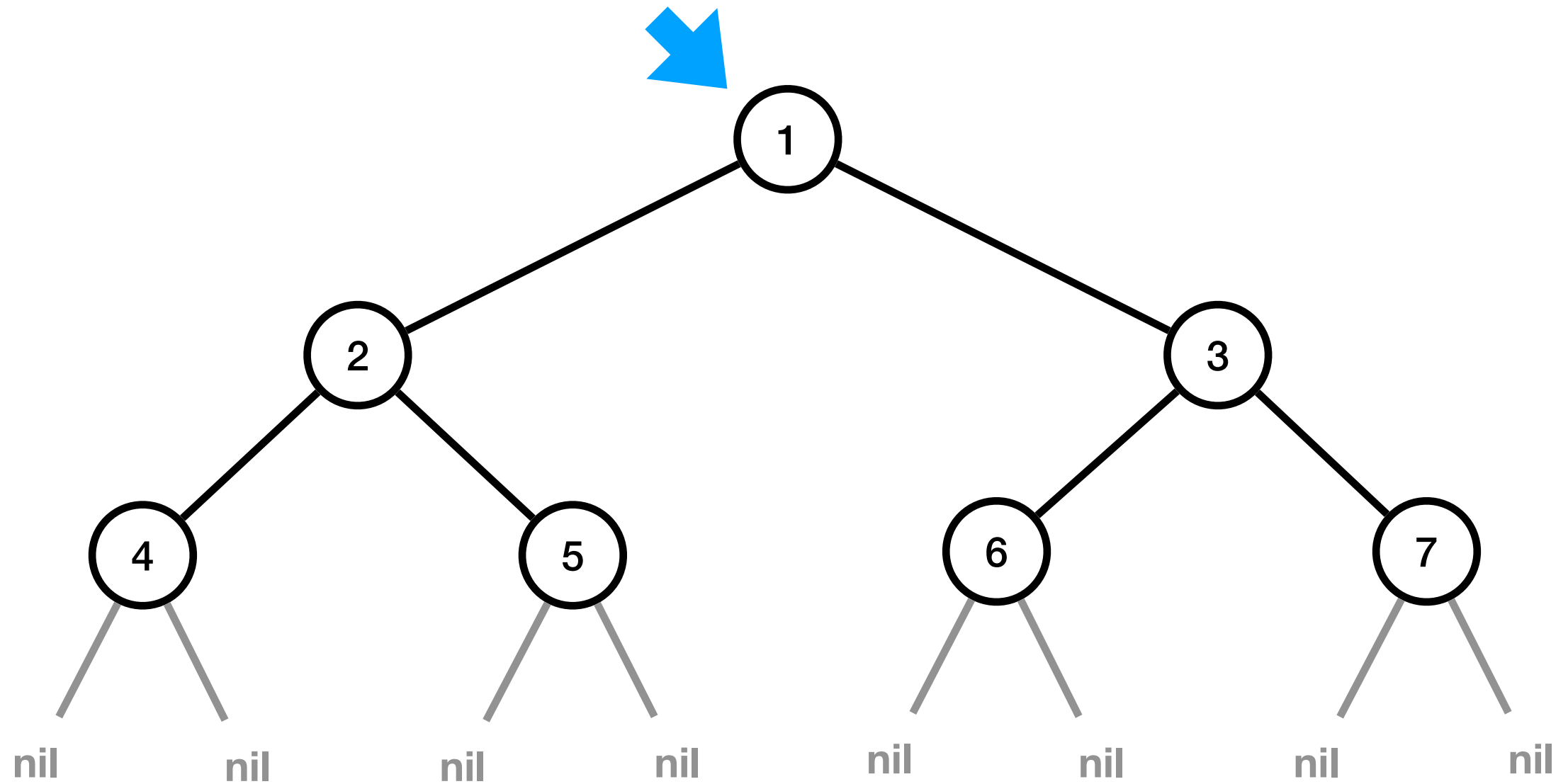
Parcours infixe



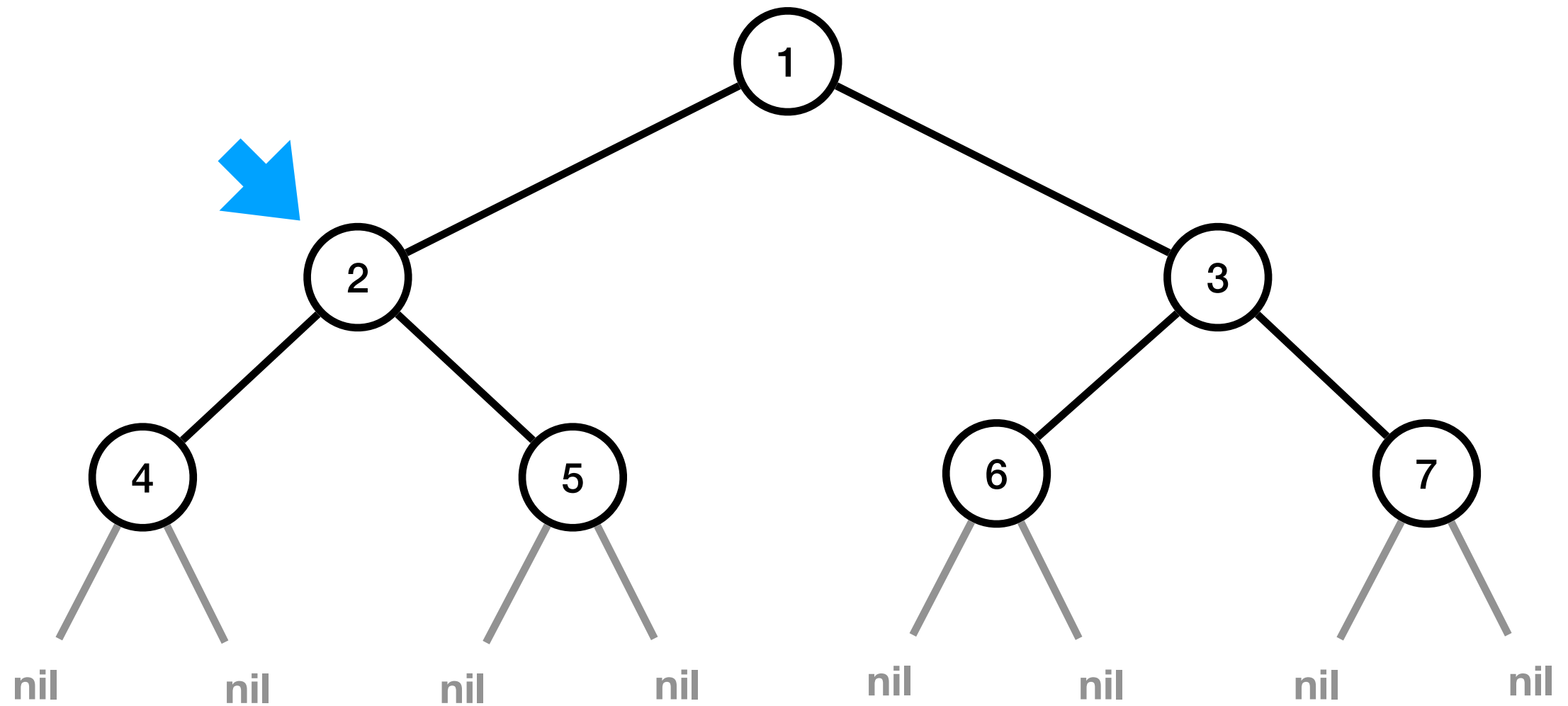
Parcours infixe



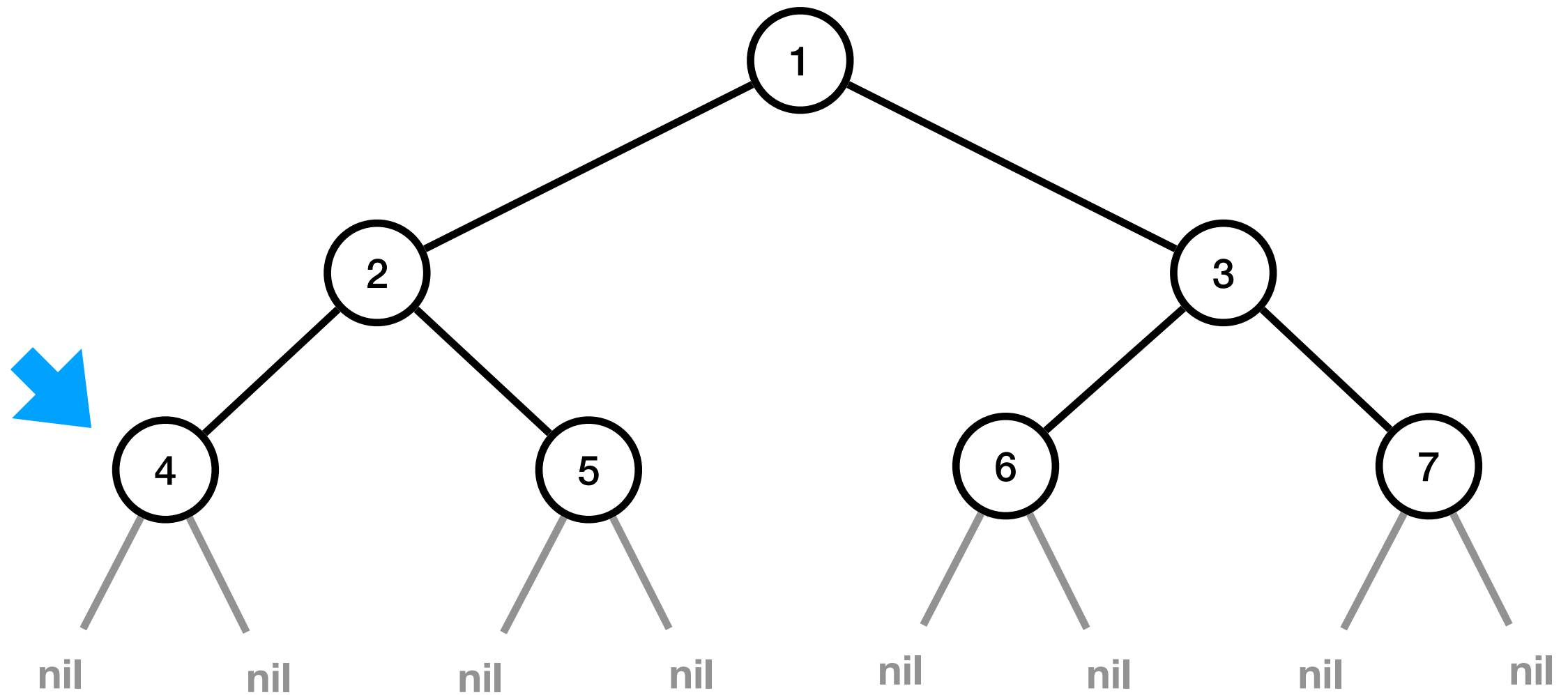
Parcours infixe



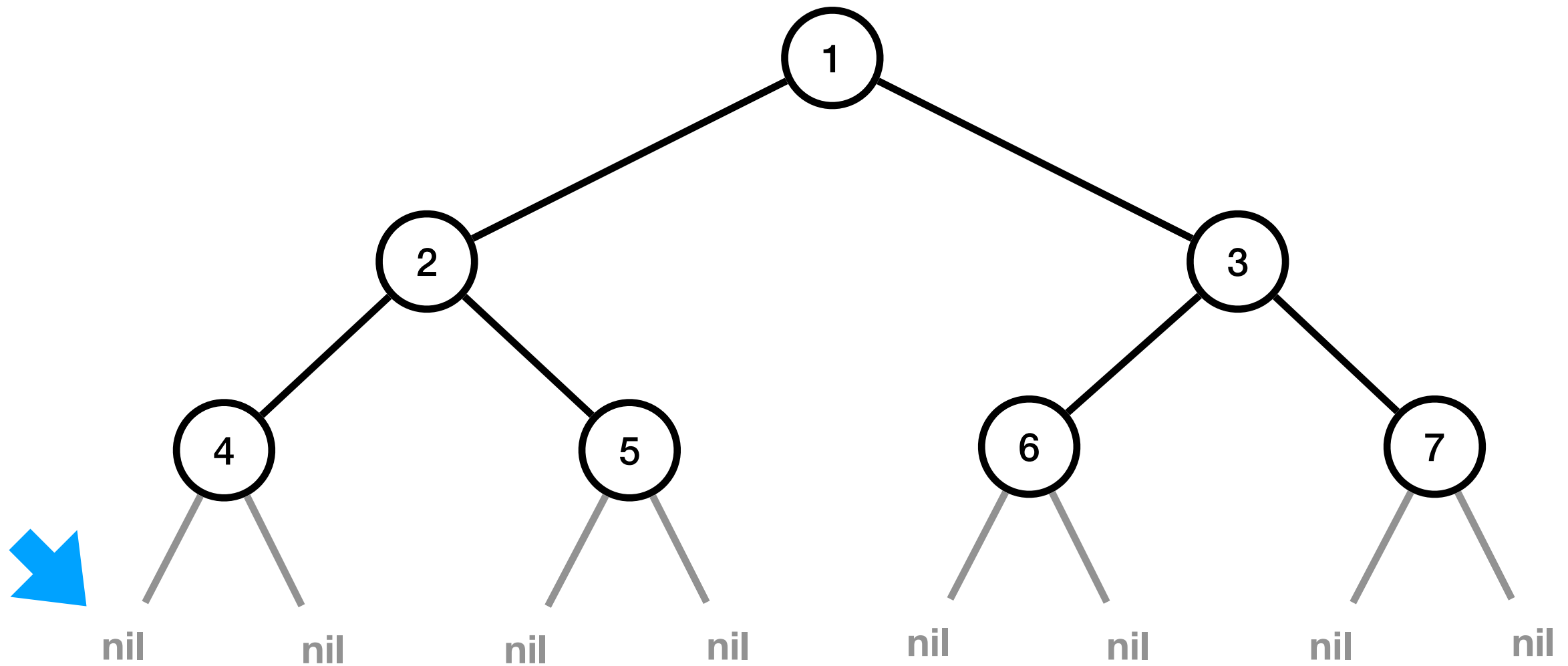
Parcours infixe



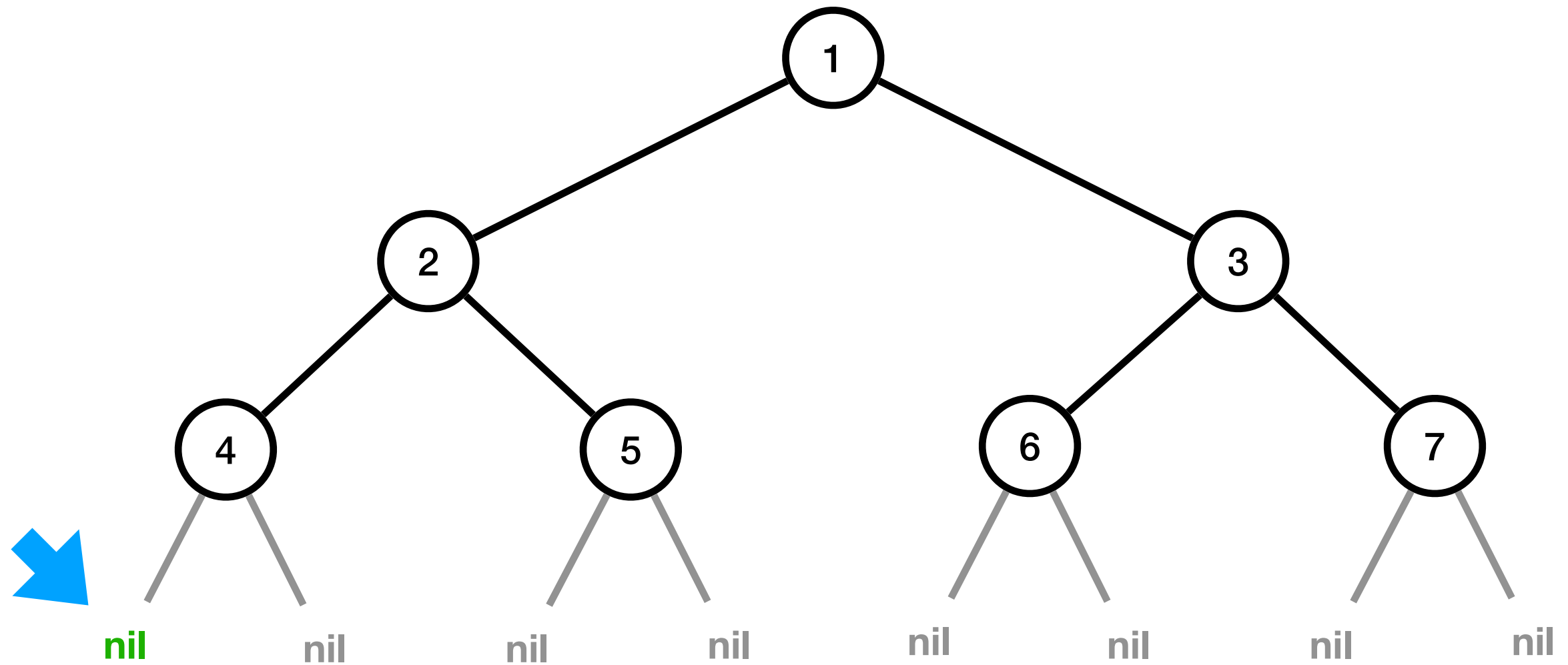
Parcours infixe



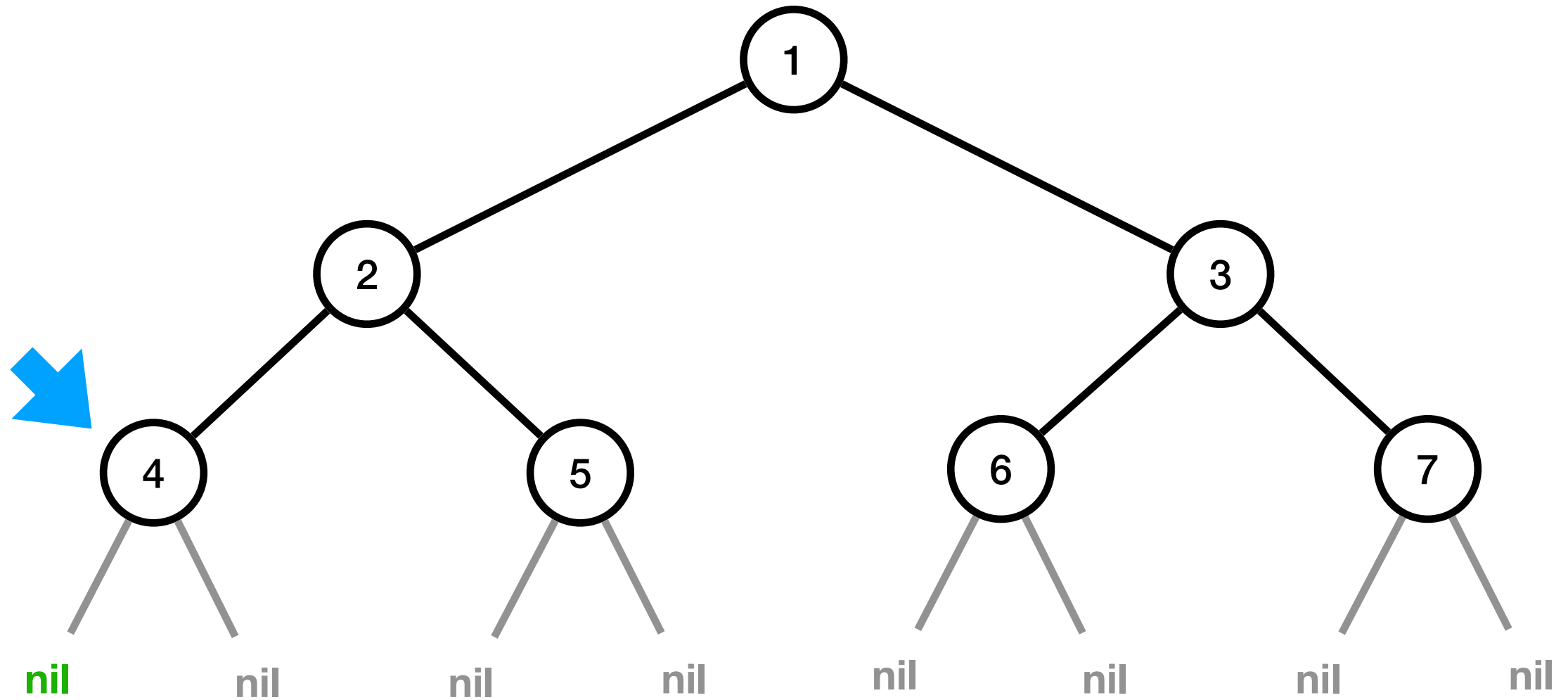
Parcours infixe



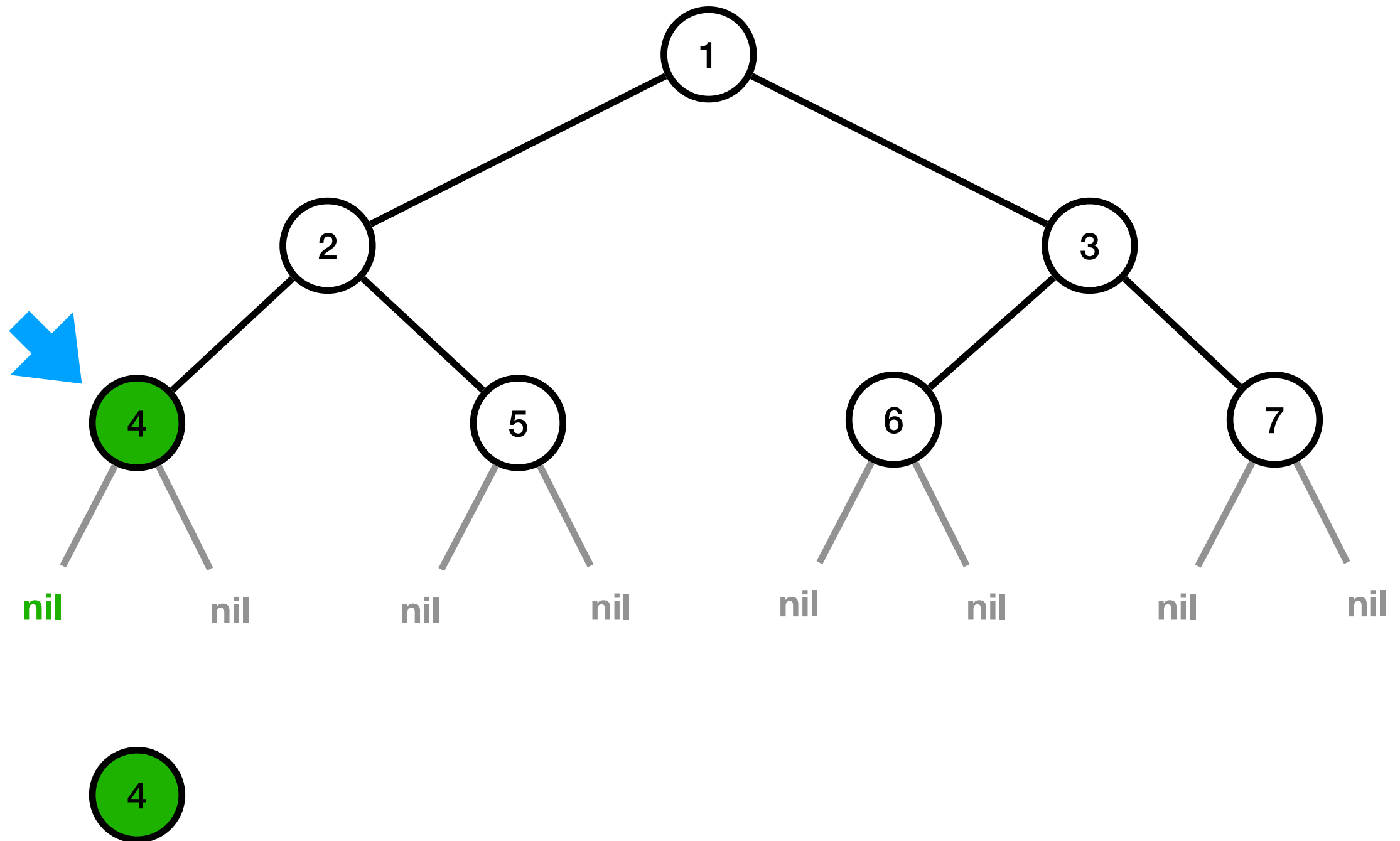
Parcours infixe



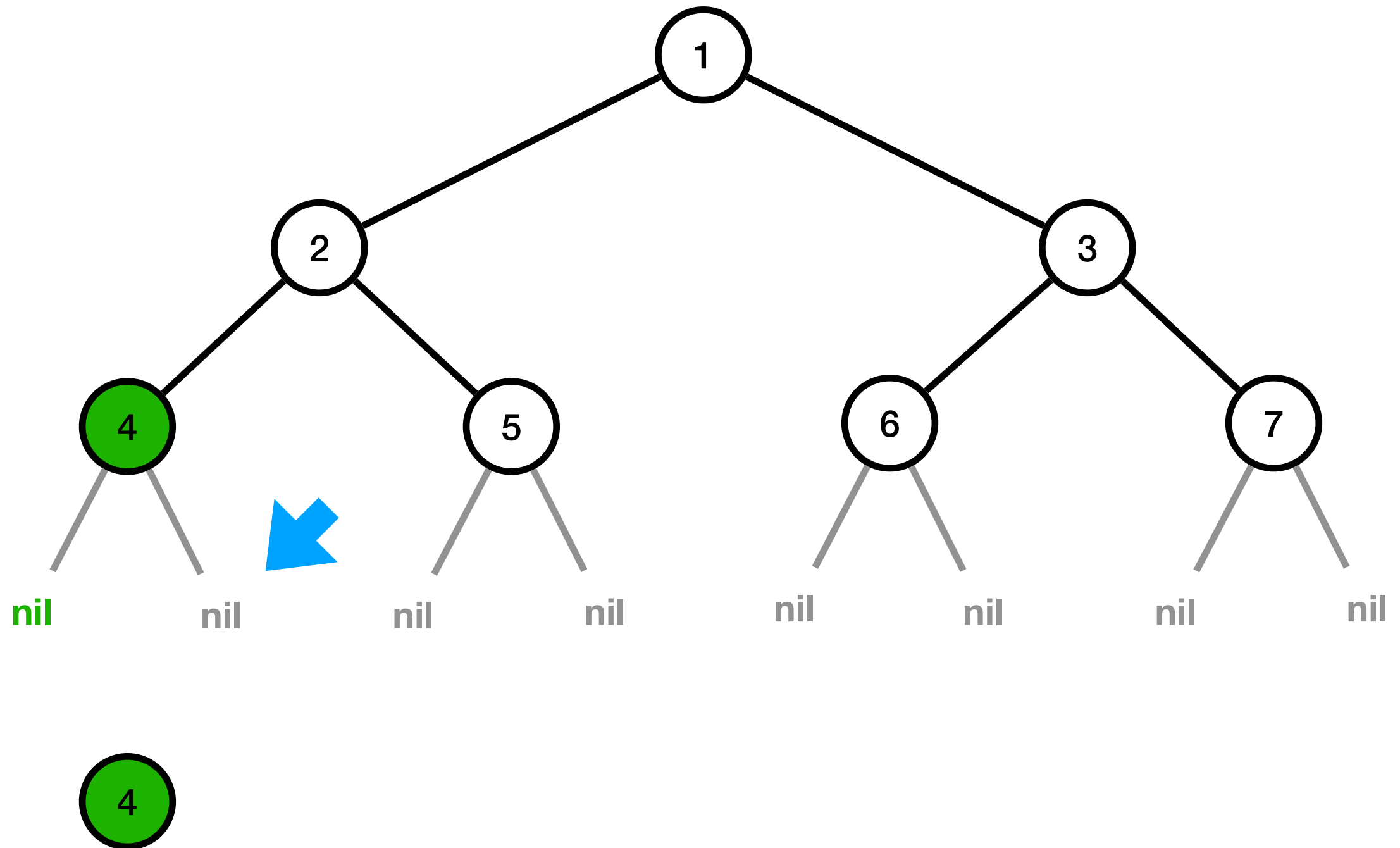
Parcours infixe



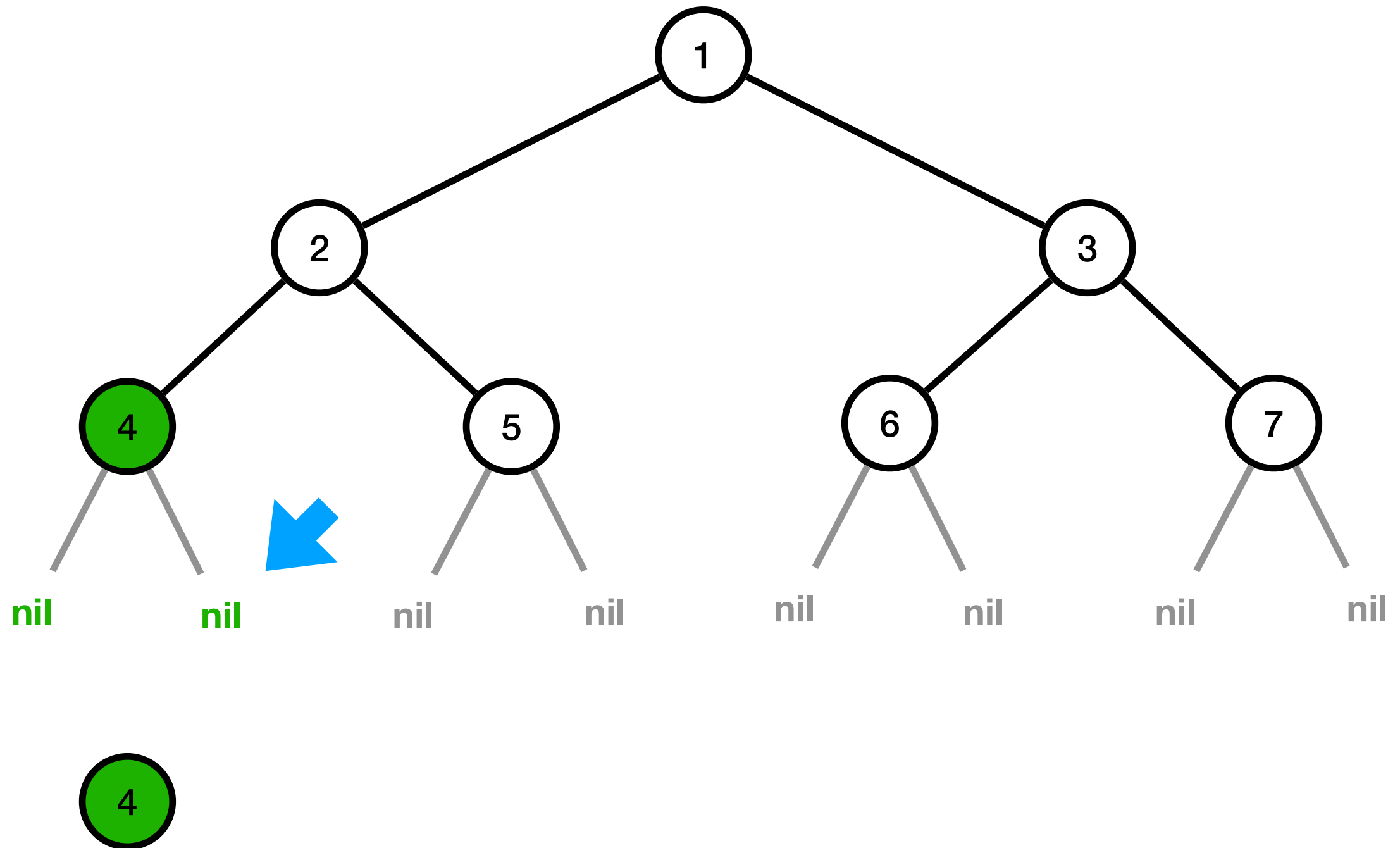
Parcours infixe



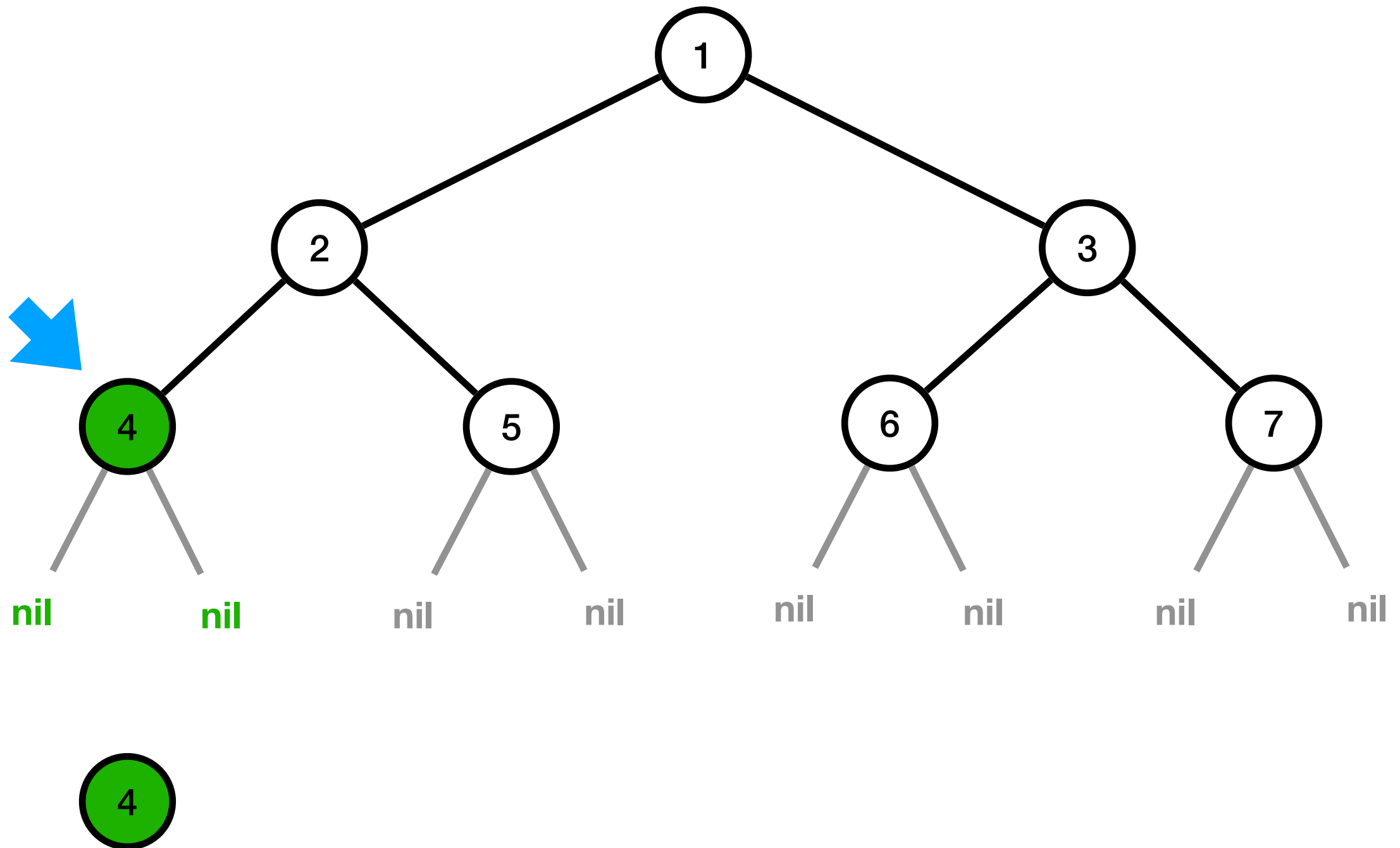
Parcours infixe



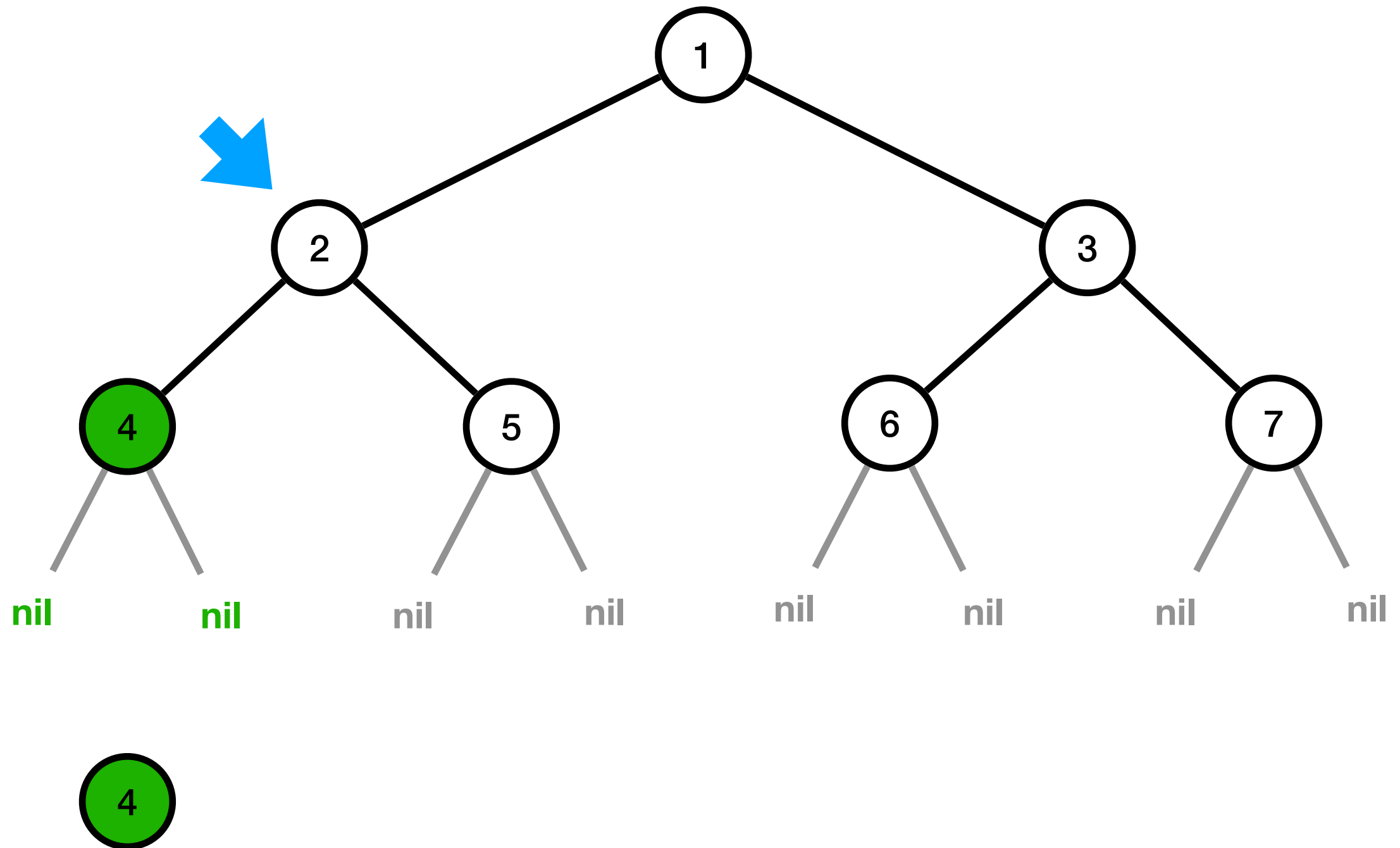
Parcours infixe



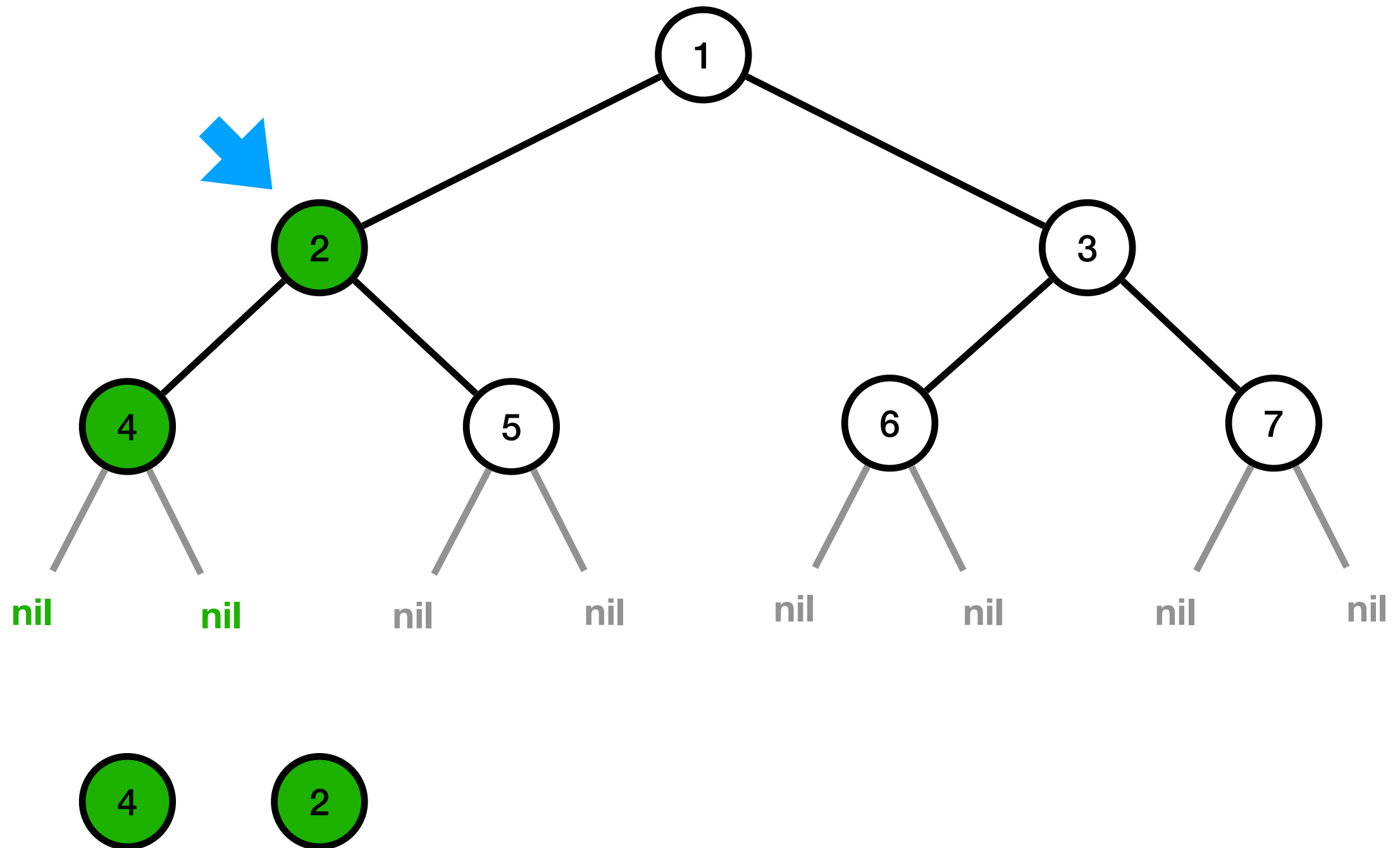
Parcours infixe



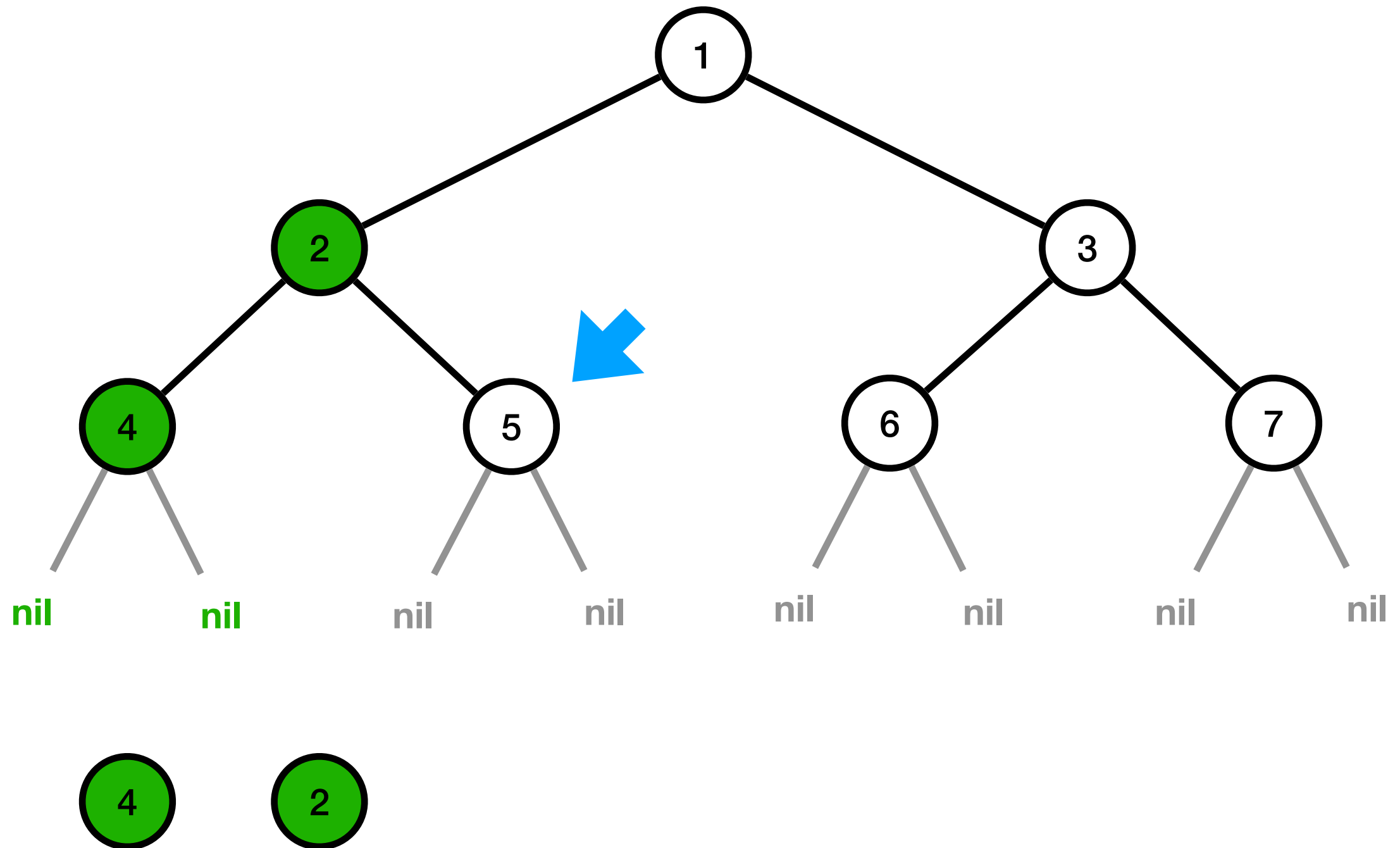
Parcours infixe



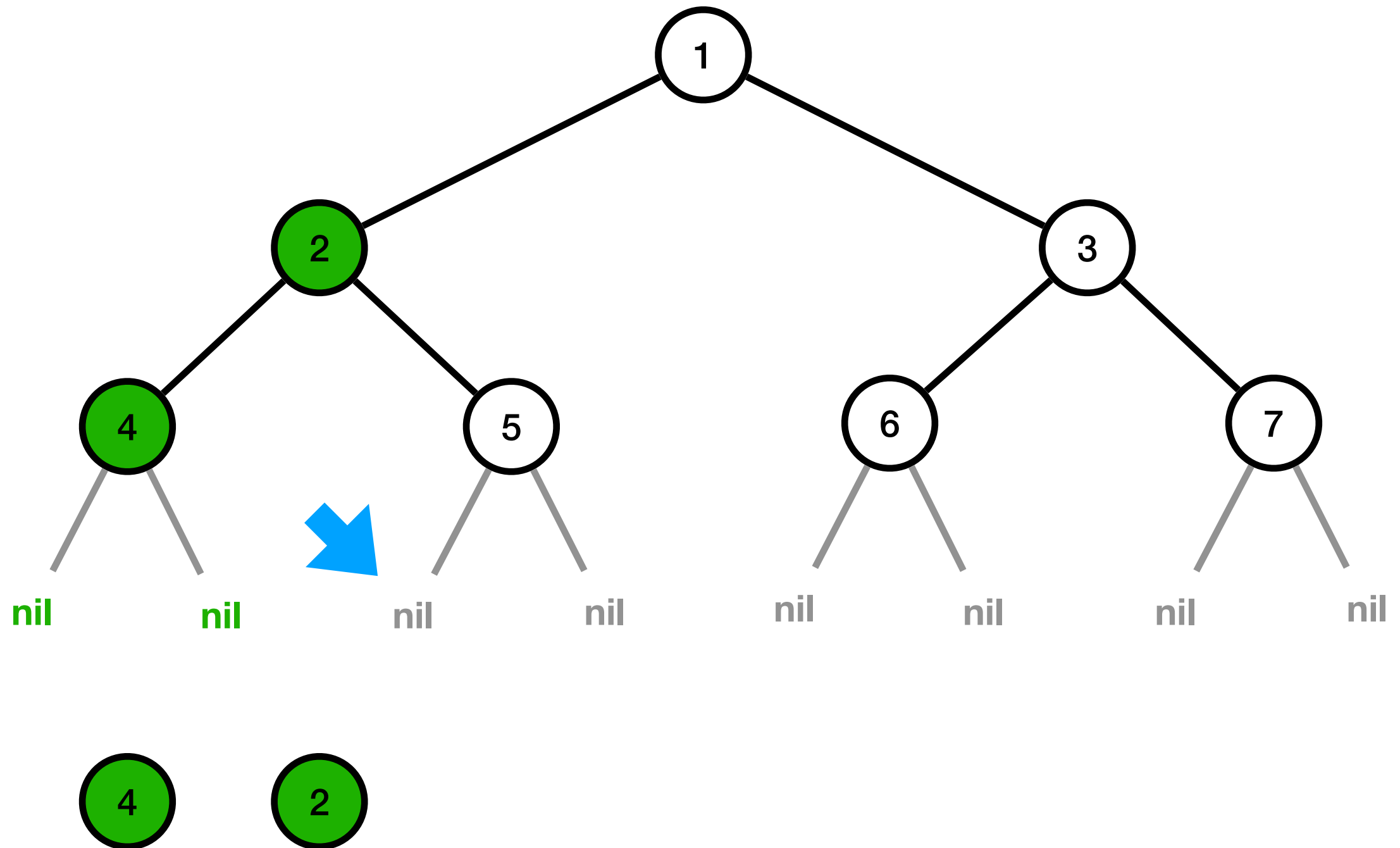
Parcours infixe



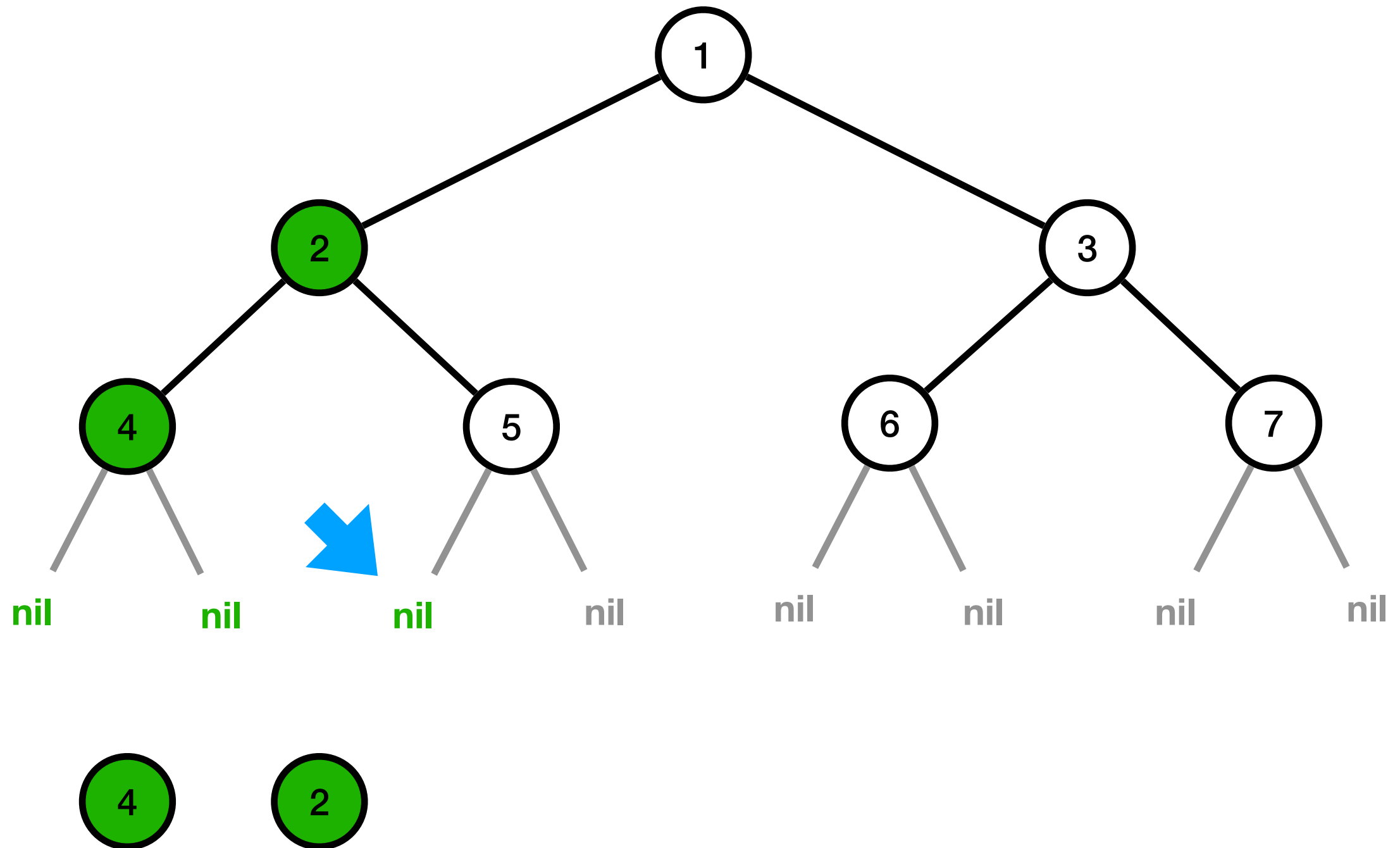
Parcours infixe



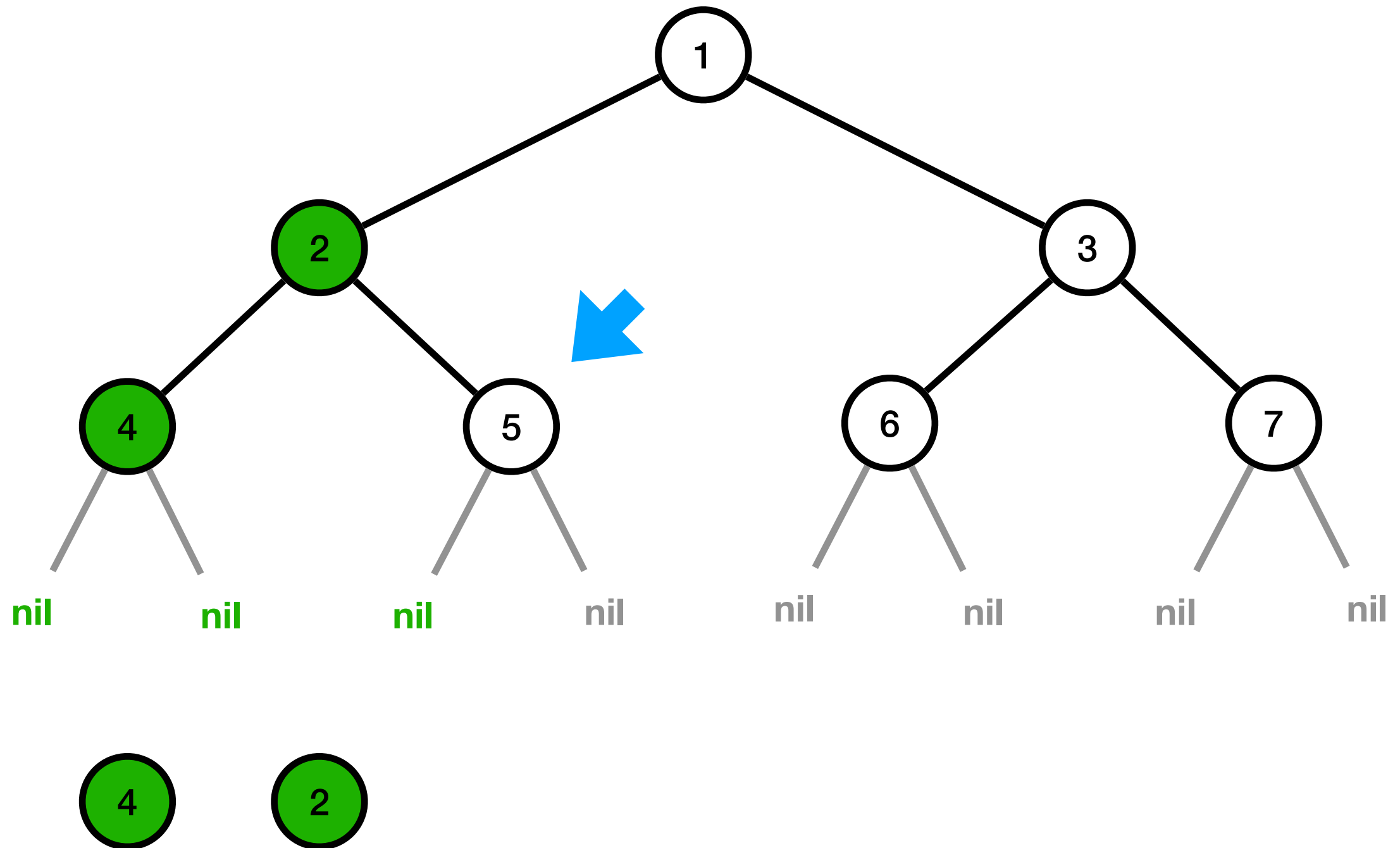
Parcours infixe



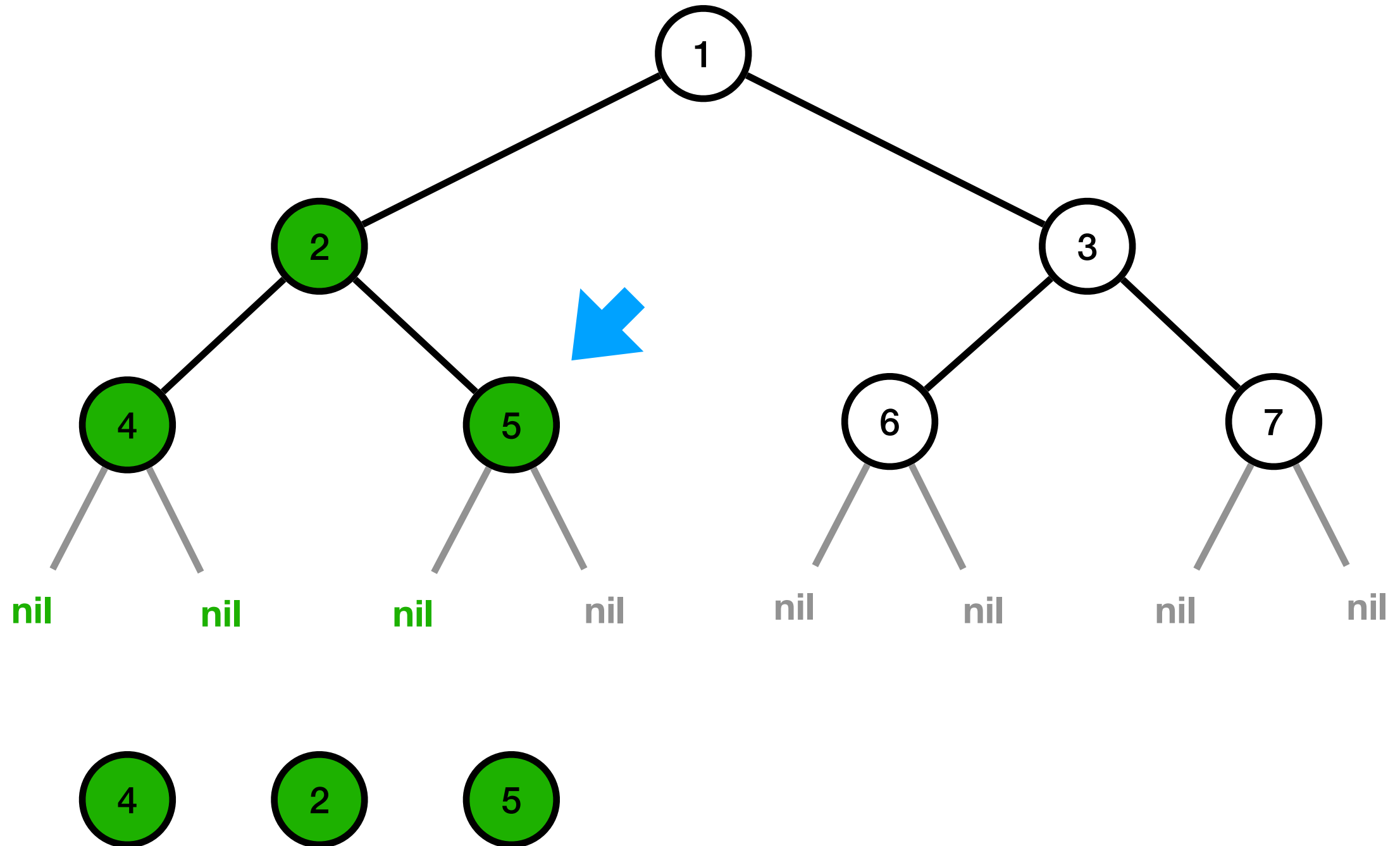
Parcours infixe



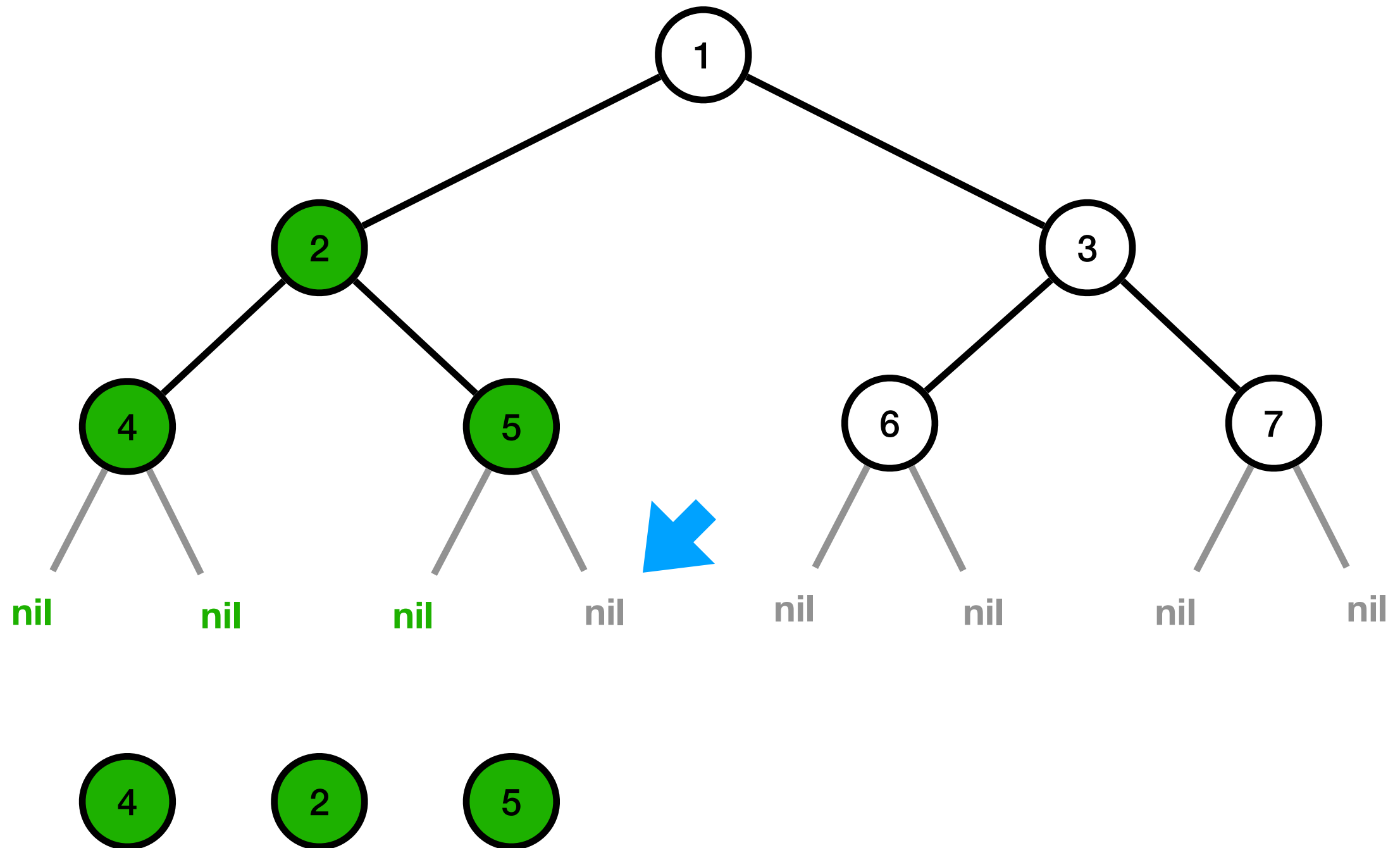
Parcours infixe



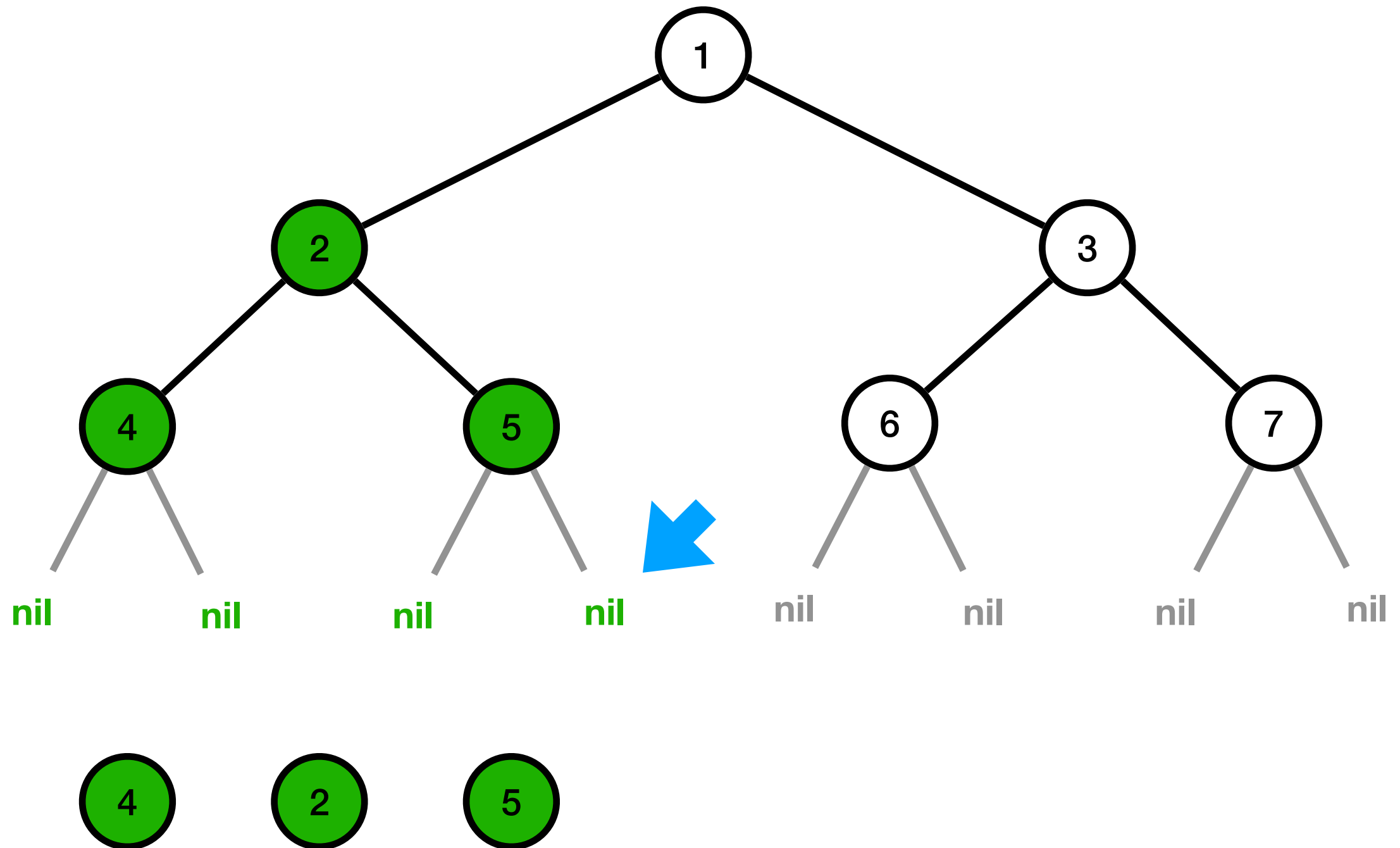
Parcours infixe



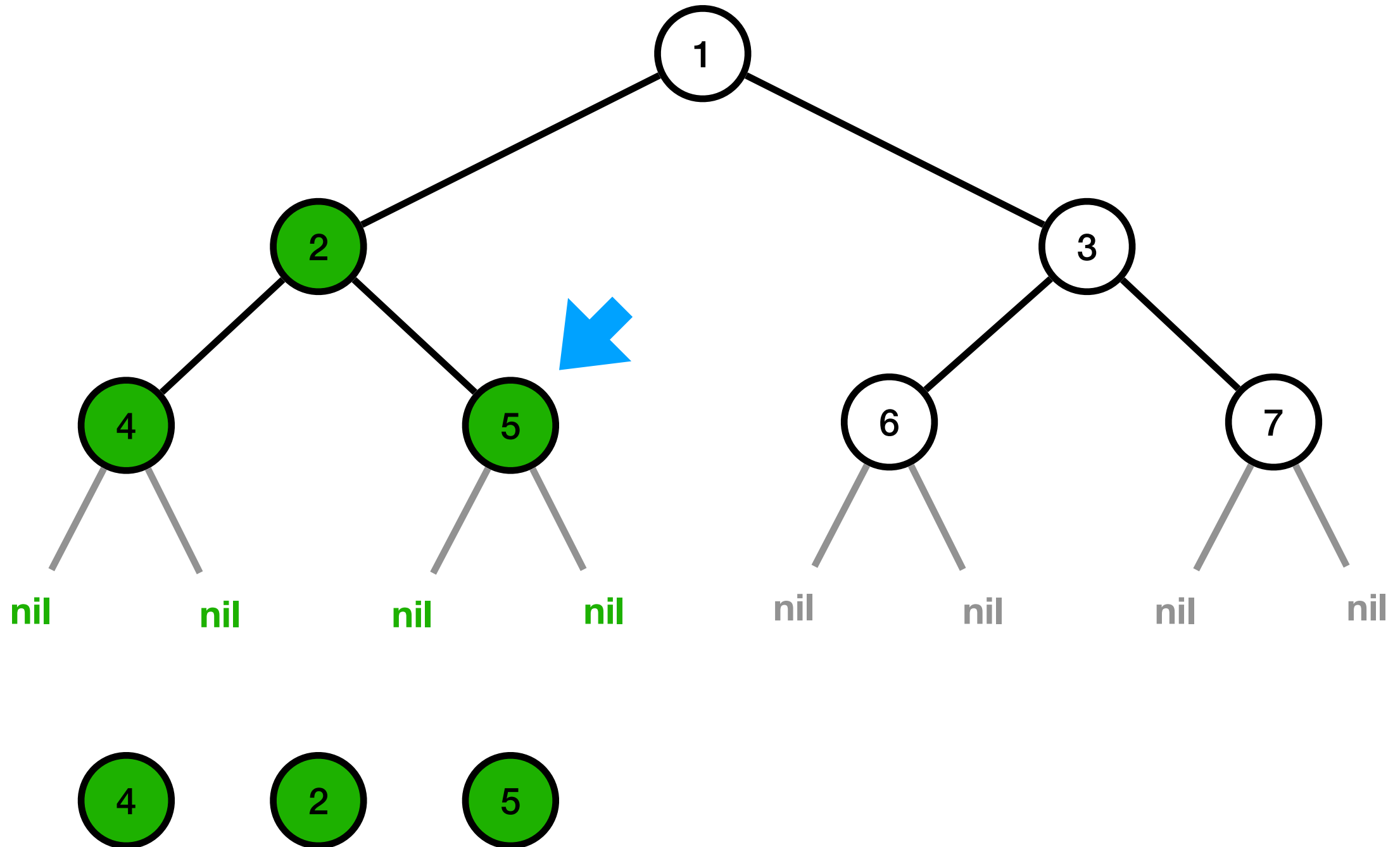
Parcours infixe



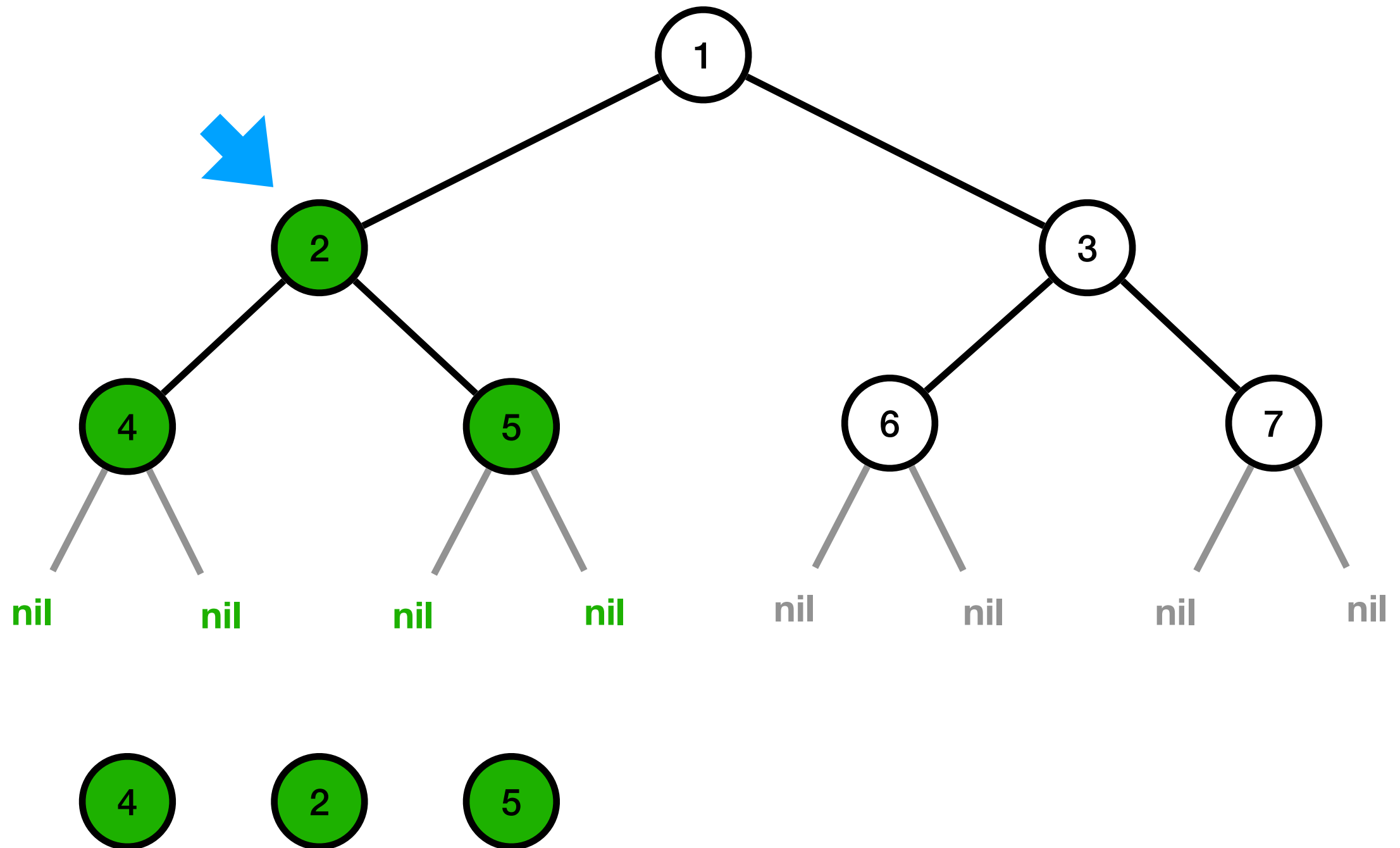
Parcours infixe



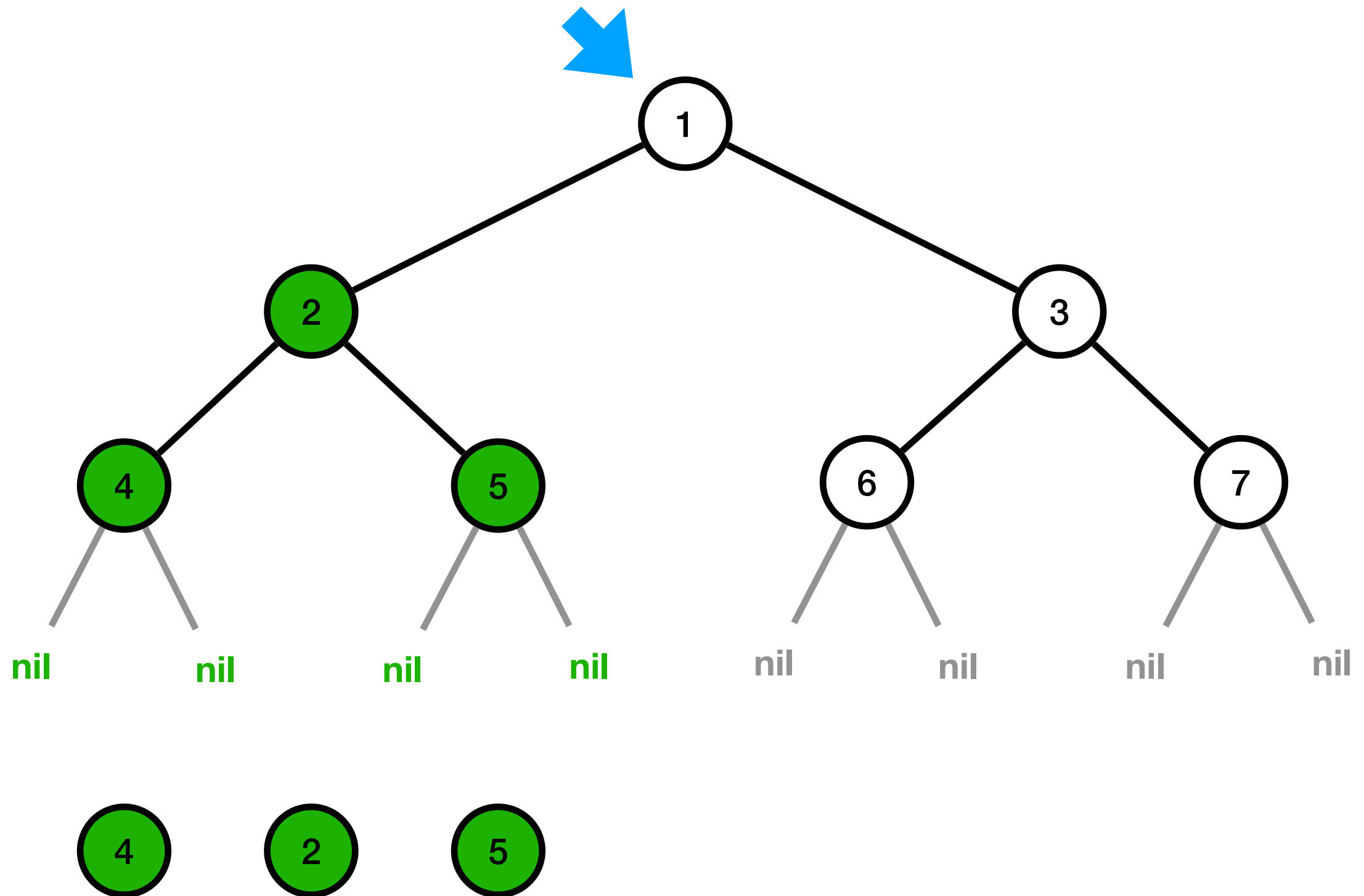
Parcours infixe



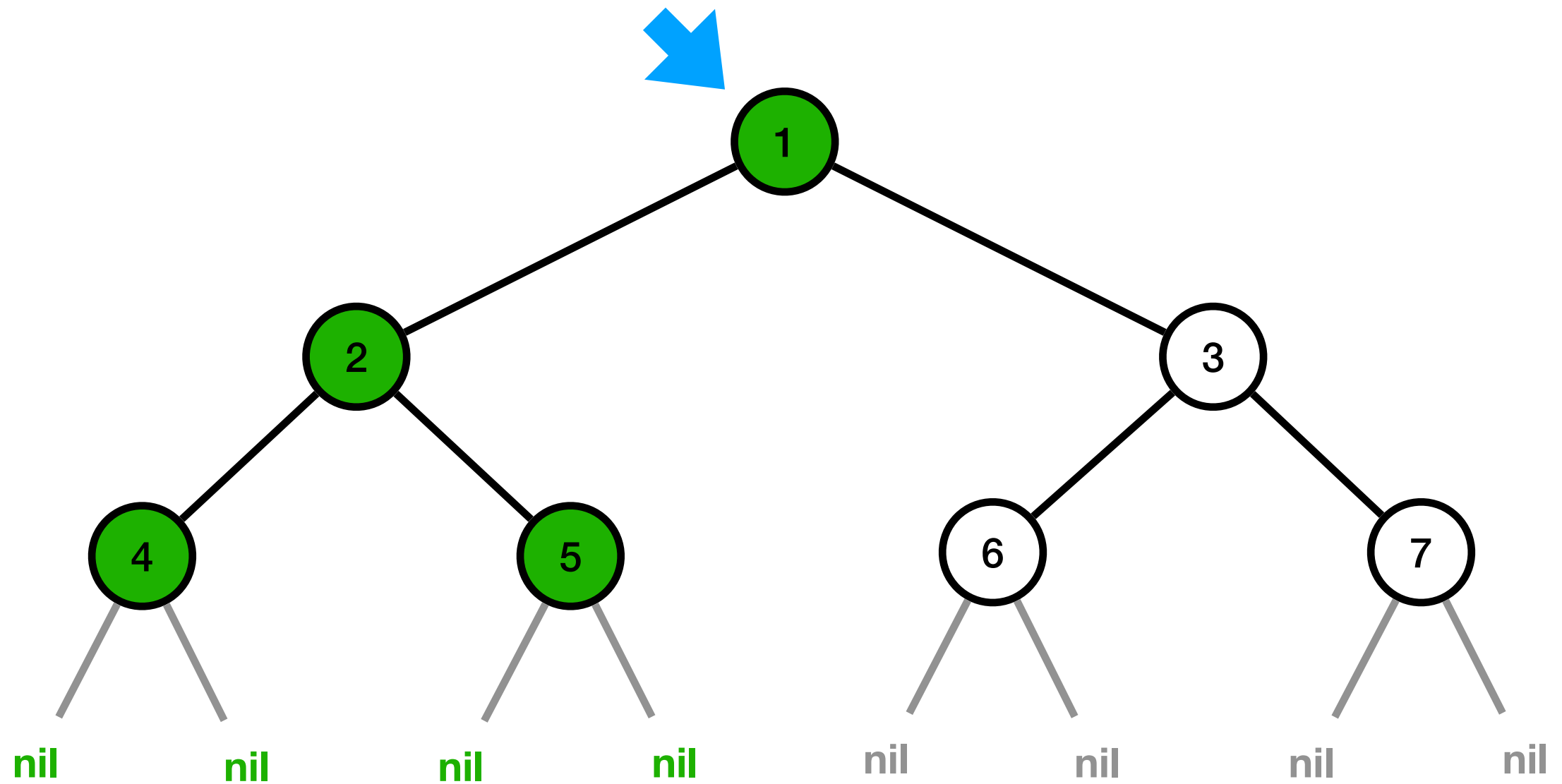
Parcours infixe



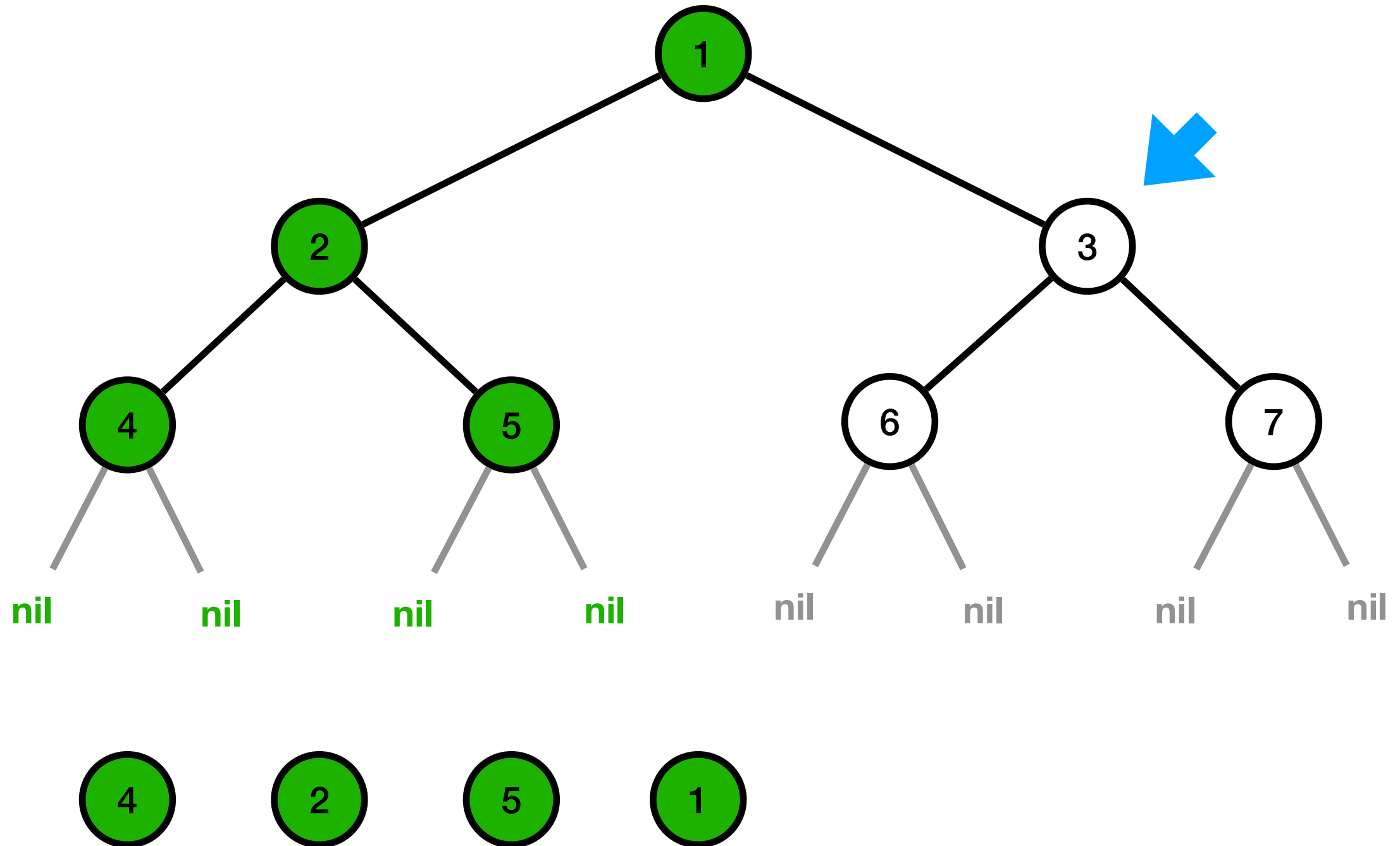
Parcours infixe



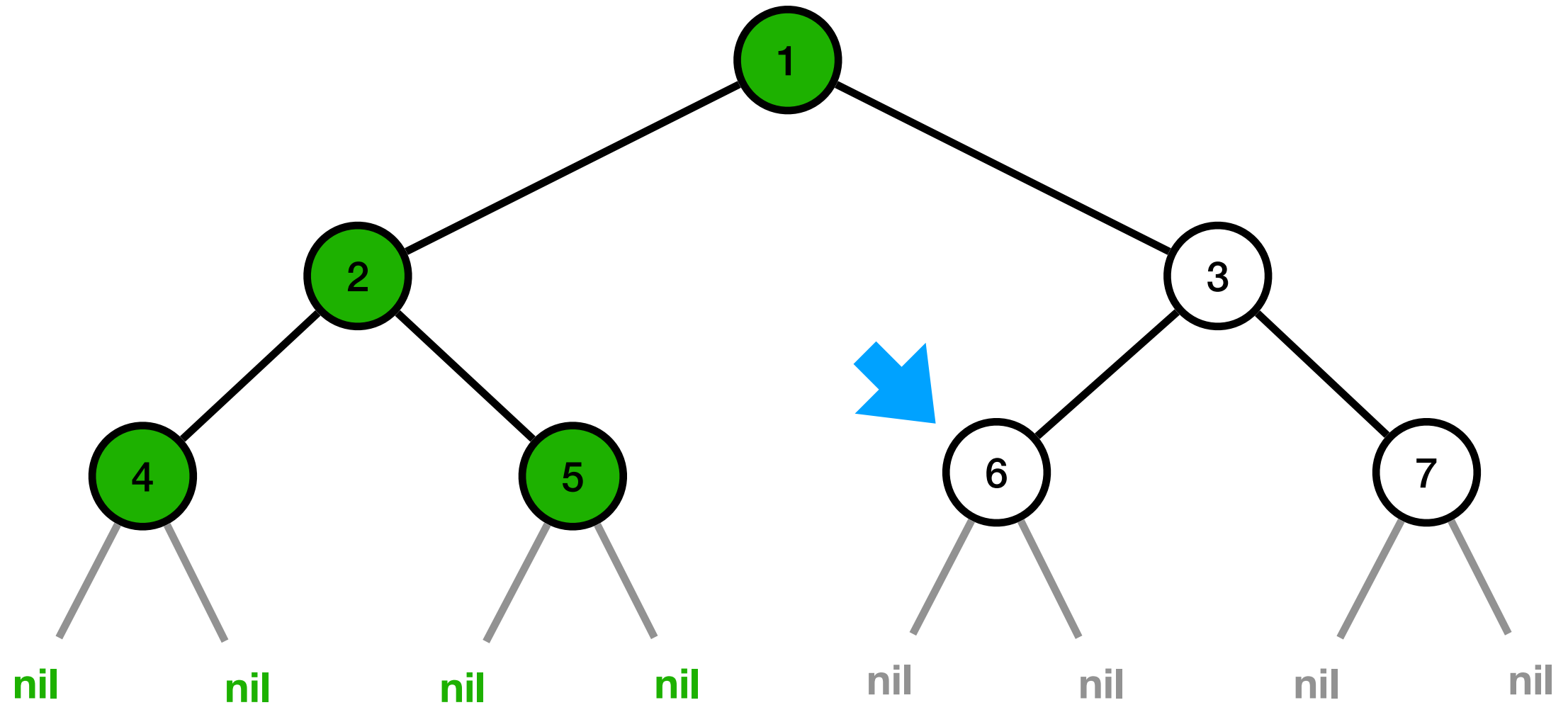
Parcours infixe



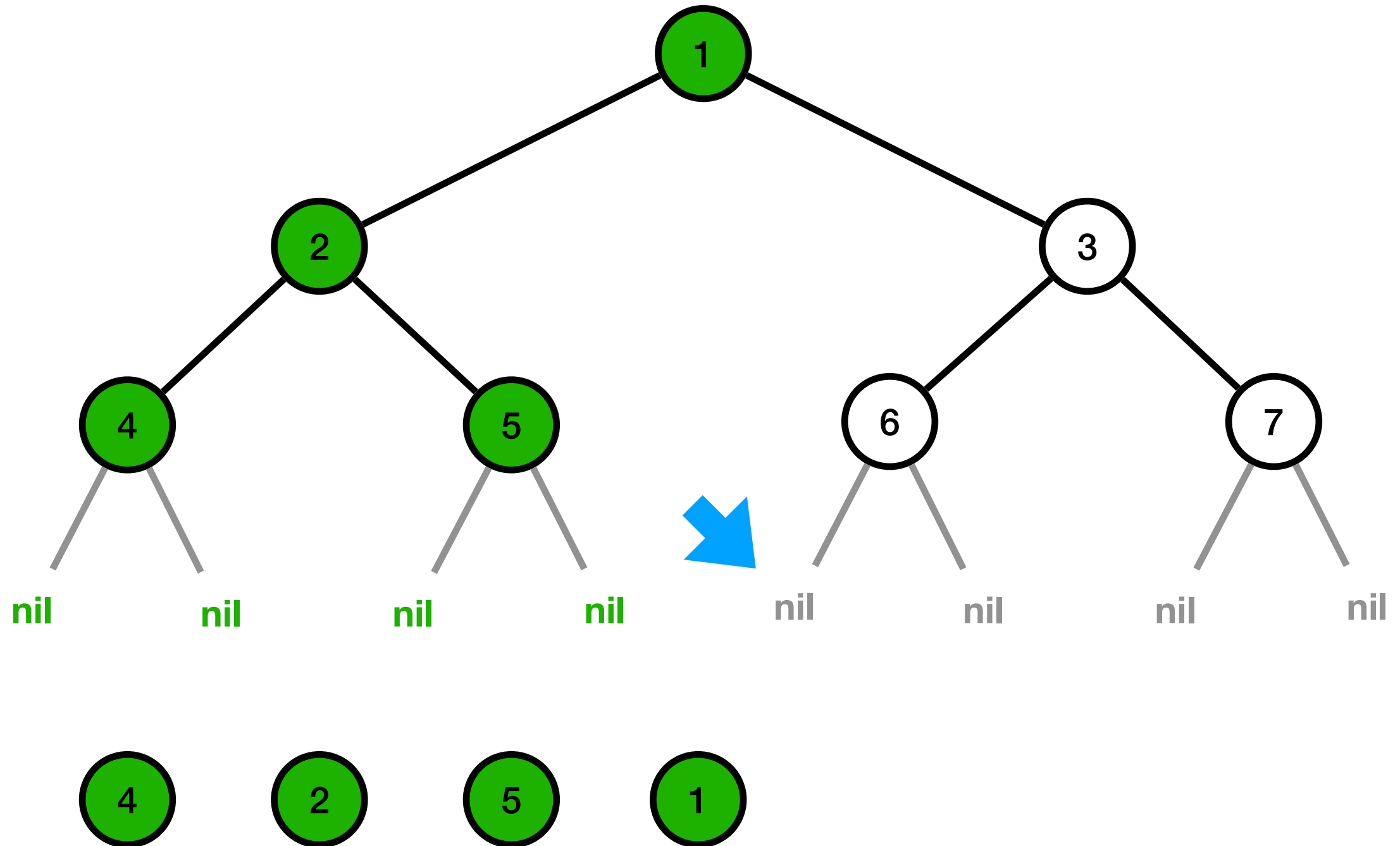
Parcours infixe



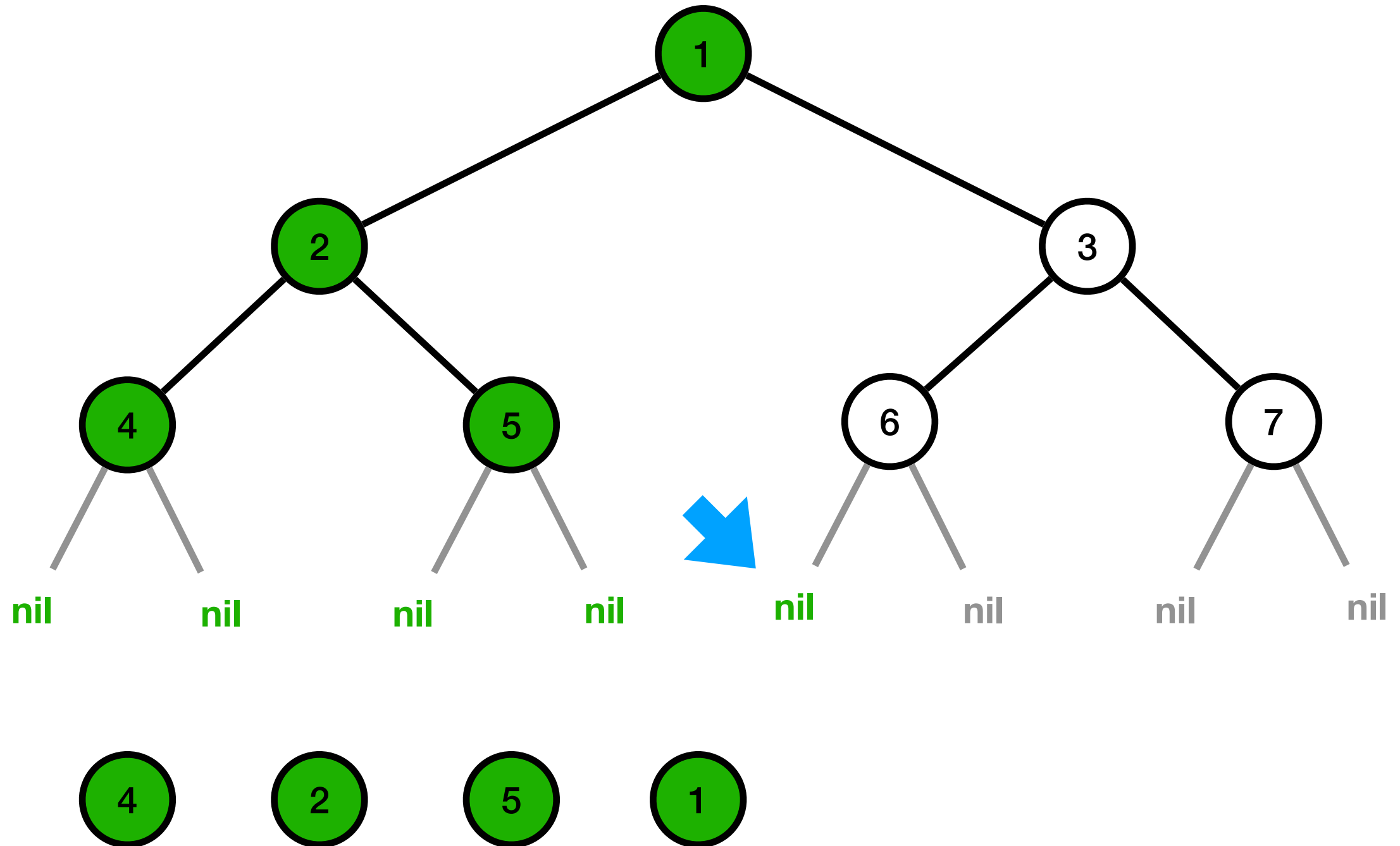
Parcours infixe



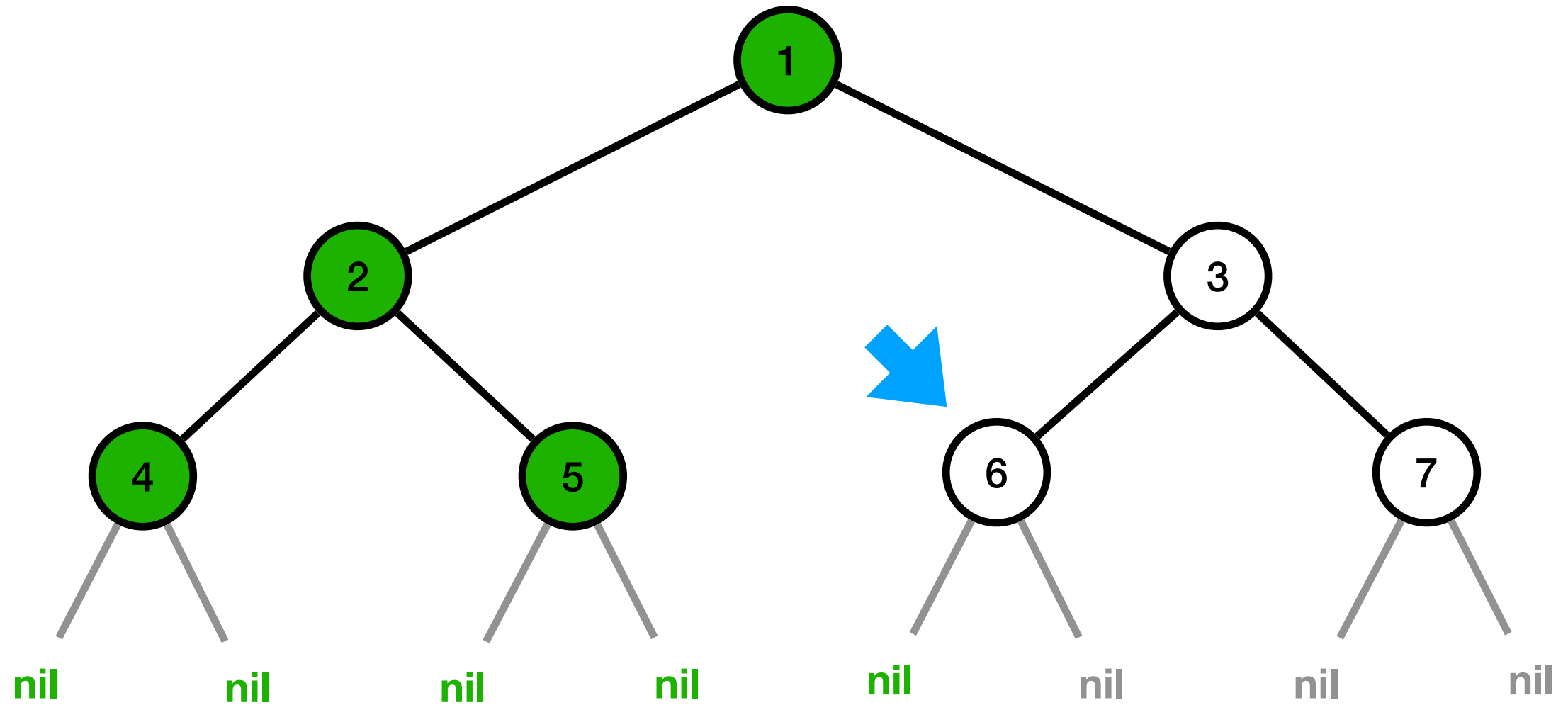
Parcours infixe



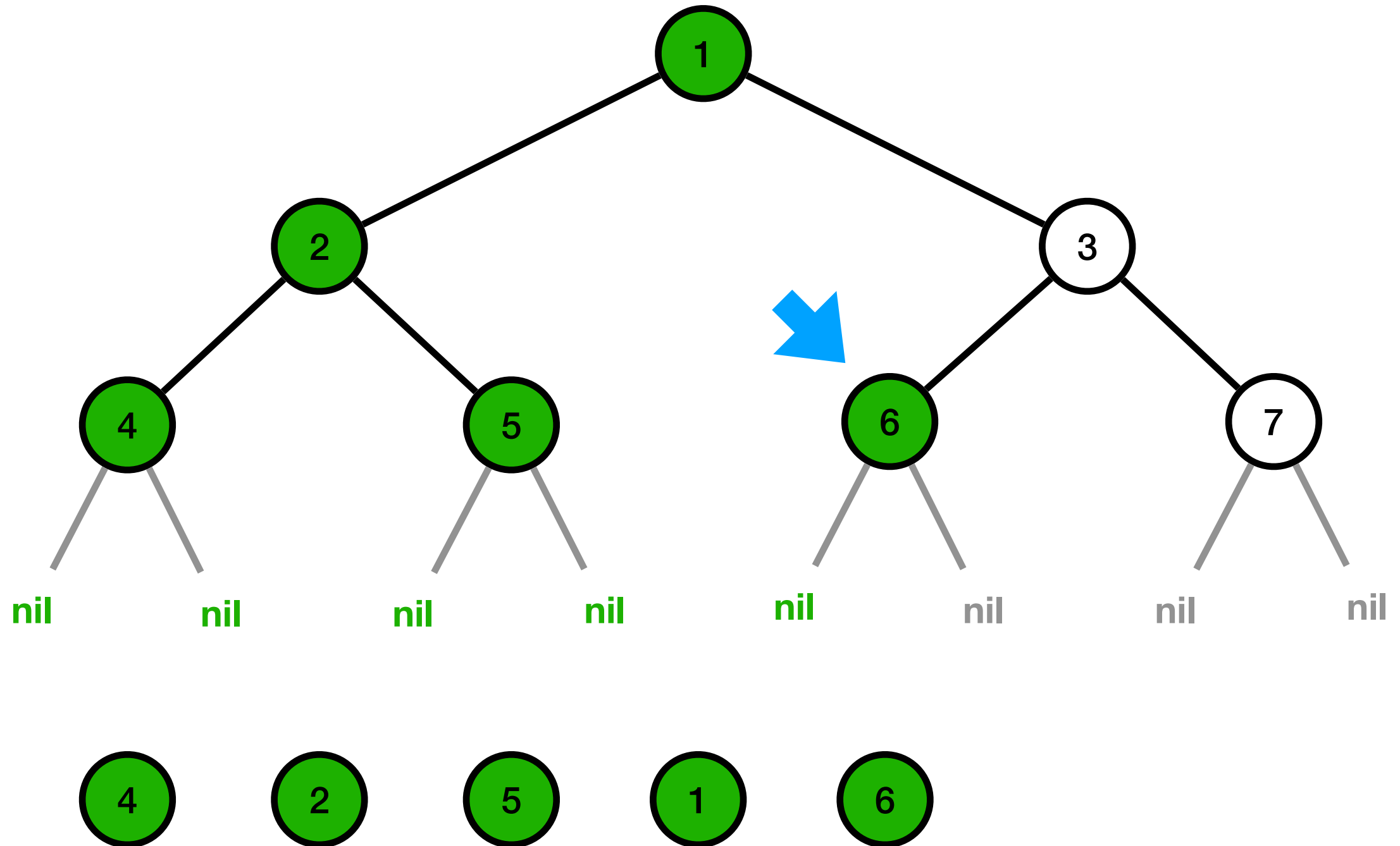
Parcours infixe



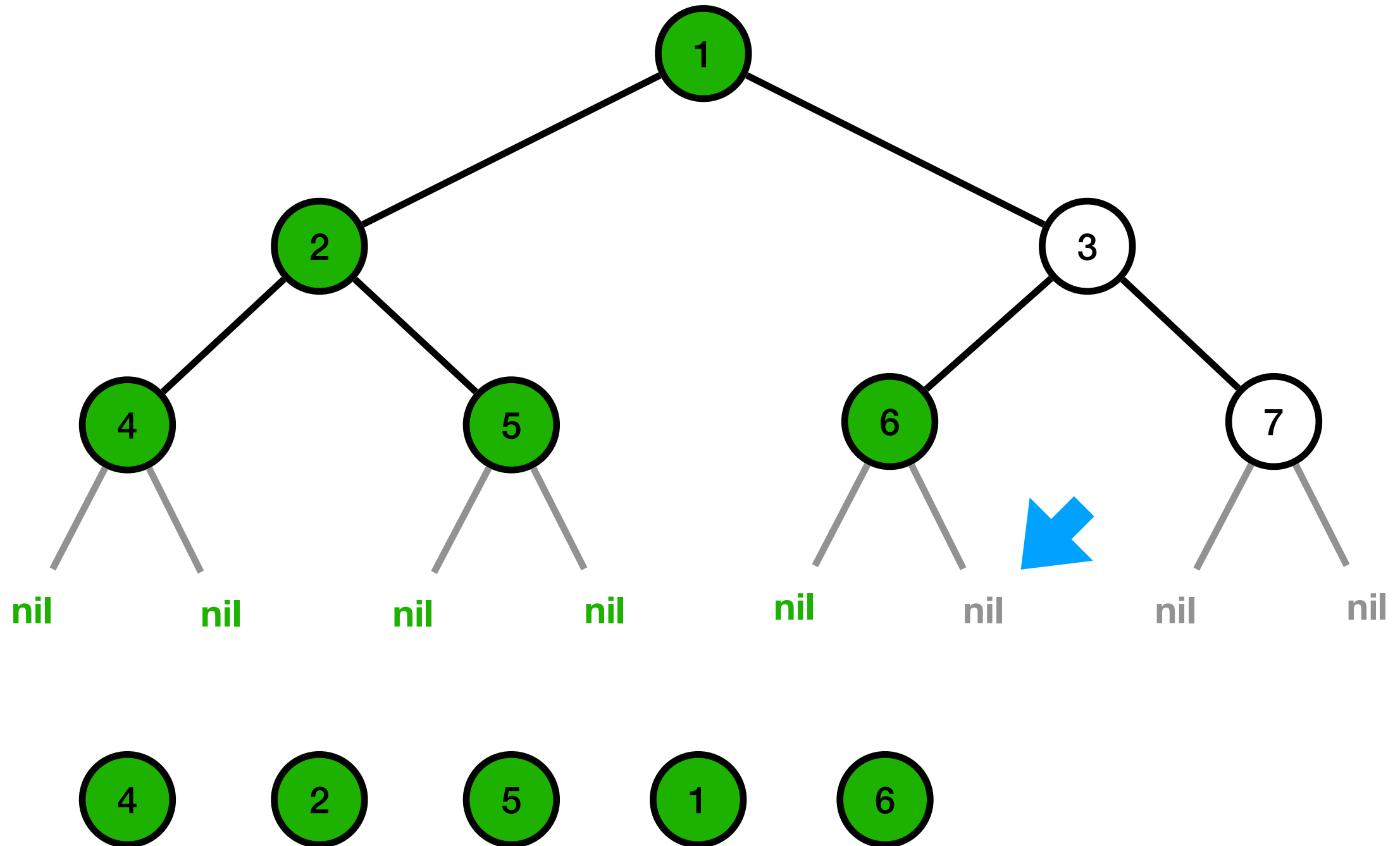
Parcours infixe



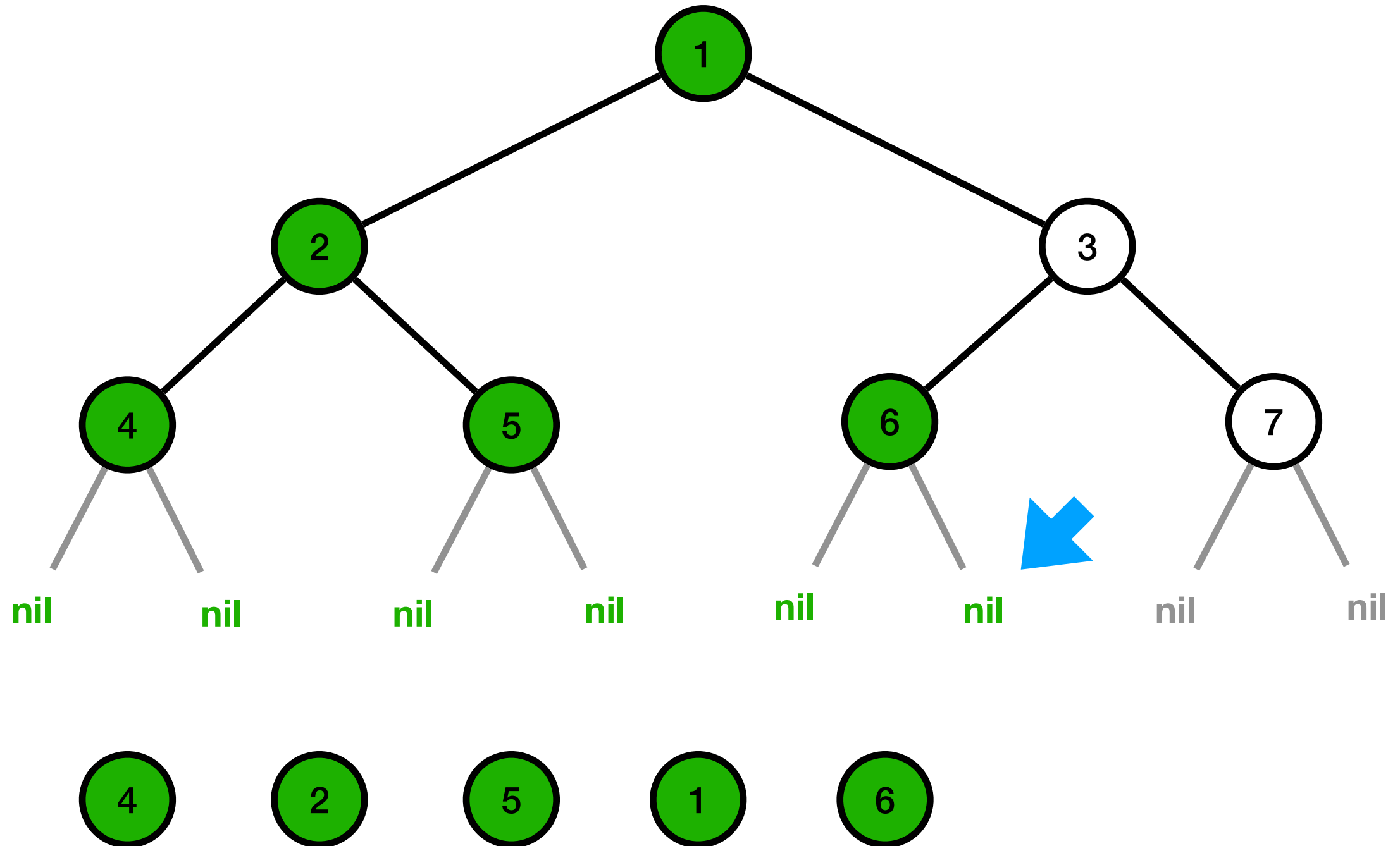
Parcours infixe



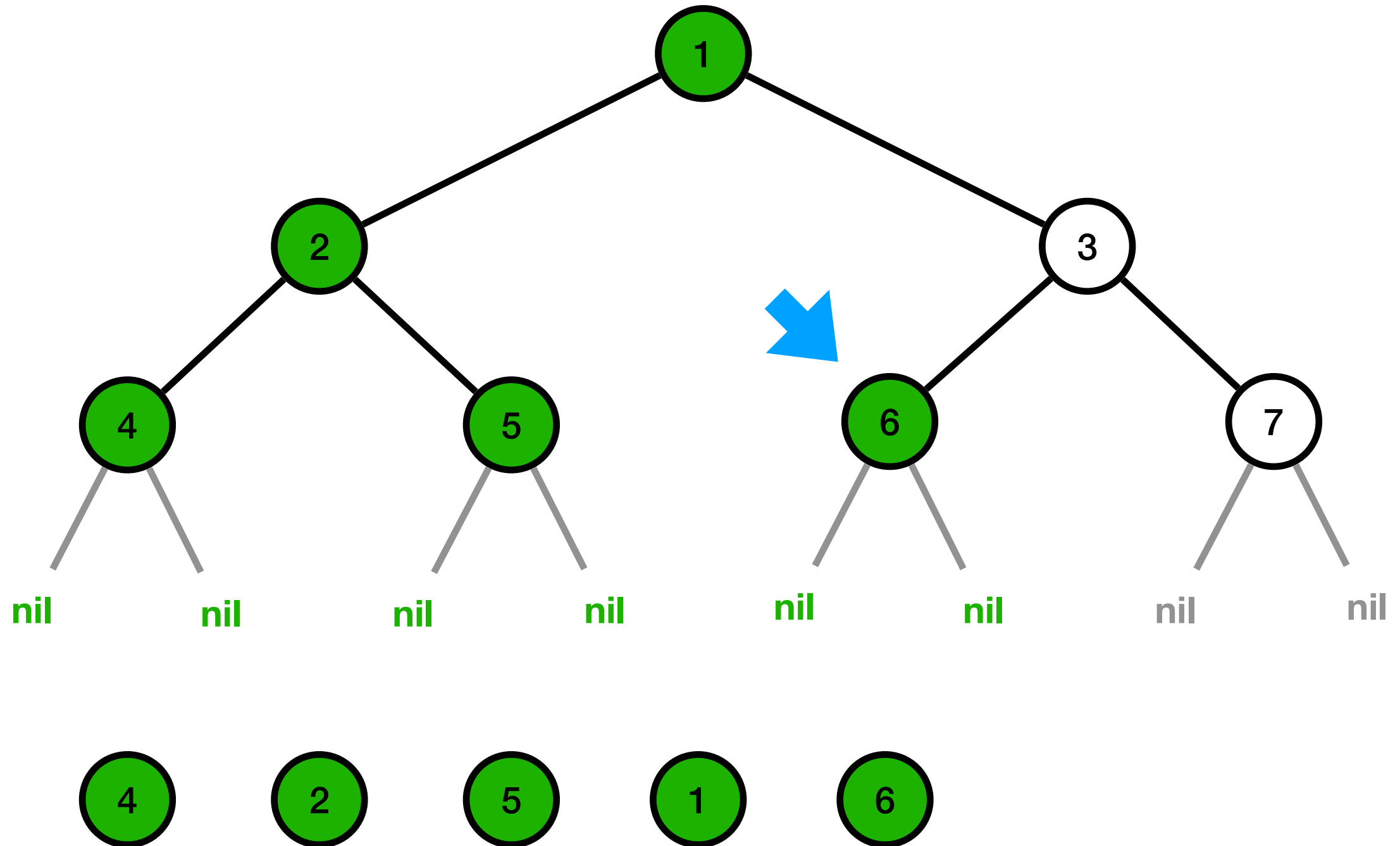
Parcours infixe



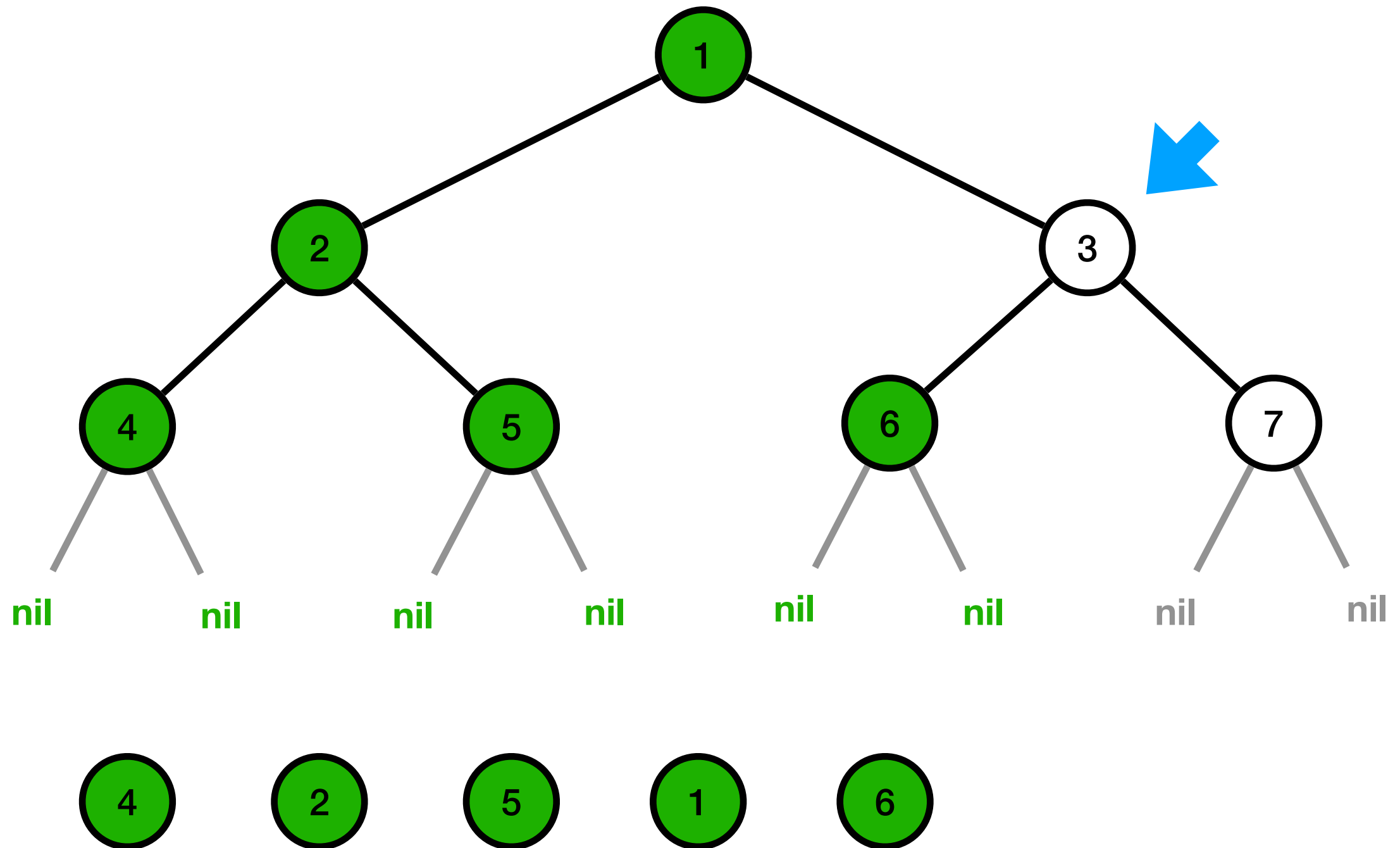
Parcours infixe



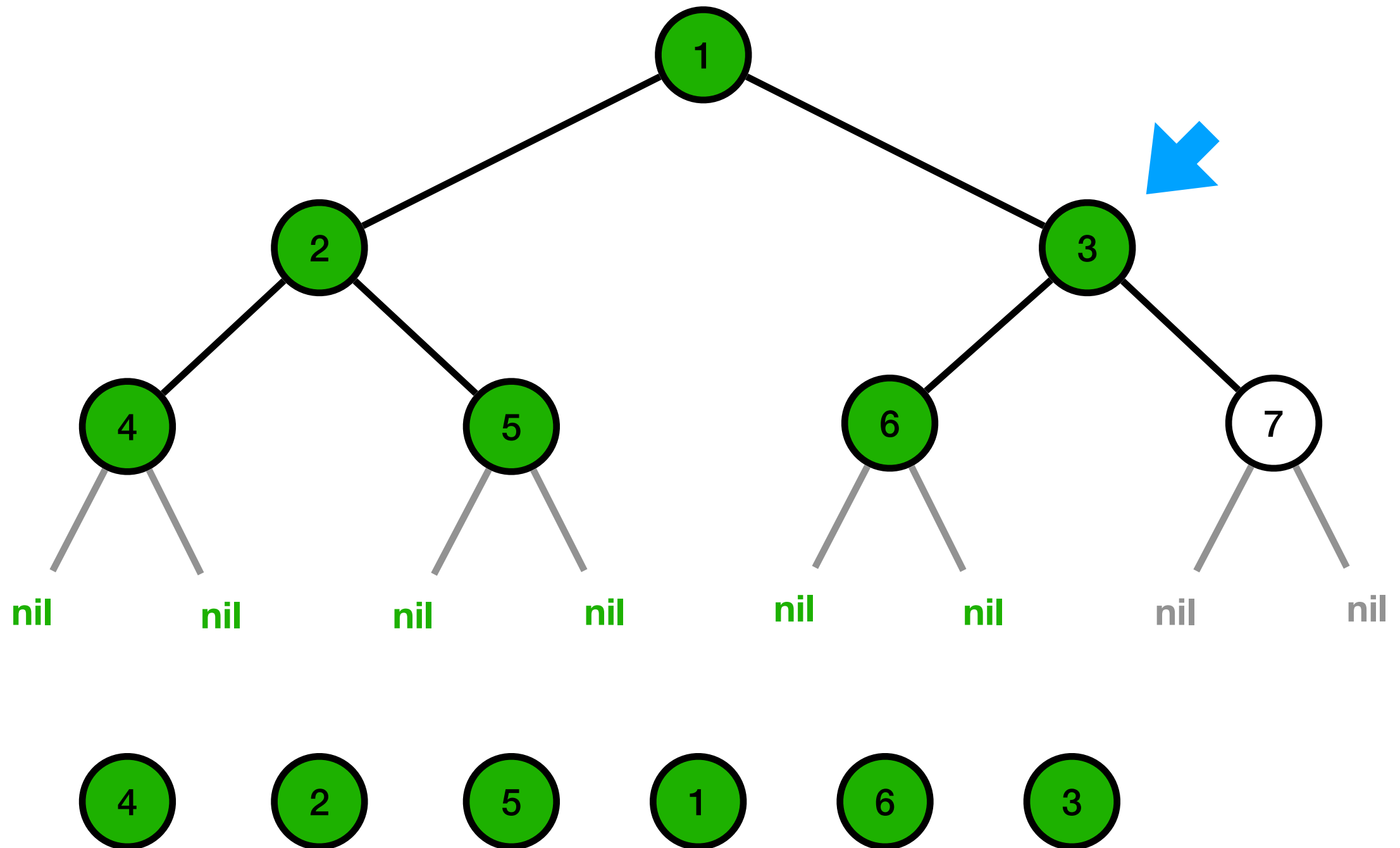
Parcours infixe



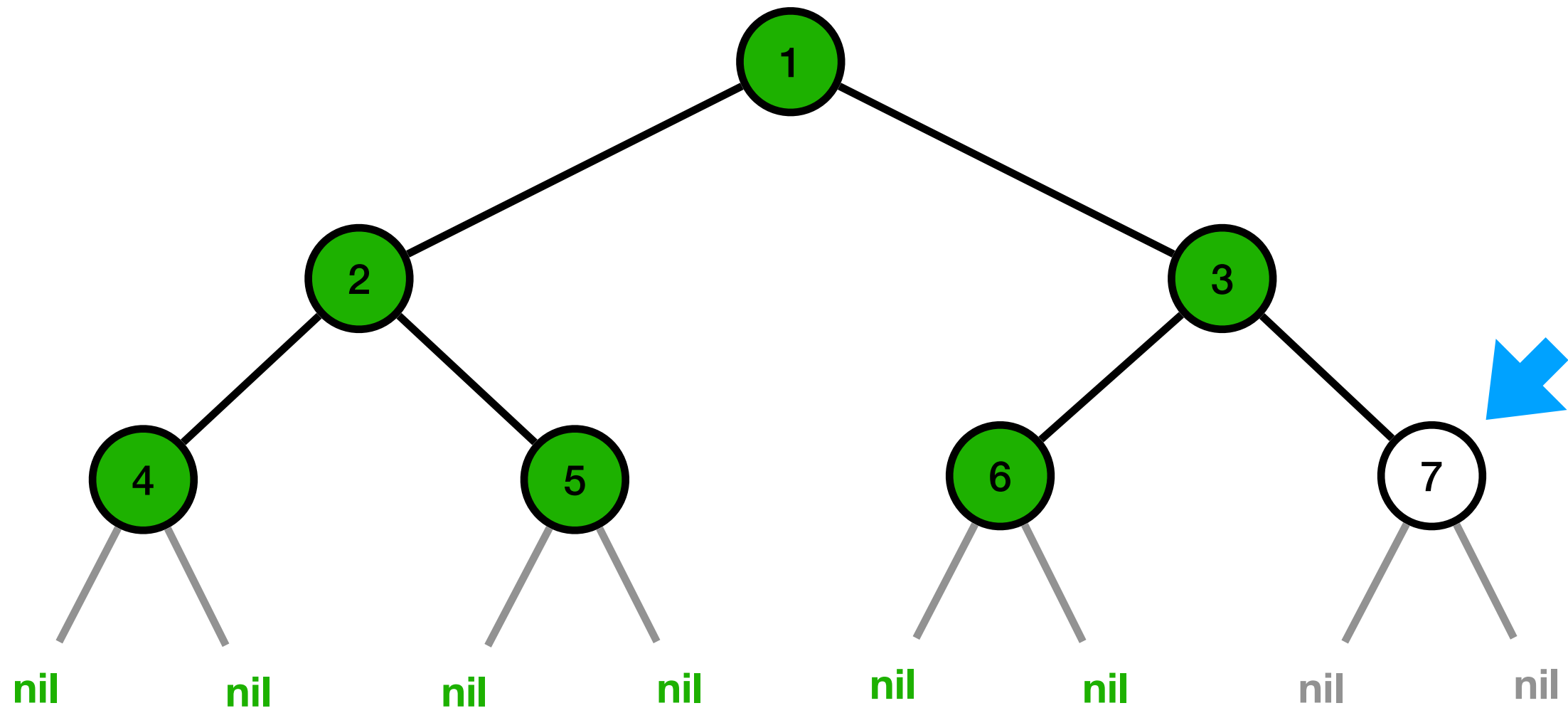
Parcours infixe



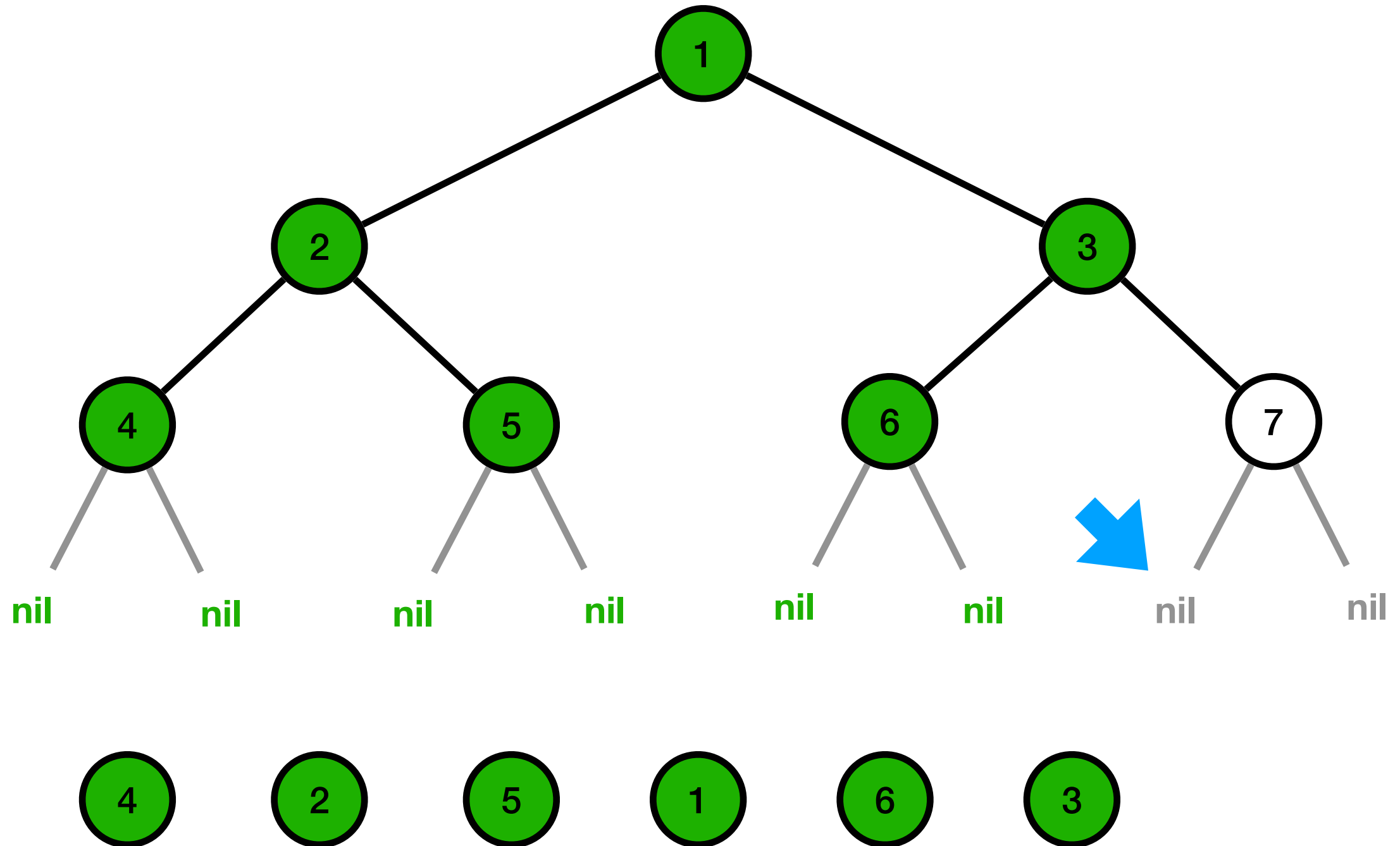
Parcours infixe



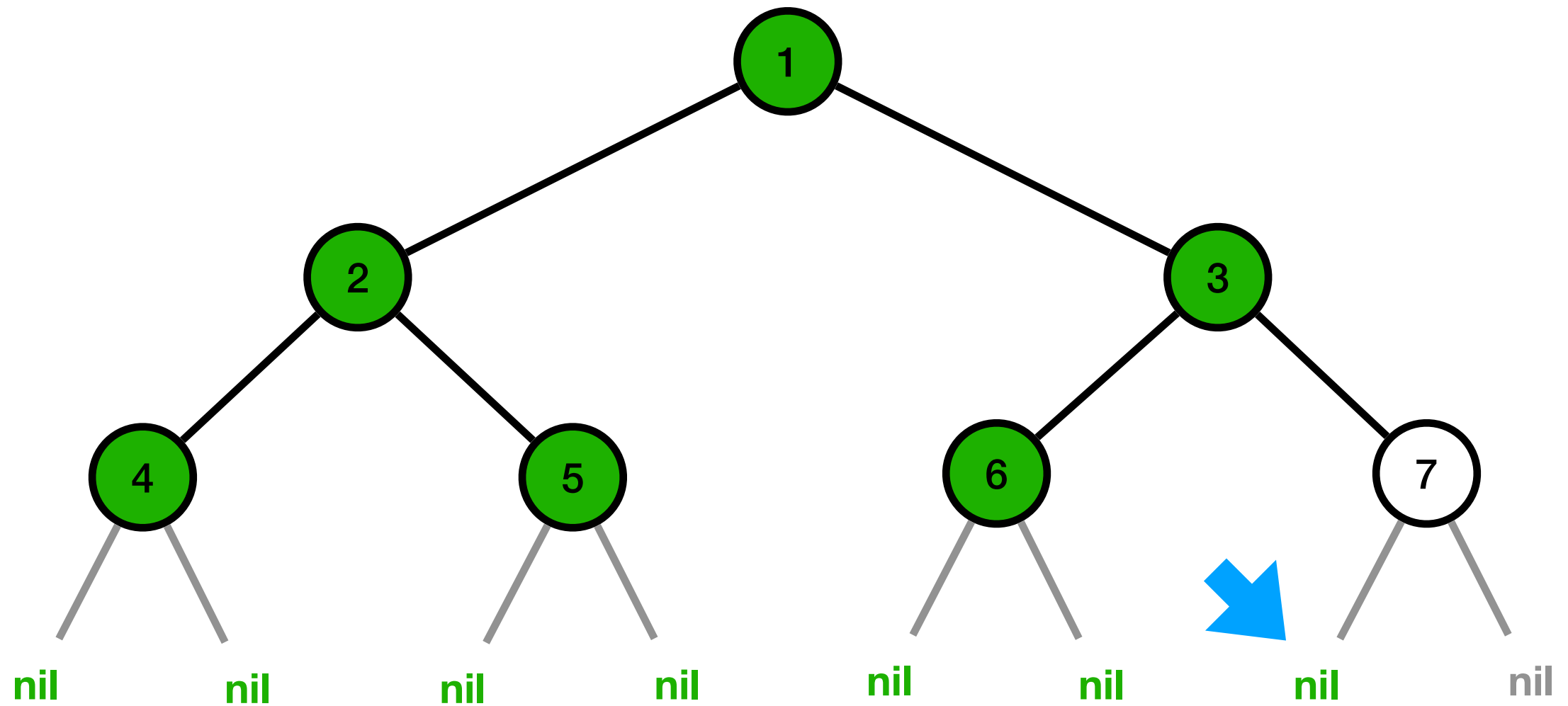
Parcours infixe



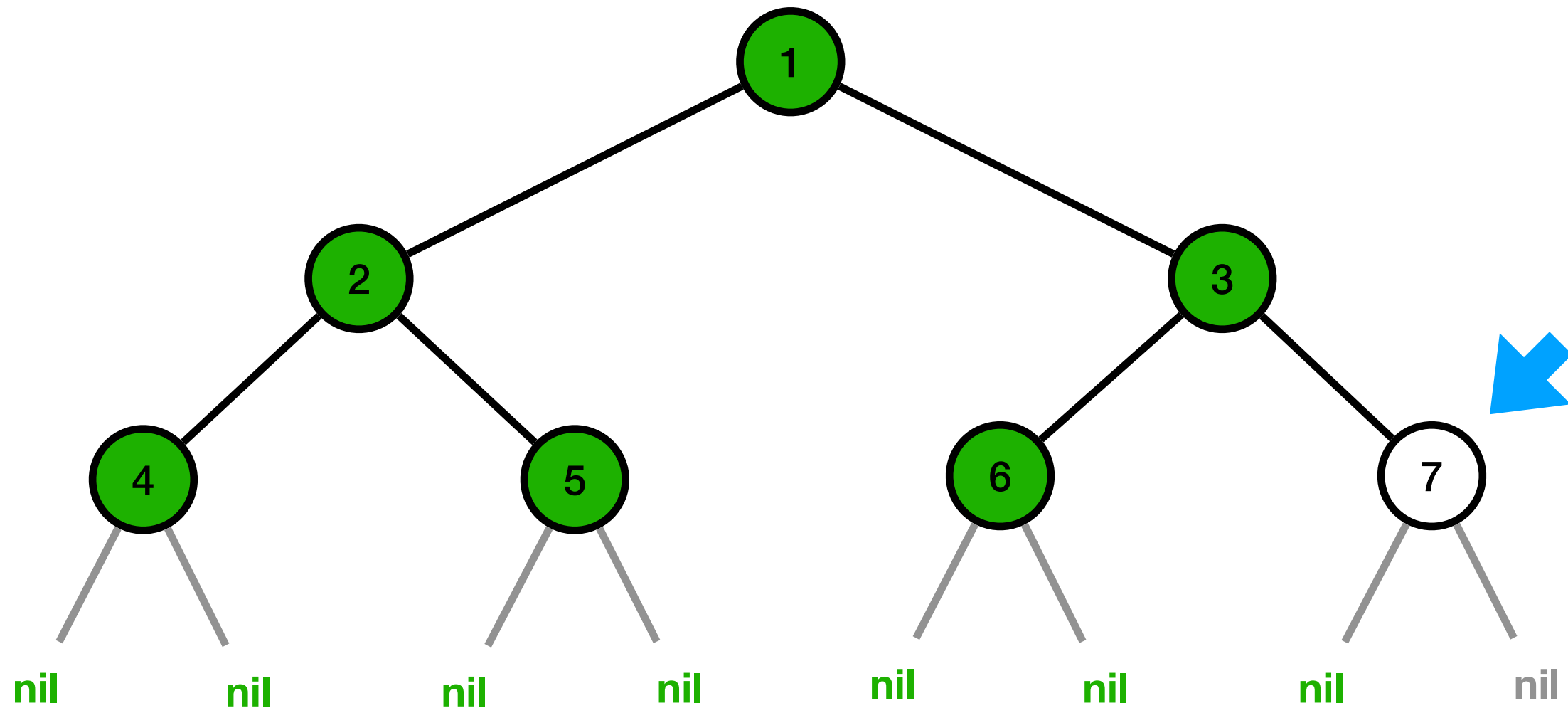
Parcours infixe



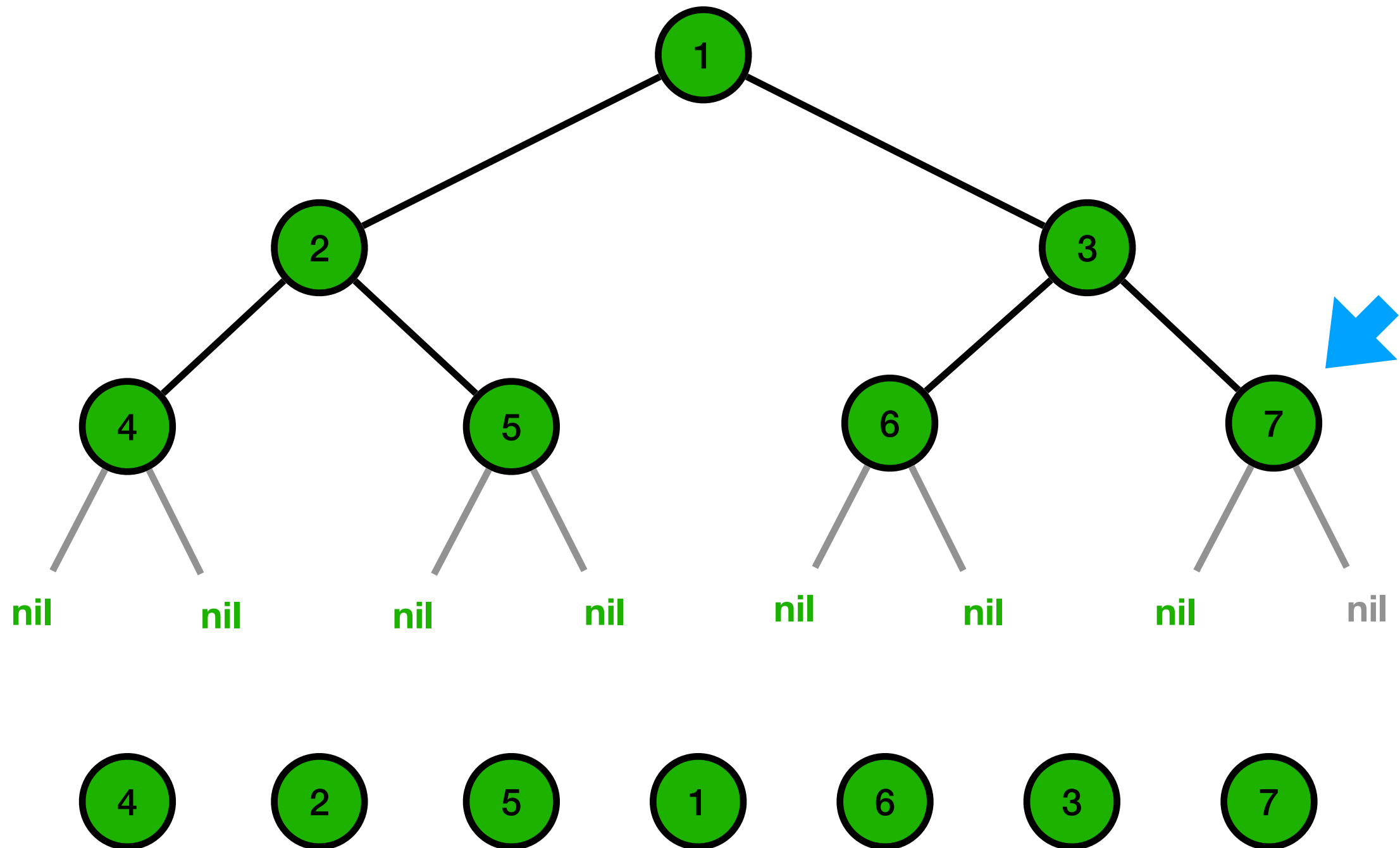
Parcours infixe



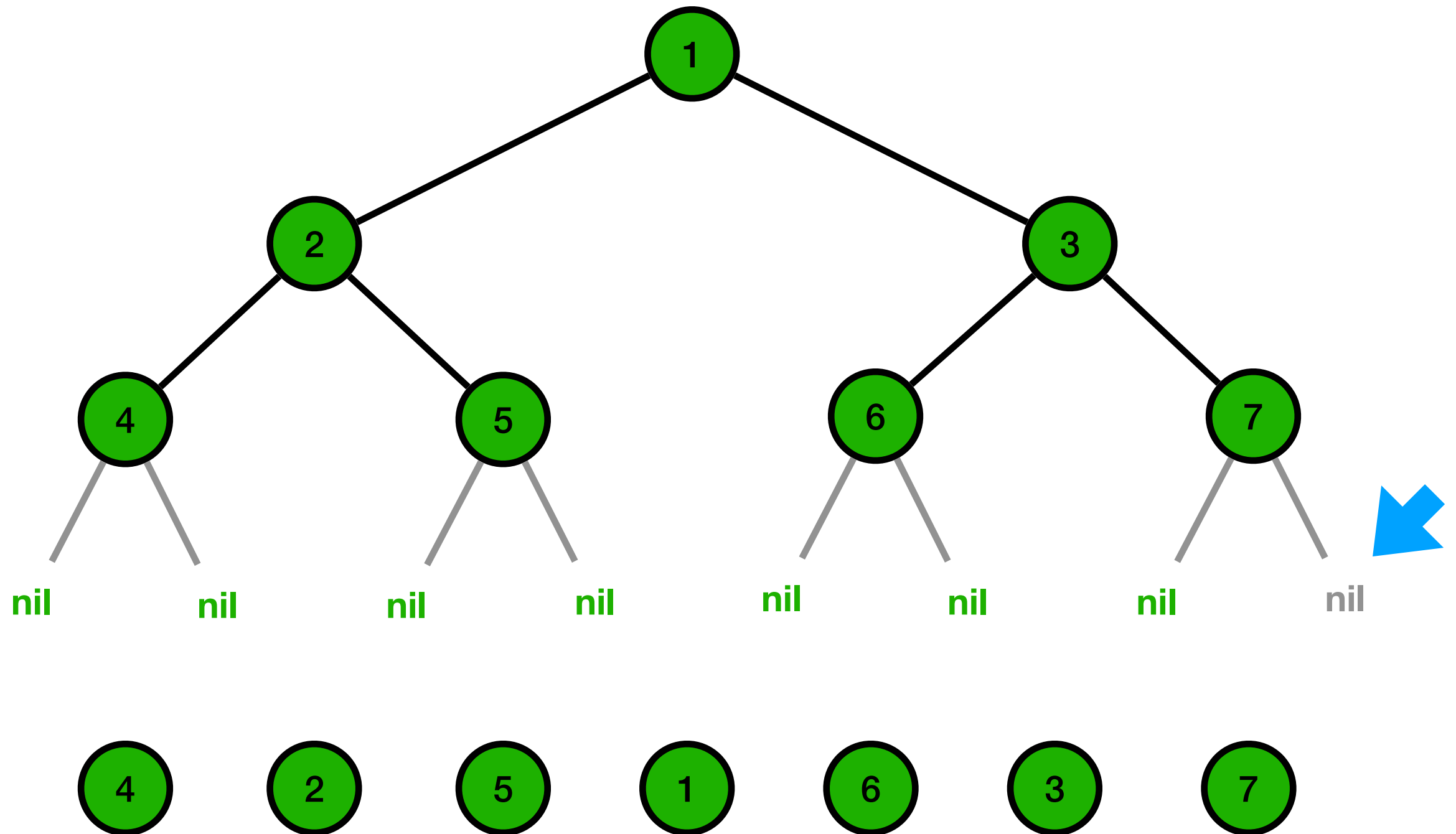
Parcours infixe



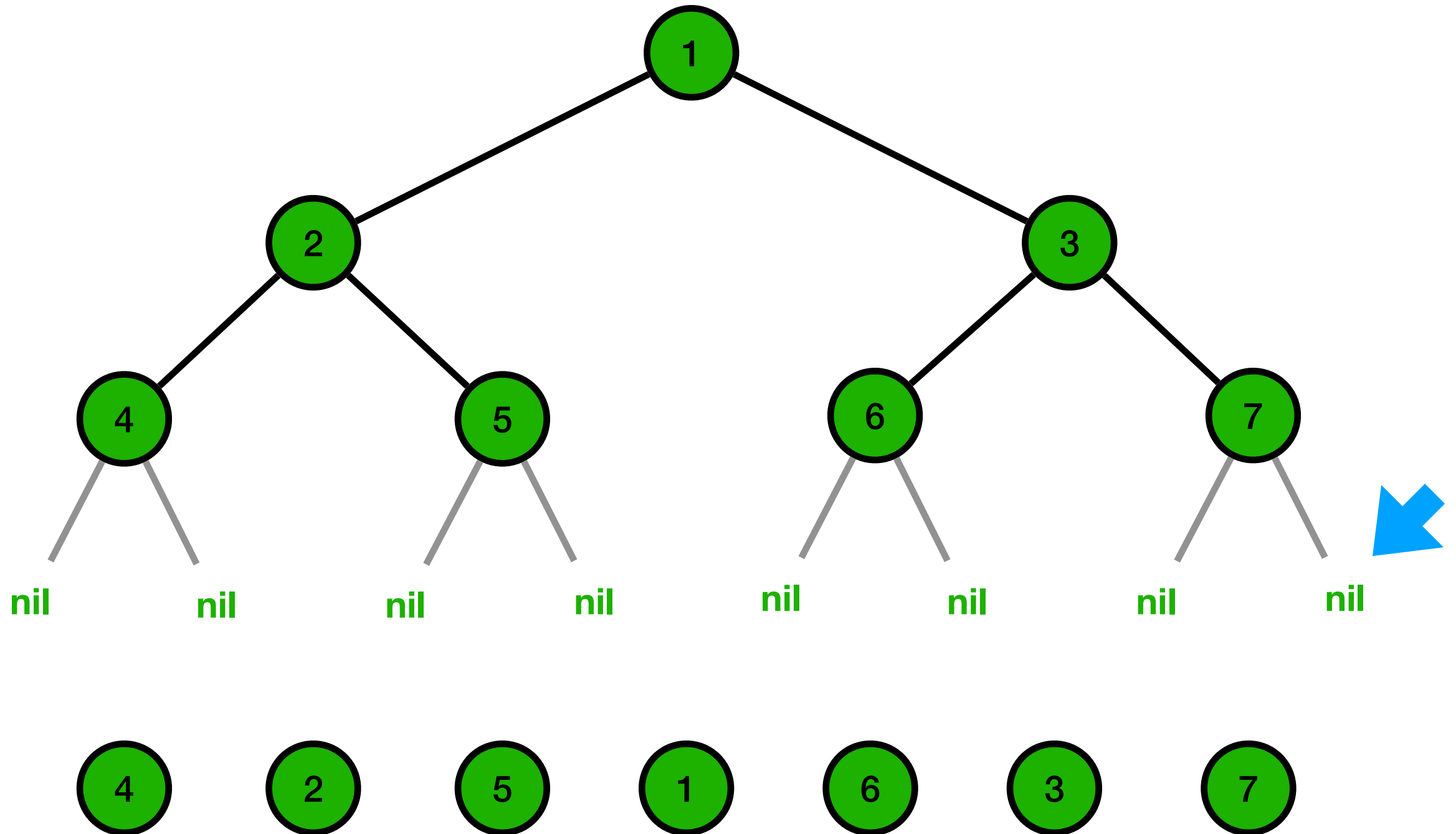
Parcours infixe



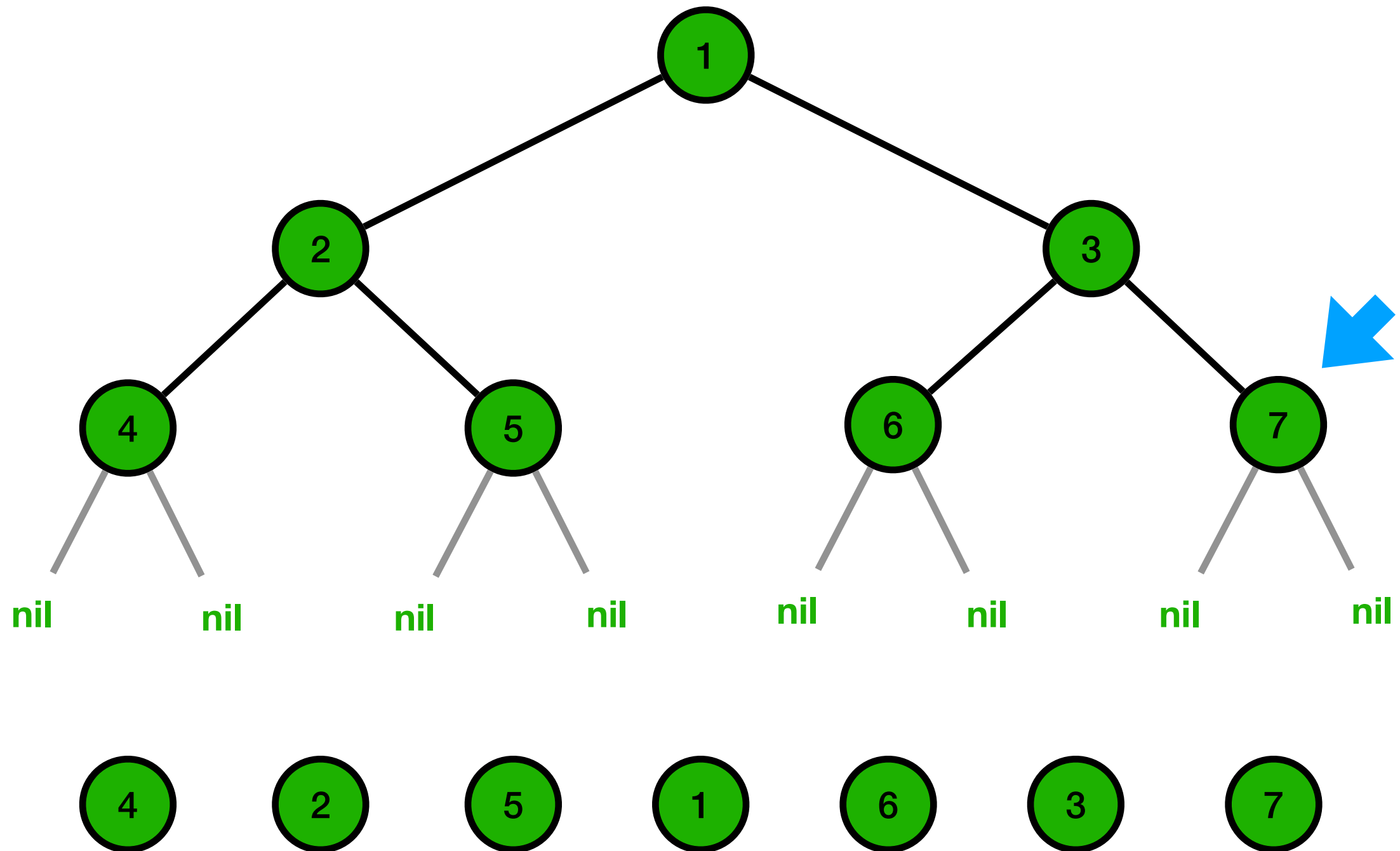
Parcours infixe



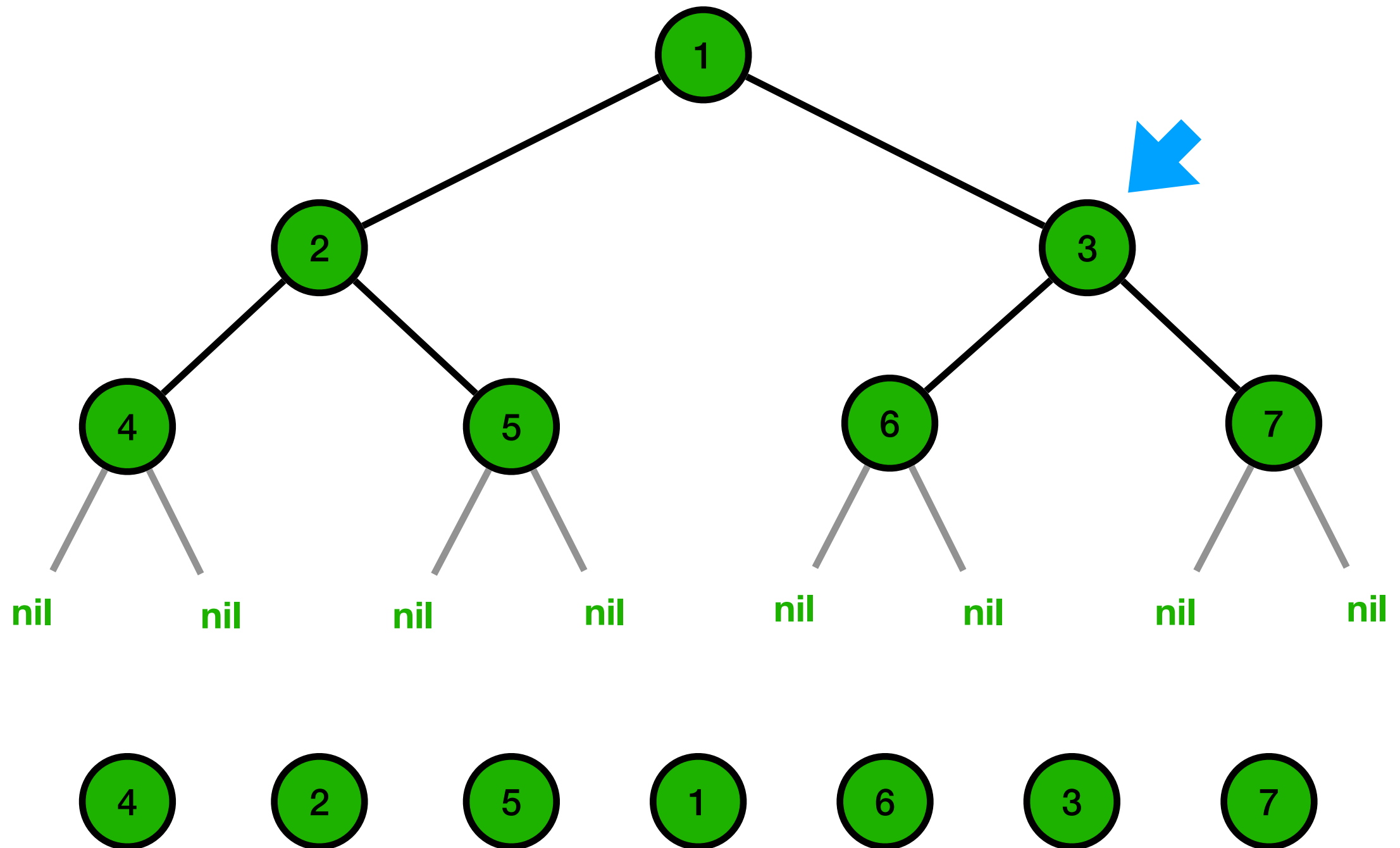
Parcours infixe



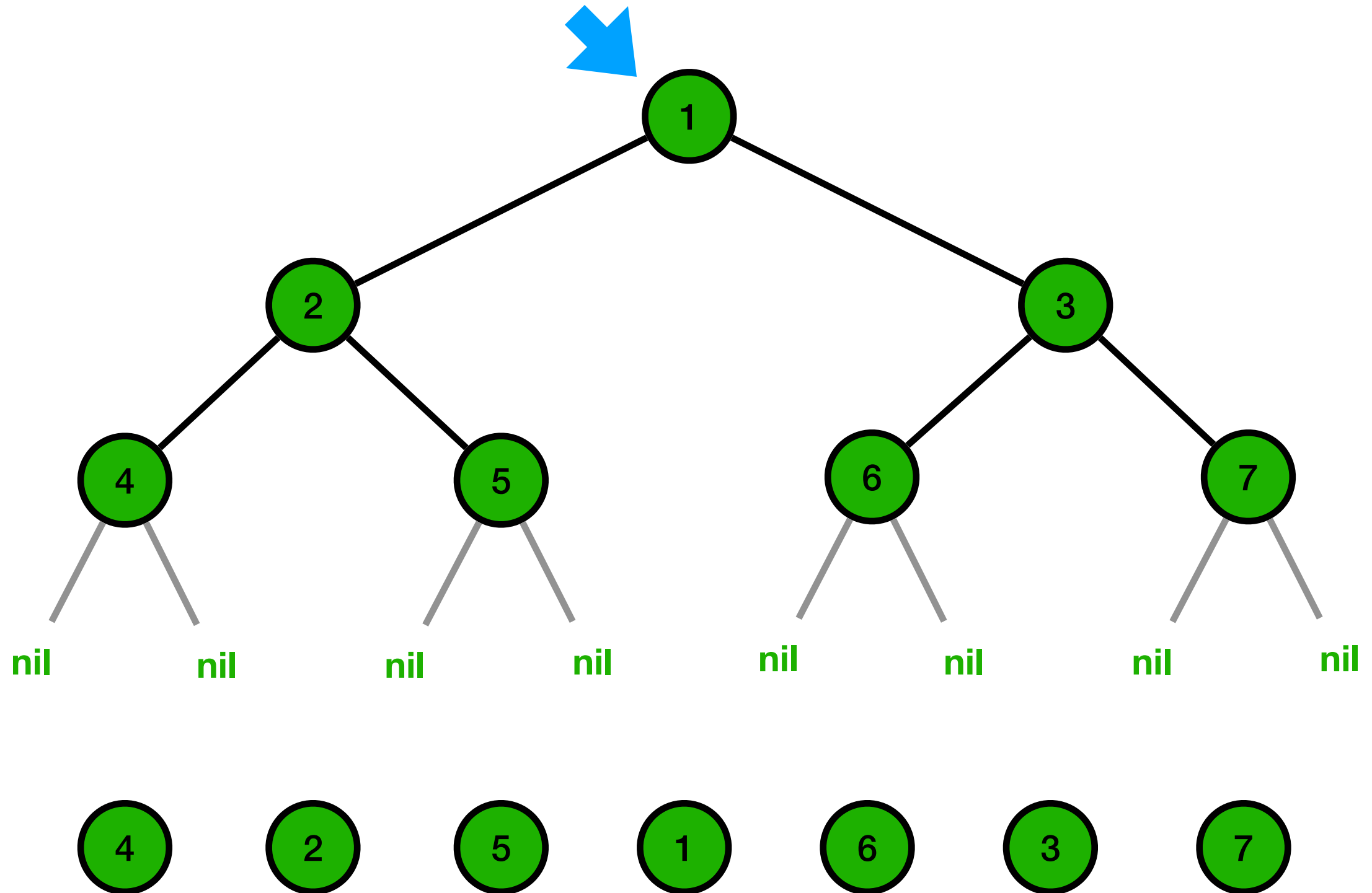
Parcours infixe



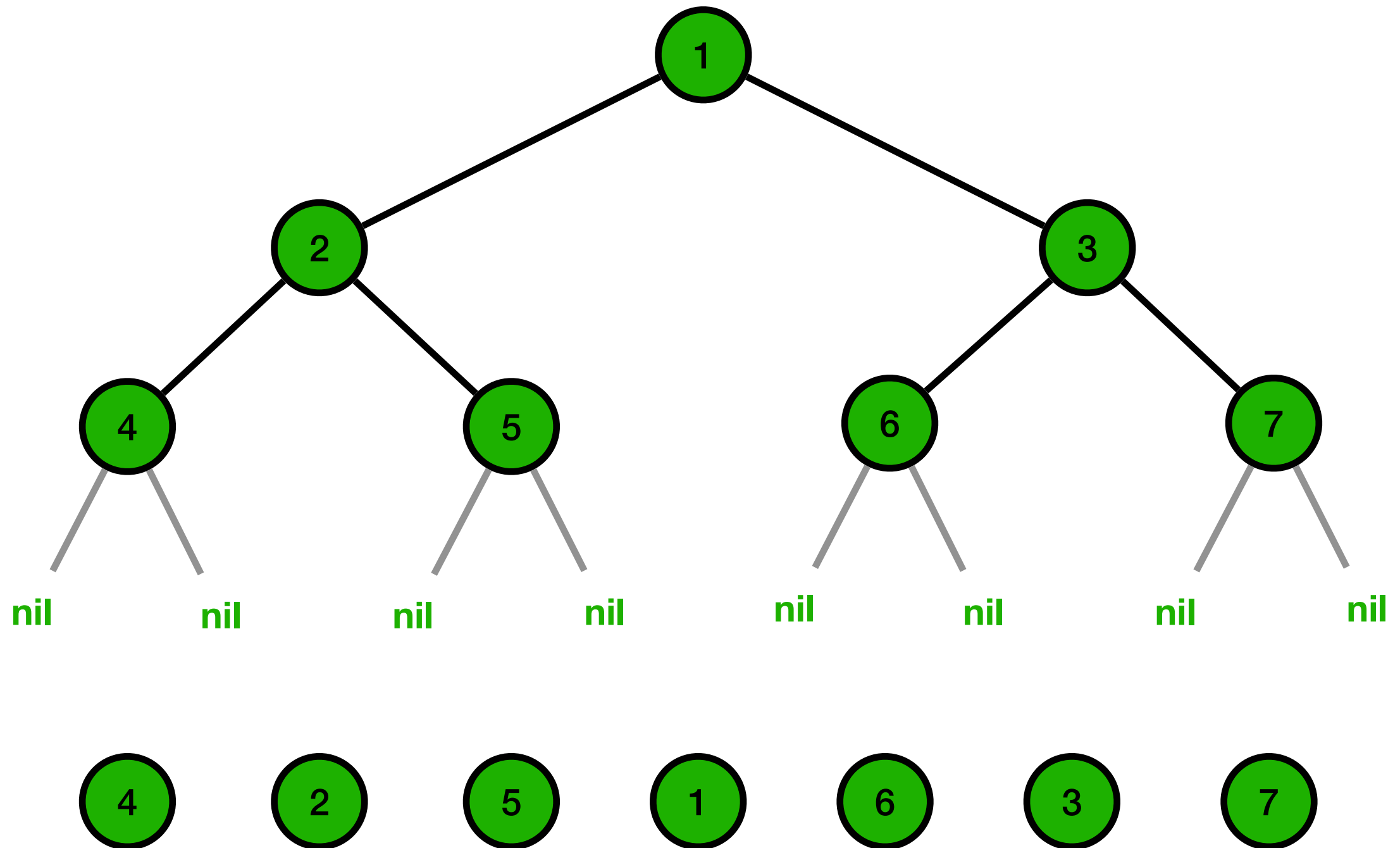
Parcours infixe



Parcours infixe



Parcours infixe



Expressions arithmétiques

Une expression arithmétique (totalement parenthésée) est :

- soit un nombre entier n
- soit une somme parenthésée $(e_1 + e_2)$, où e_1 et e_2 sont deux sub-expressions (totalement parenthésée)
- soit une soustraction parenthésée $(e_1 - e_2)$
- soit un produit parenthésé $(e_1 \times e_2)$
- soit une division parenthésée $(e_1 \div e_2)$

Exemples d'expressions

Exemples d'expressions

5

3

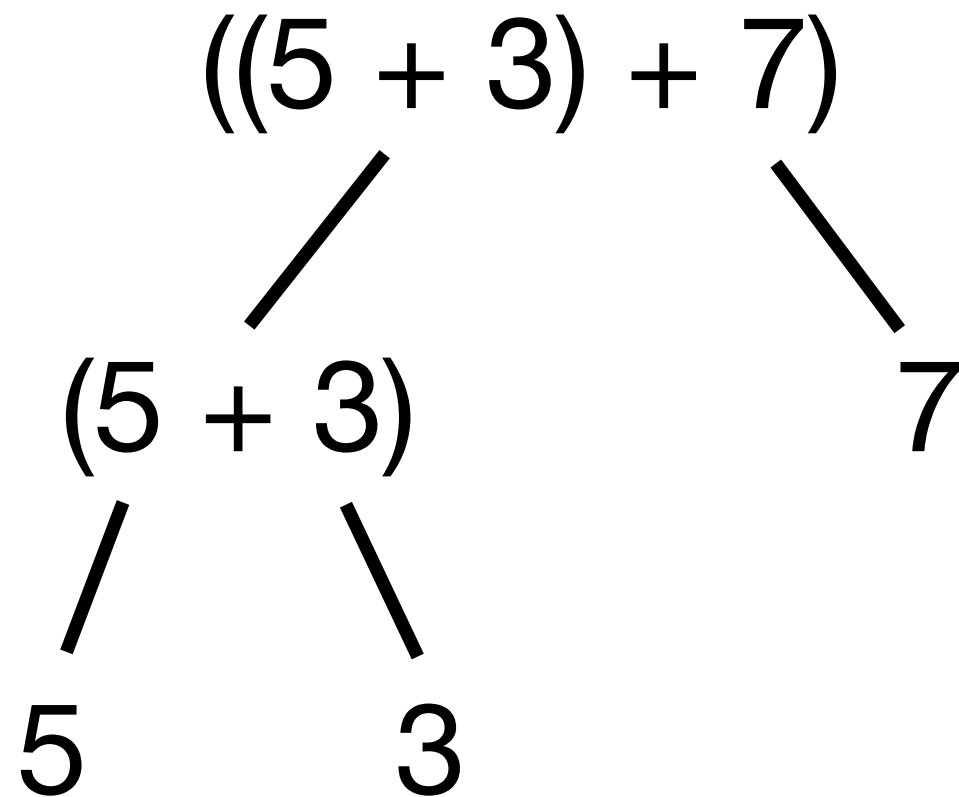
Exemples d'expressions

$$\begin{array}{c} (5 + 3) \\ / \quad \backslash \\ 5 \quad 3 \end{array}$$

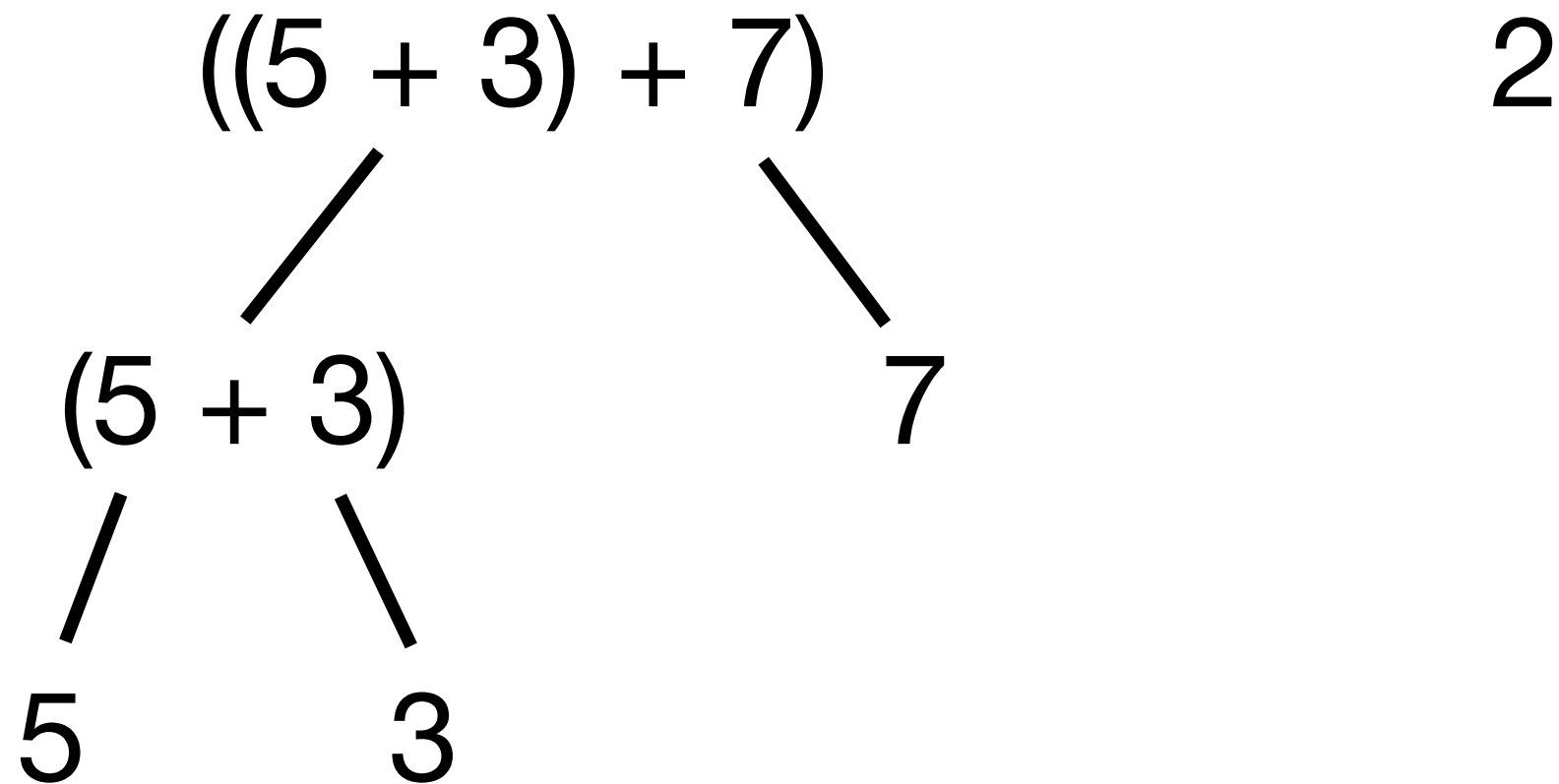
Exemples d'expressions

$$\begin{array}{c} (5 + 3) \\ / \quad \backslash \\ 5 \quad 3 \end{array} \quad 7$$

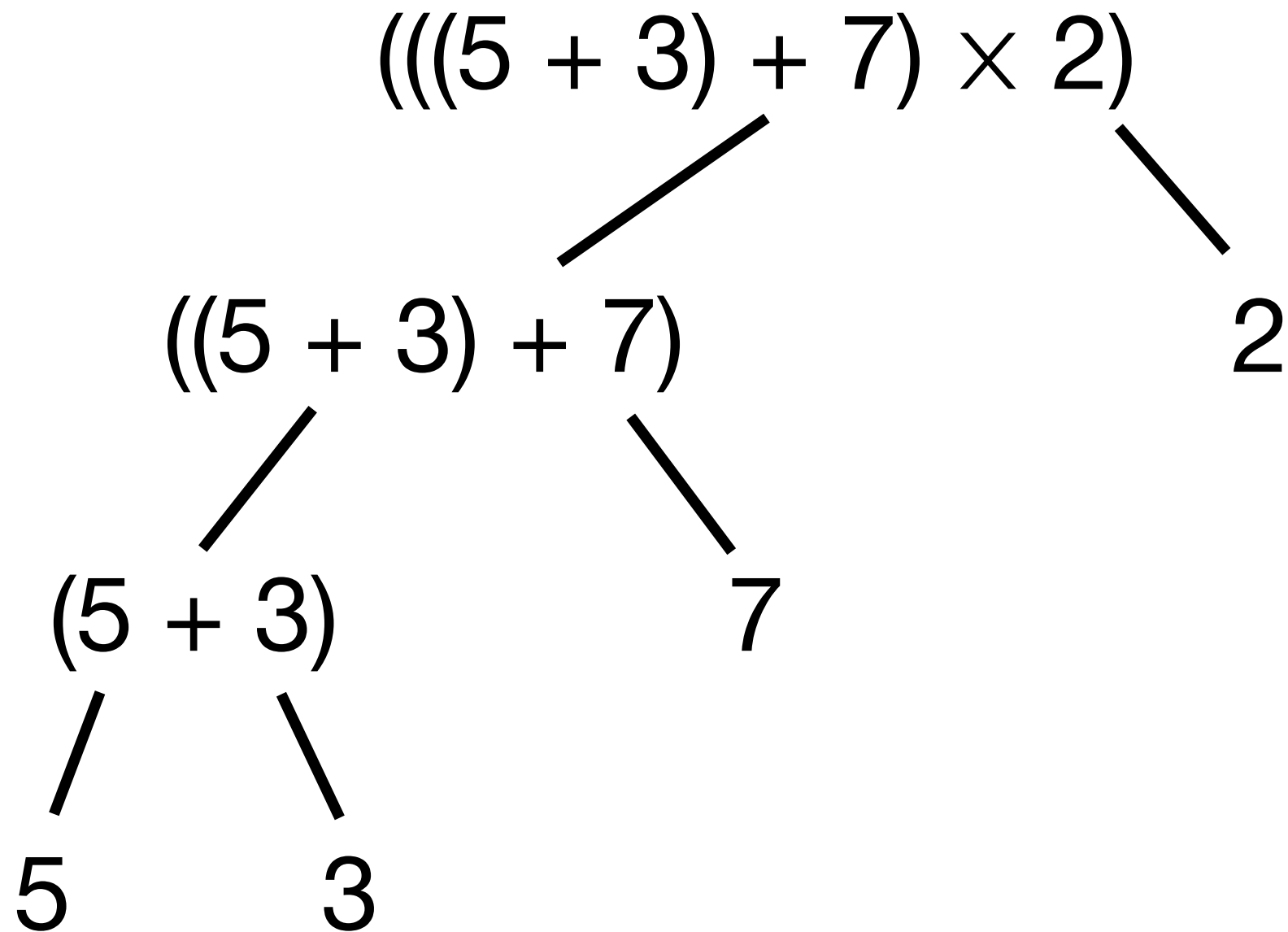
Exemples d'expressions



Exemples d'expressions



Exemples d'expressions



Exemples d'expressions

5

3

7

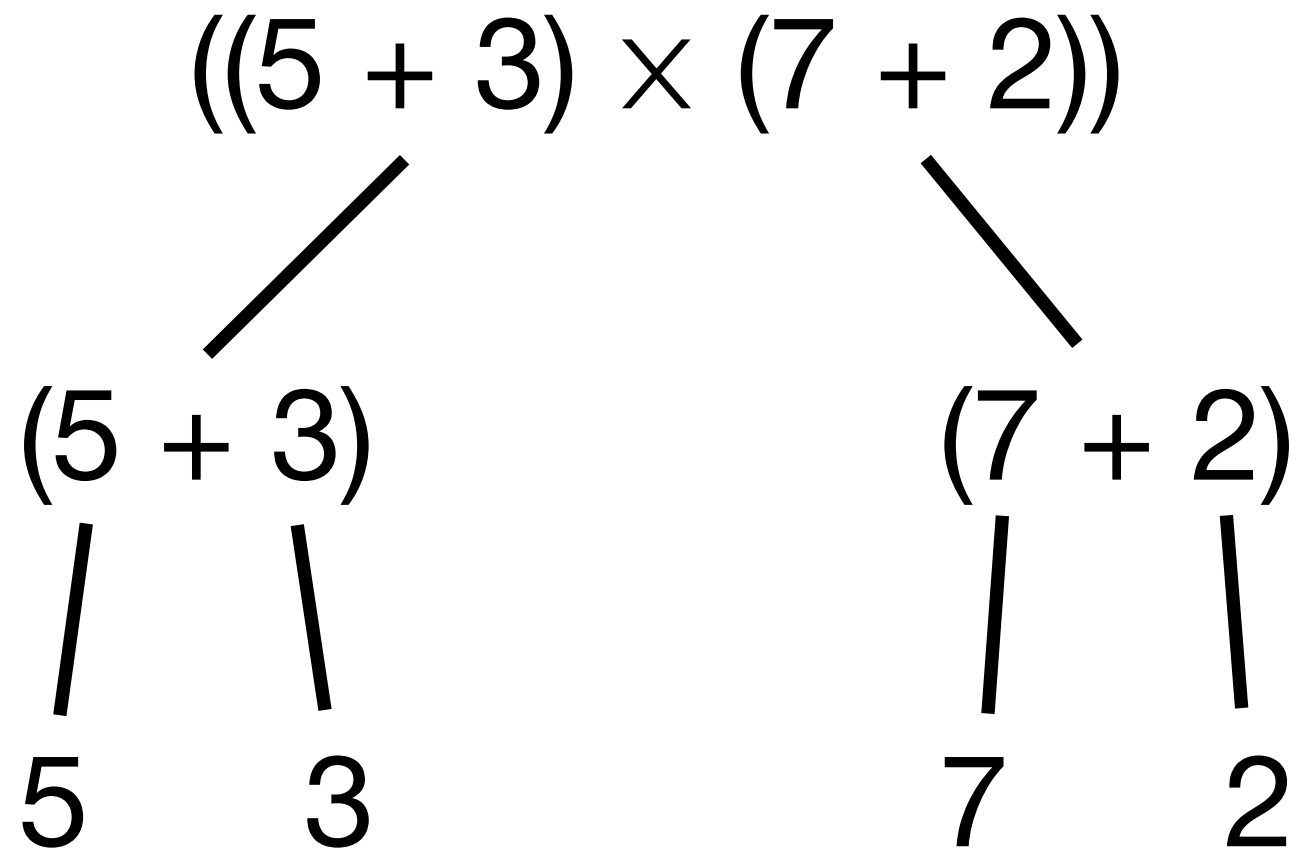
2

Exemples d'expressions

$$\begin{array}{c} (5 + 3) \\ | \quad \backslash \\ 5 \quad 3 \end{array}$$

$$\begin{array}{c} (7 + 2) \\ | \quad \backslash \\ 7 \quad 2 \end{array}$$

Exemples d'expressions



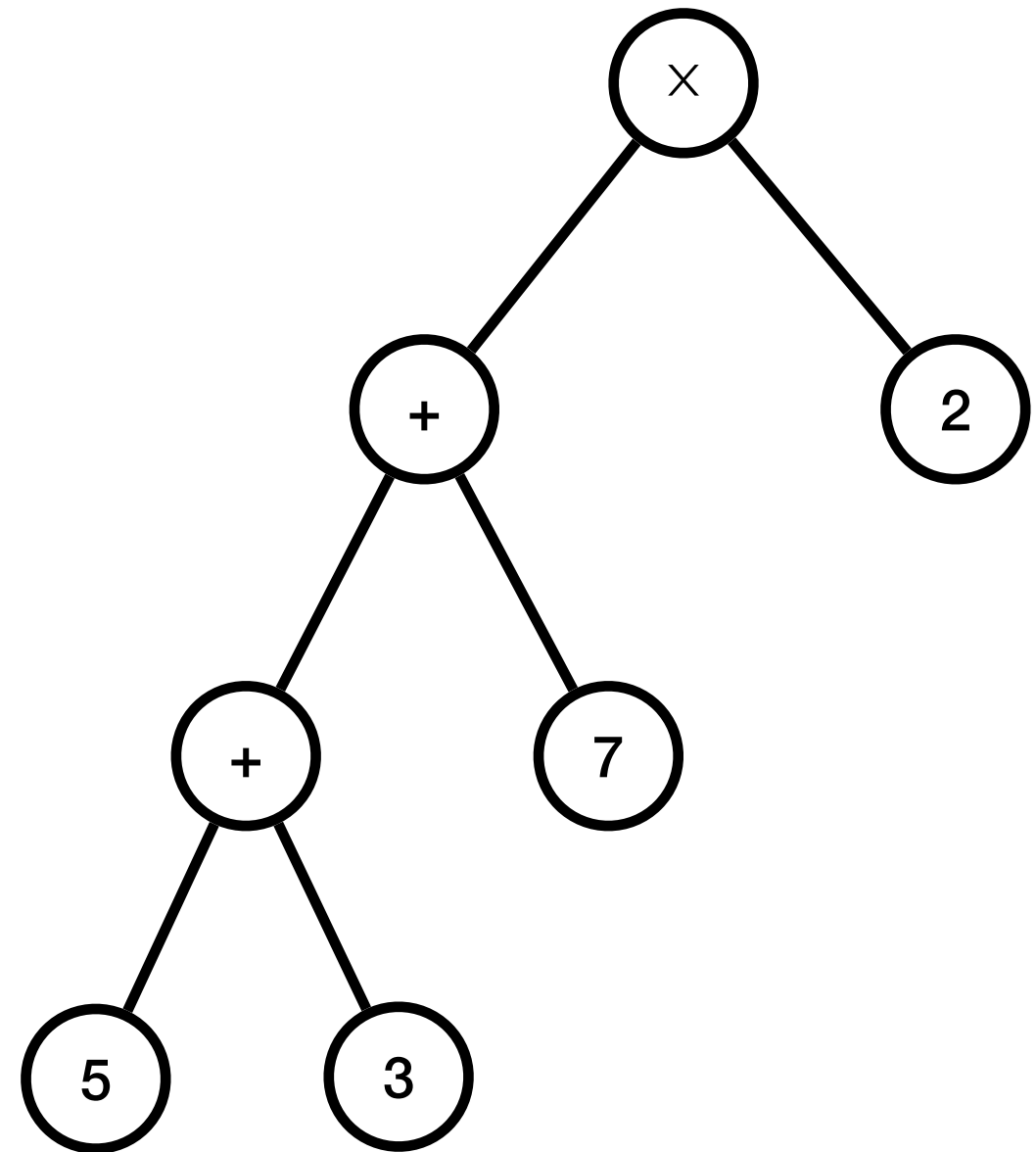
Représentation d'expressions

$((5 + 3) + 7) \times 2$

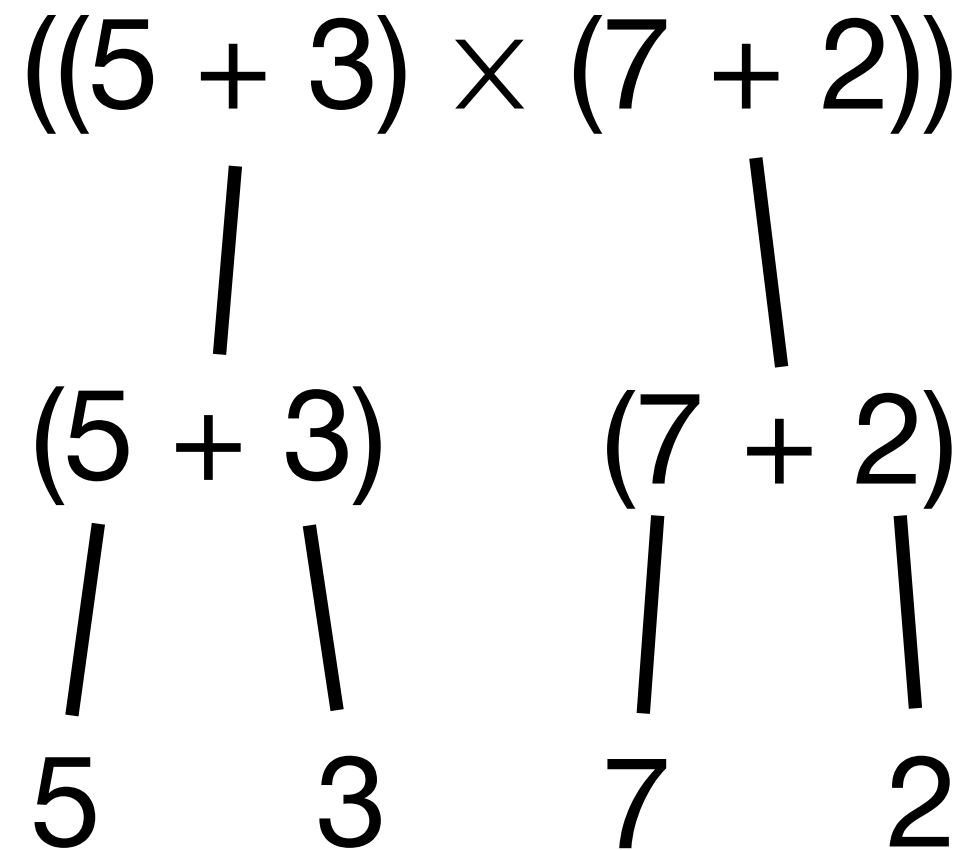
$((5 + 3) + 7)$ 2

$(5 + 3)$ 7

5 3

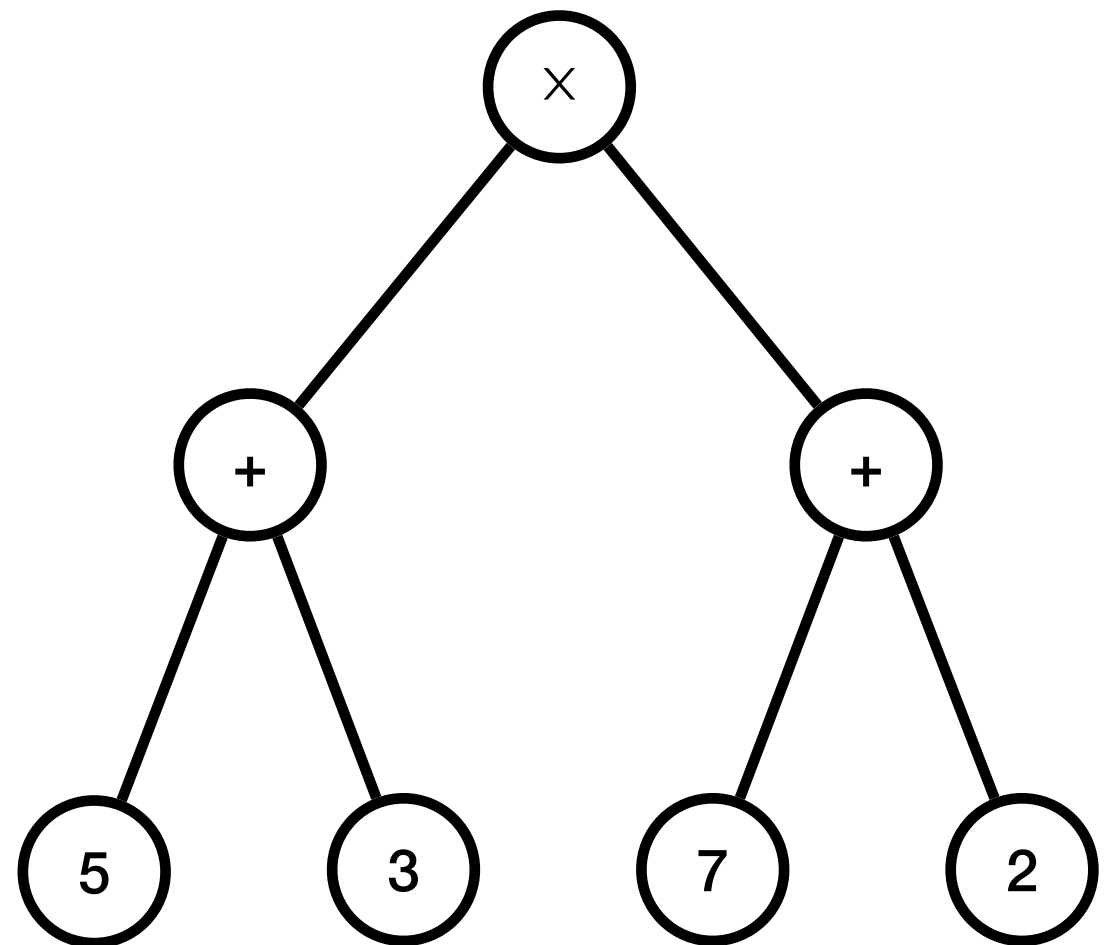
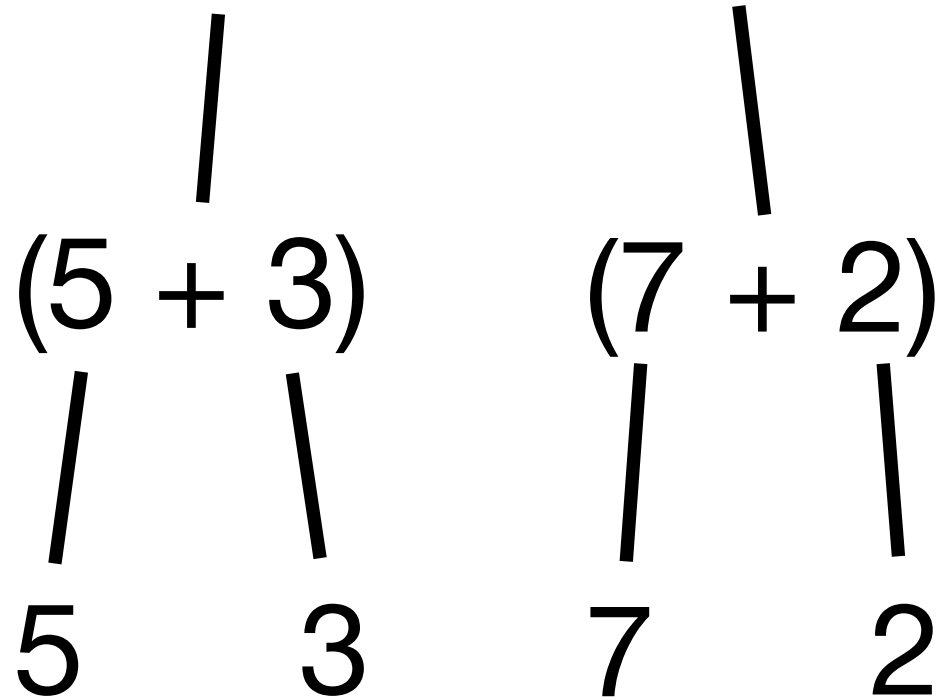


Représentation d'expressions

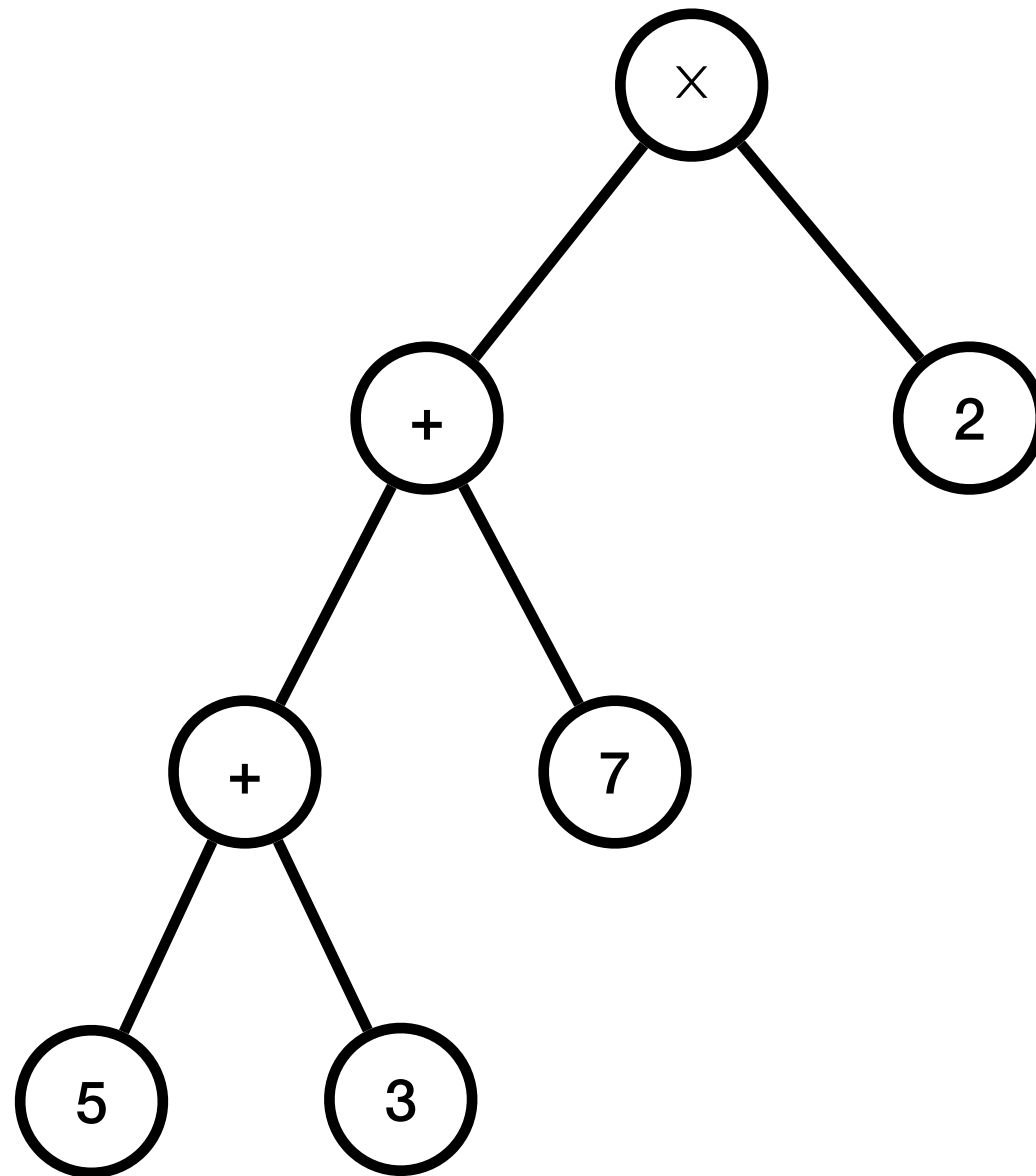


Représentation d'expressions

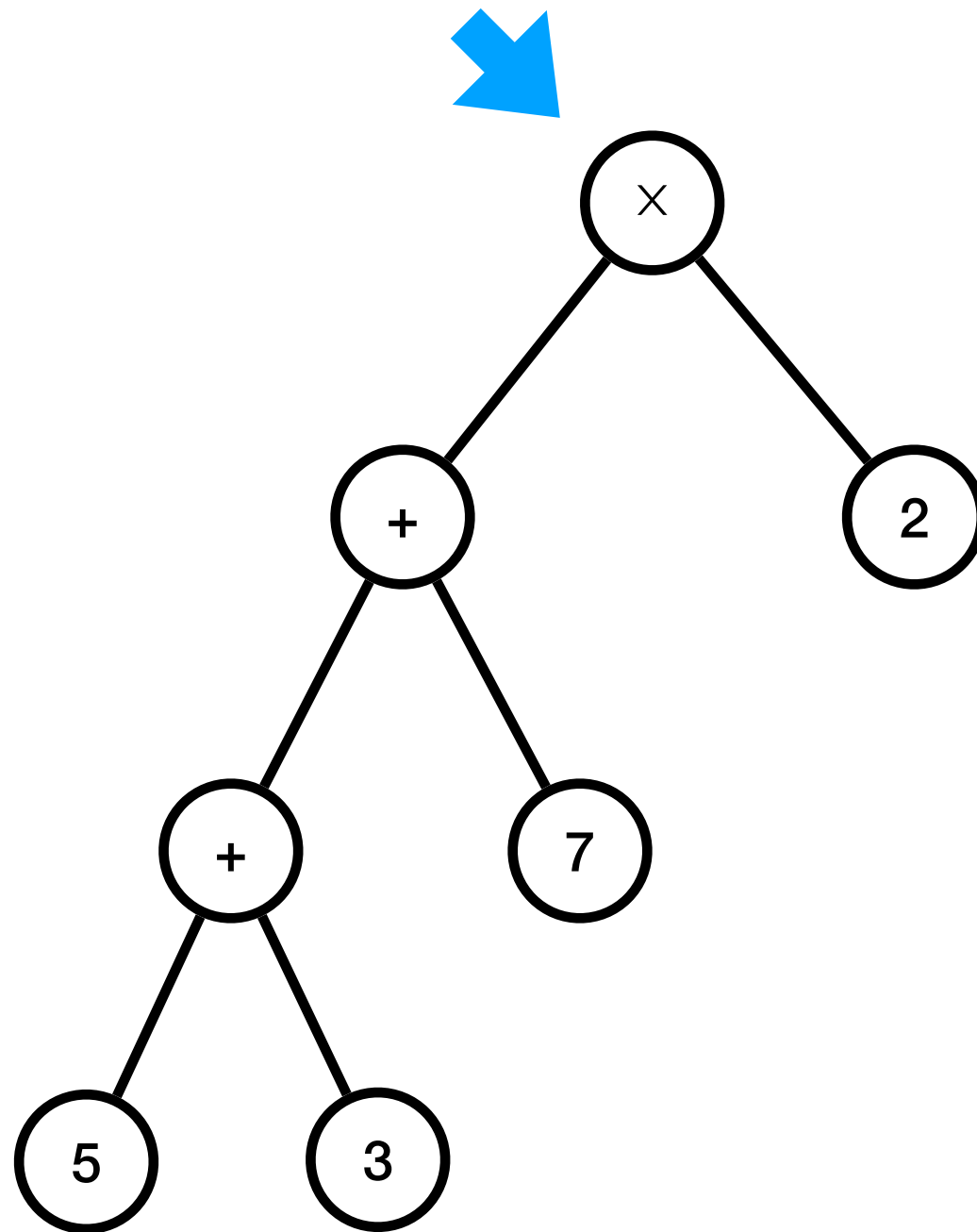
$((5 + 3) \times (7 + 2))$



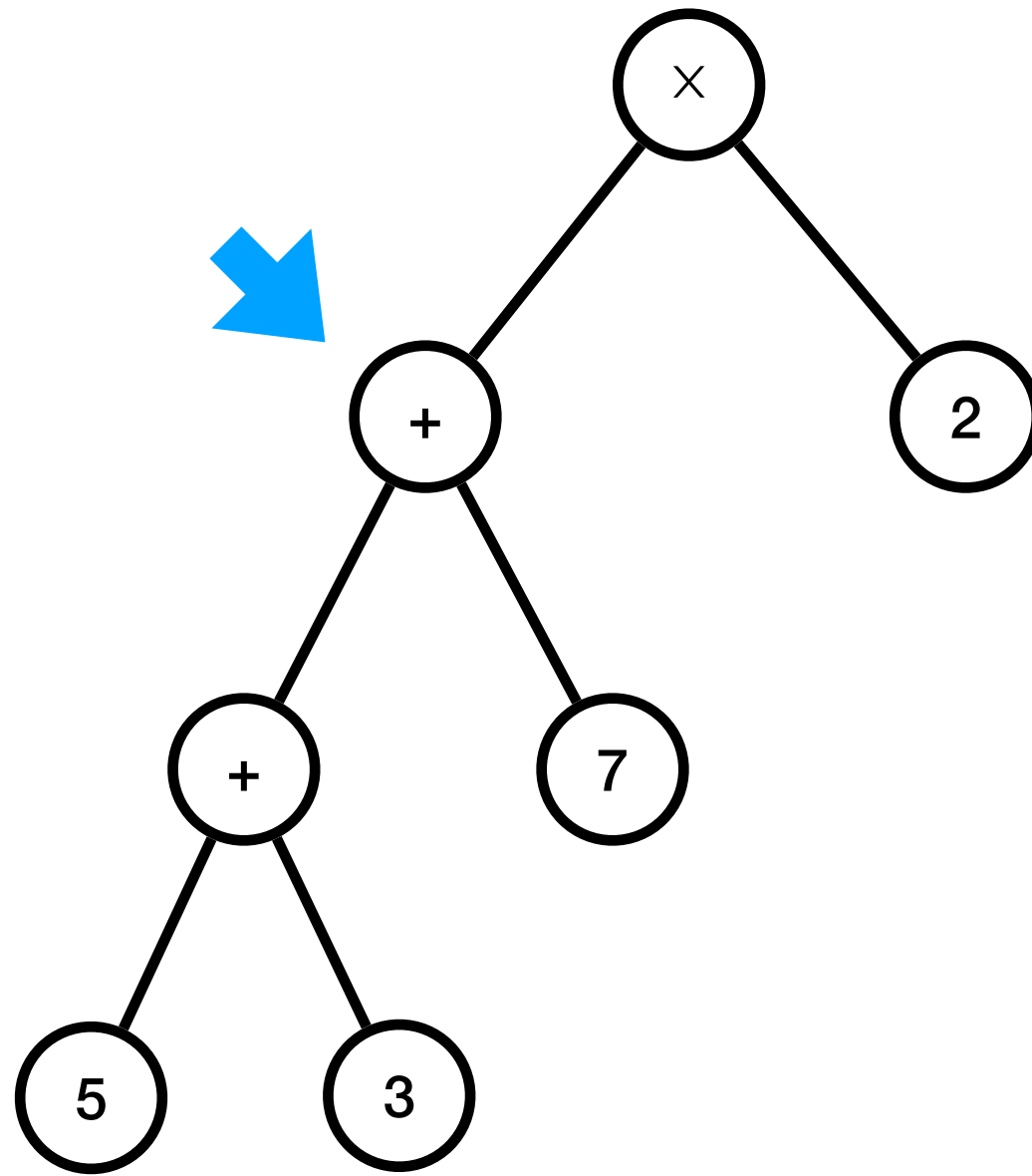
Évaluer $((5 + 3) + 7) \times 2$



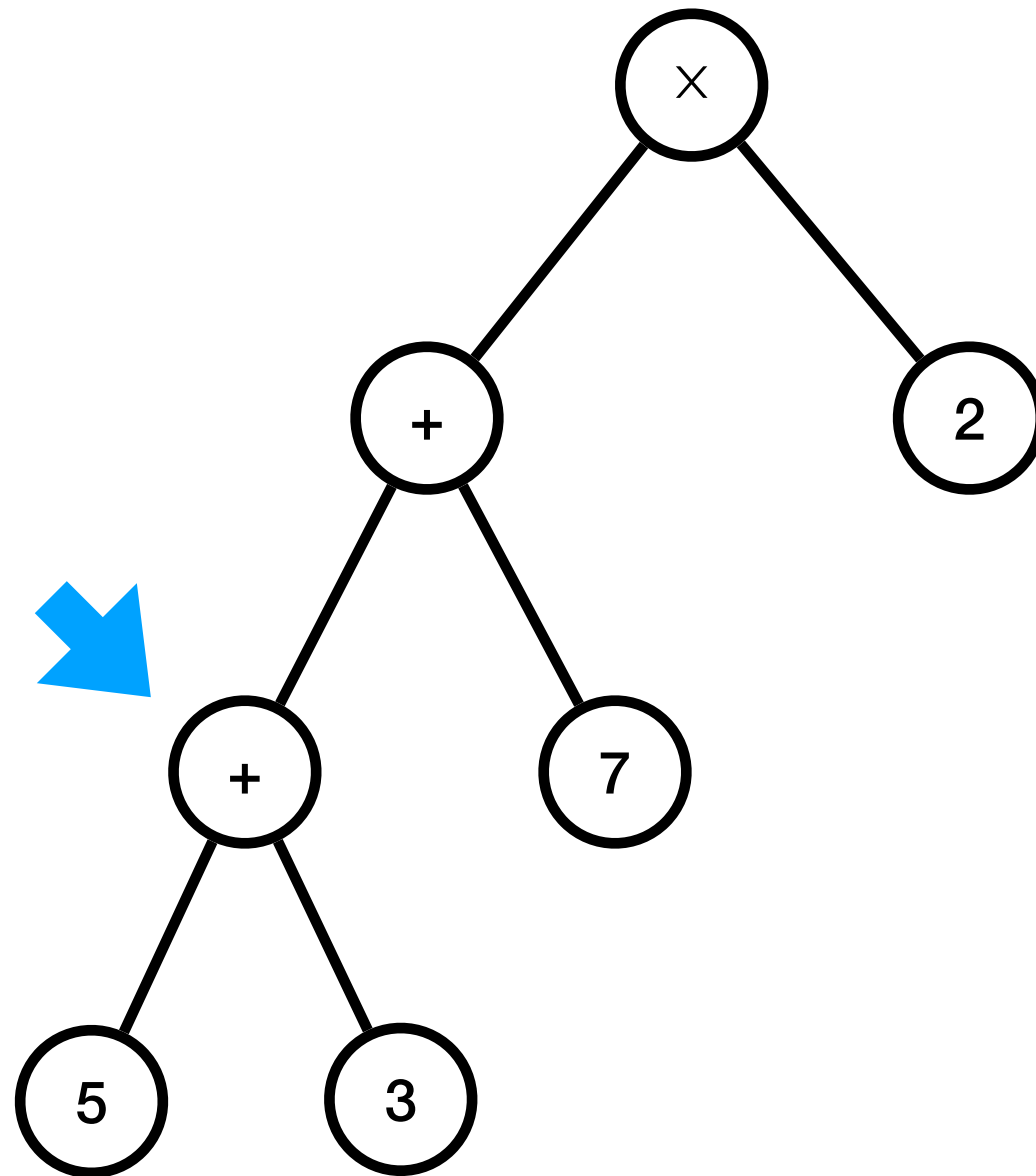
Évaluer $((5 + 3) + 7) \times 2$



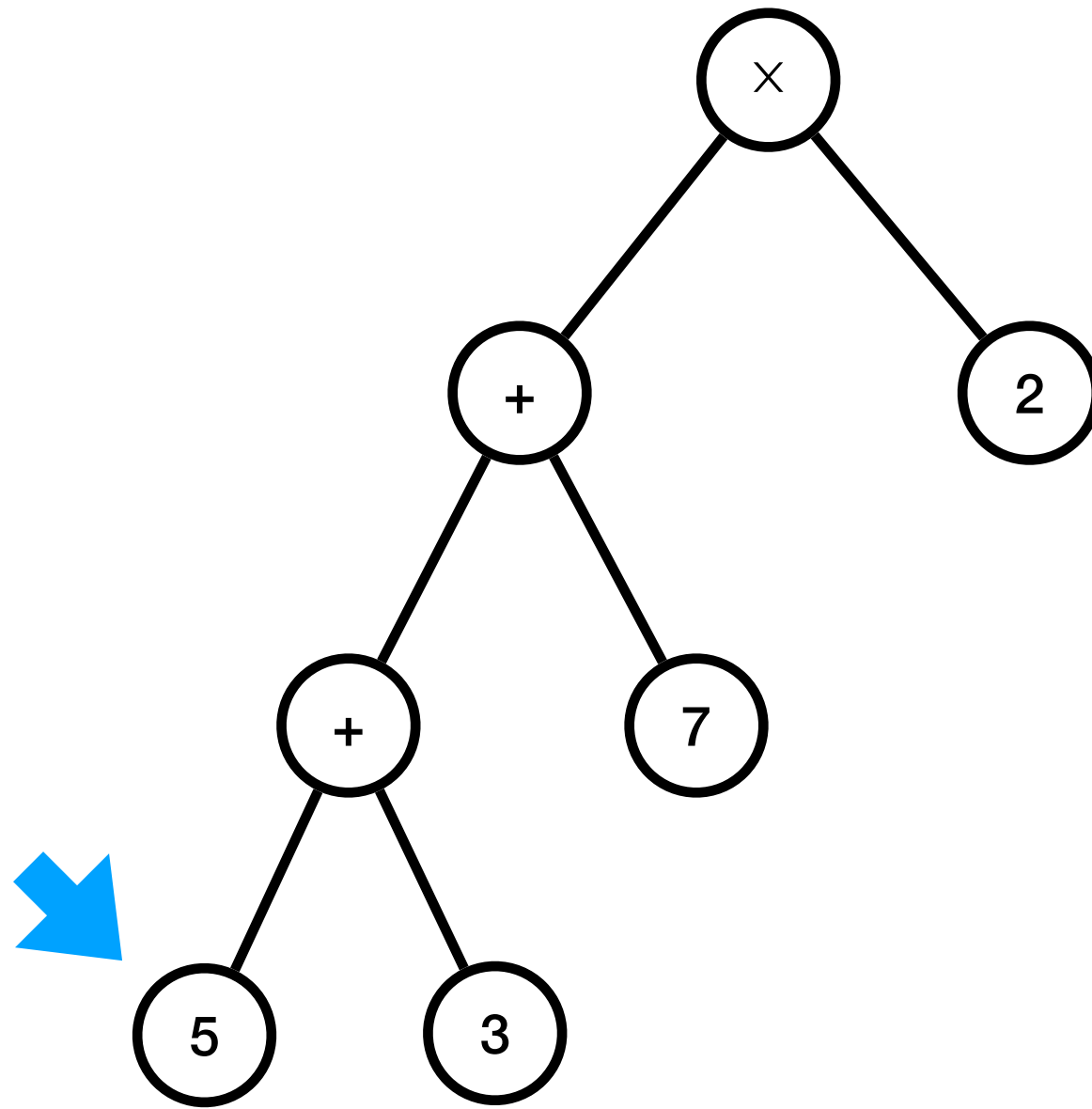
Évaluer $((5 + 3) + 7) \times 2$



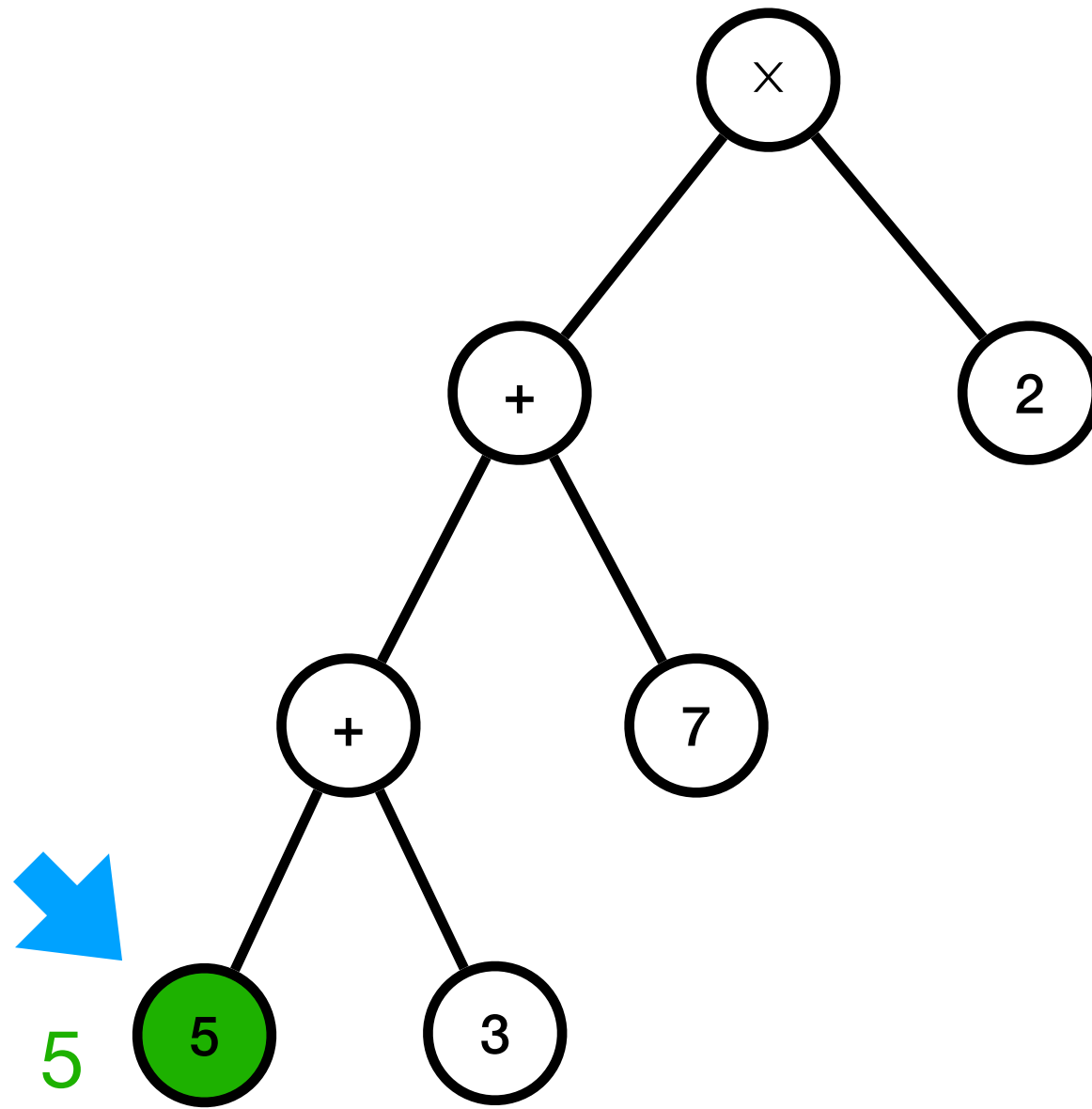
Évaluer $((5 + 3) + 7) \times 2$



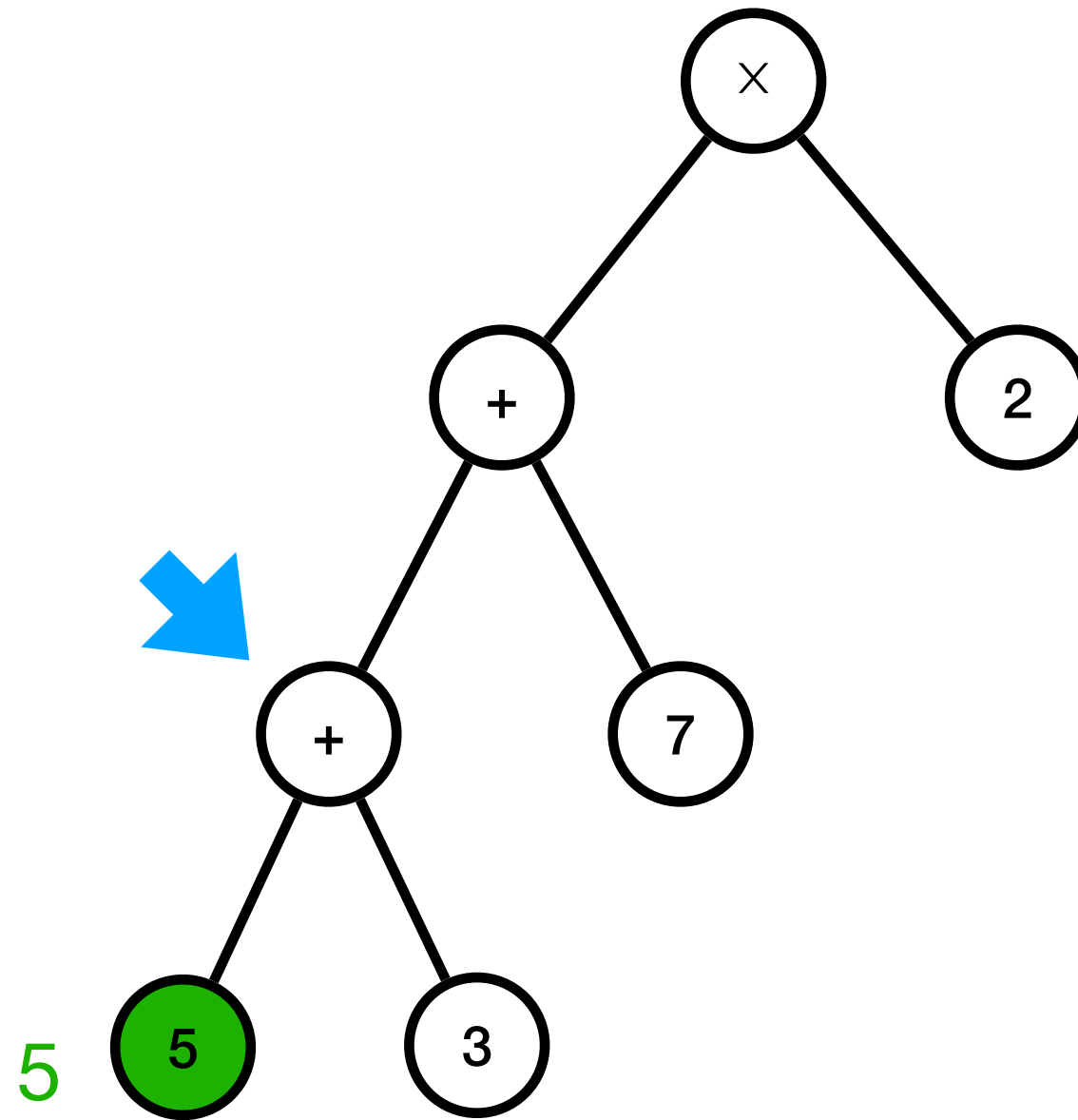
Évaluer $((5 + 3) + 7) \times 2$



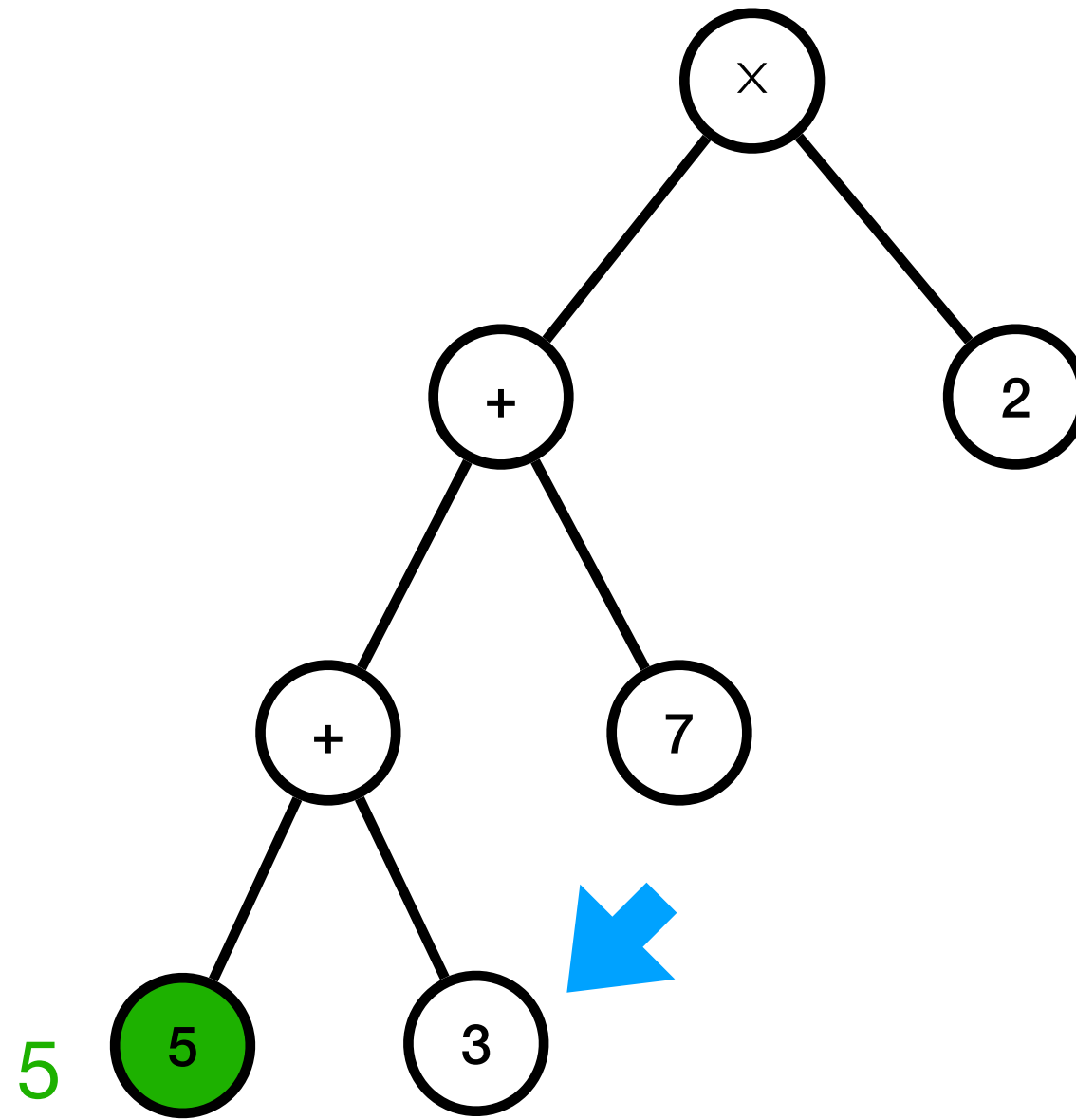
Évaluer $((5 + 3) + 7) \times 2$



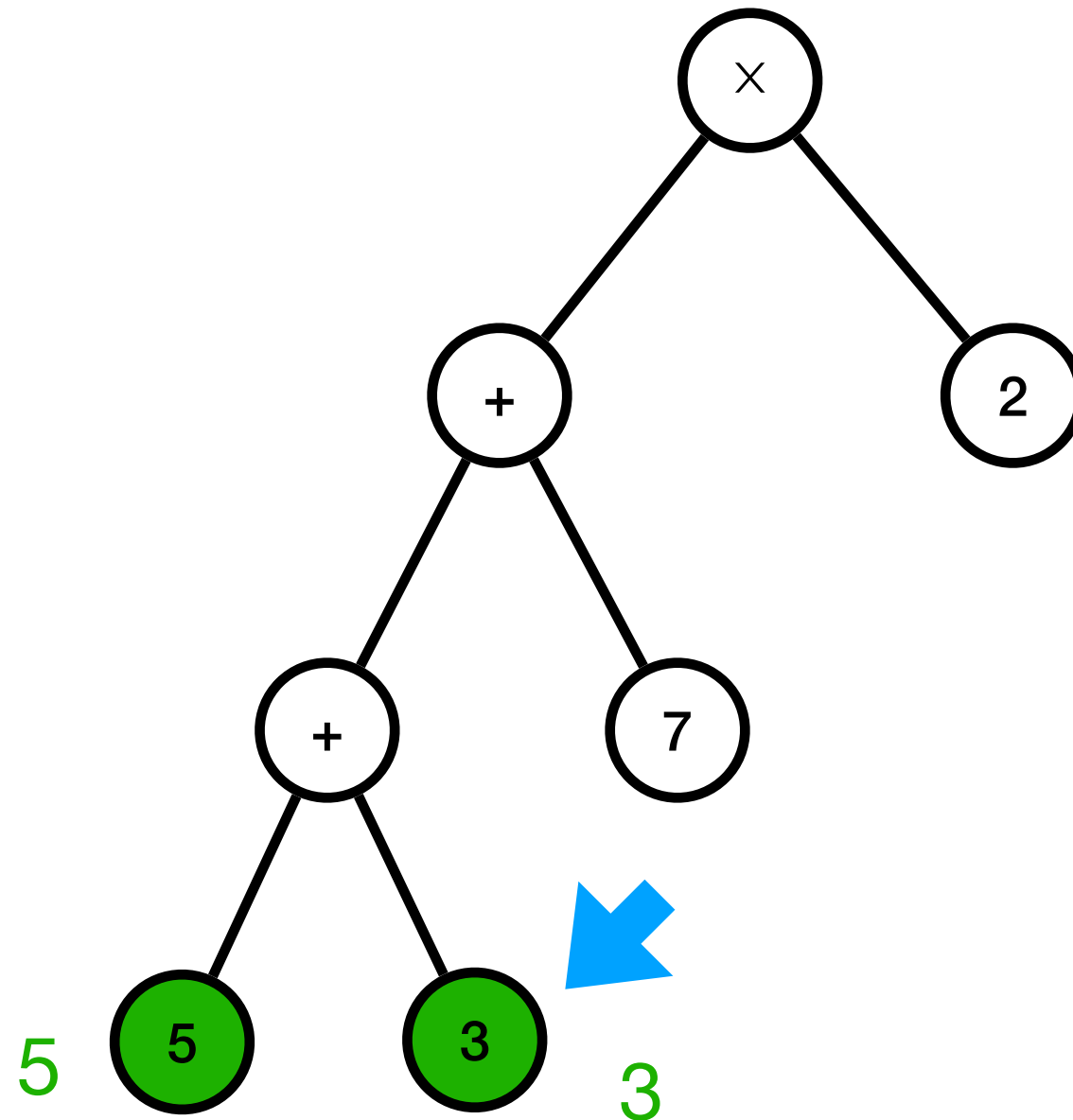
Évaluer $((5 + 3) + 7) \times 2$



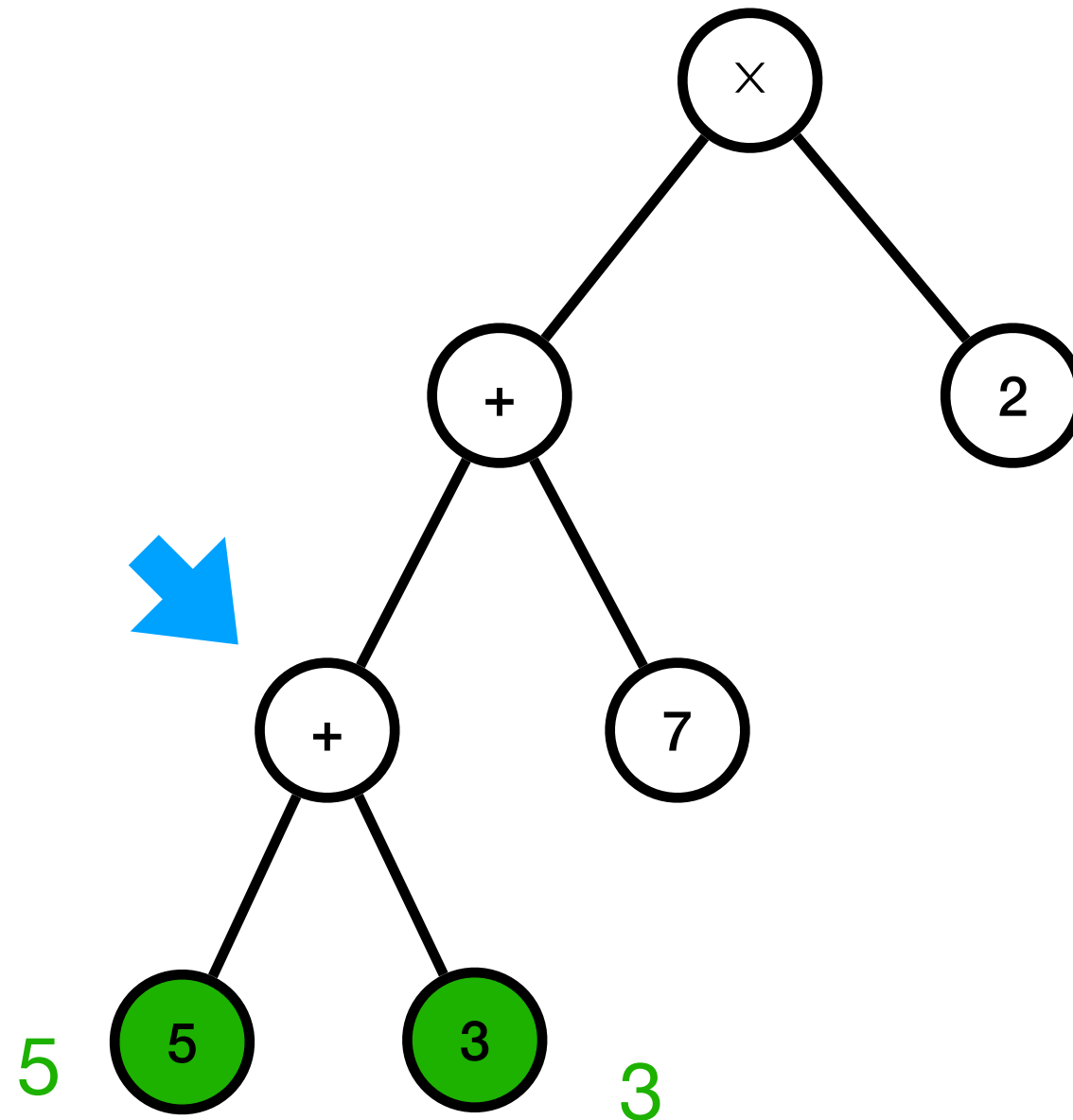
Évaluer $((5 + 3) + 7) \times 2$



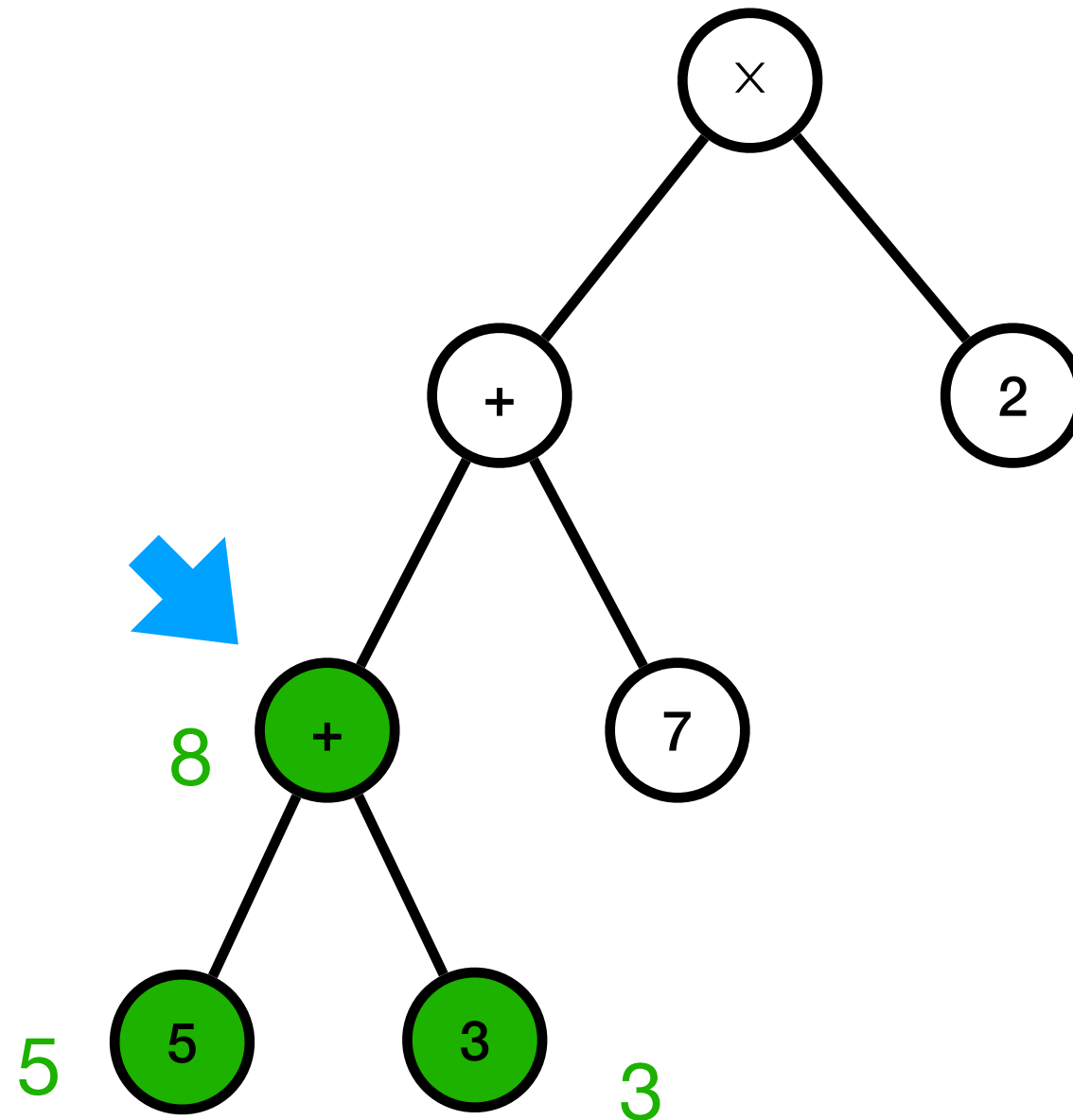
Évaluer $((5 + 3) + 7) \times 2$



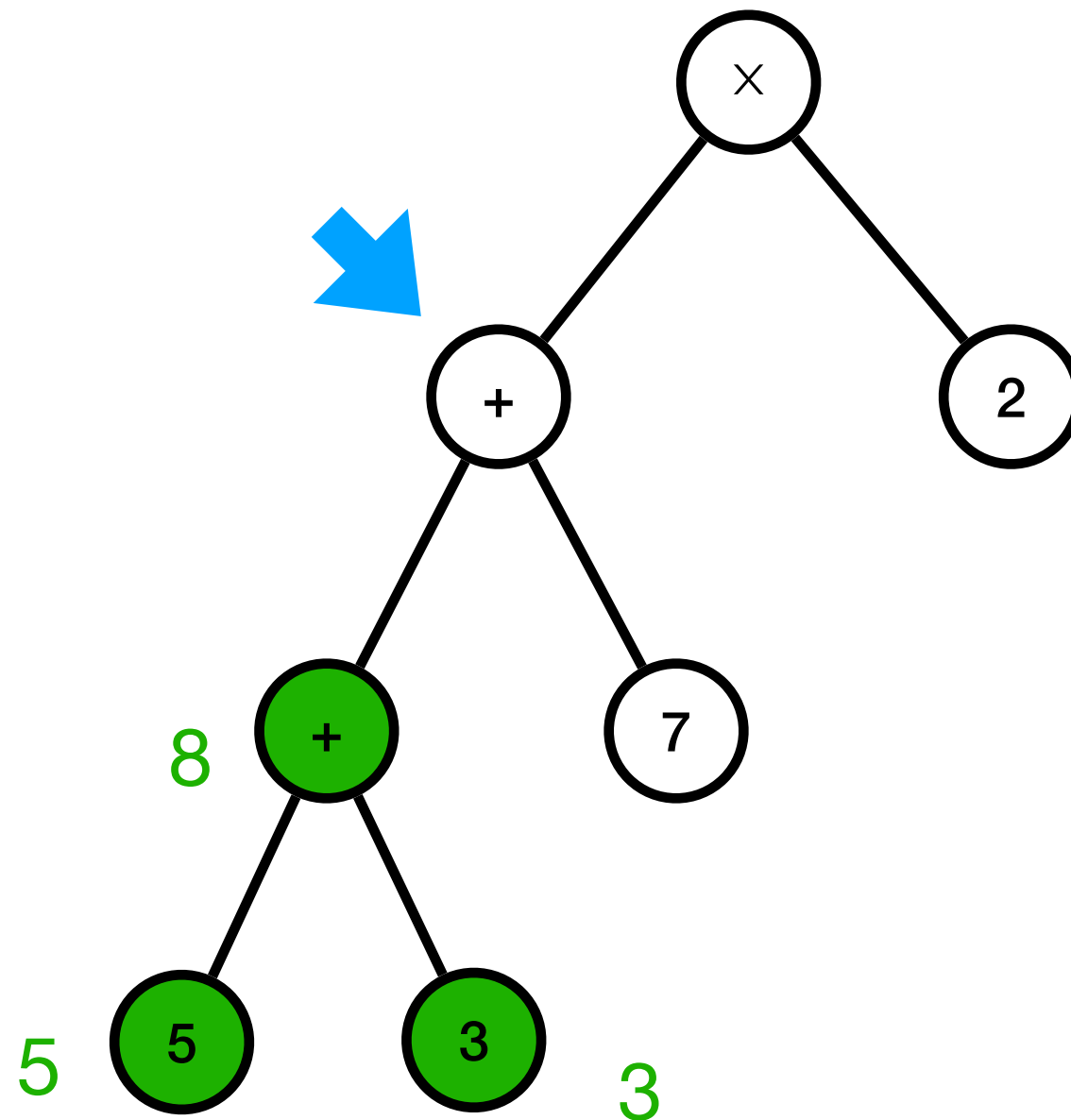
Évaluer $((5 + 3) + 7) \times 2$



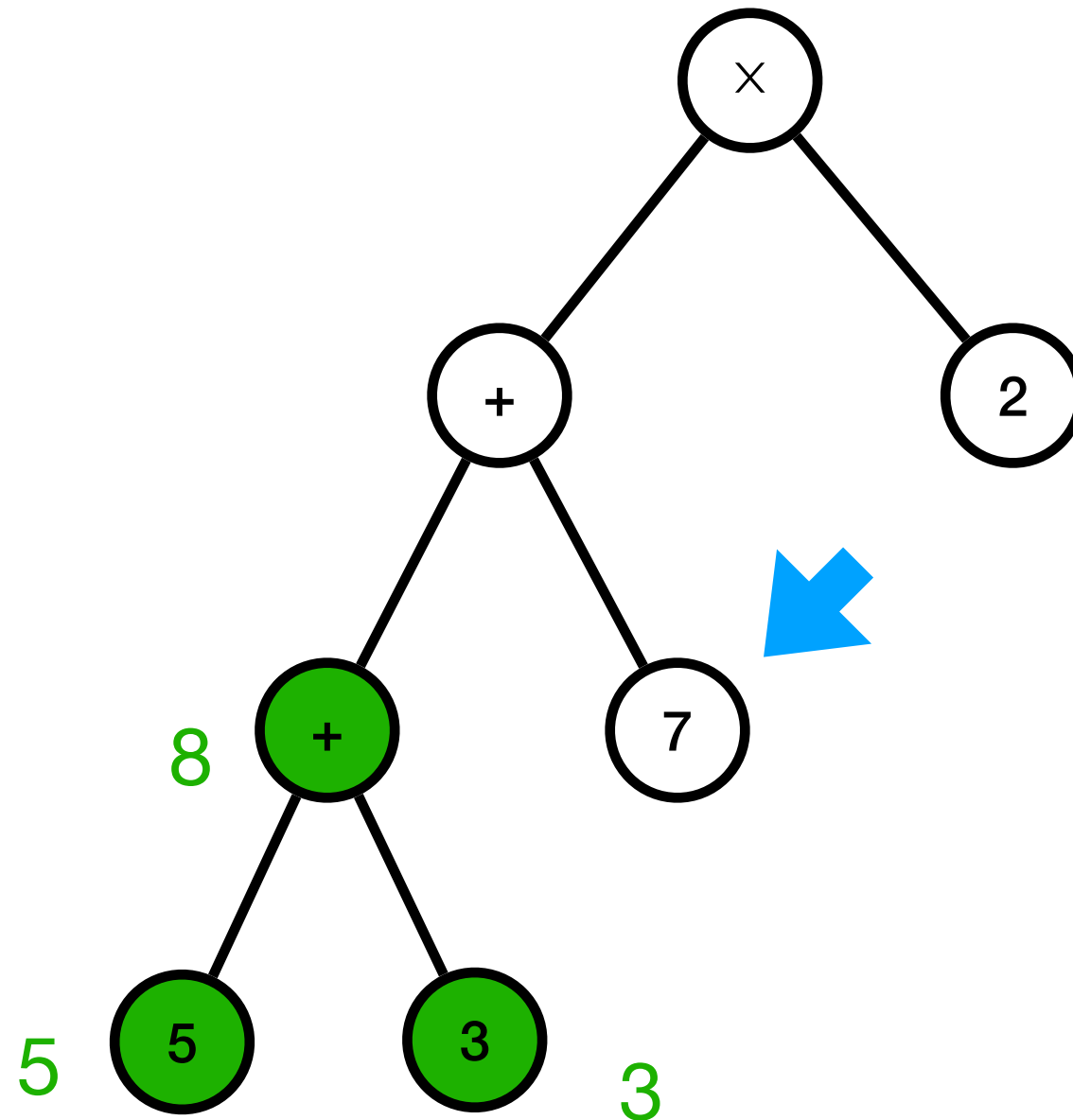
Évaluer $((5 + 3) + 7) \times 2$



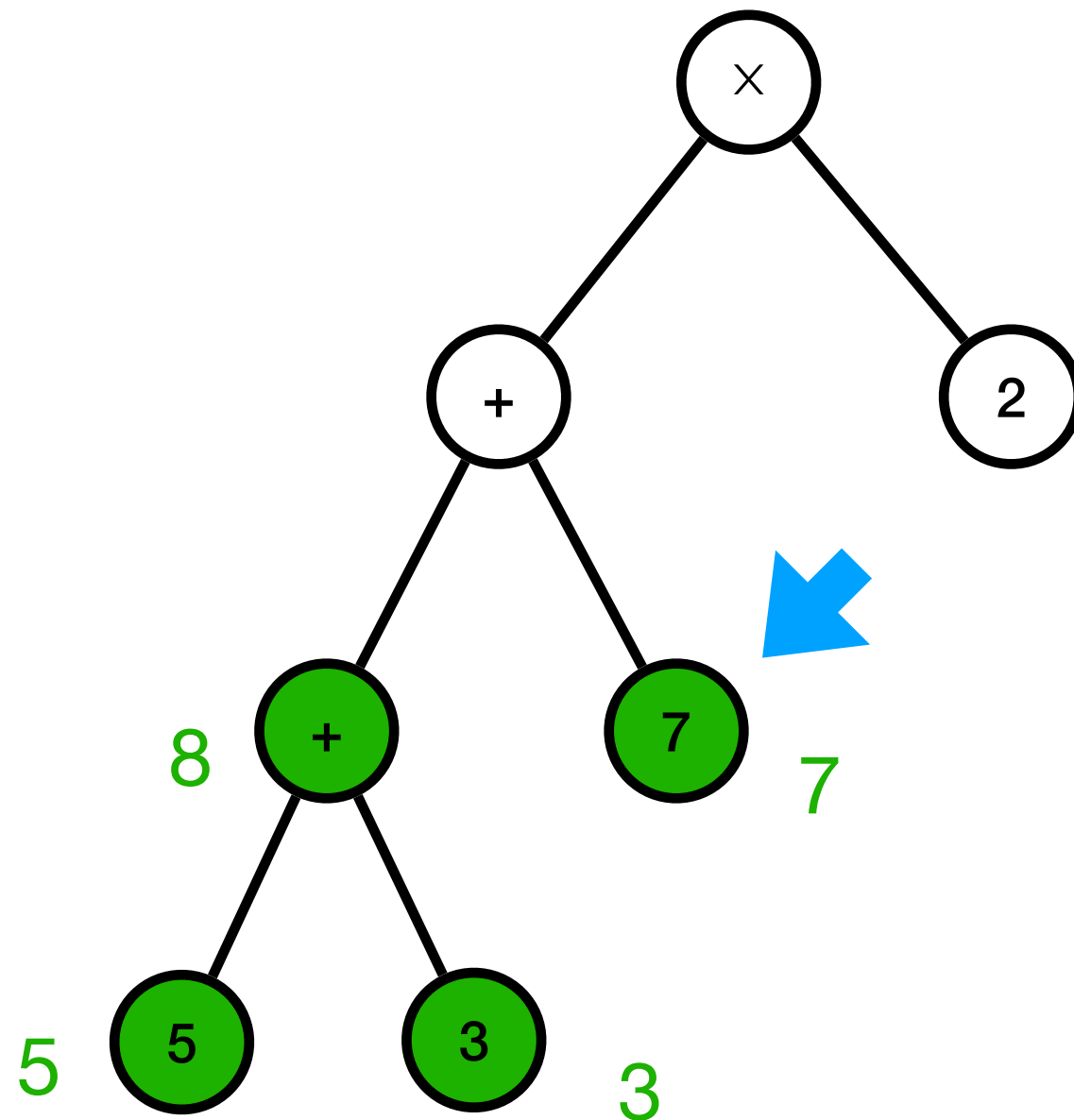
Évaluer $((5 + 3) + 7) \times 2$



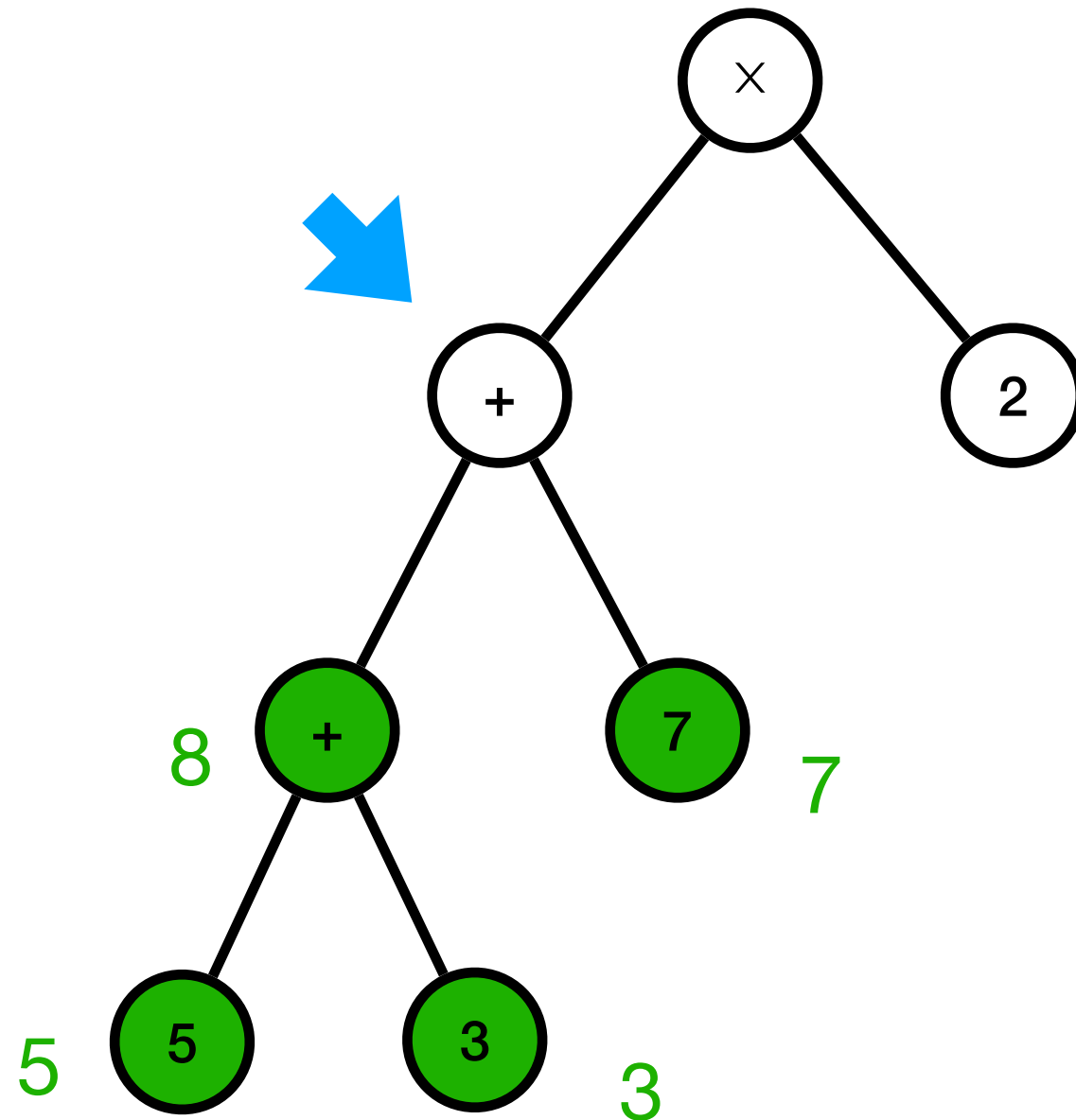
Évaluer $((5 + 3) + 7) \times 2$



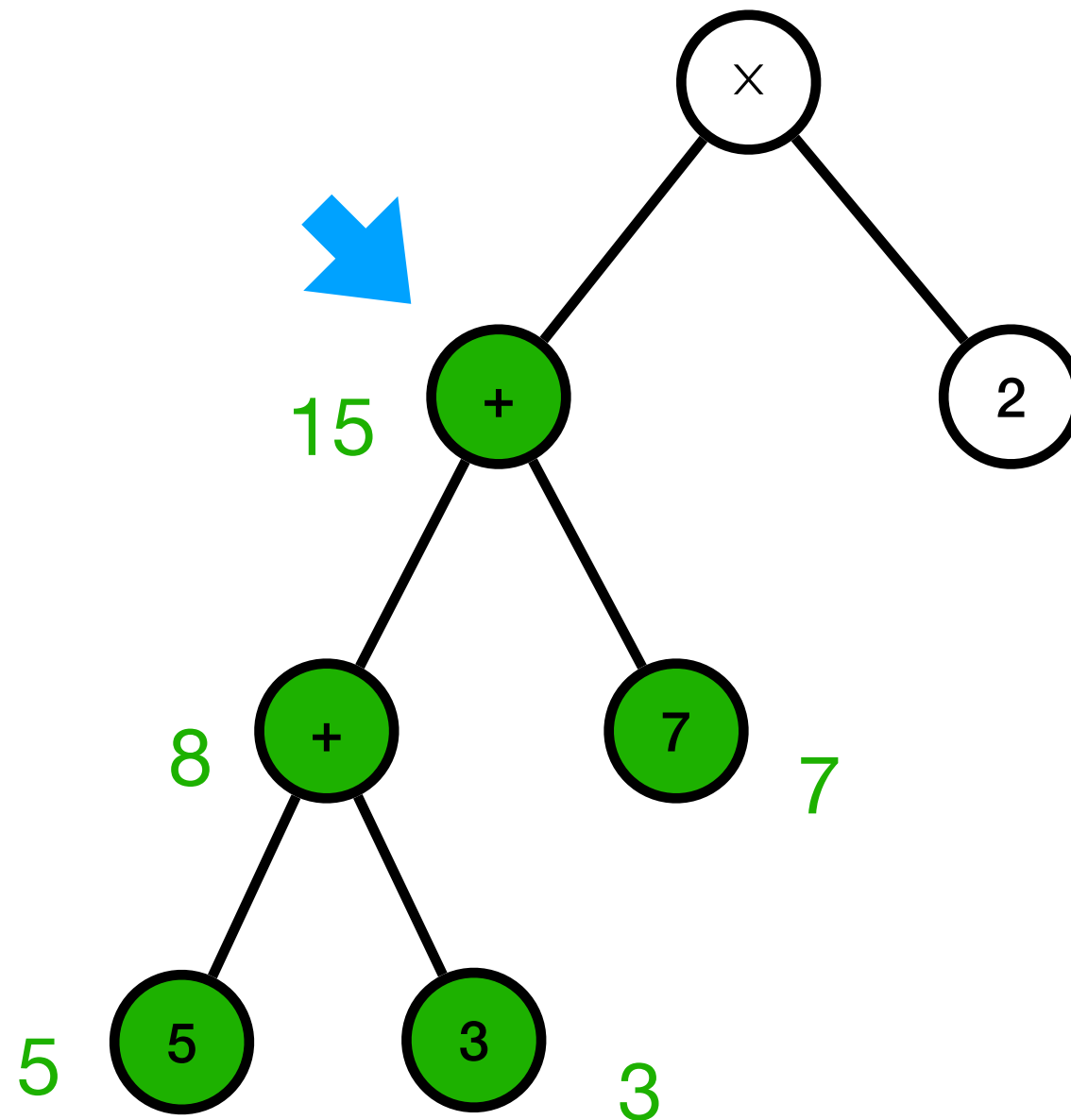
Évaluer $((5 + 3) + 7) \times 2$



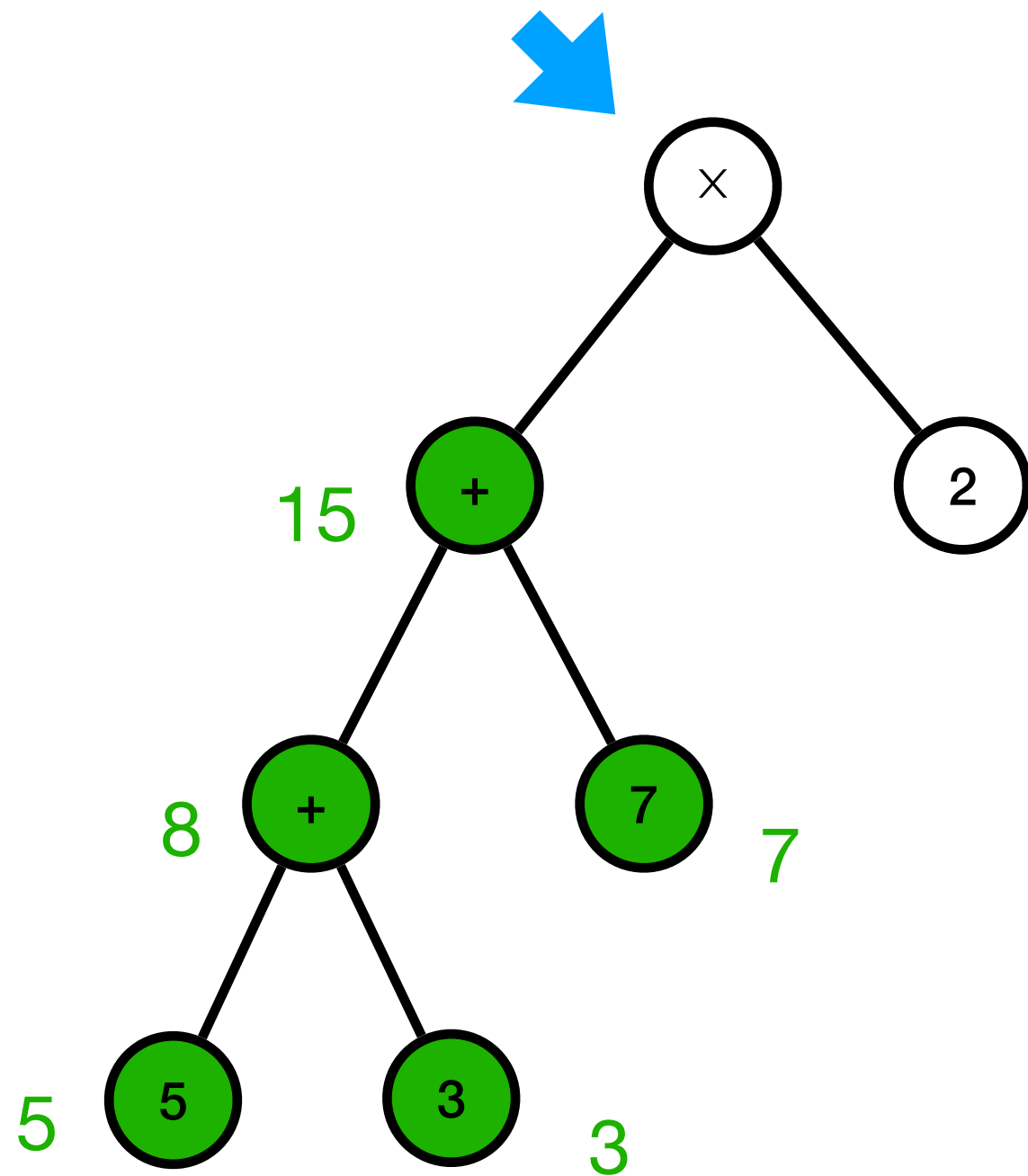
Évaluer $((5 + 3) + 7) \times 2$



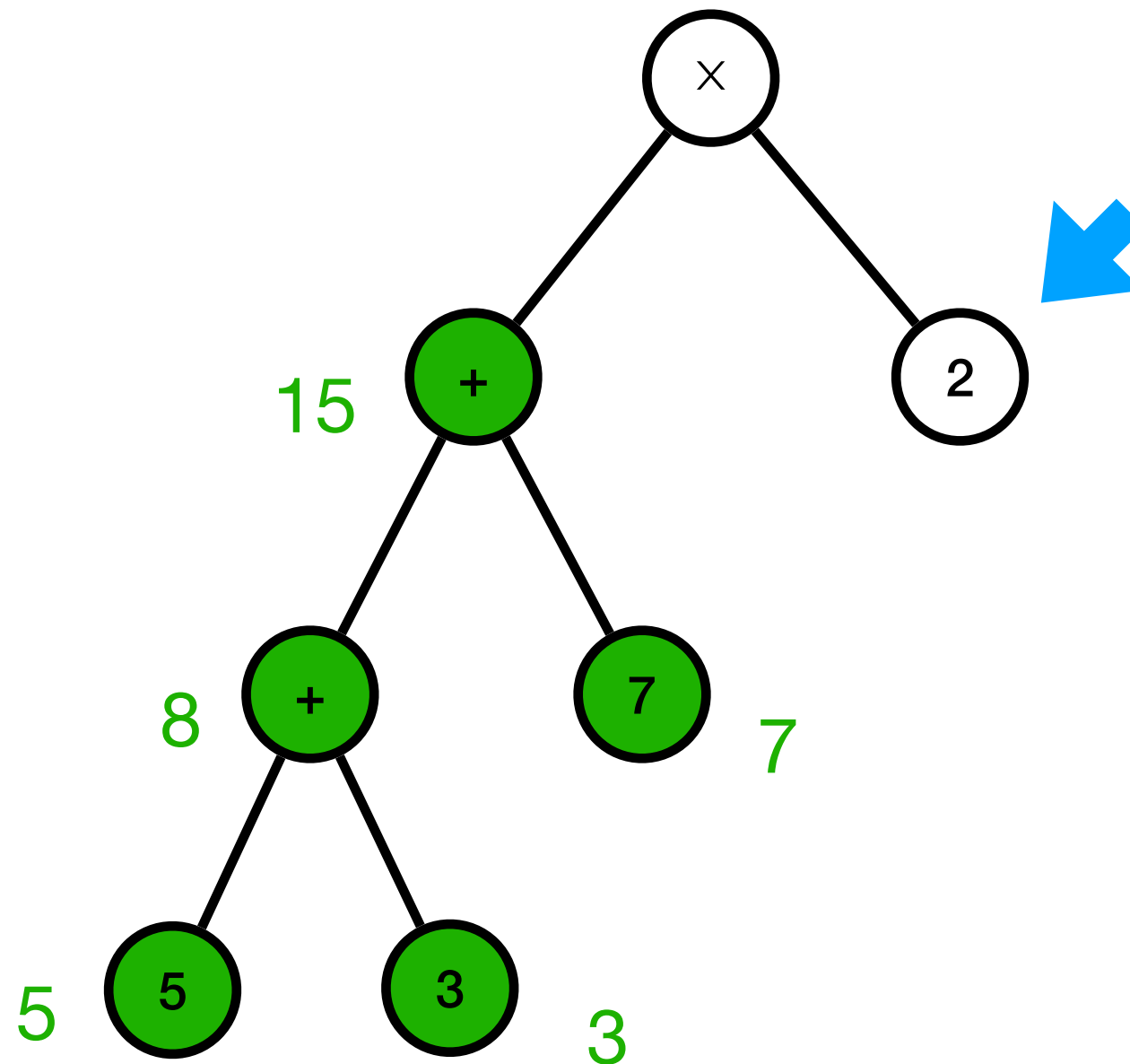
Évaluer $((5 + 3) + 7) \times 2$



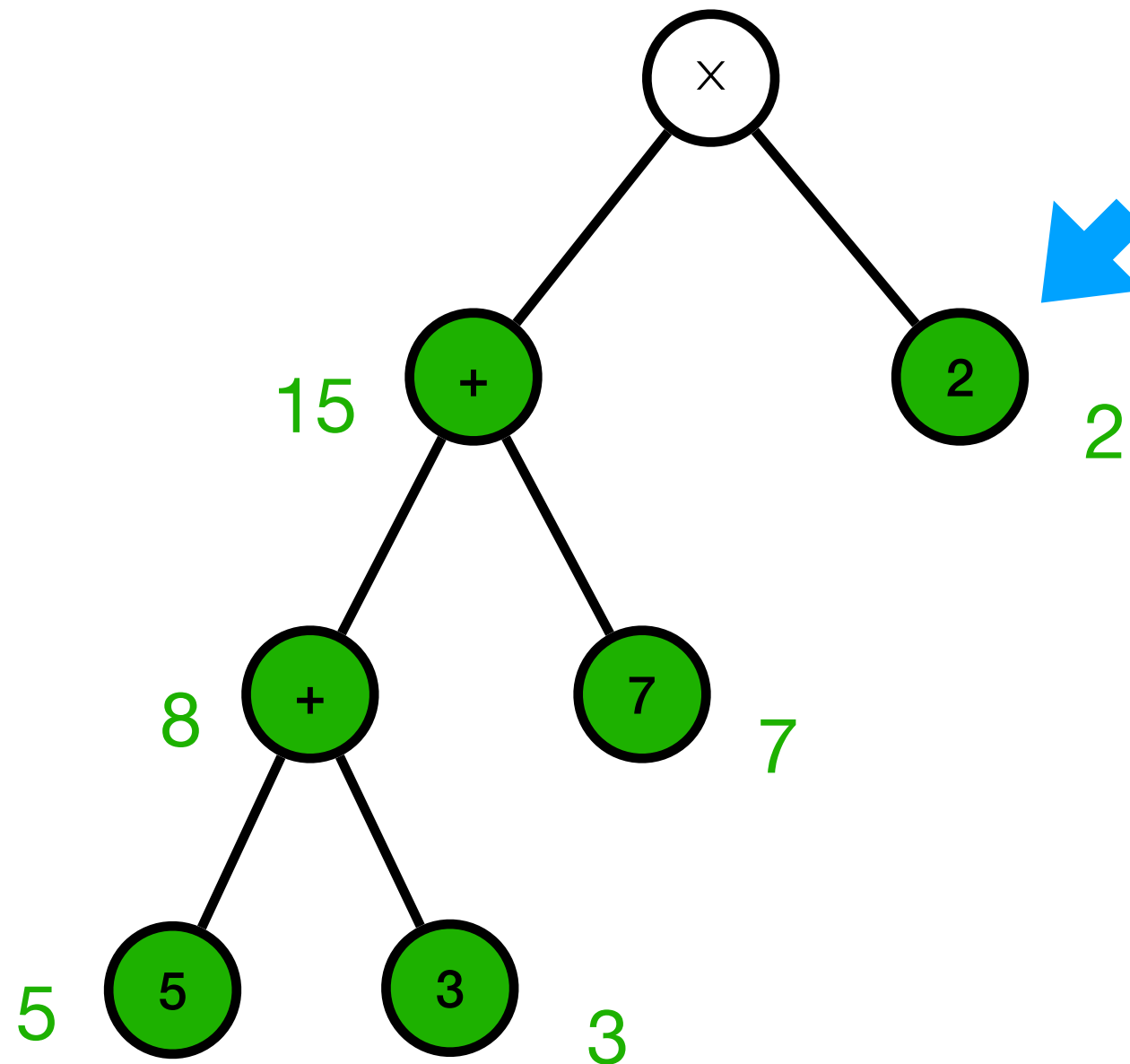
Évaluer $((5 + 3) + 7) \times 2$



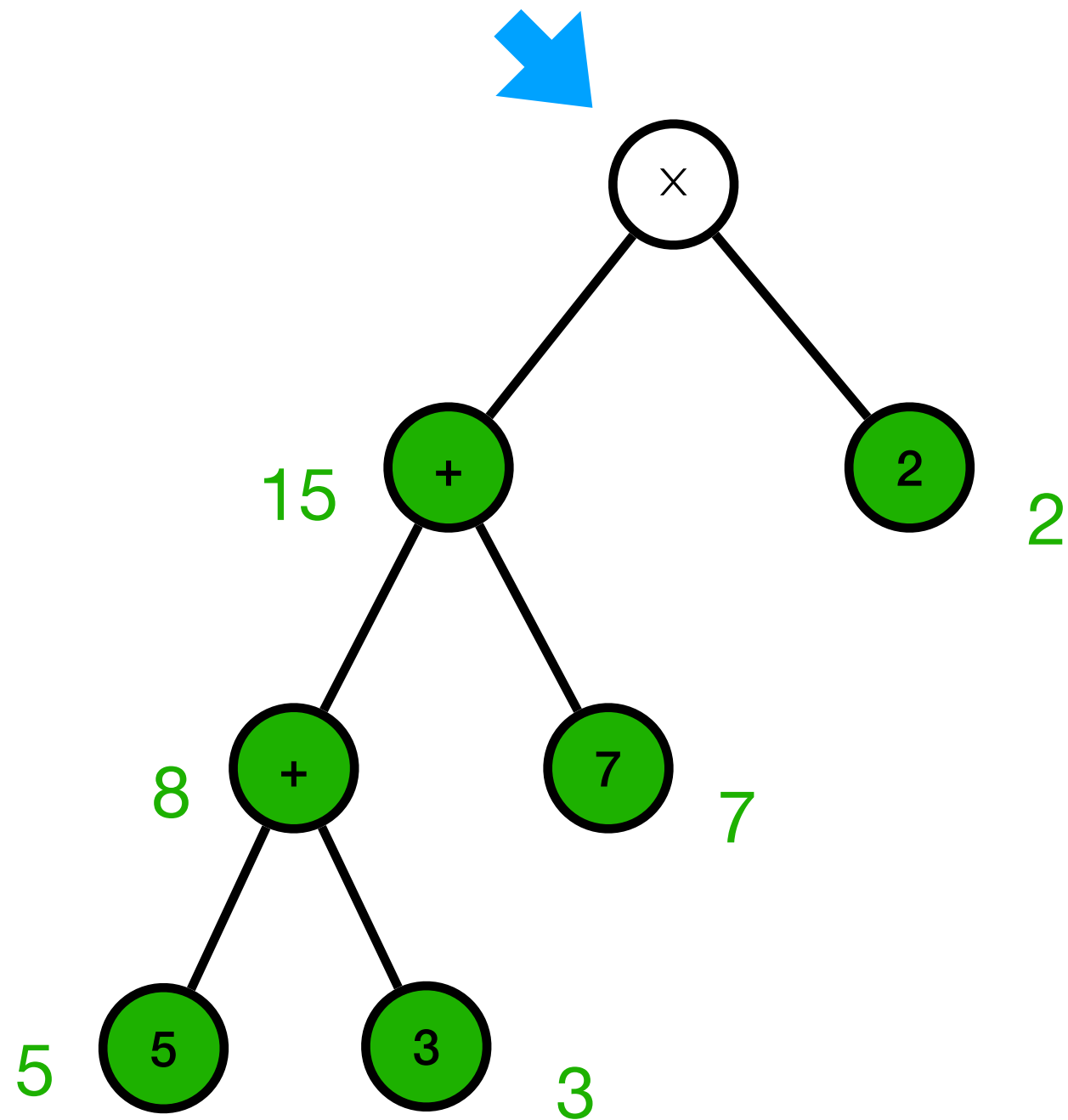
Évaluer $((5 + 3) + 7) \times 2$



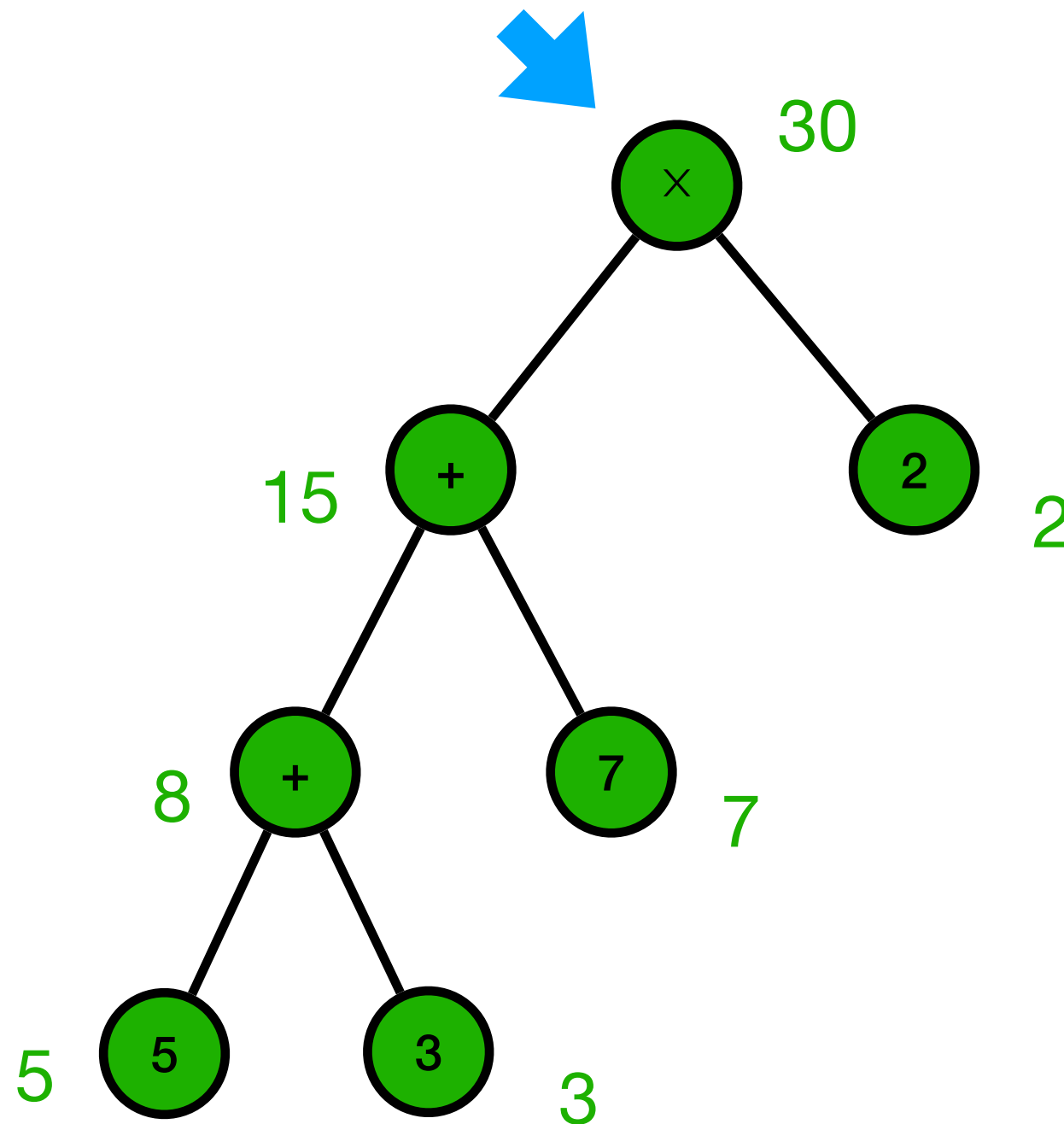
Évaluer $((5 + 3) + 7) \times 2$



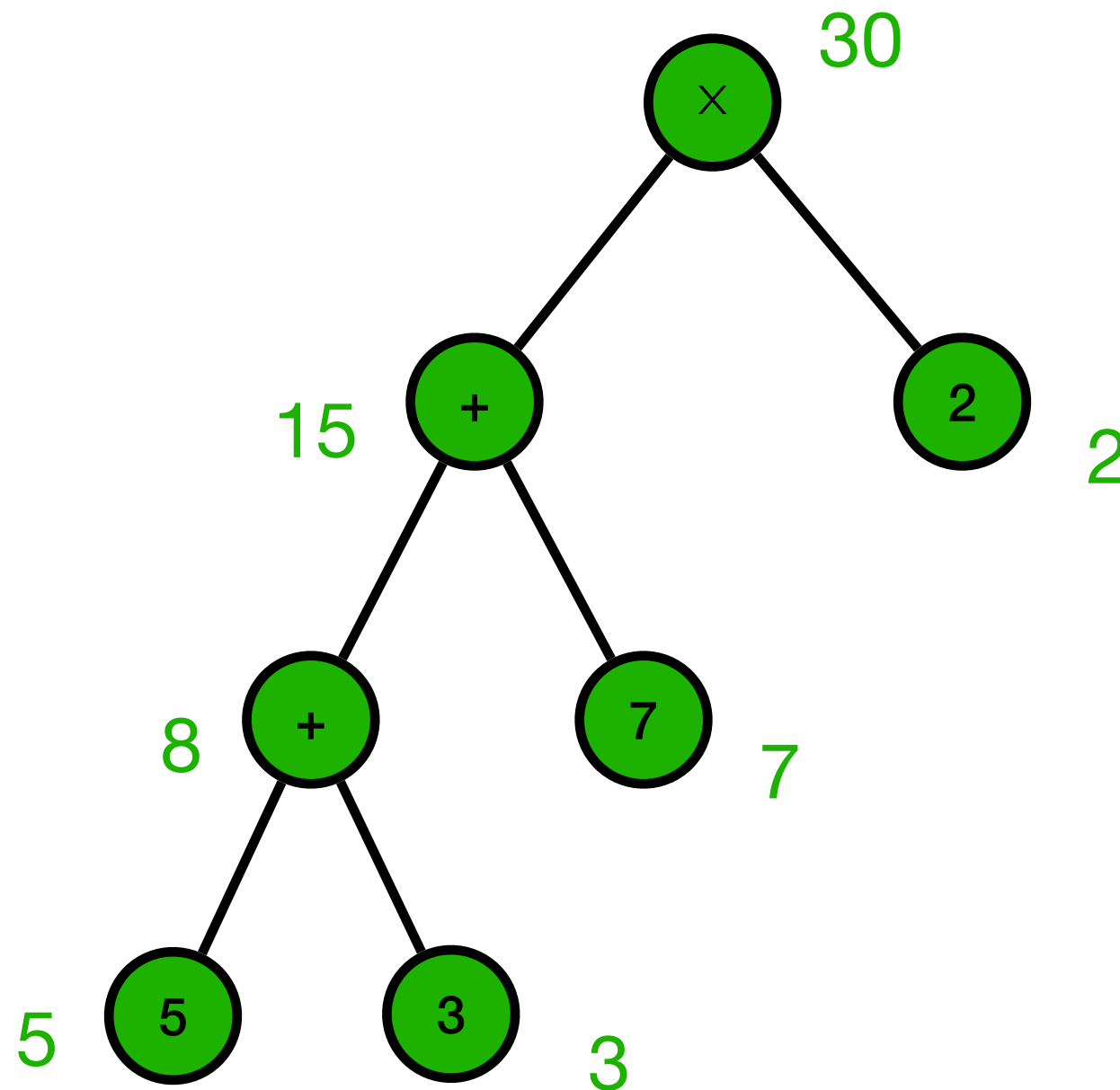
Évaluer $((5 + 3) + 7) \times 2$



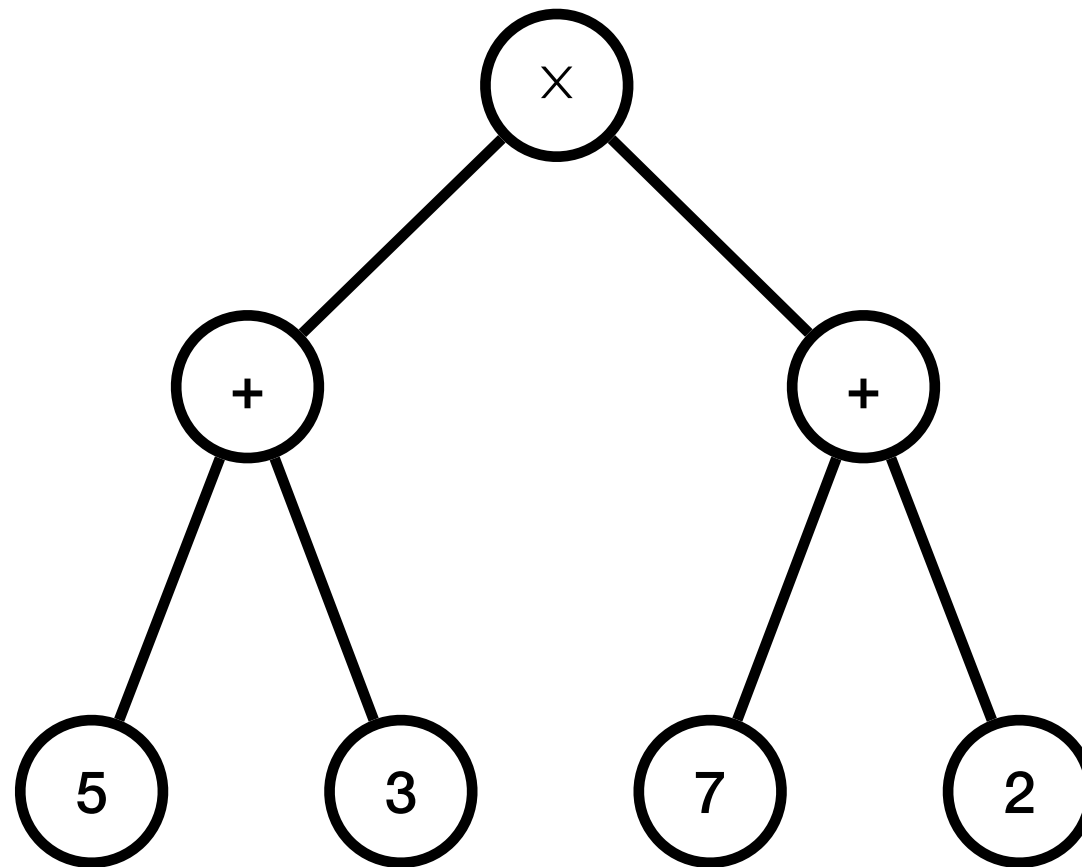
Évaluer $((5 + 3) + 7) \times 2$



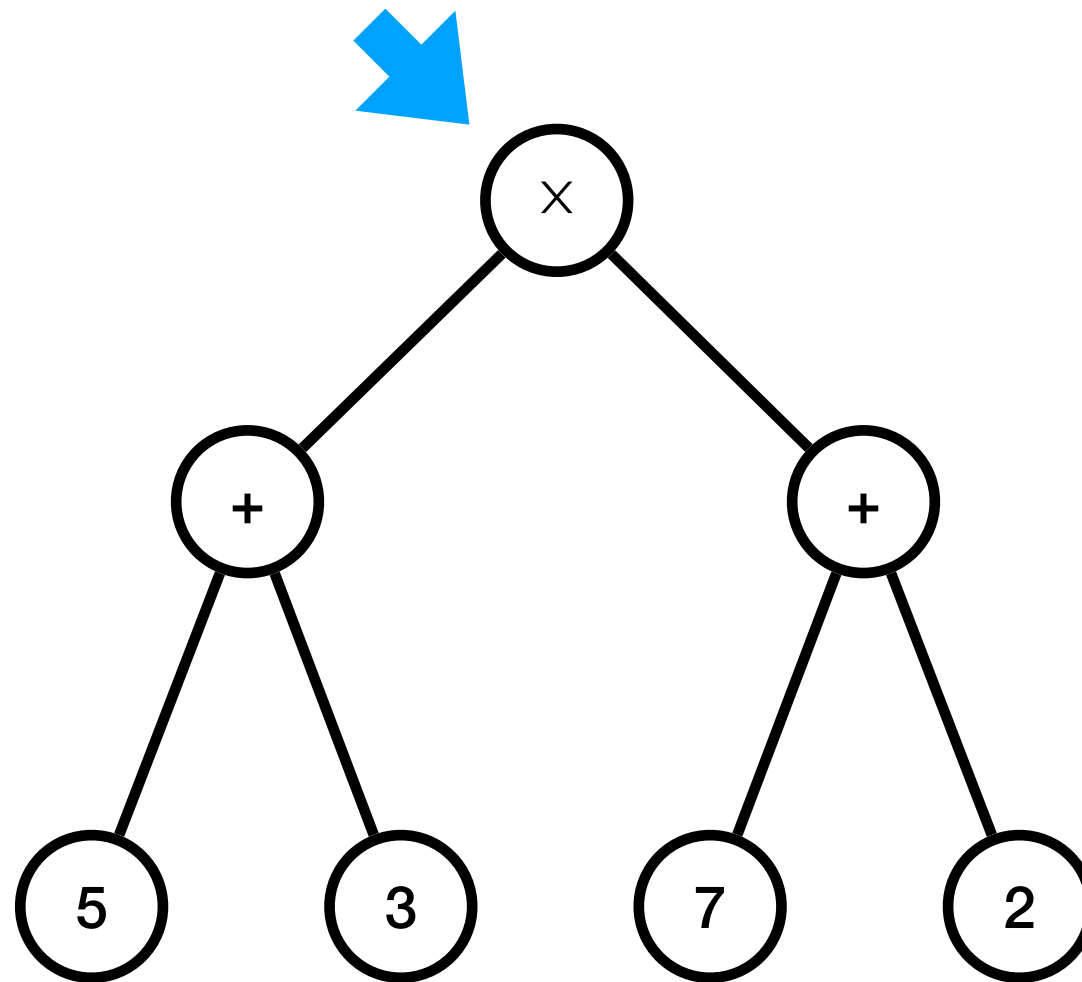
Évaluer $((5 + 3) + 7) \times 2$



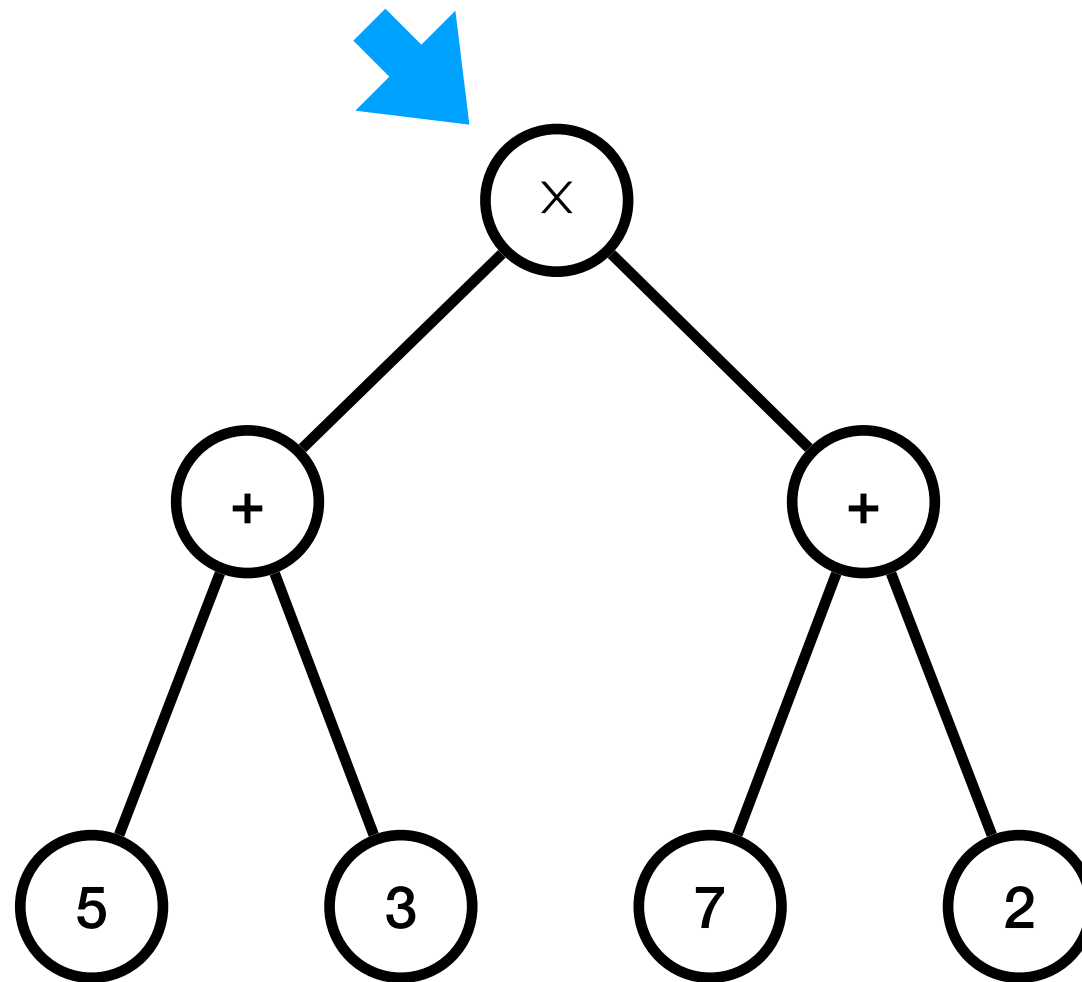
Afficher $((5 + 3) \times (7 + 2))$



Afficher $((5 + 3) \times (7 + 2))$

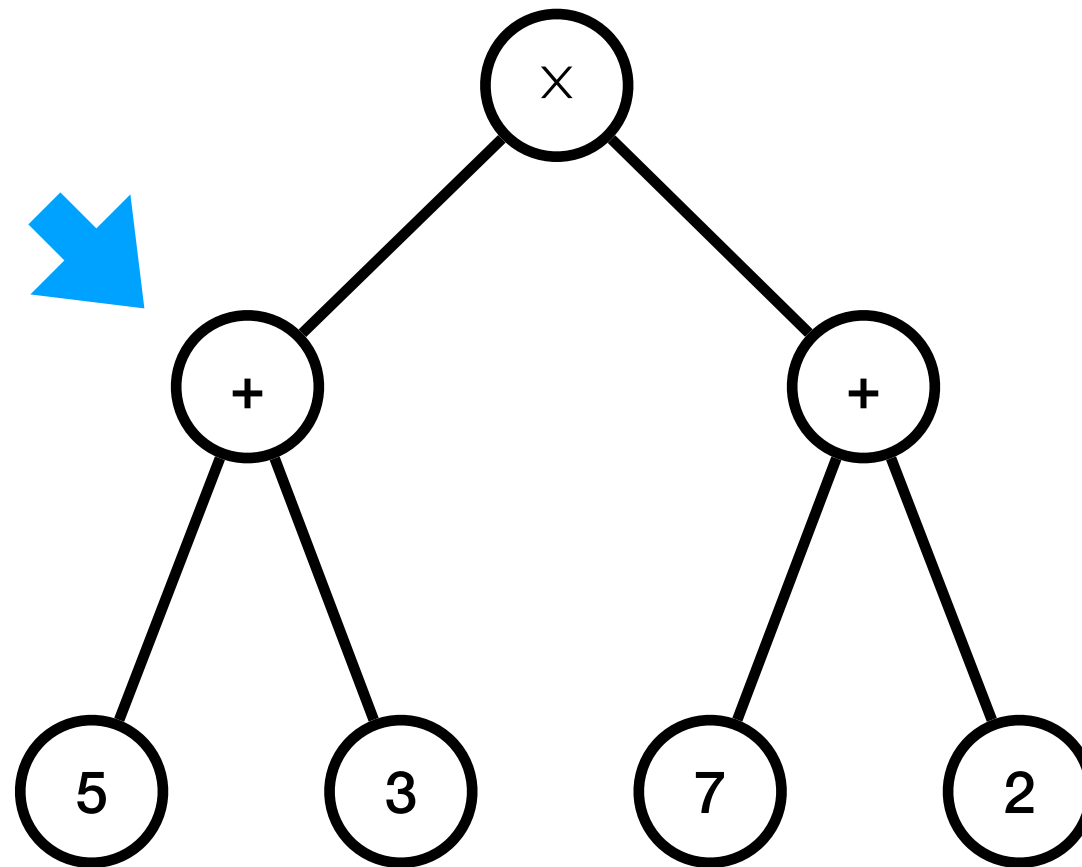


Afficher $((5 + 3) \times (7 + 2))$



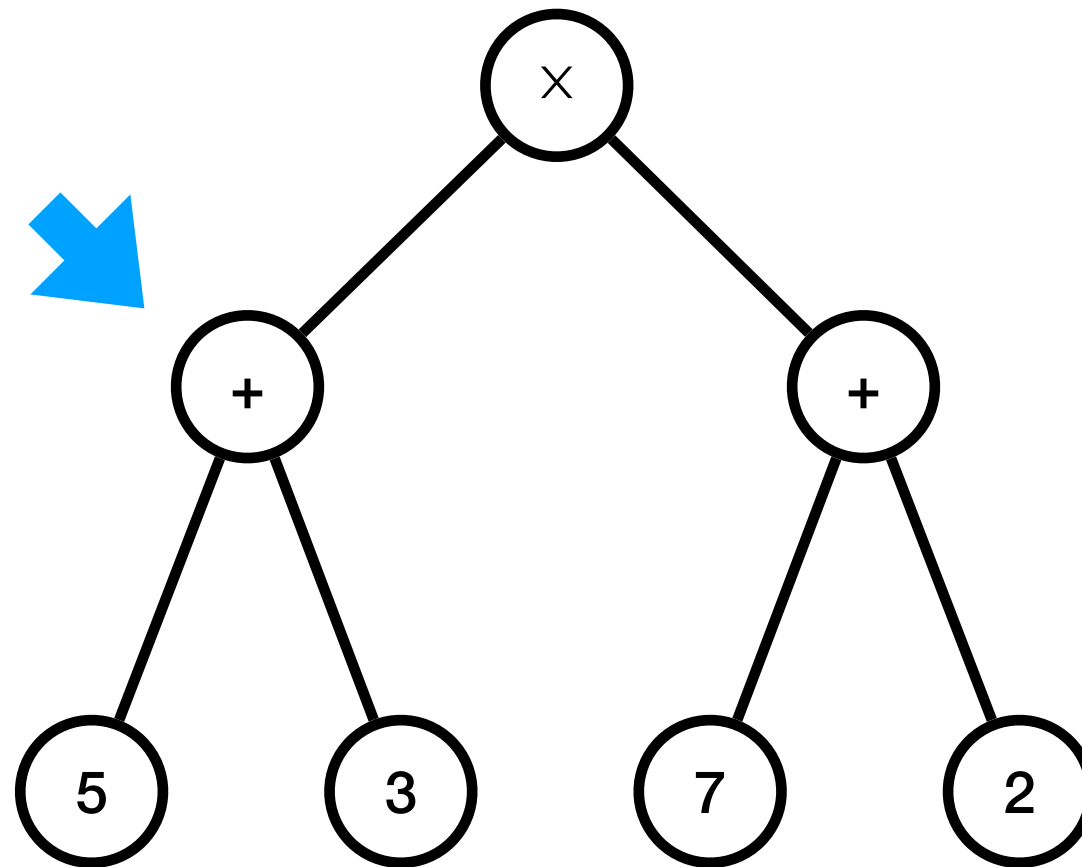
(

Afficher $((5 + 3) \times (7 + 2))$



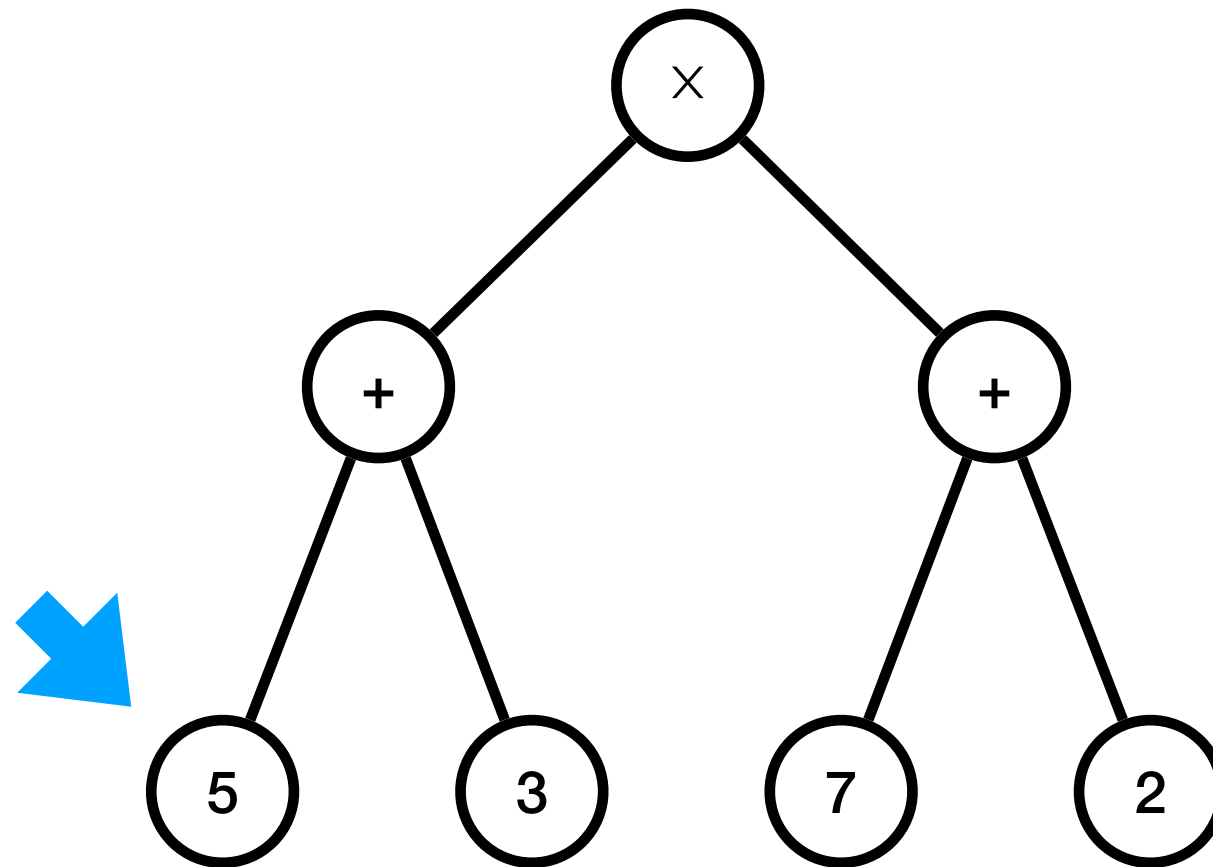
(

Afficher $((5 + 3) \times (7 + 2))$



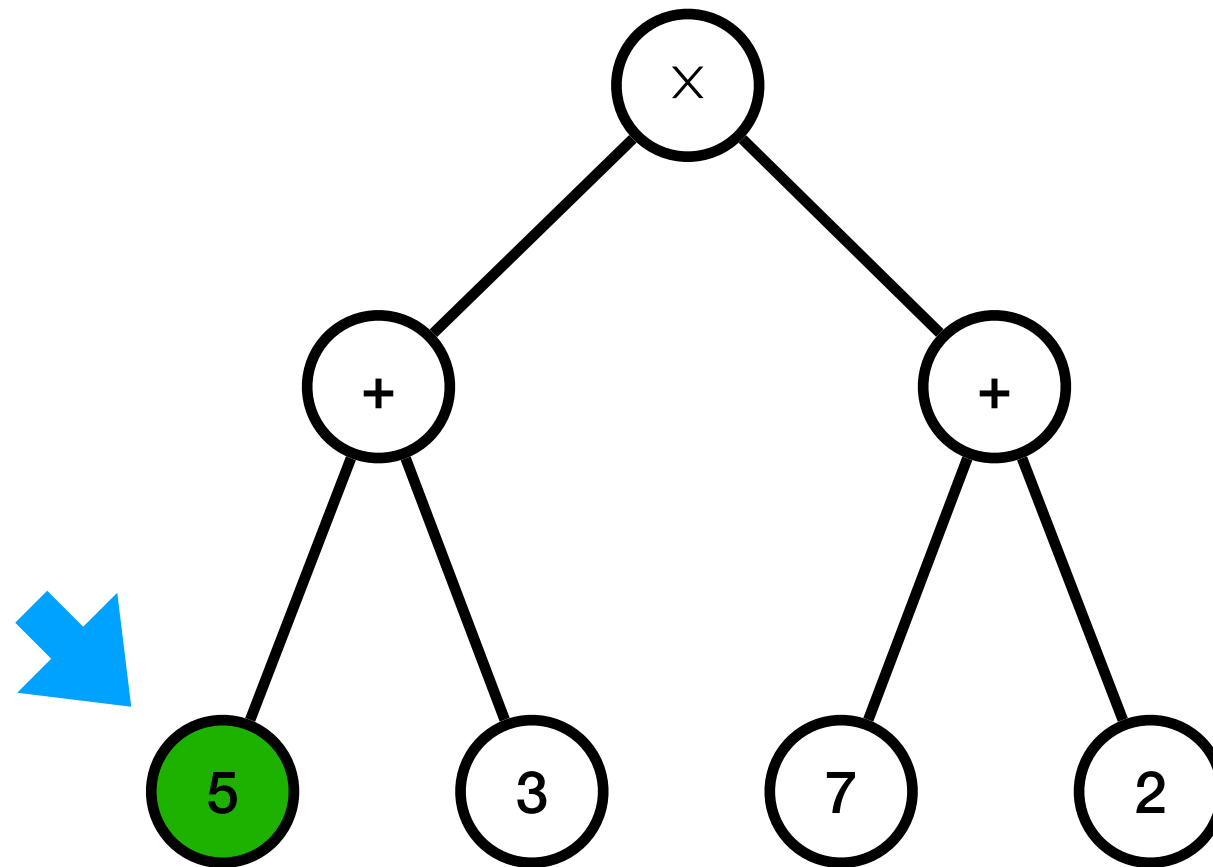
((

Afficher $((5 + 3) \times (7 + 2))$



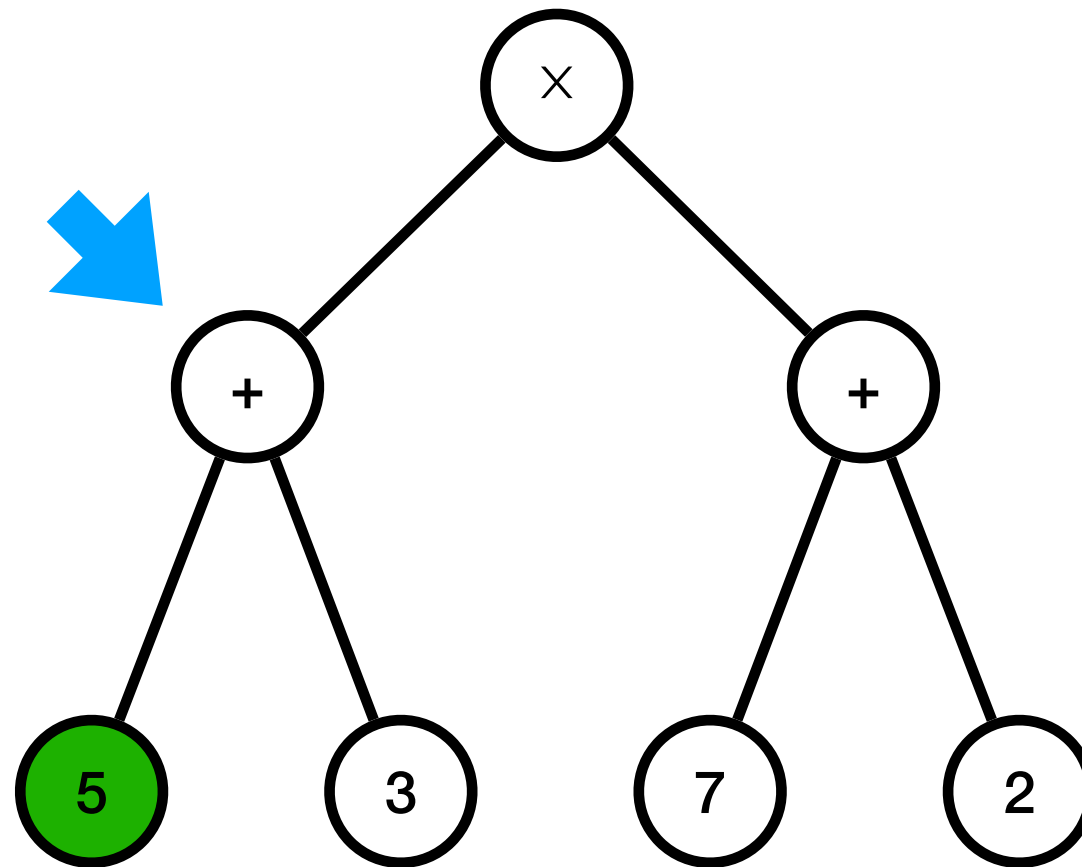
((

Afficher $((5 + 3) \times (7 + 2))$



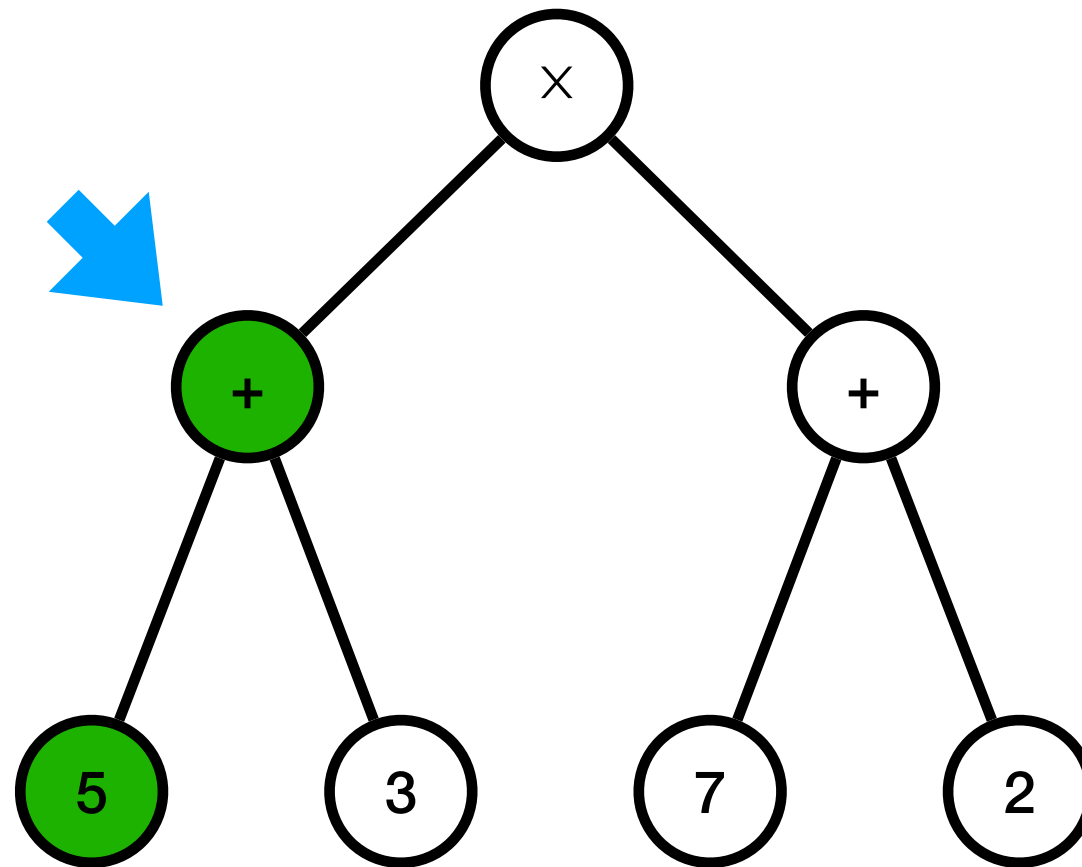
$((5$

Afficher $((5 + 3) \times (7 + 2))$



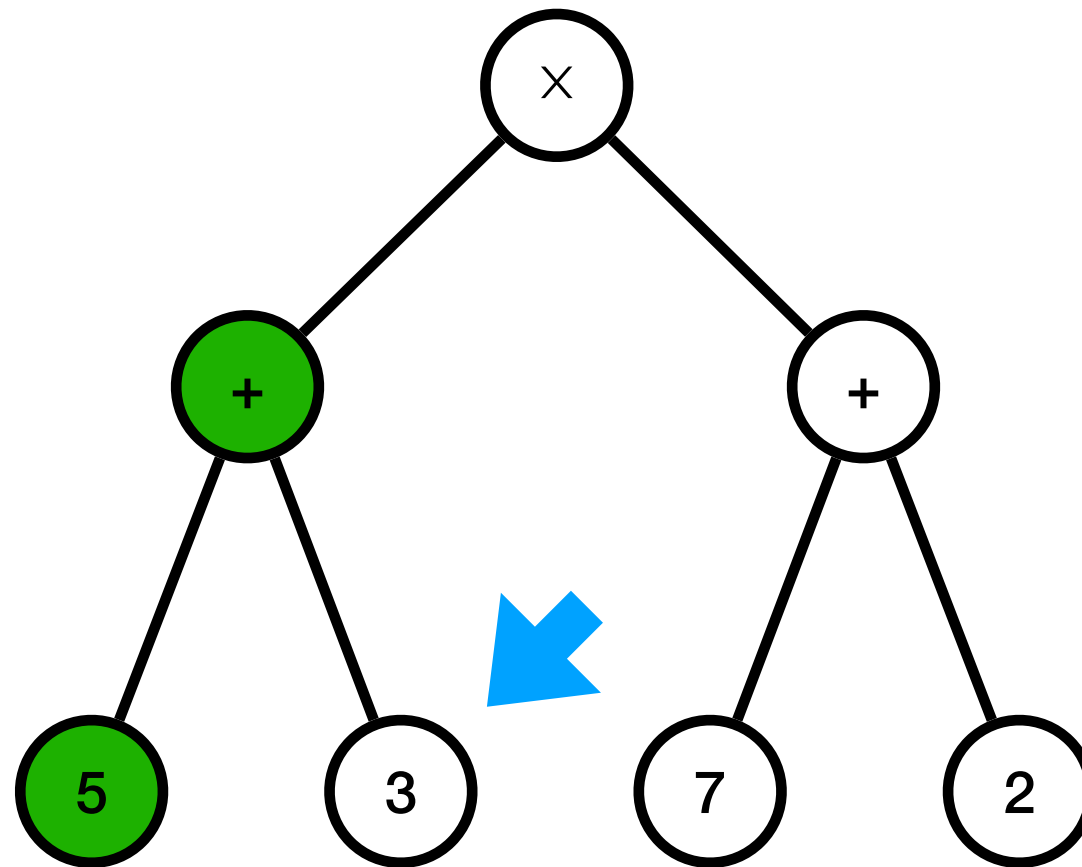
$((5$

Afficher $((5 + 3) \times (7 + 2))$



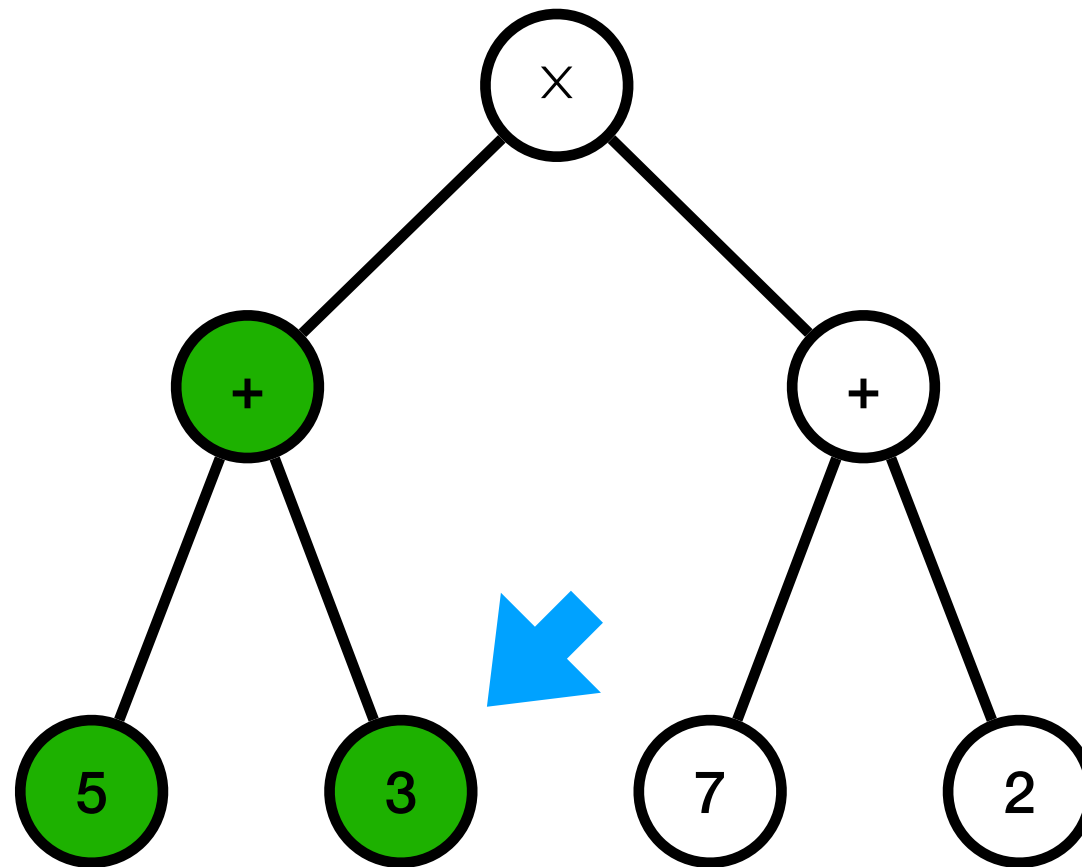
$((5 +$

Afficher $((5 + 3) \times (7 + 2))$



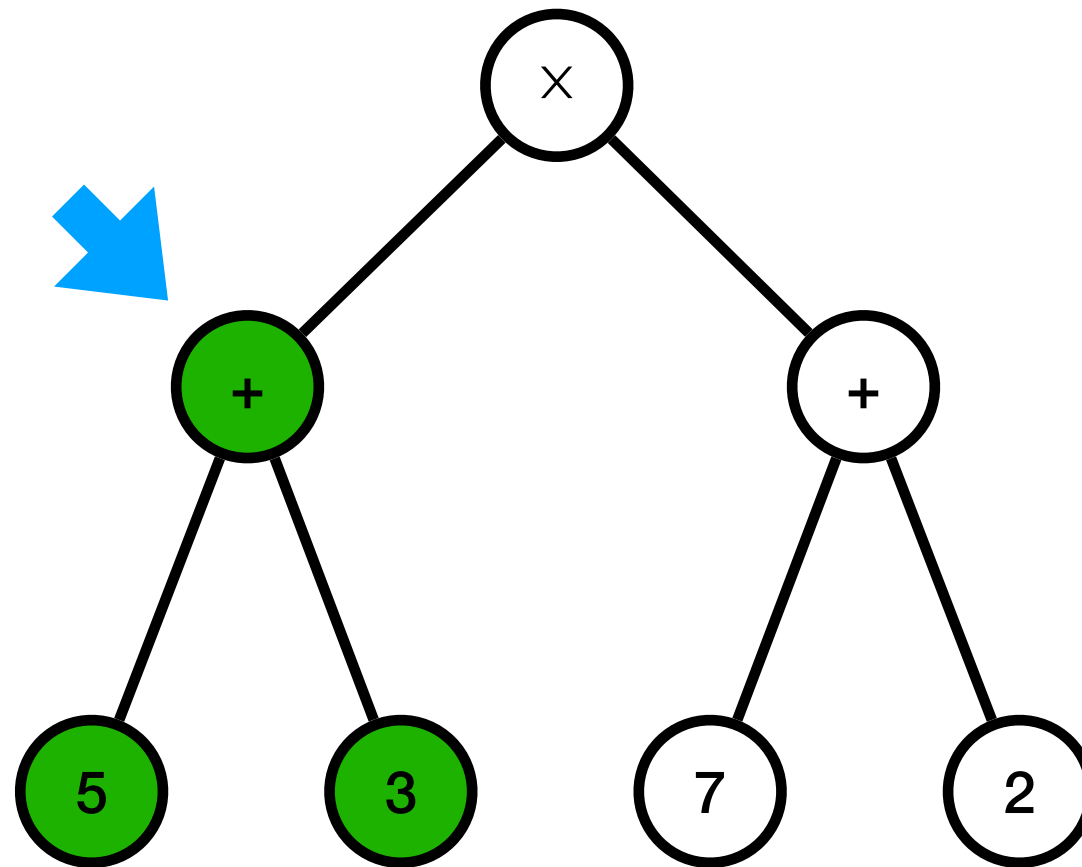
$((5 +$

Afficher $((5 + 3) \times (7 + 2))$



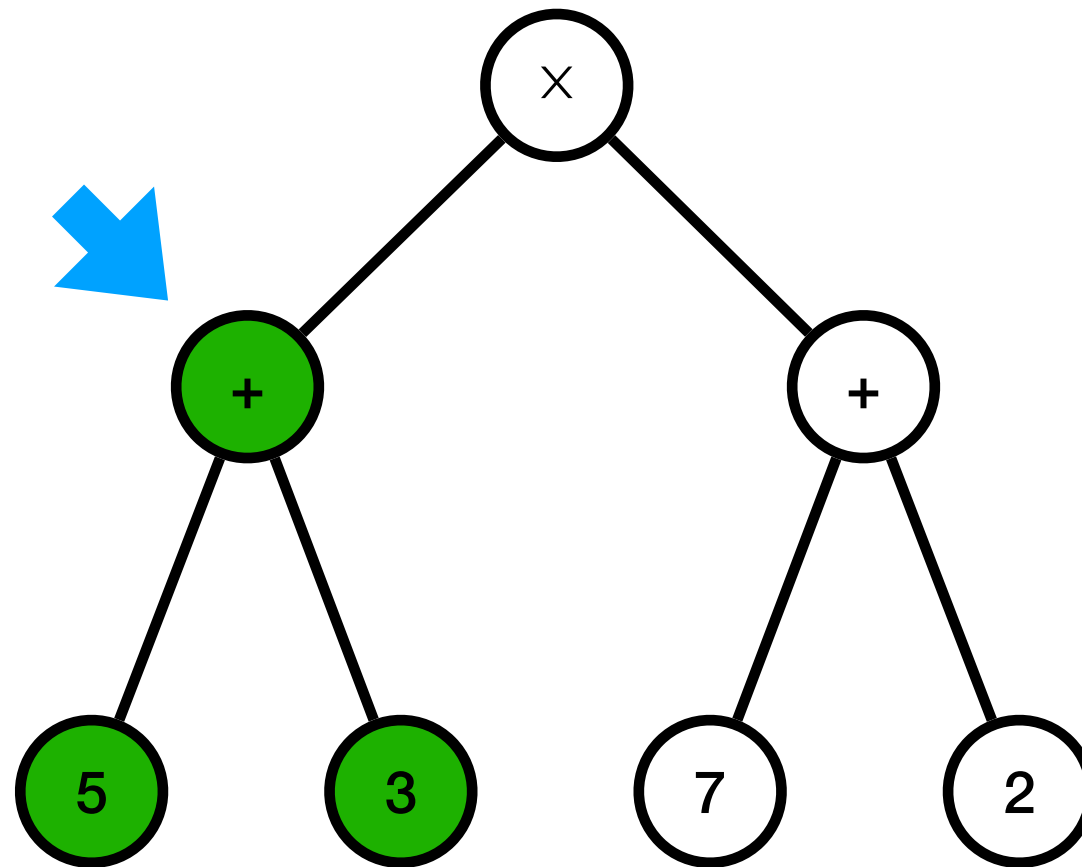
$((5 + 3$

Afficher $((5 + 3) \times (7 + 2))$



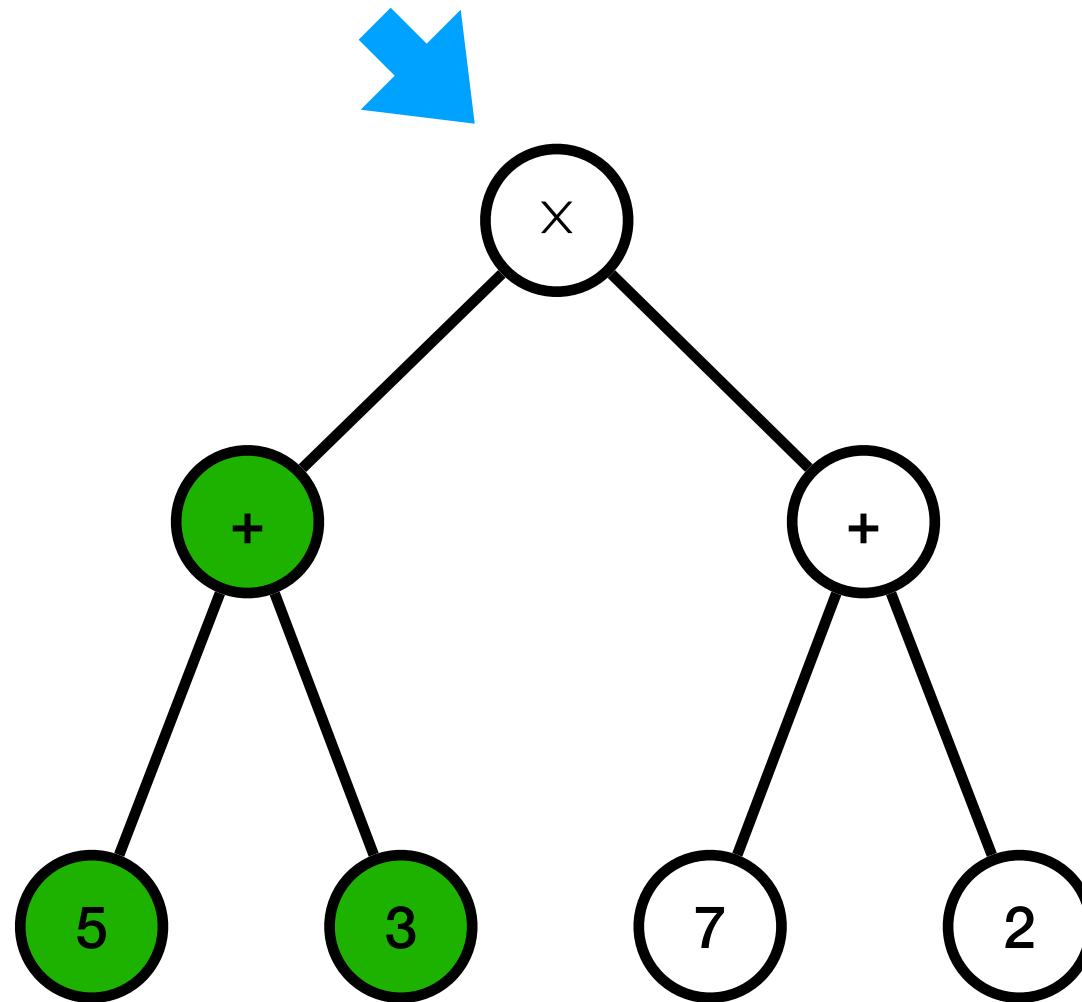
$((5 + 3$

Afficher $((5 + 3) \times (7 + 2))$



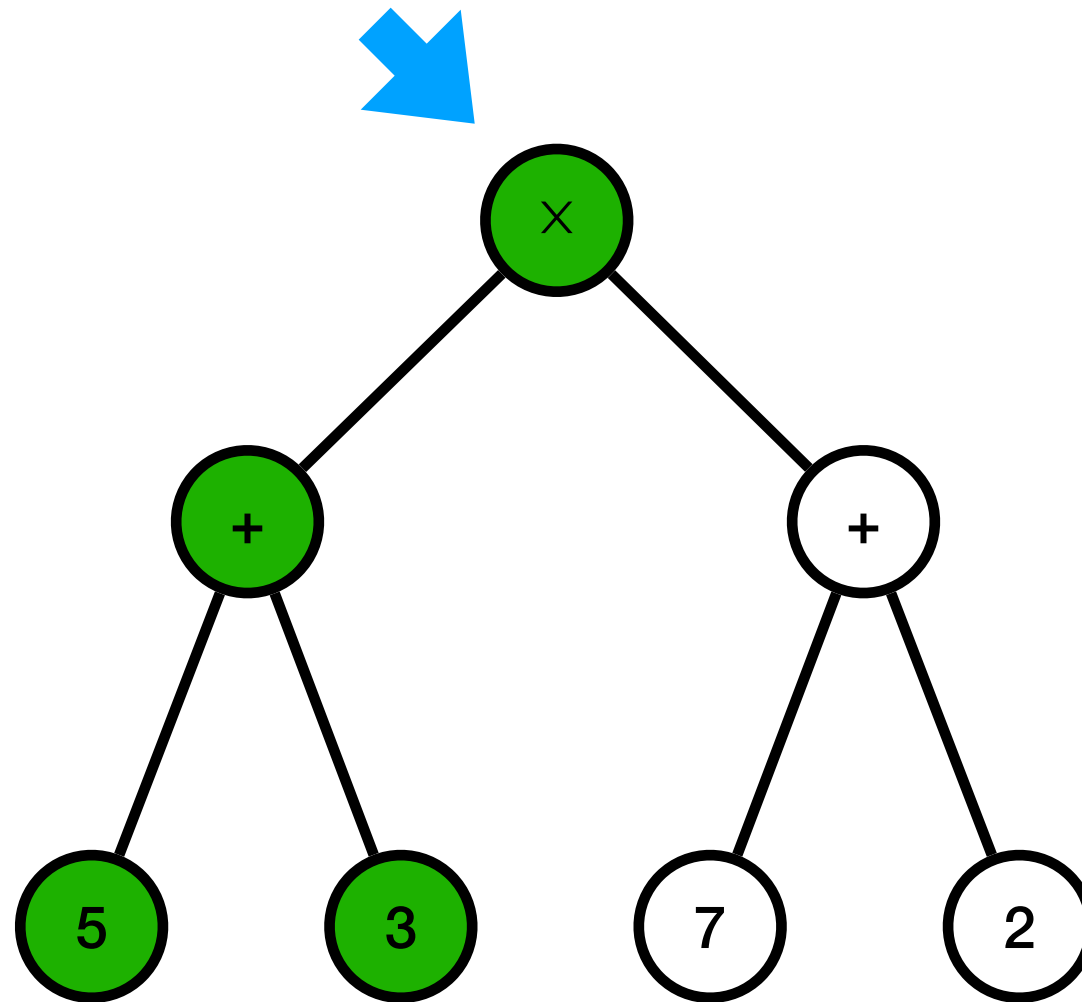
$((5 + 3))$

Afficher $((5 + 3) \times (7 + 2))$



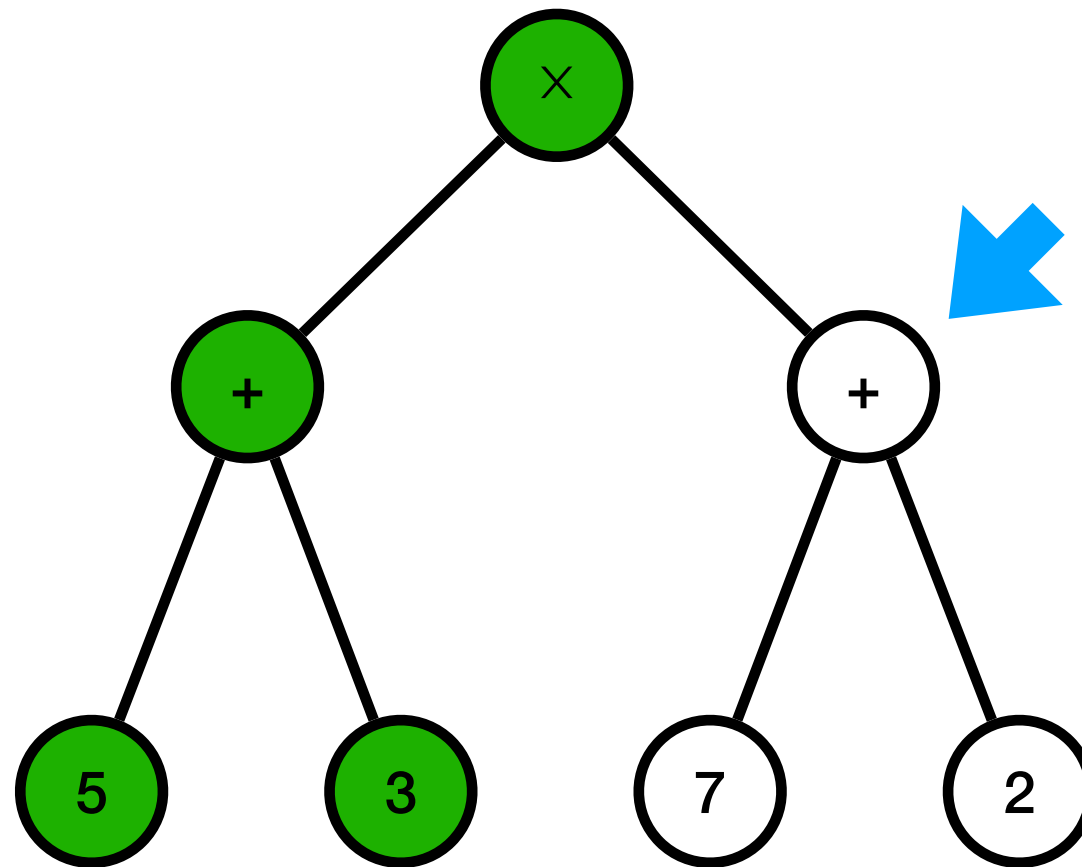
$((5 + 3))$

Afficher $((5 + 3) \times (7 + 2))$



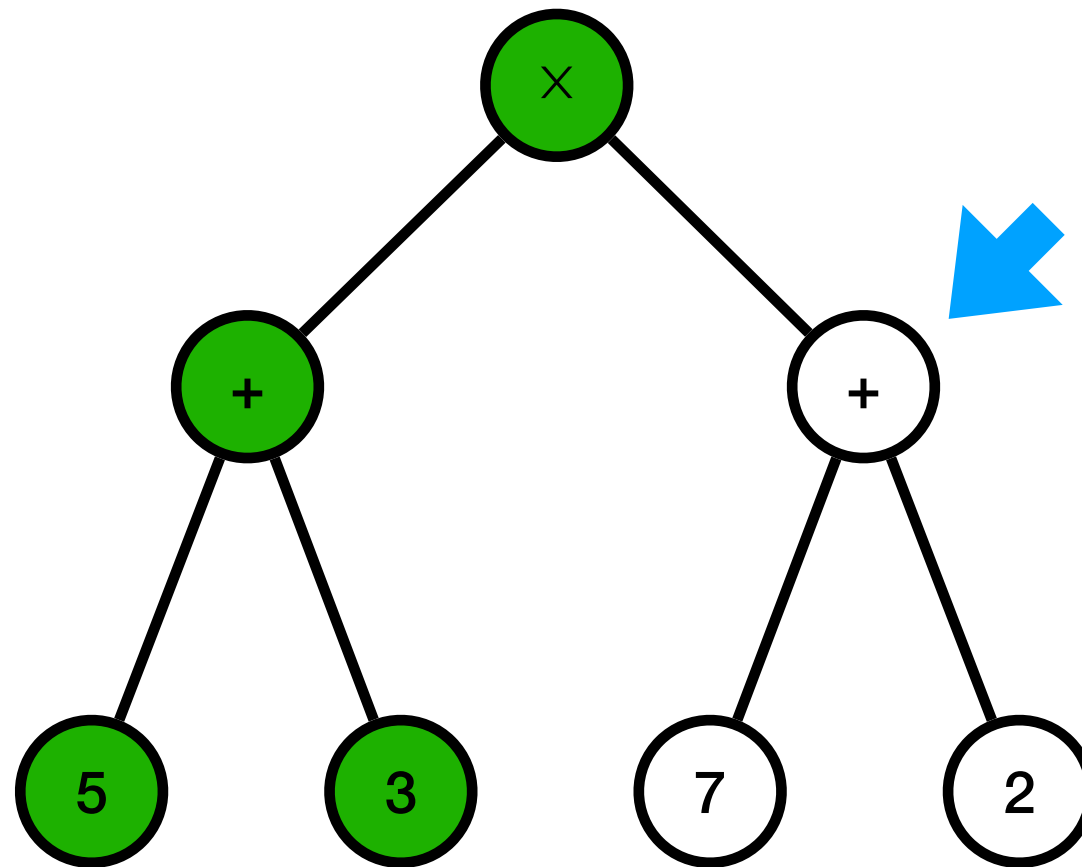
$((5 + 3) \times$

Afficher $((5 + 3) \times (7 + 2))$



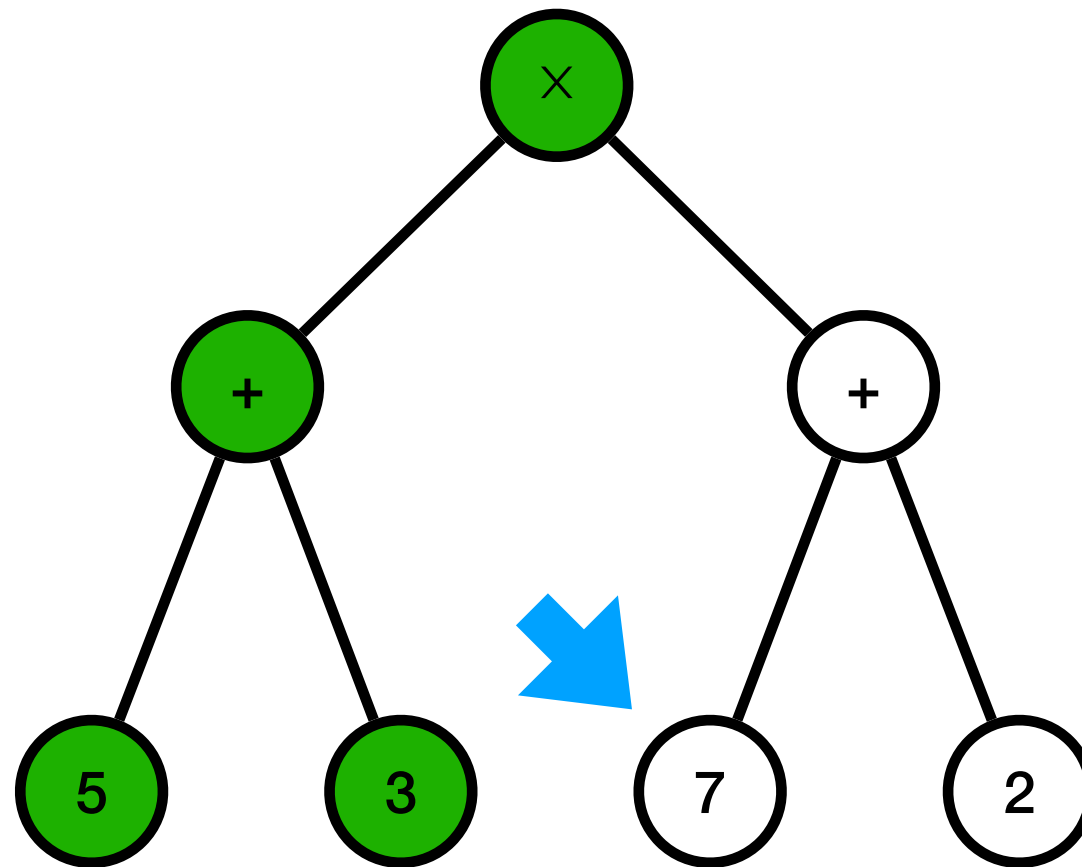
$((5 + 3) \times$

Afficher $((5 + 3) \times (7 + 2))$



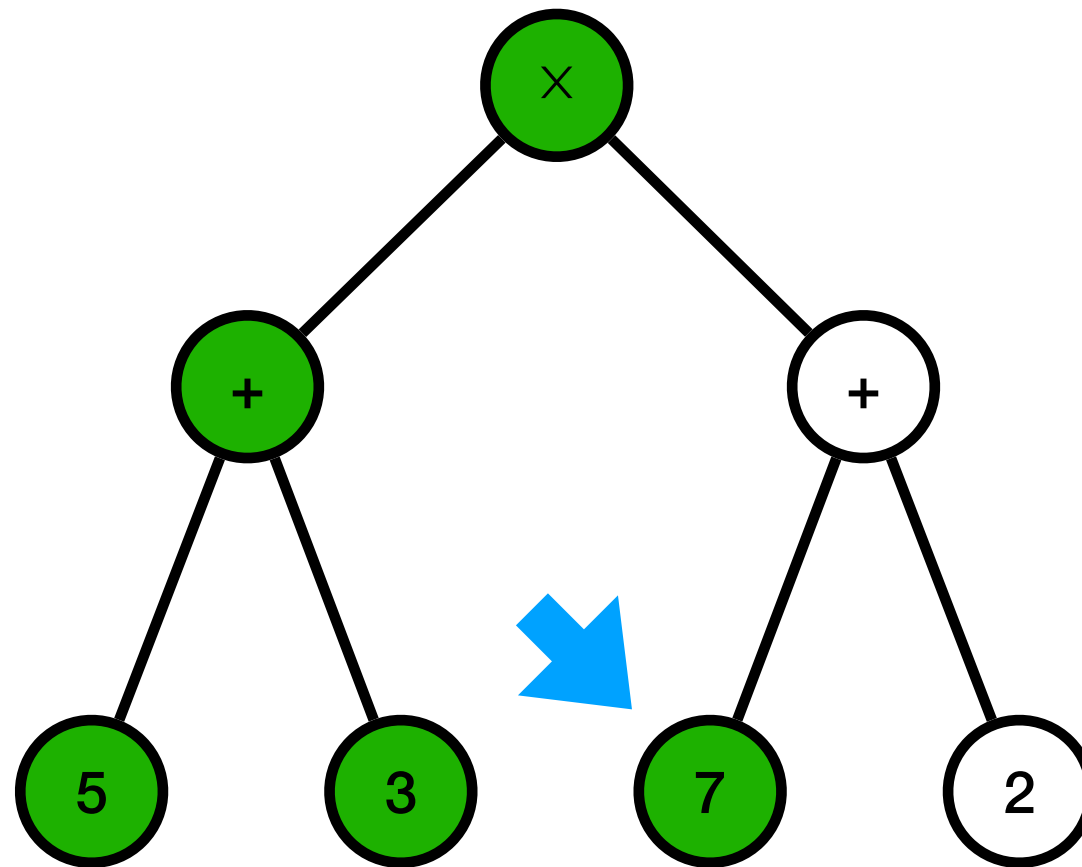
$((5 + 3) \times ($

Afficher $((5 + 3) \times (7 + 2))$



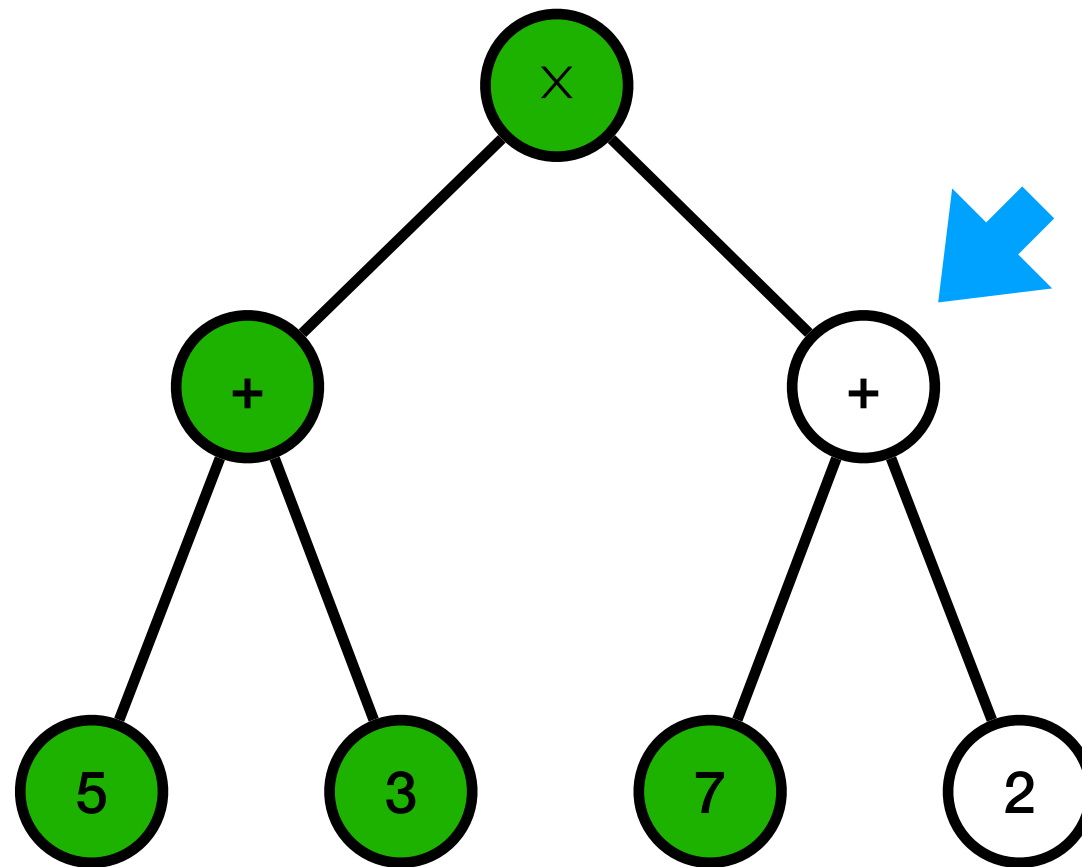
$((5 + 3) \times ($

Afficher $((5 + 3) \times (7 + 2))$



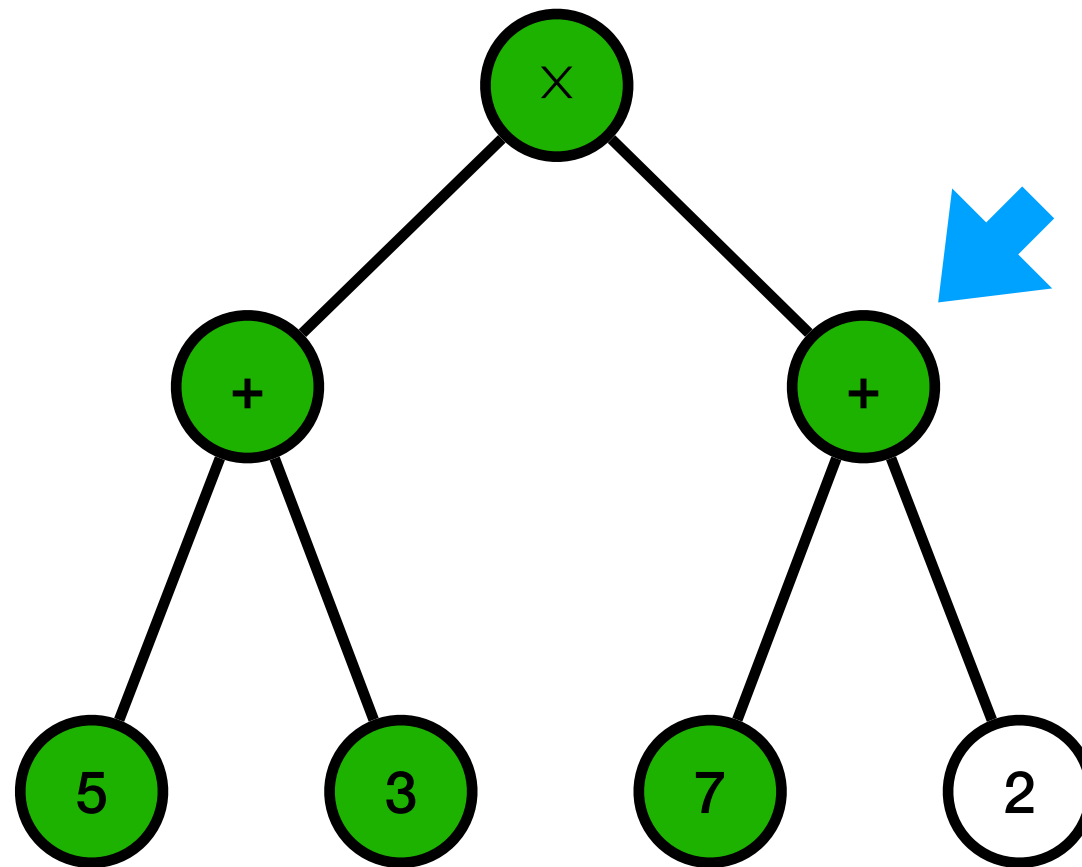
$((5 + 3) \times (7$

Afficher $((5 + 3) \times (7 + 2))$



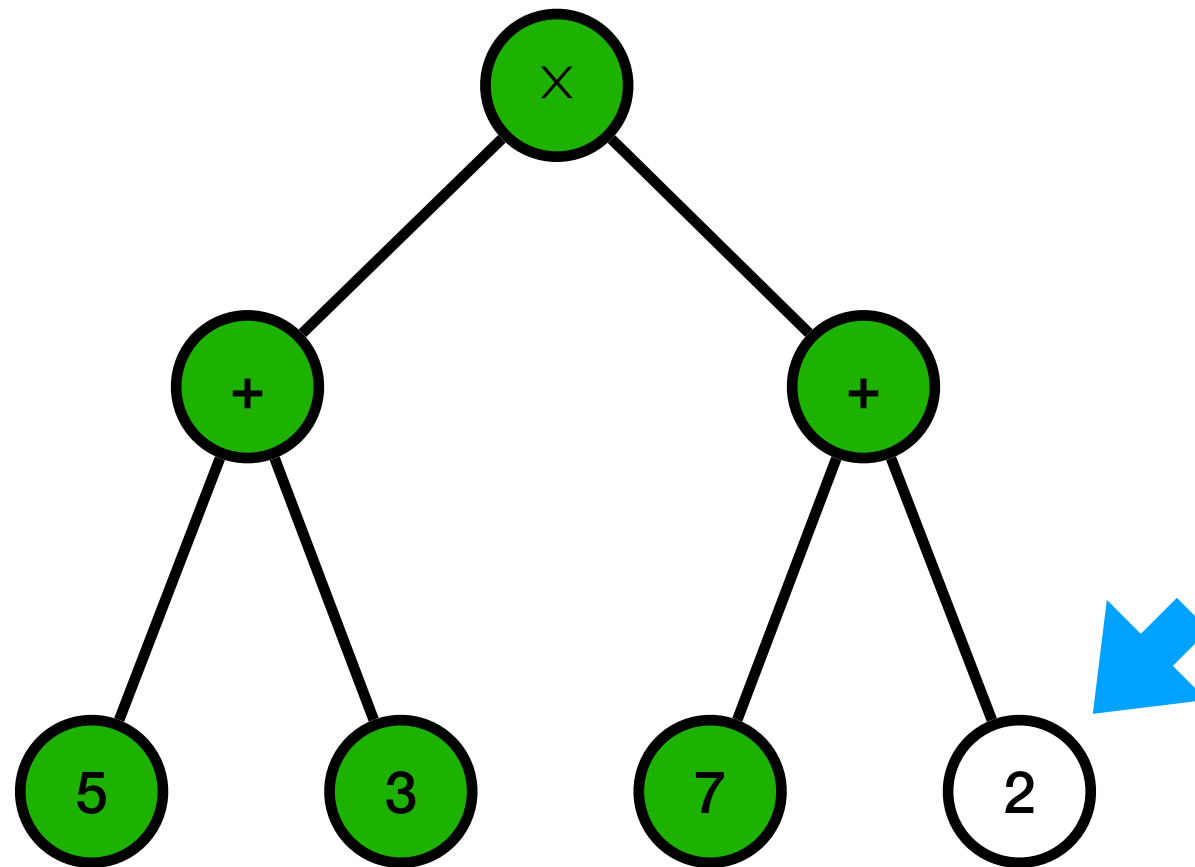
$((5 + 3) \times (7$

Afficher $((5 + 3) \times (7 + 2))$



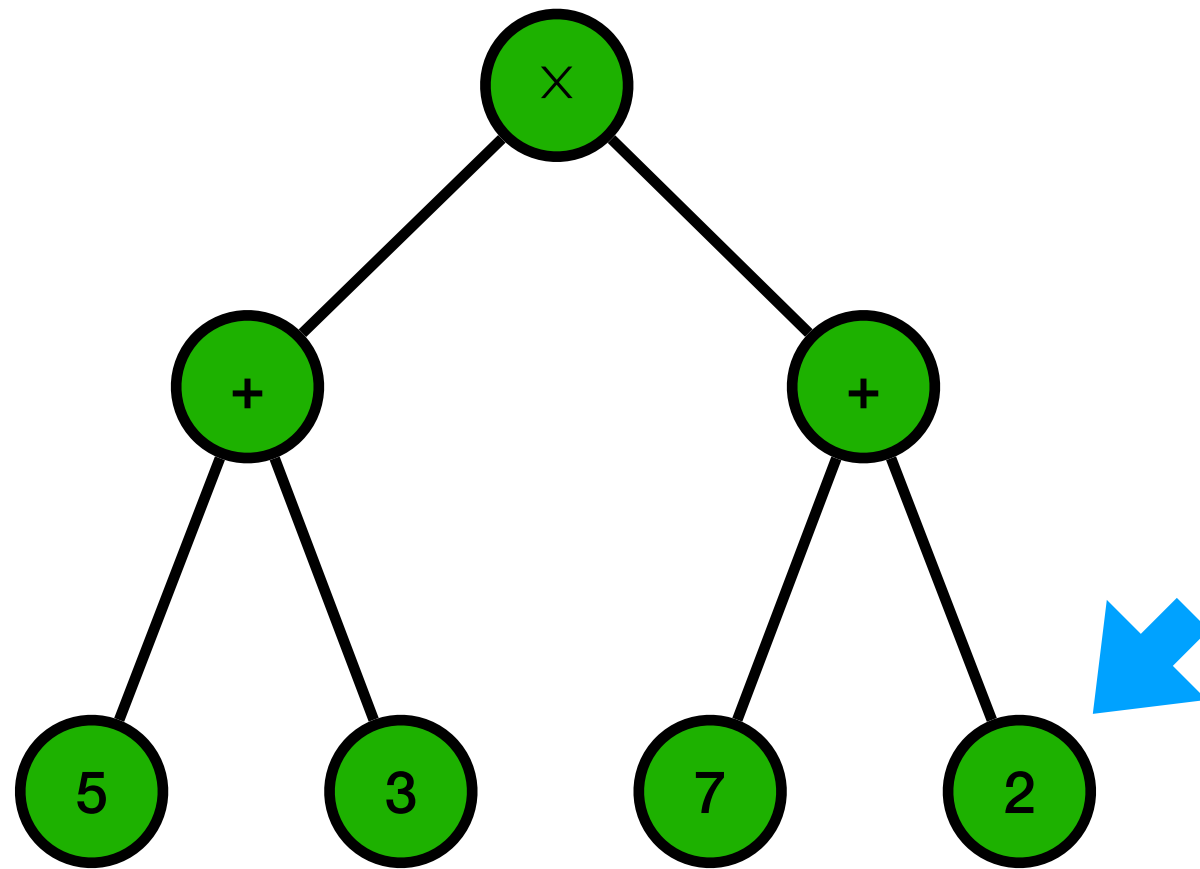
$((5 + 3) \times (7 +$

Afficher $((5 + 3) \times (7 + 2))$



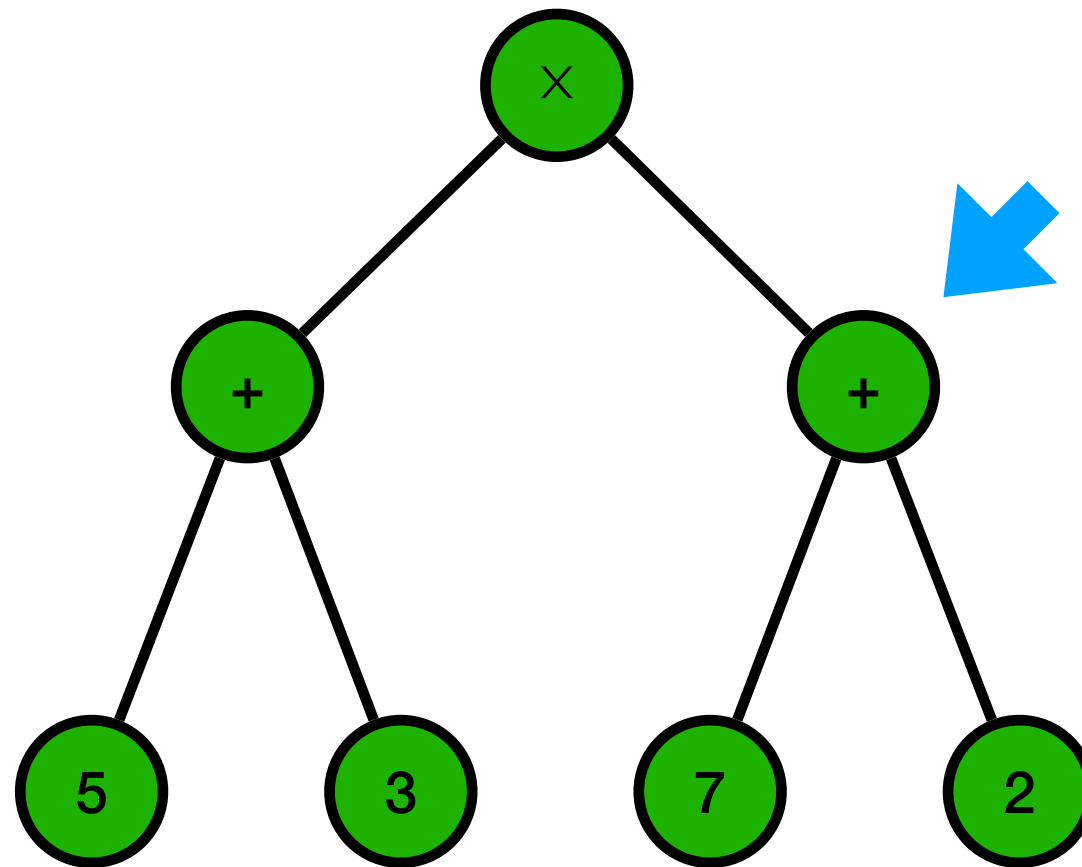
$((5 + 3) \times (7 +$

Afficher $((5 + 3) \times (7 + 2))$



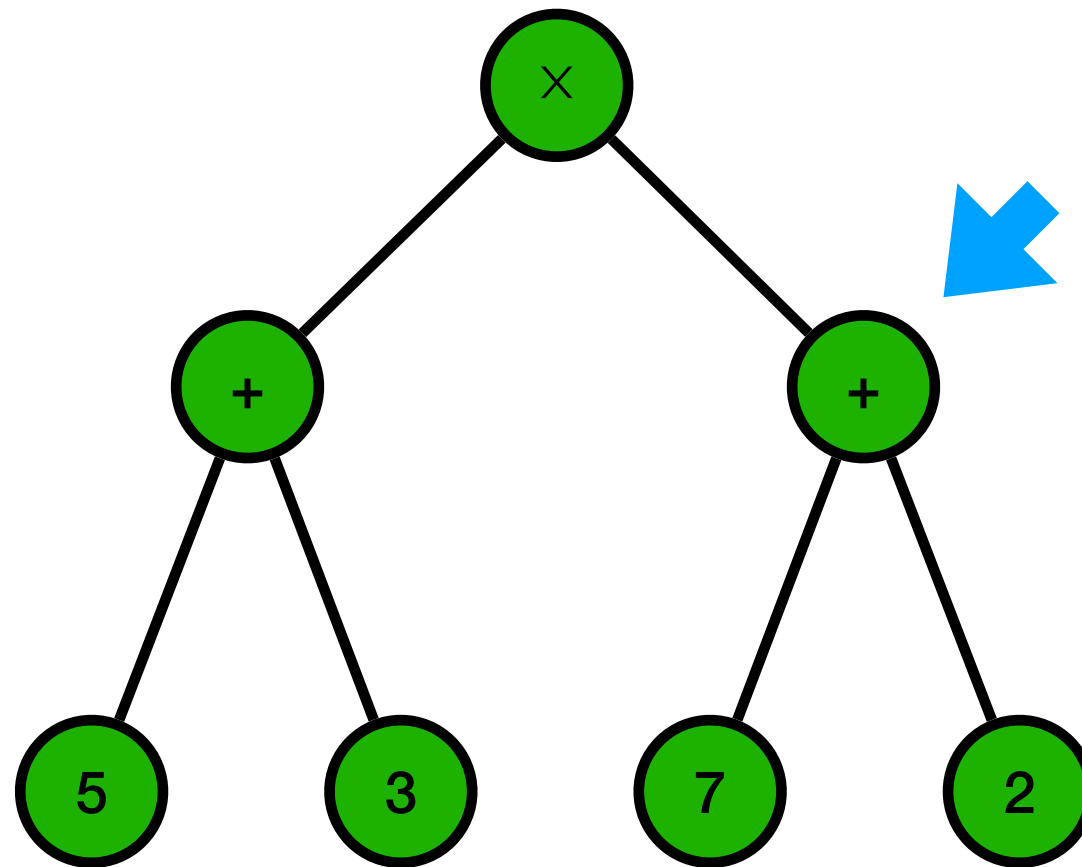
$((5 + 3) \times (7 + 2$

Afficher $((5 + 3) \times (7 + 2))$



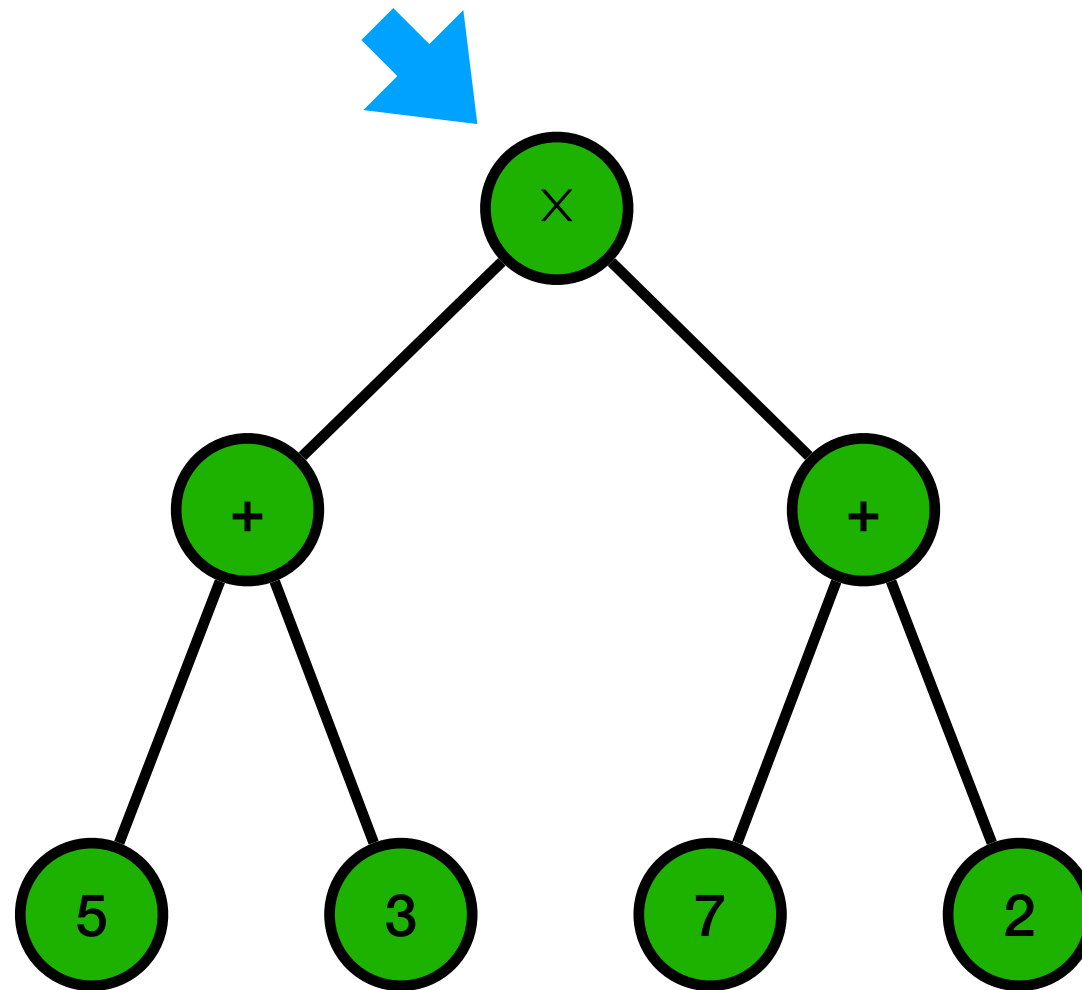
$((5 + 3) \times (7 + 2$

Afficher $((5 + 3) \times (7 + 2))$



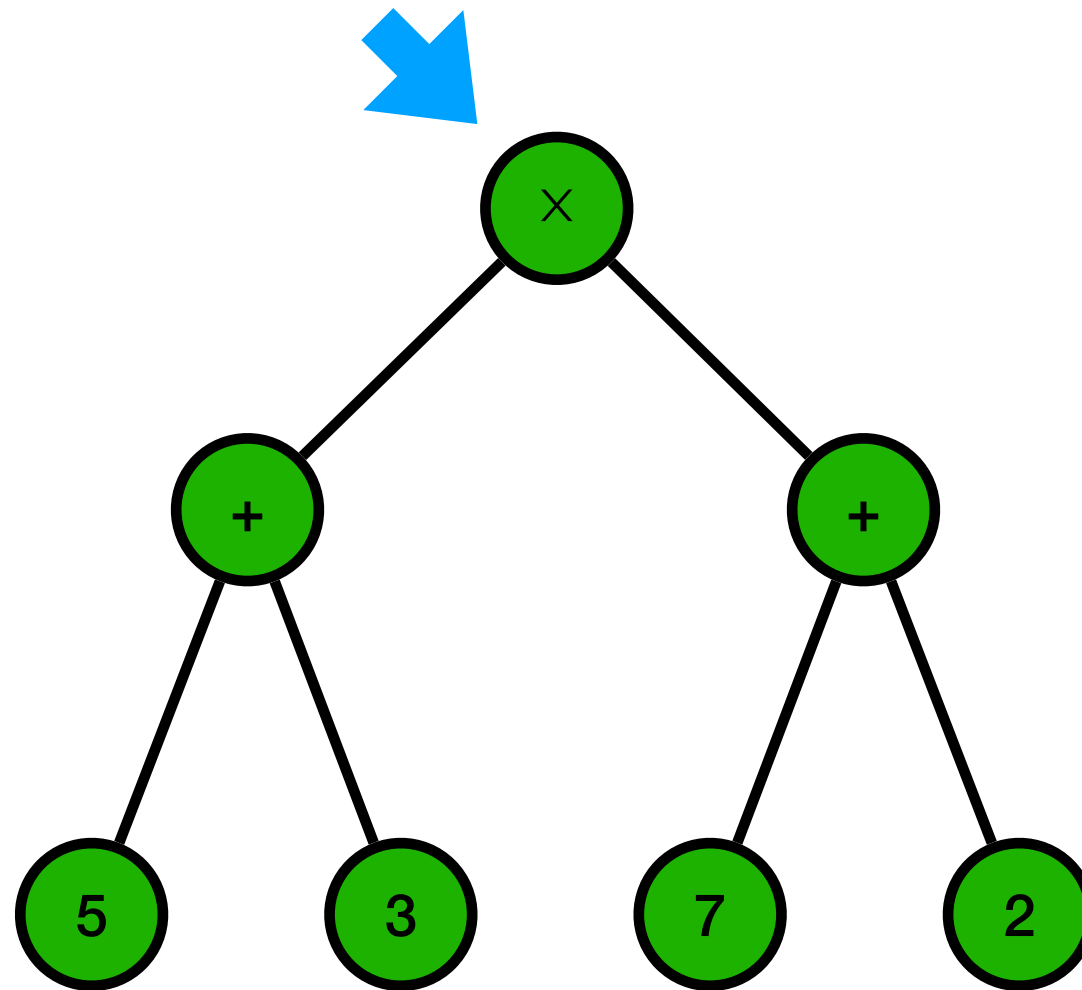
$((5 + 3) \times (7 + 2))$

Afficher $((5 + 3) \times (7 + 2))$



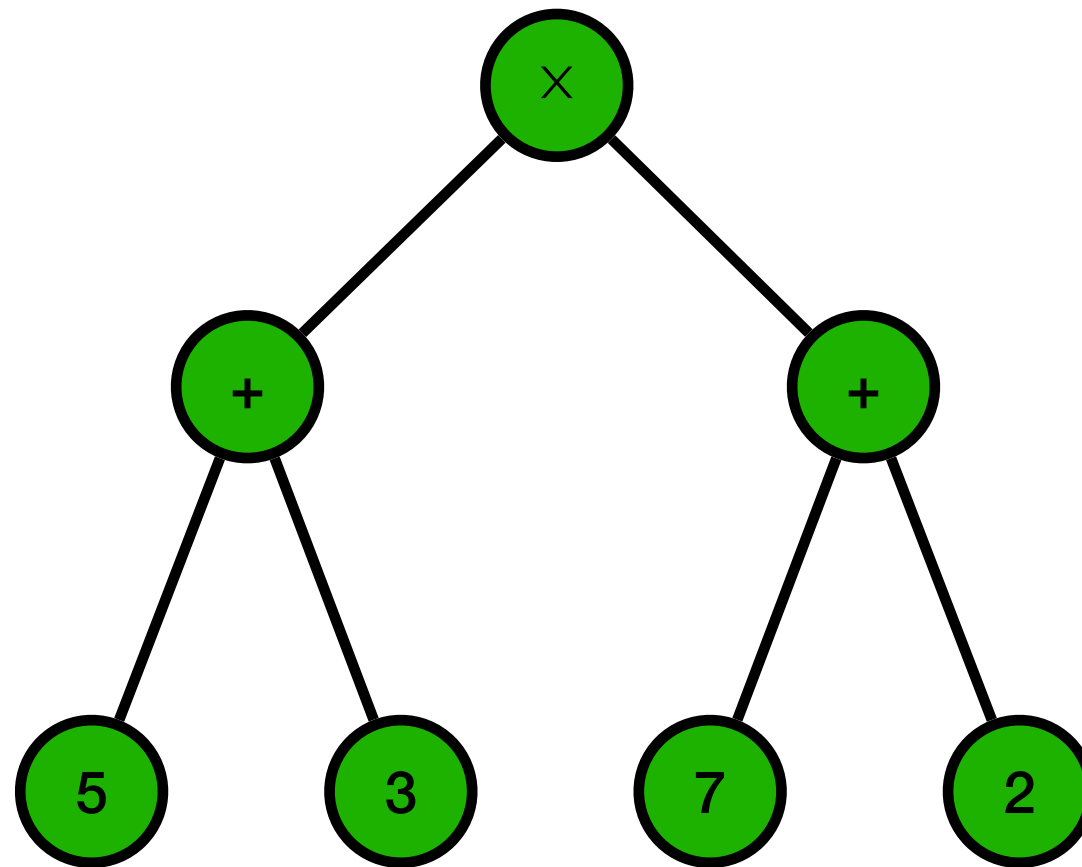
$((5 + 3) \times (7 + 2))$

Afficher $((5 + 3) \times (7 + 2))$



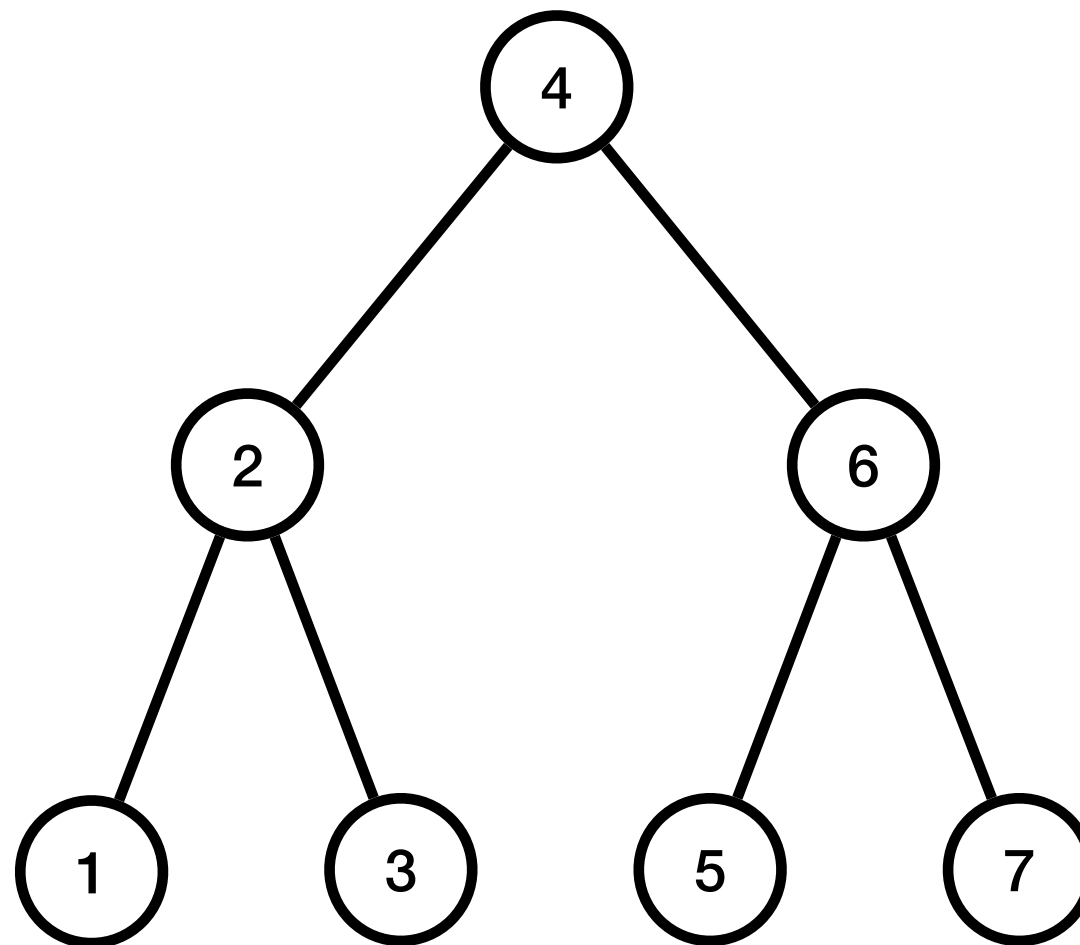
$((5 + 3) \times (7 + 2))$

Afficher $((5 + 3) \times (7 + 2))$

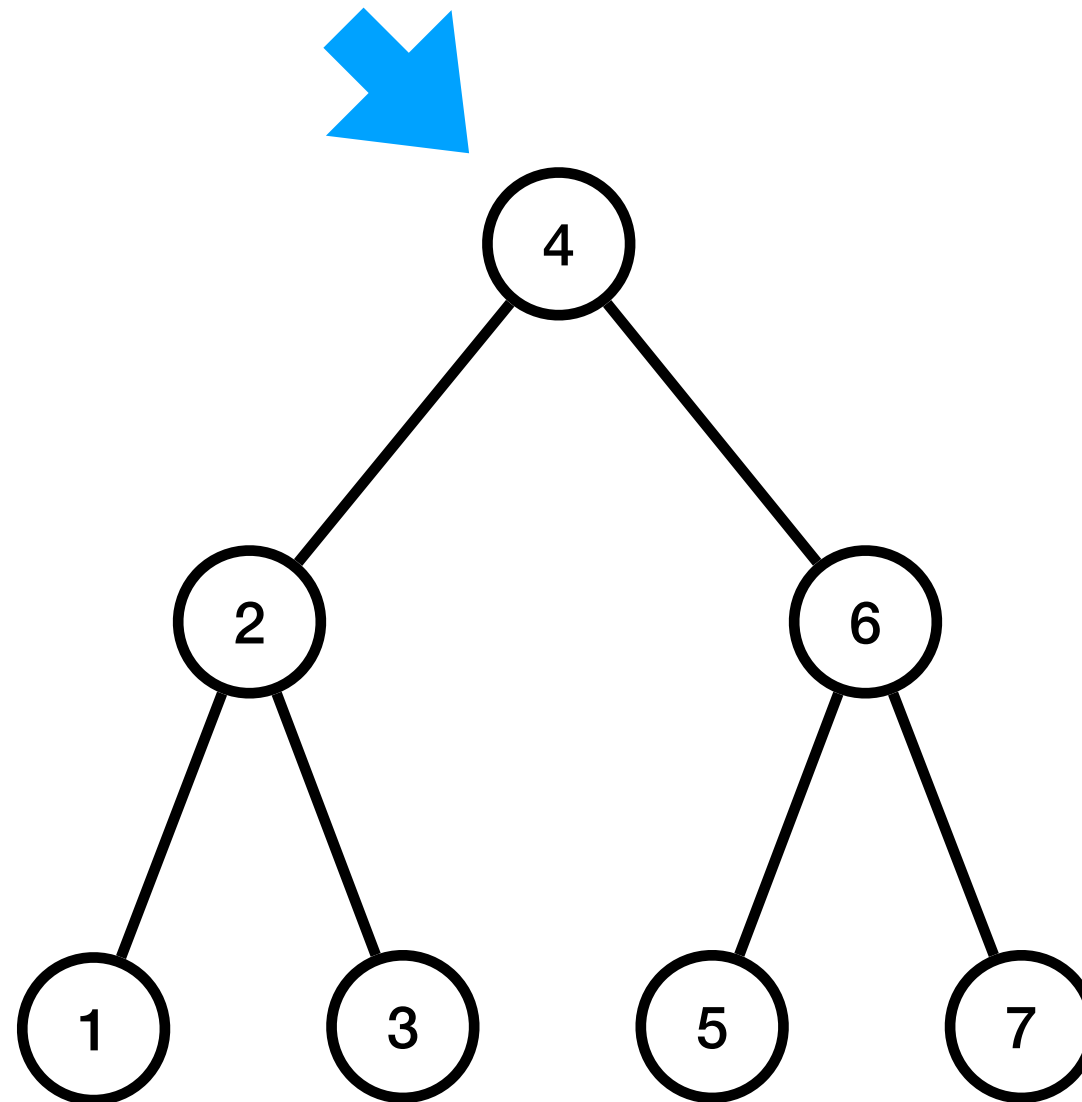


$((5 + 3) \times (7 + 2))$

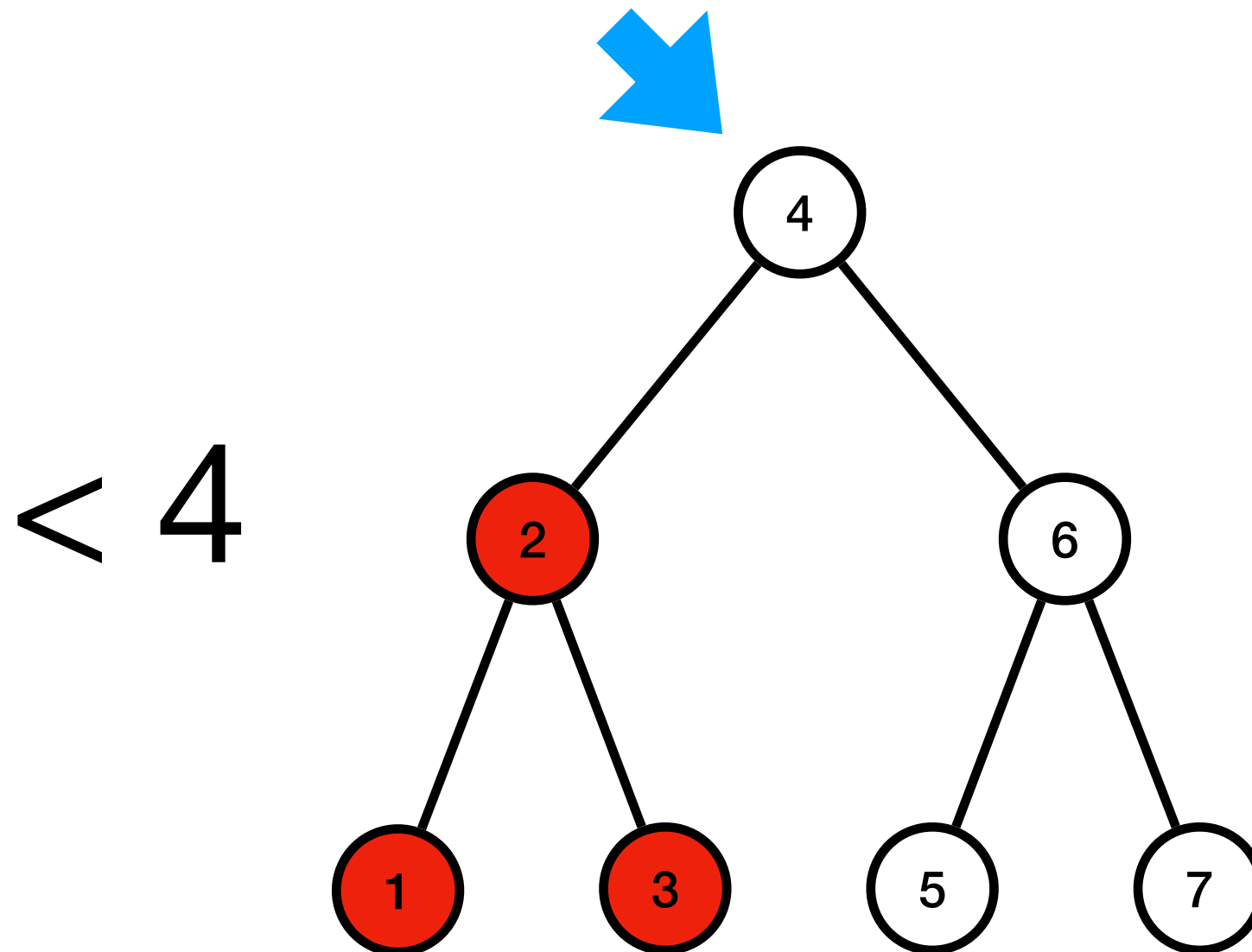
Arbres binaires de recherche



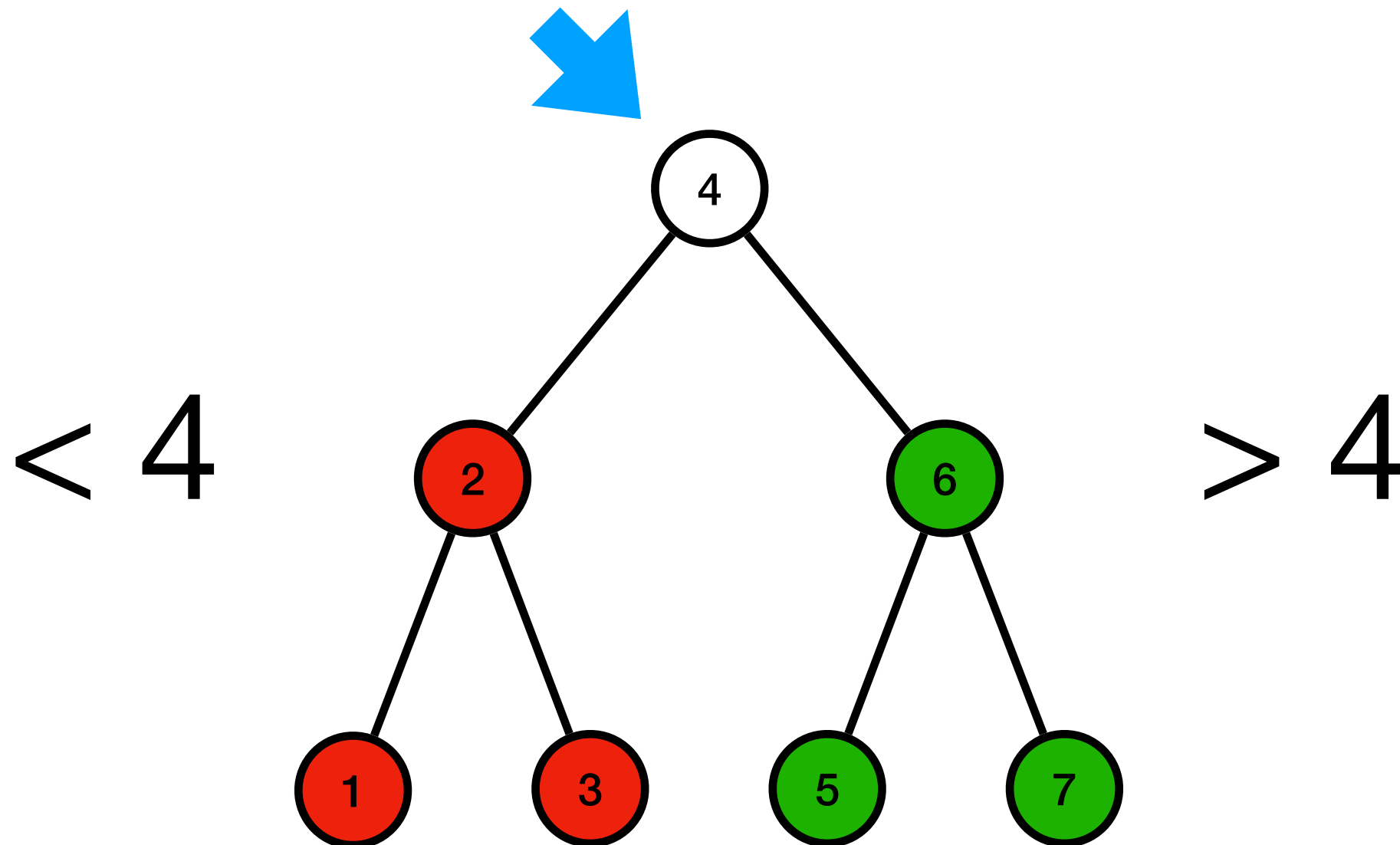
Arbres binaires de recherche



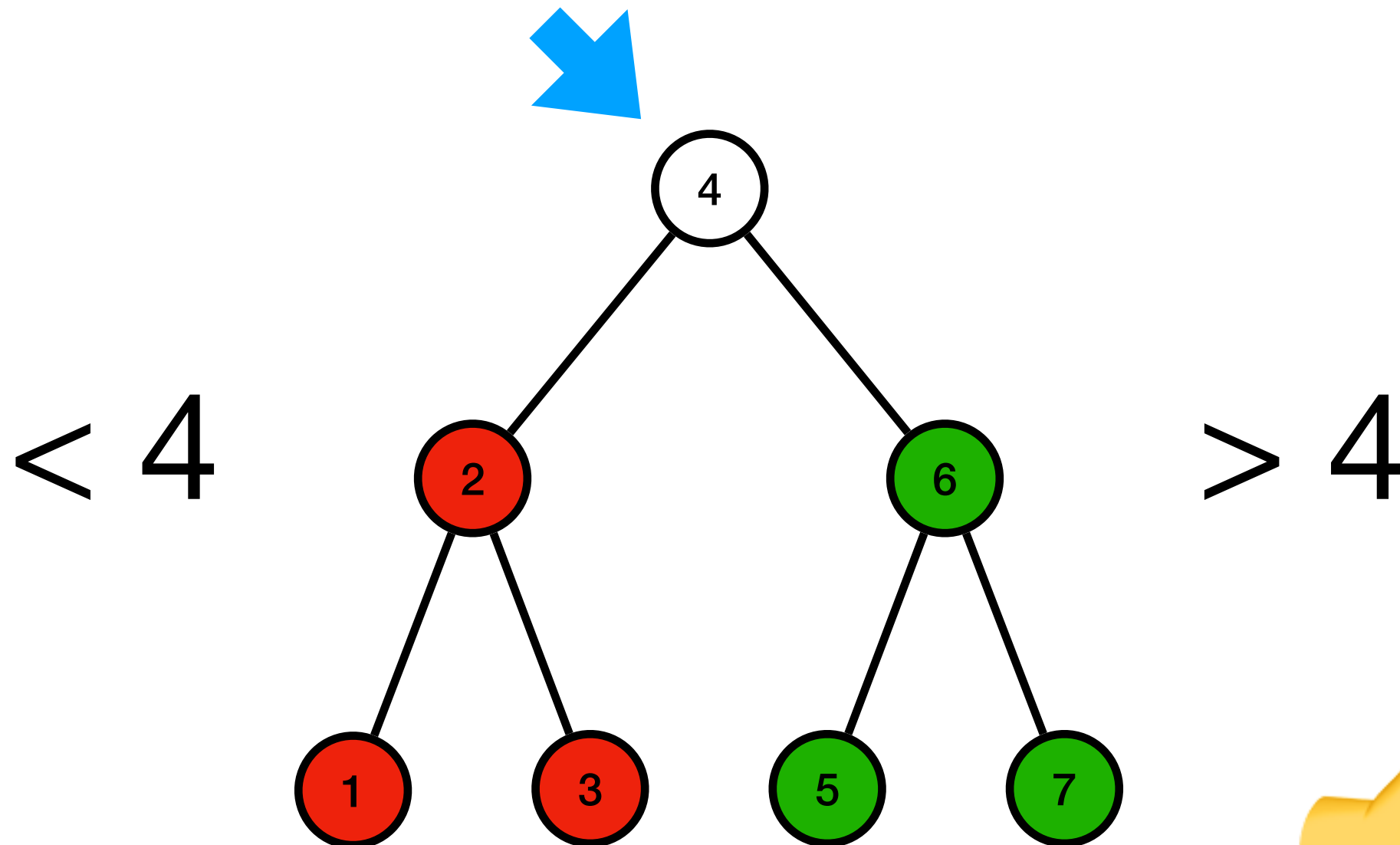
Arbres binaires de recherche



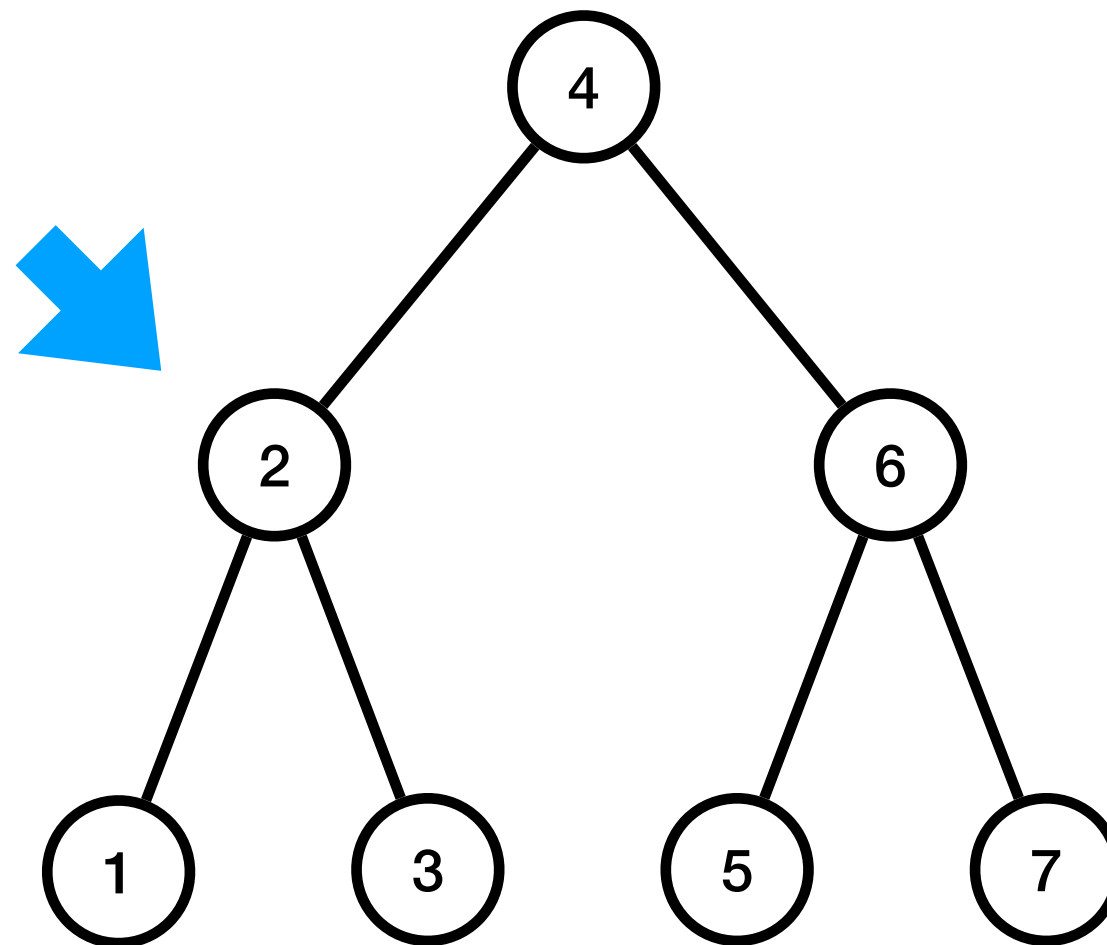
Arbres binaires de recherche



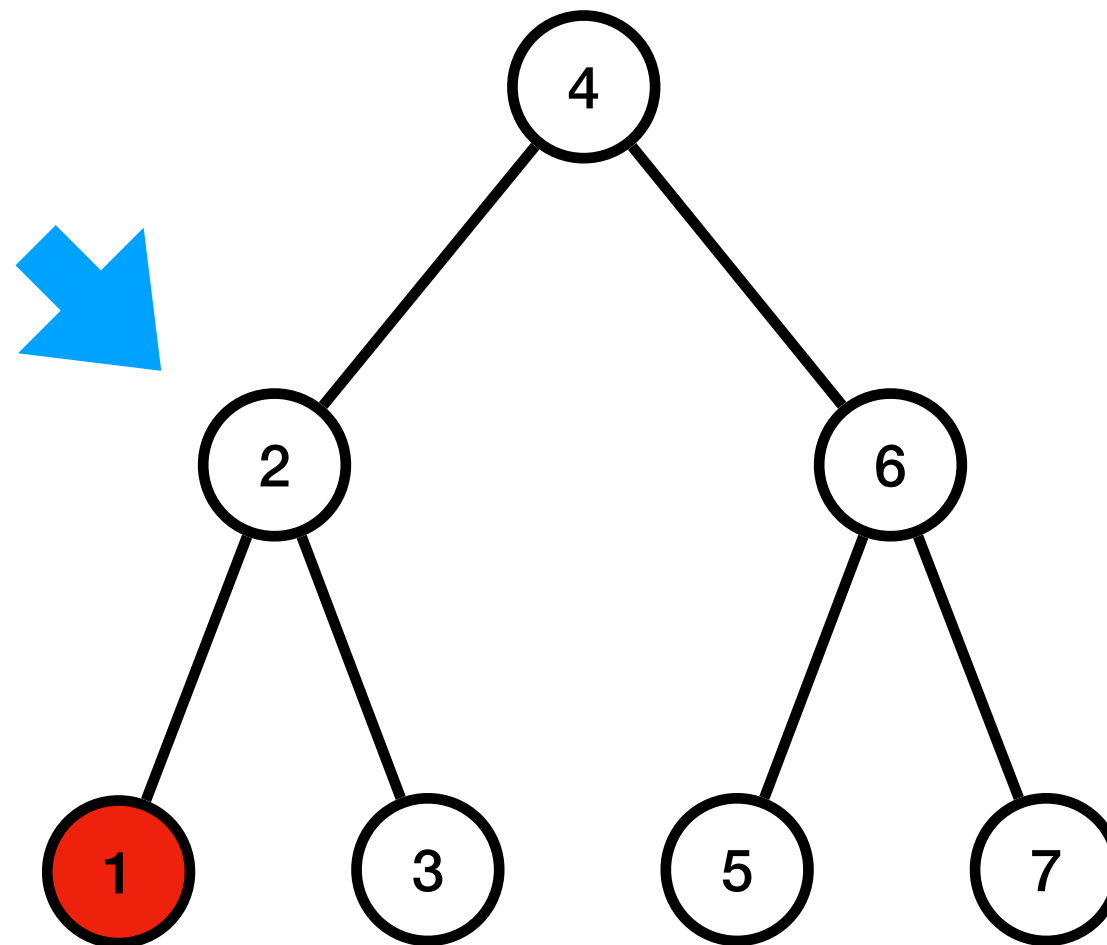
Arbres binaires de recherche



Arbres binaires de recherche

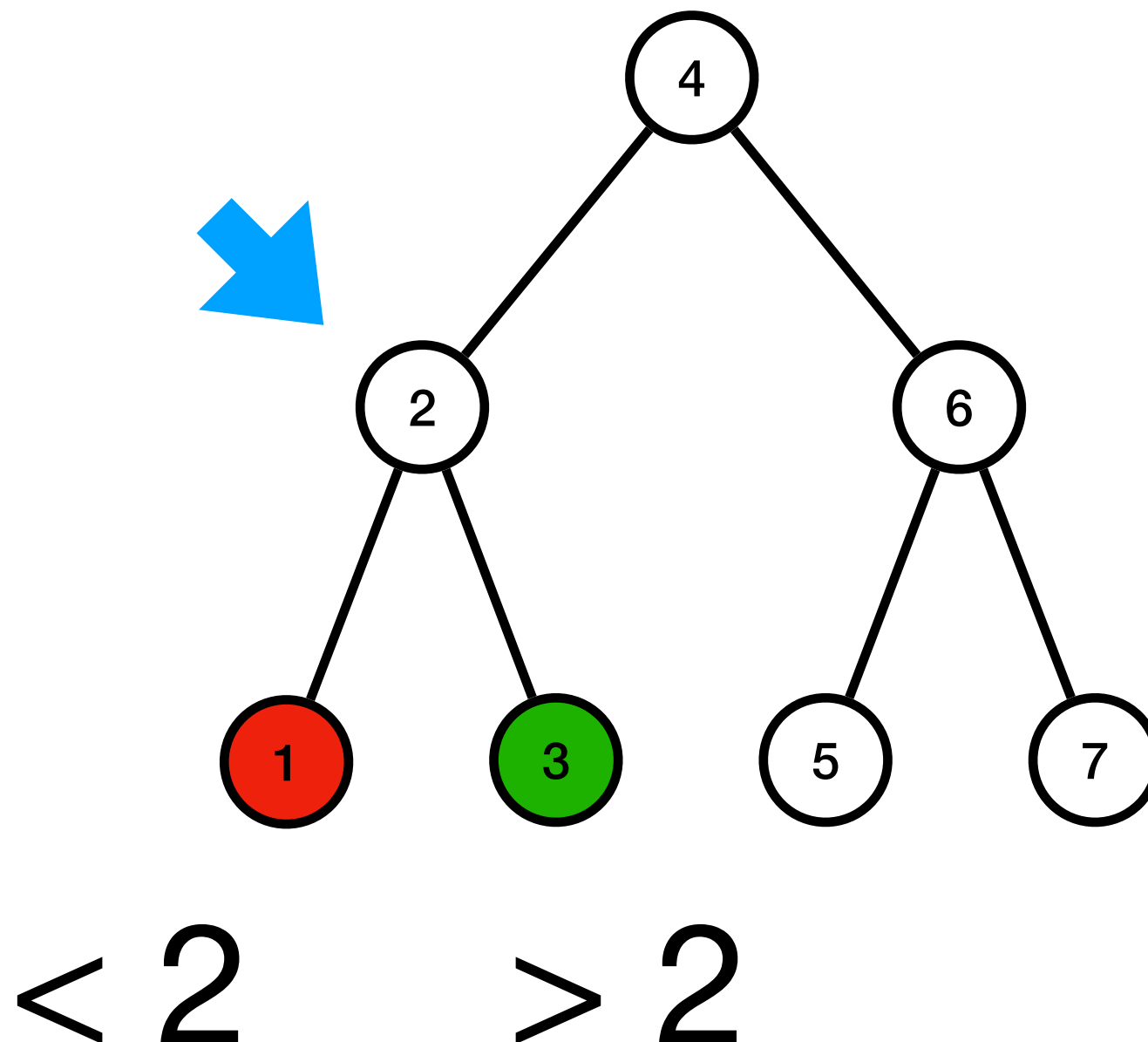


Arbres binaires de recherche

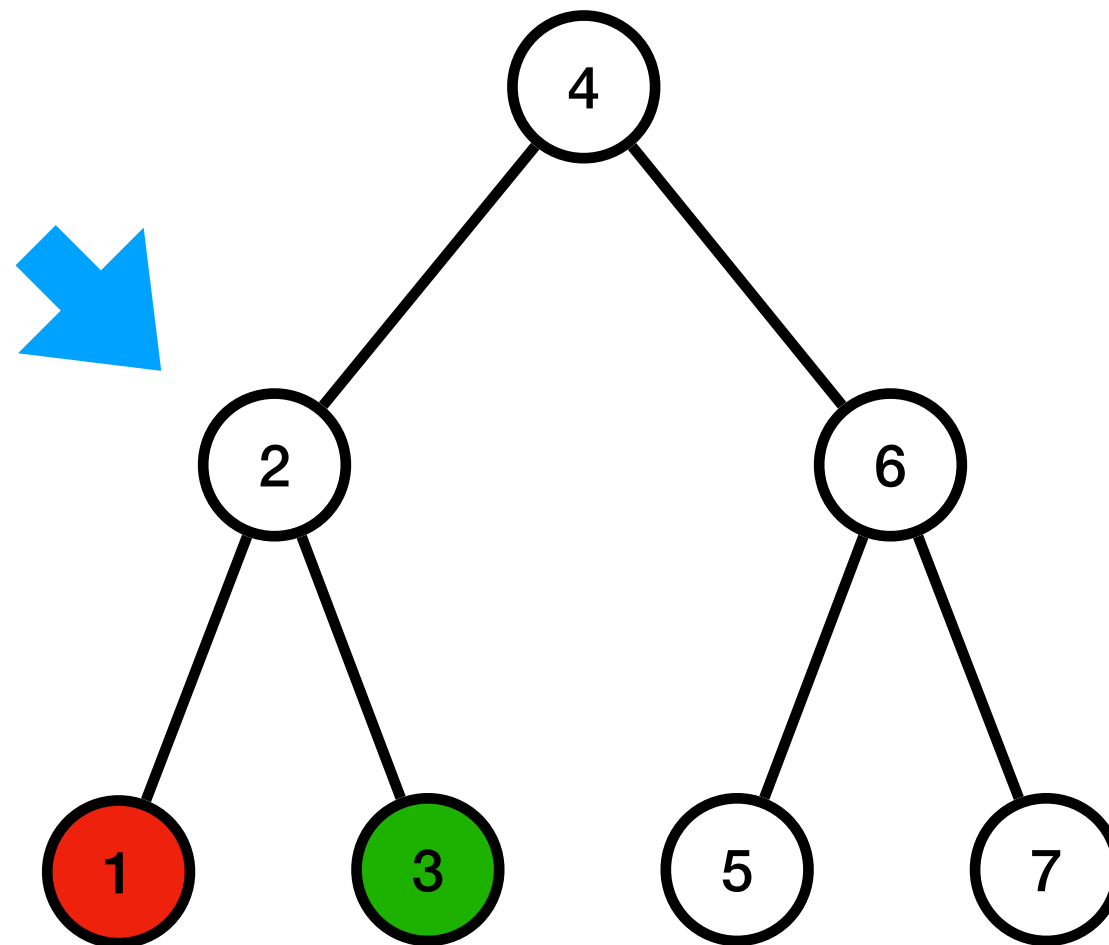


< 2

Arbres binaires de recherche



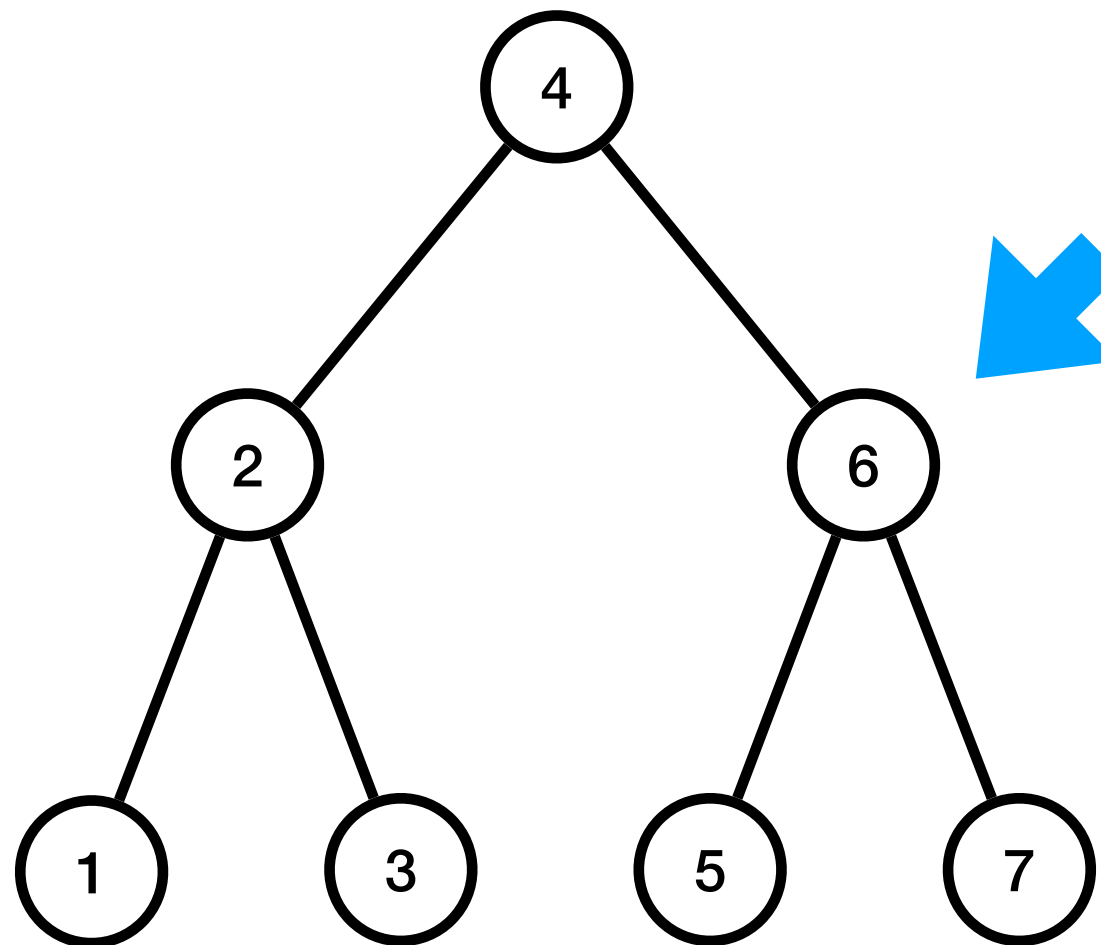
Arbres binaires de recherche



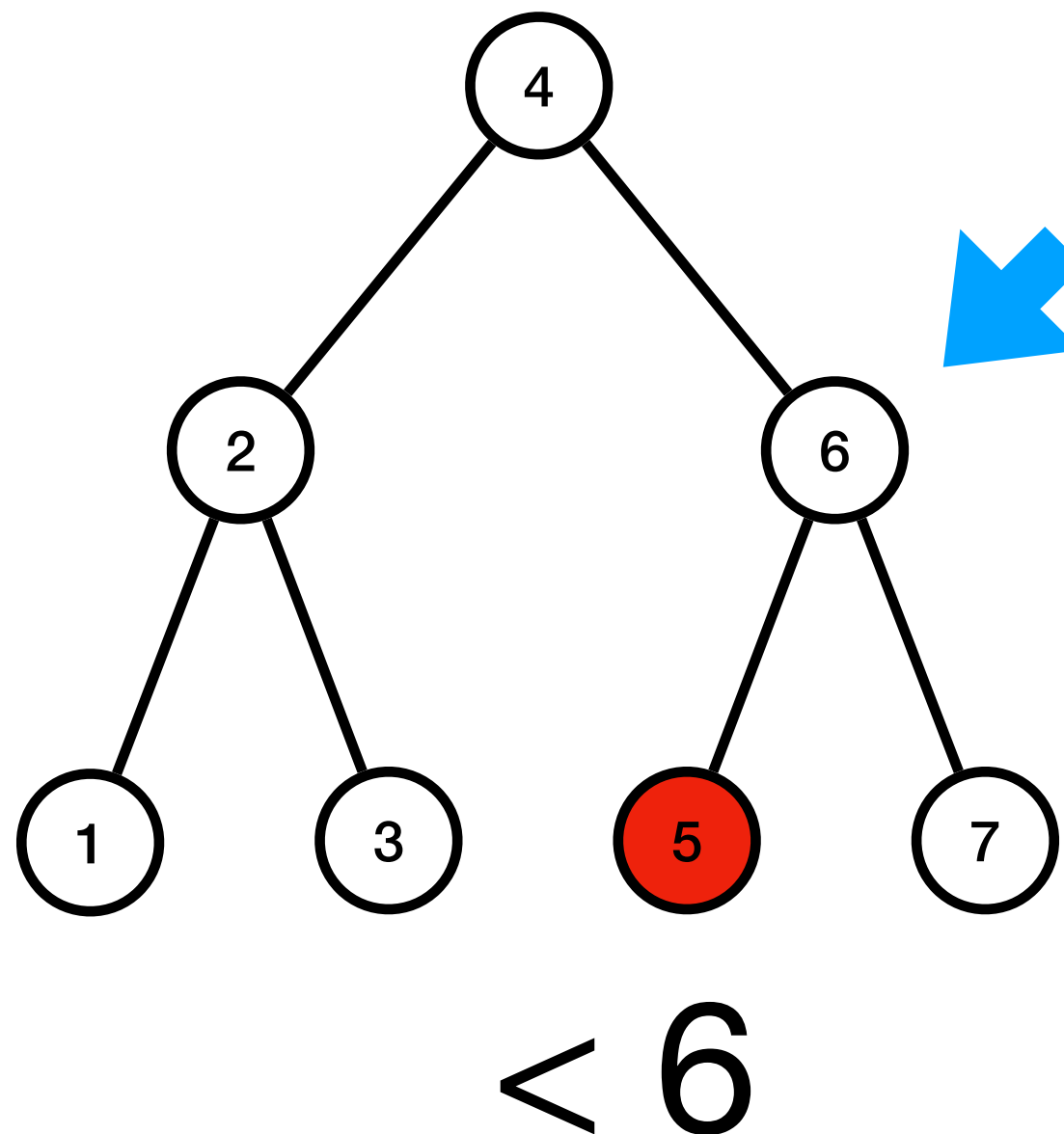
< 2

> 2

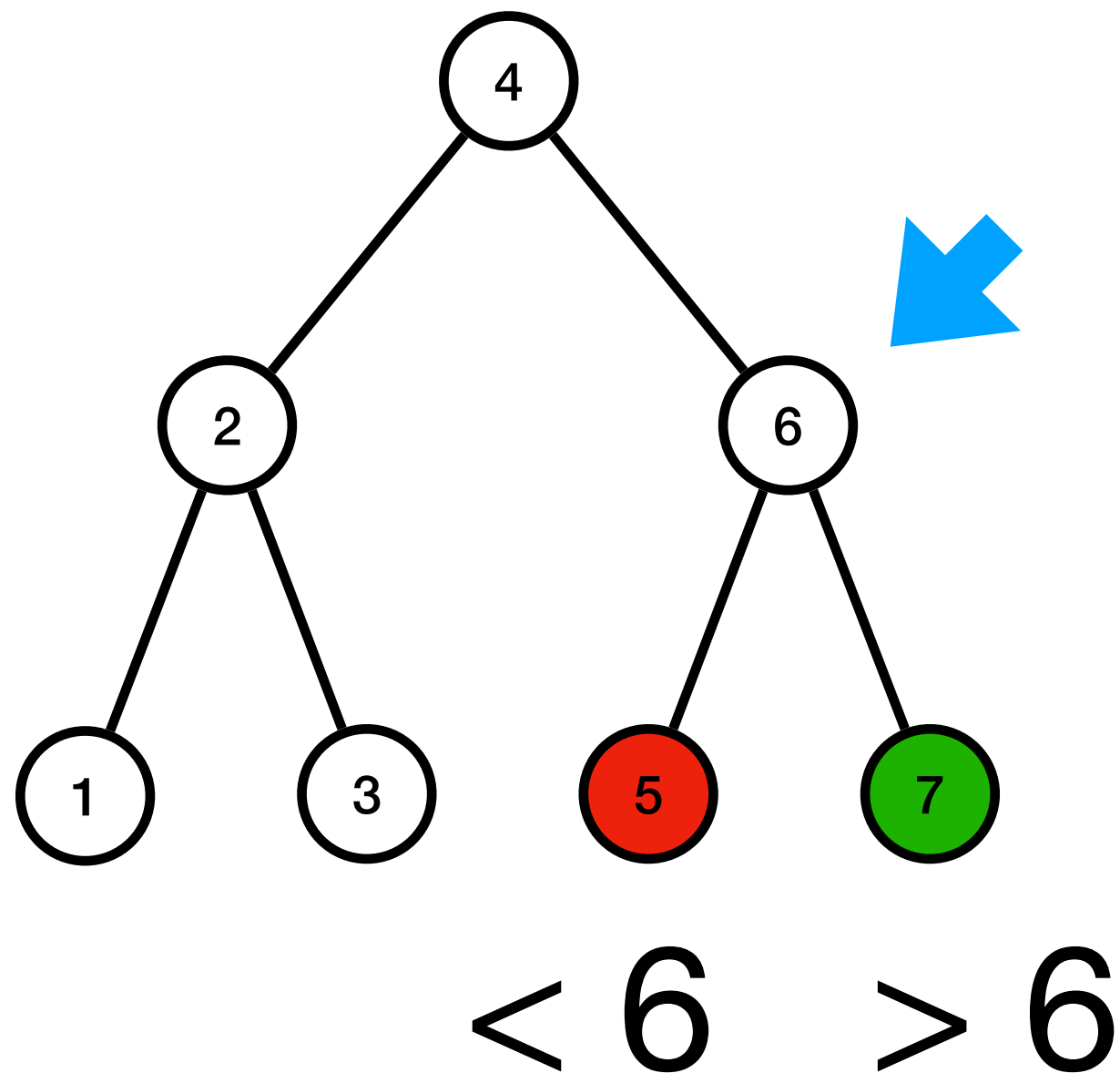
Arbres binaires de recherche



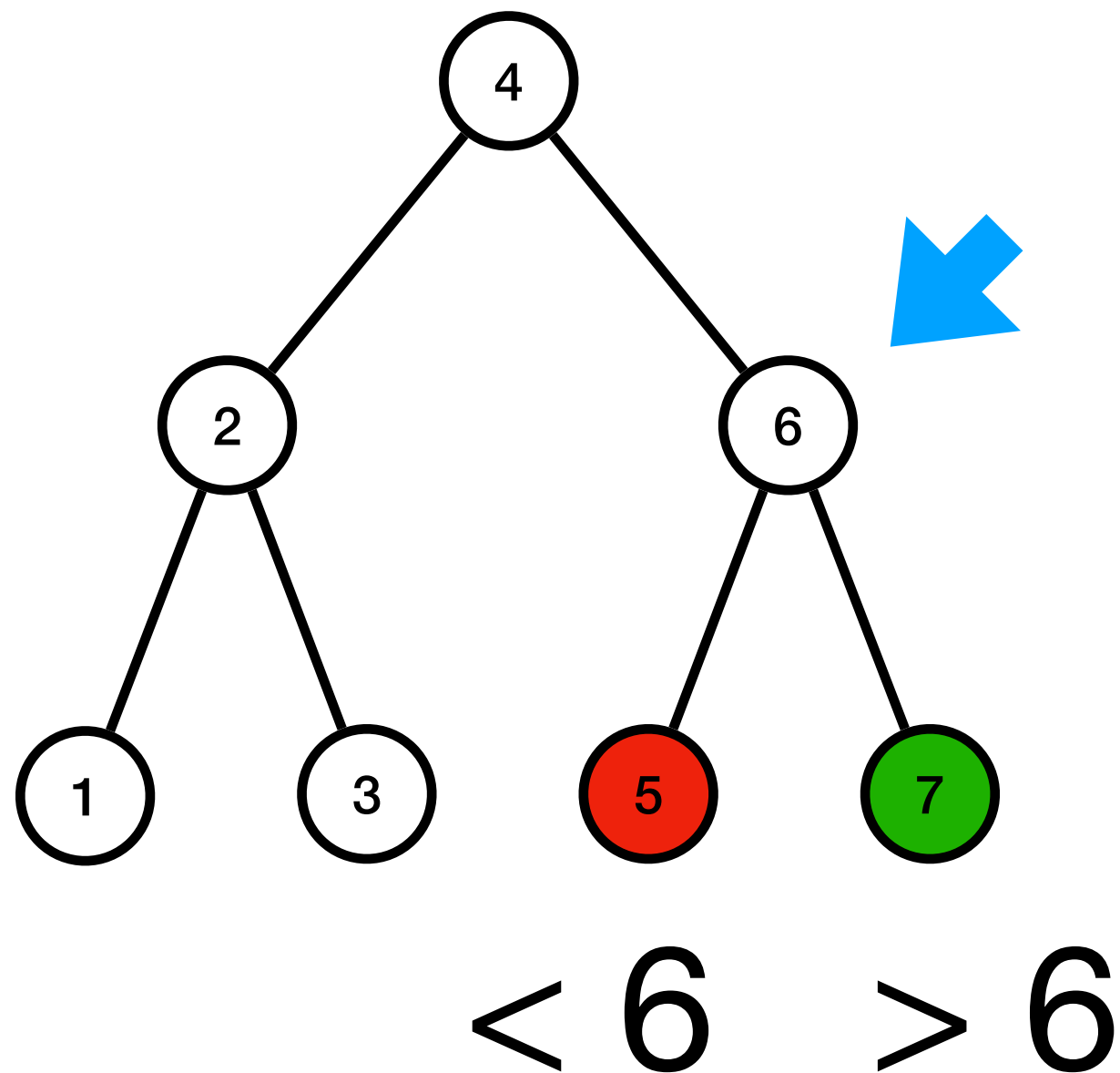
Arbres binaires de recherche



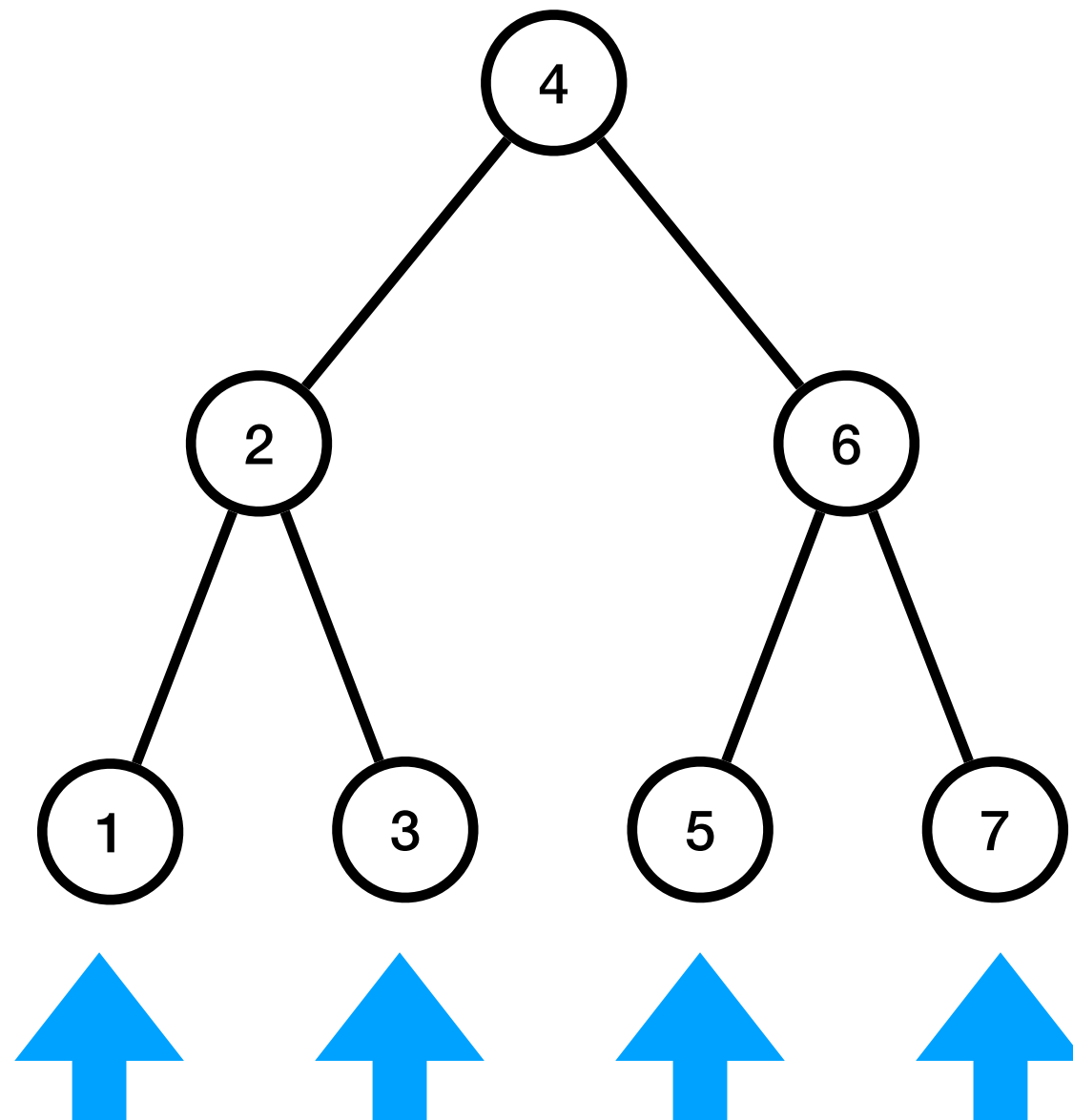
Arbres binaires de recherche



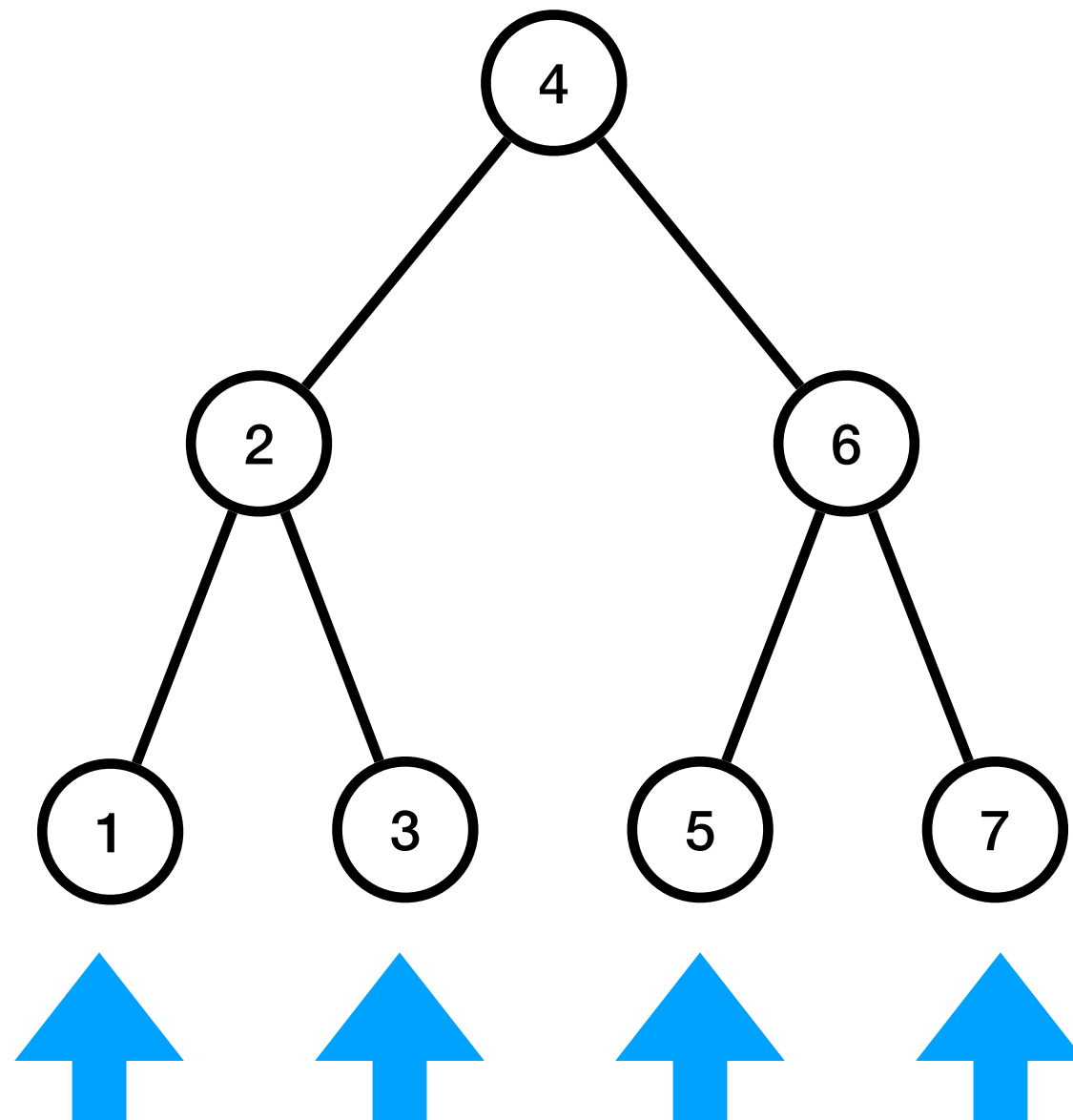
Arbres binaires de recherche



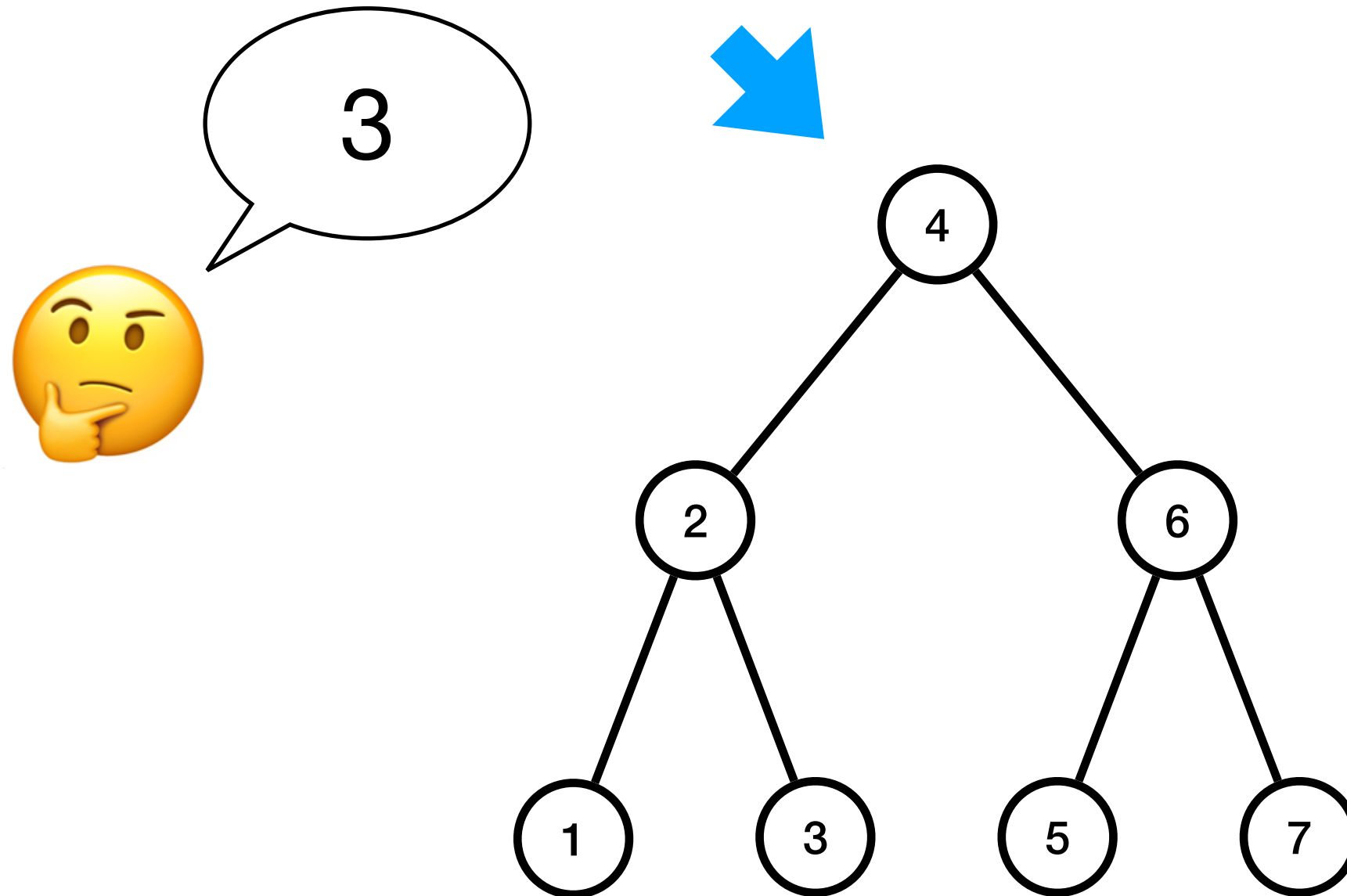
Arbres binaires de recherche



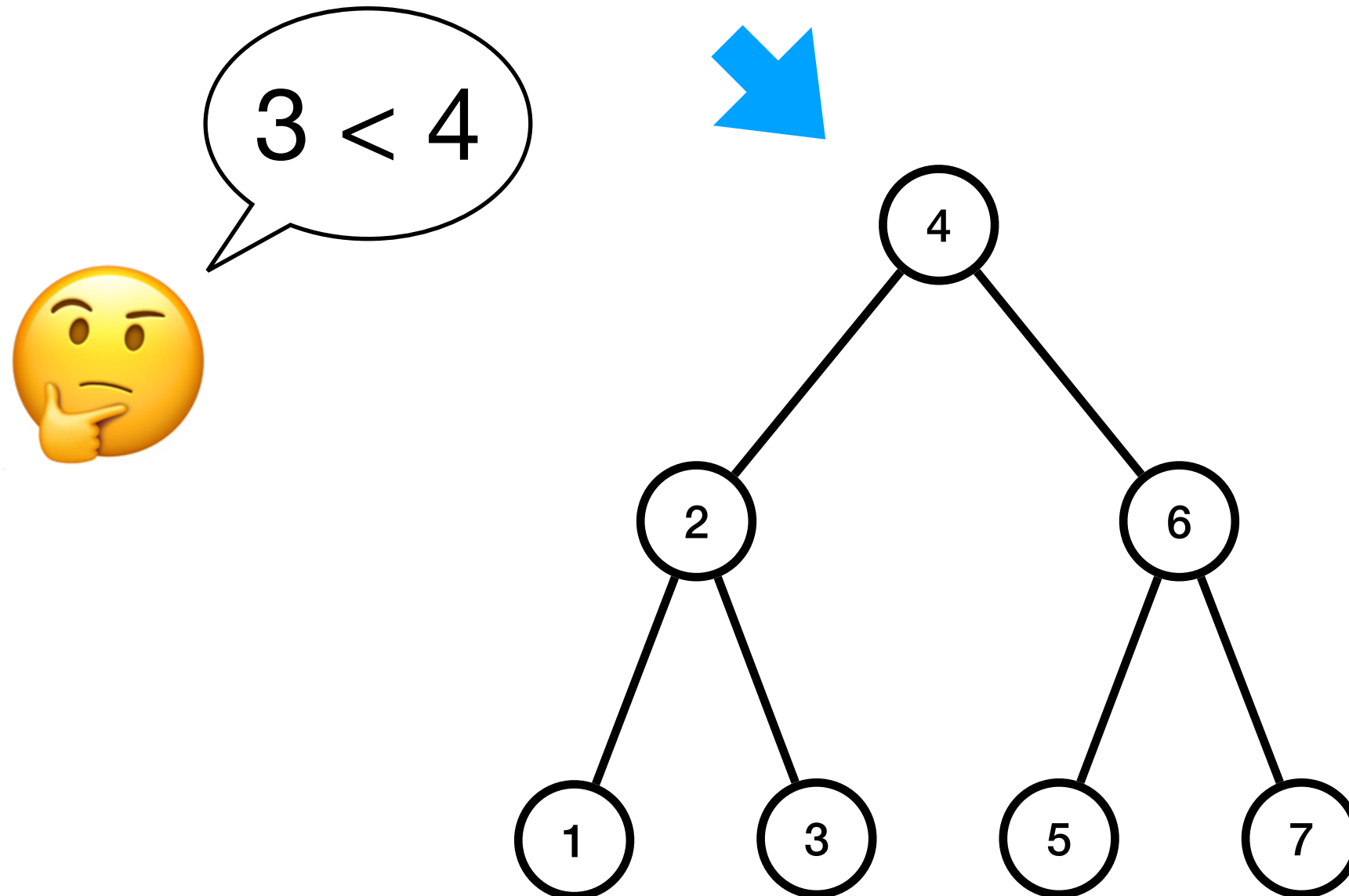
Arbres binaires de recherche



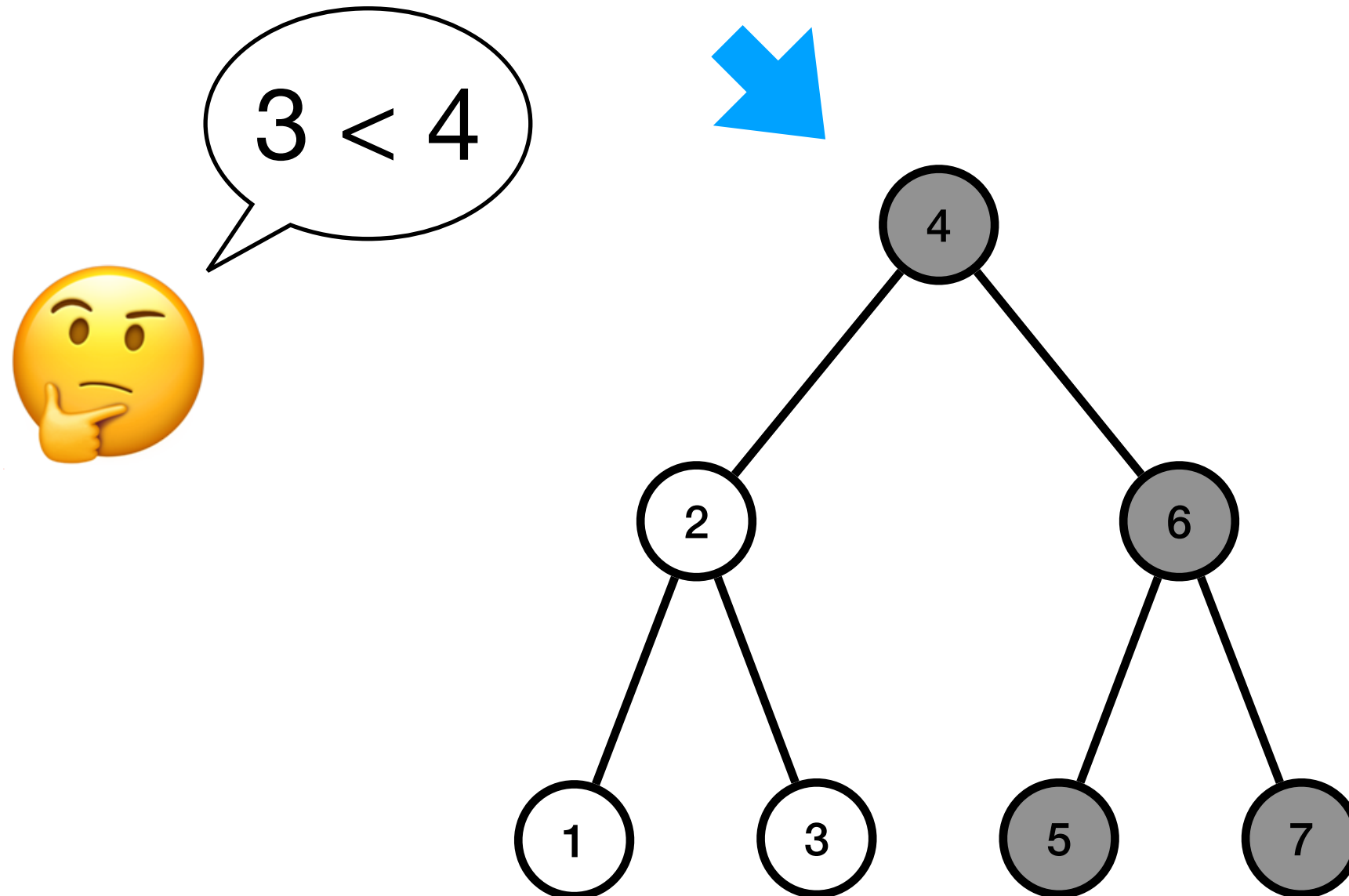
Recherche dans un ABR



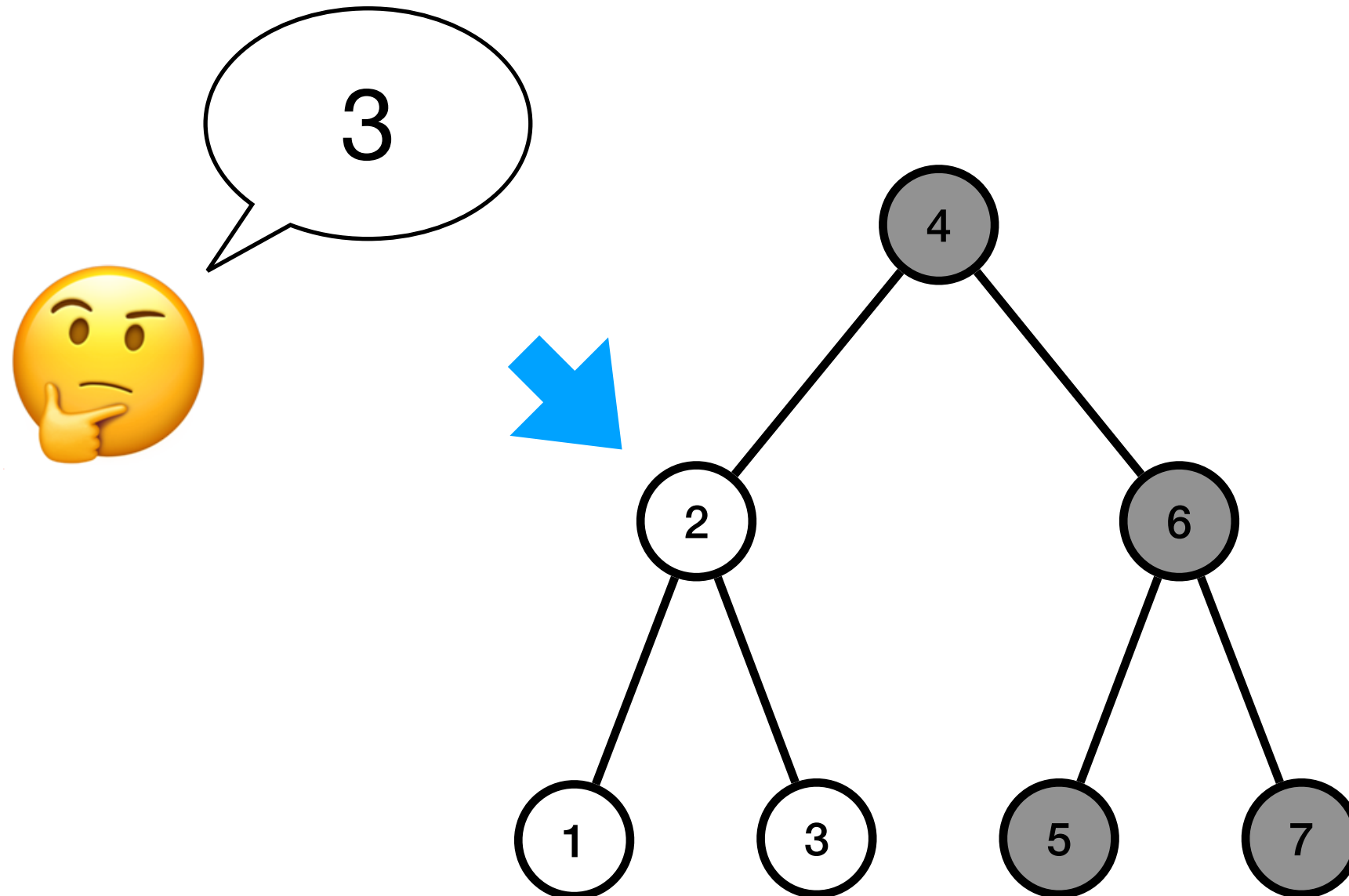
Recherche dans un ABR



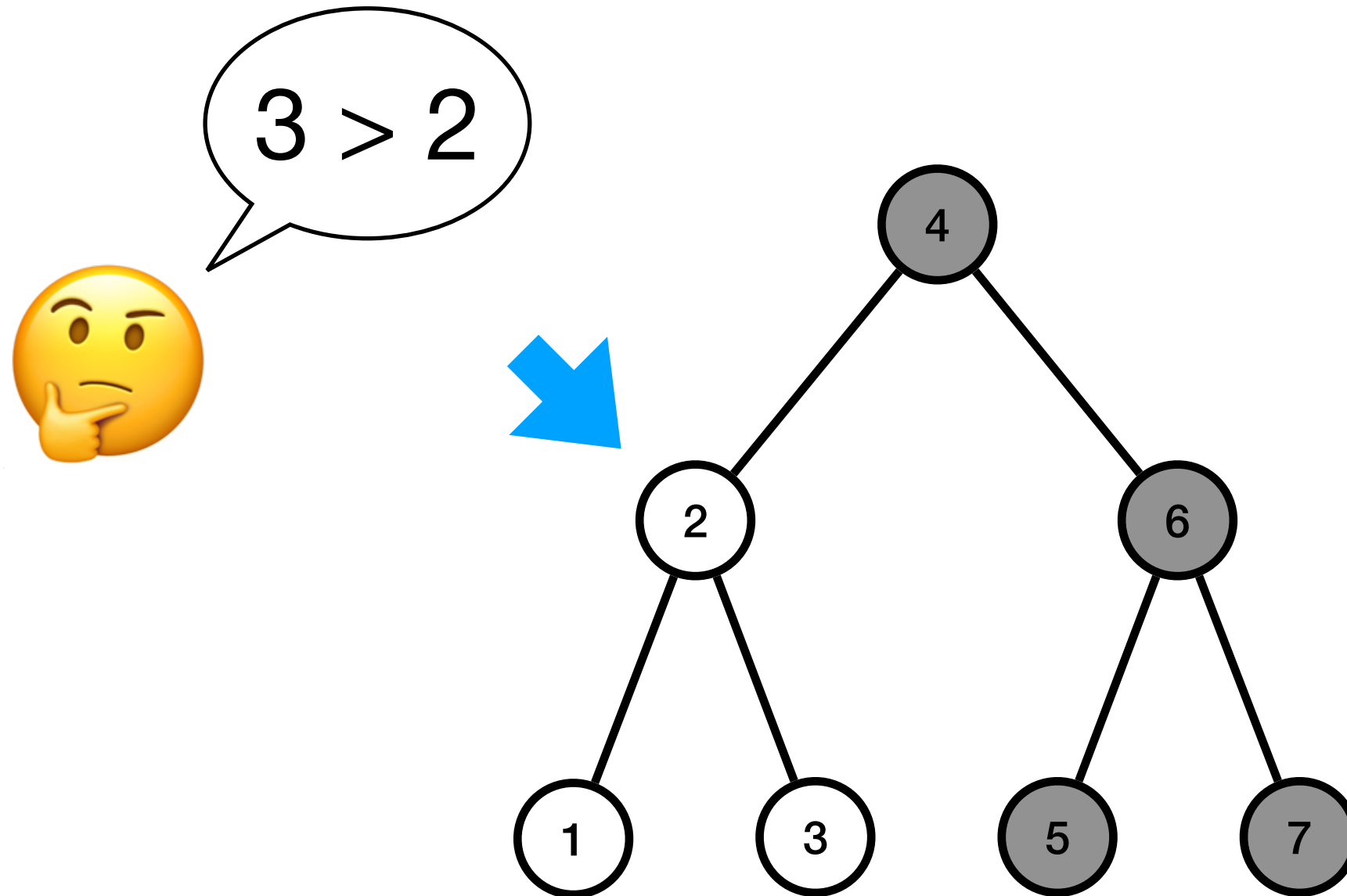
Recherche dans un ABR



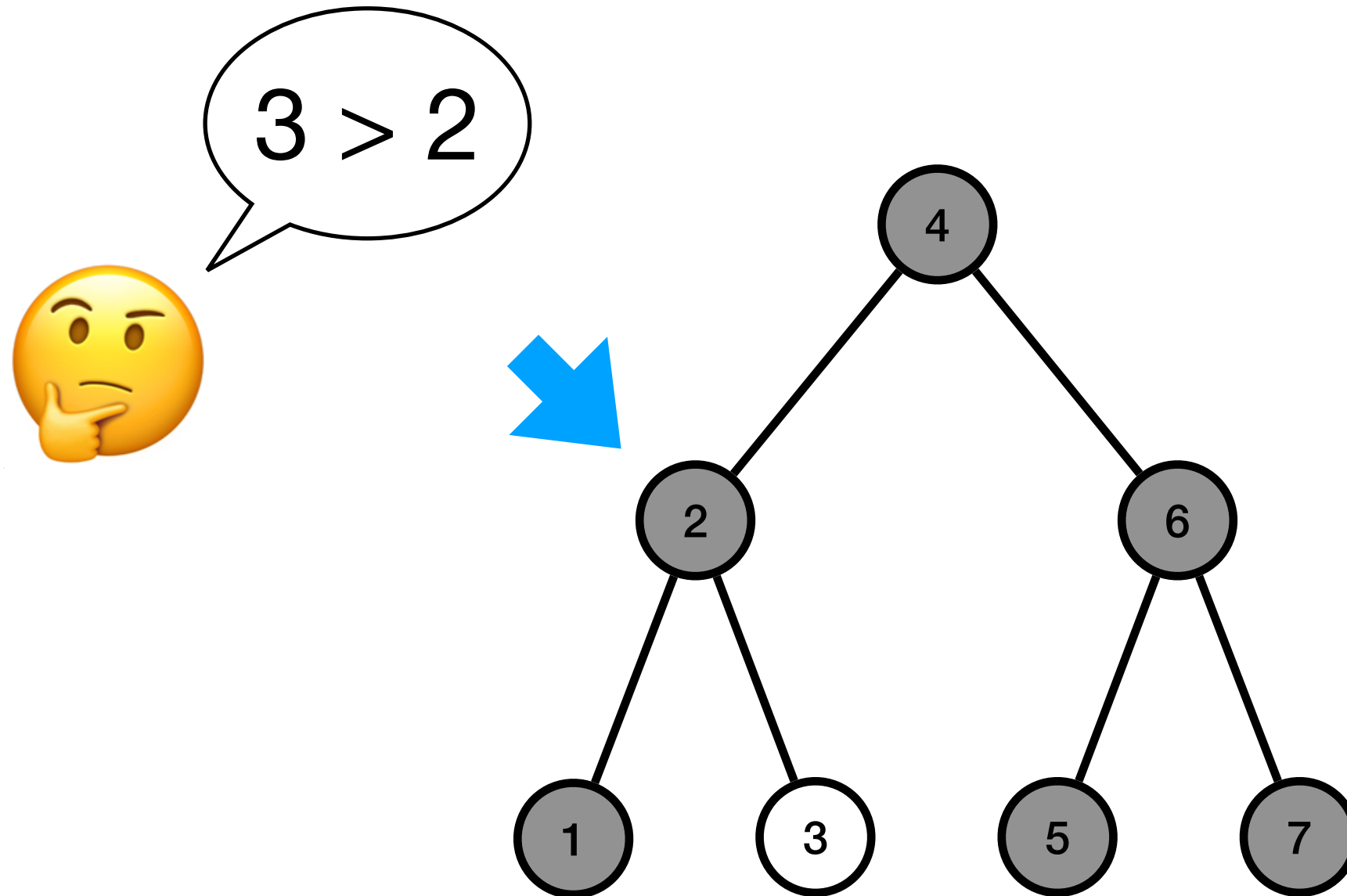
Recherche dans un ABR



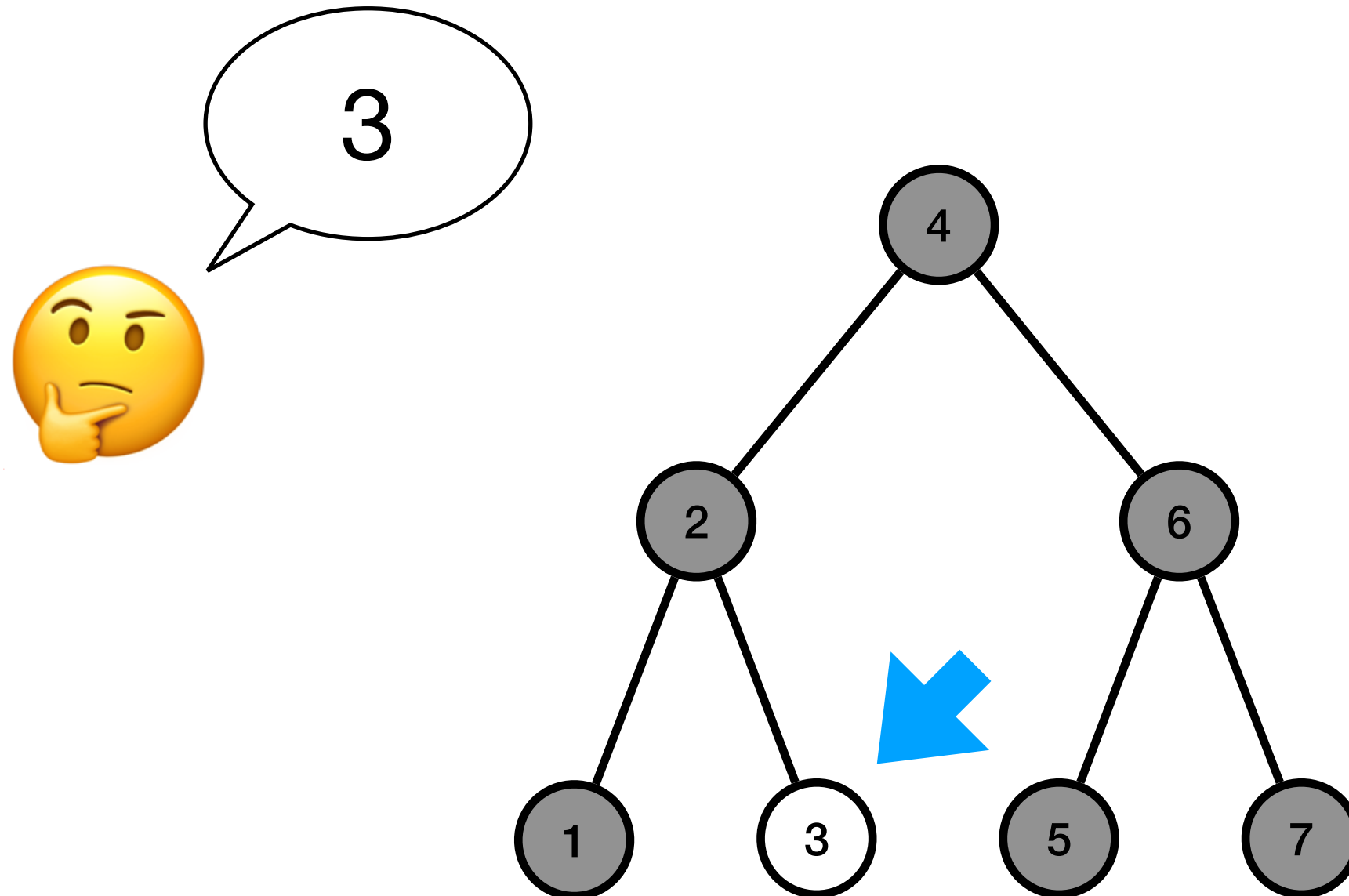
Recherche dans un ABR



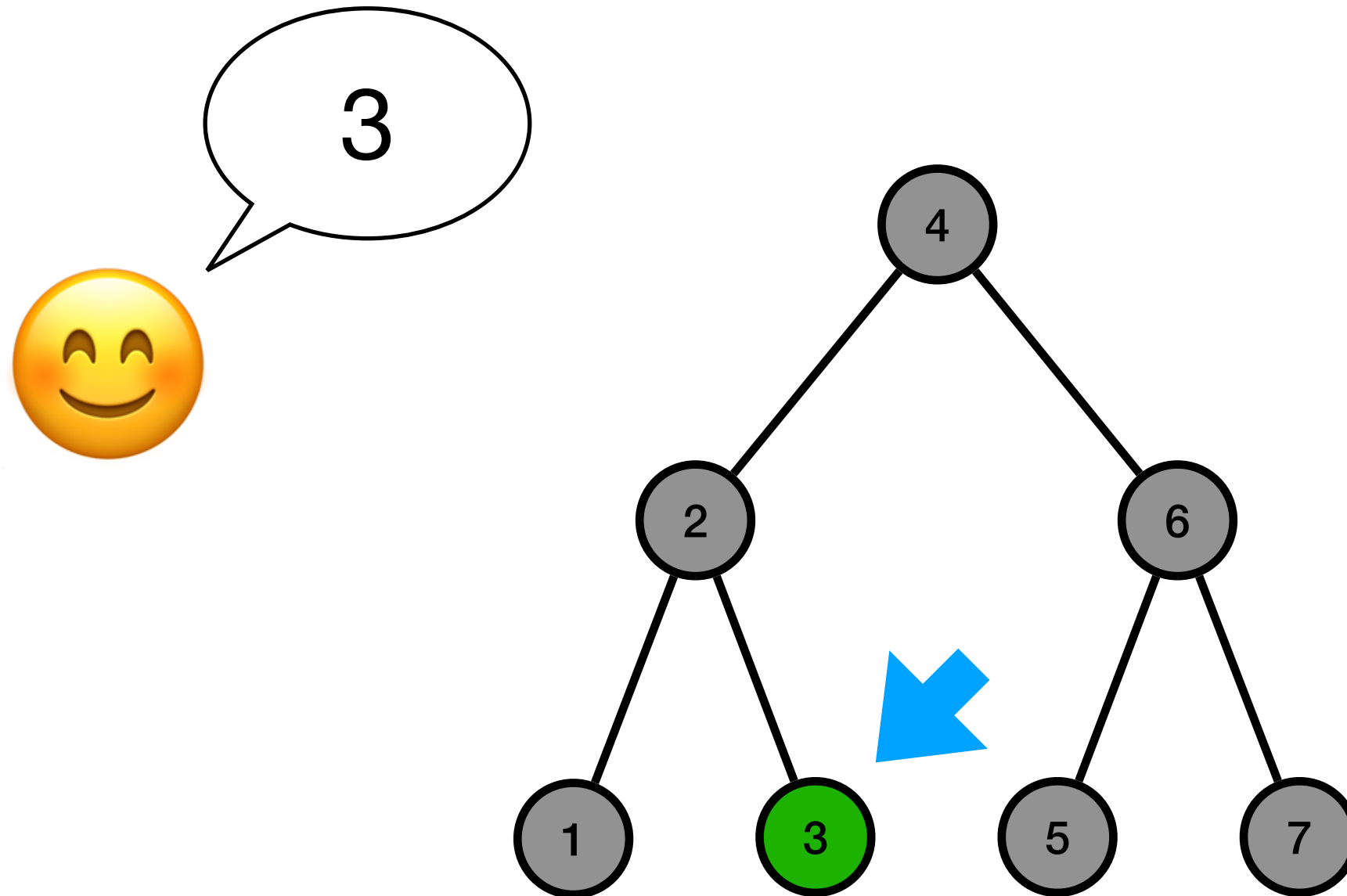
Recherche dans un ABR



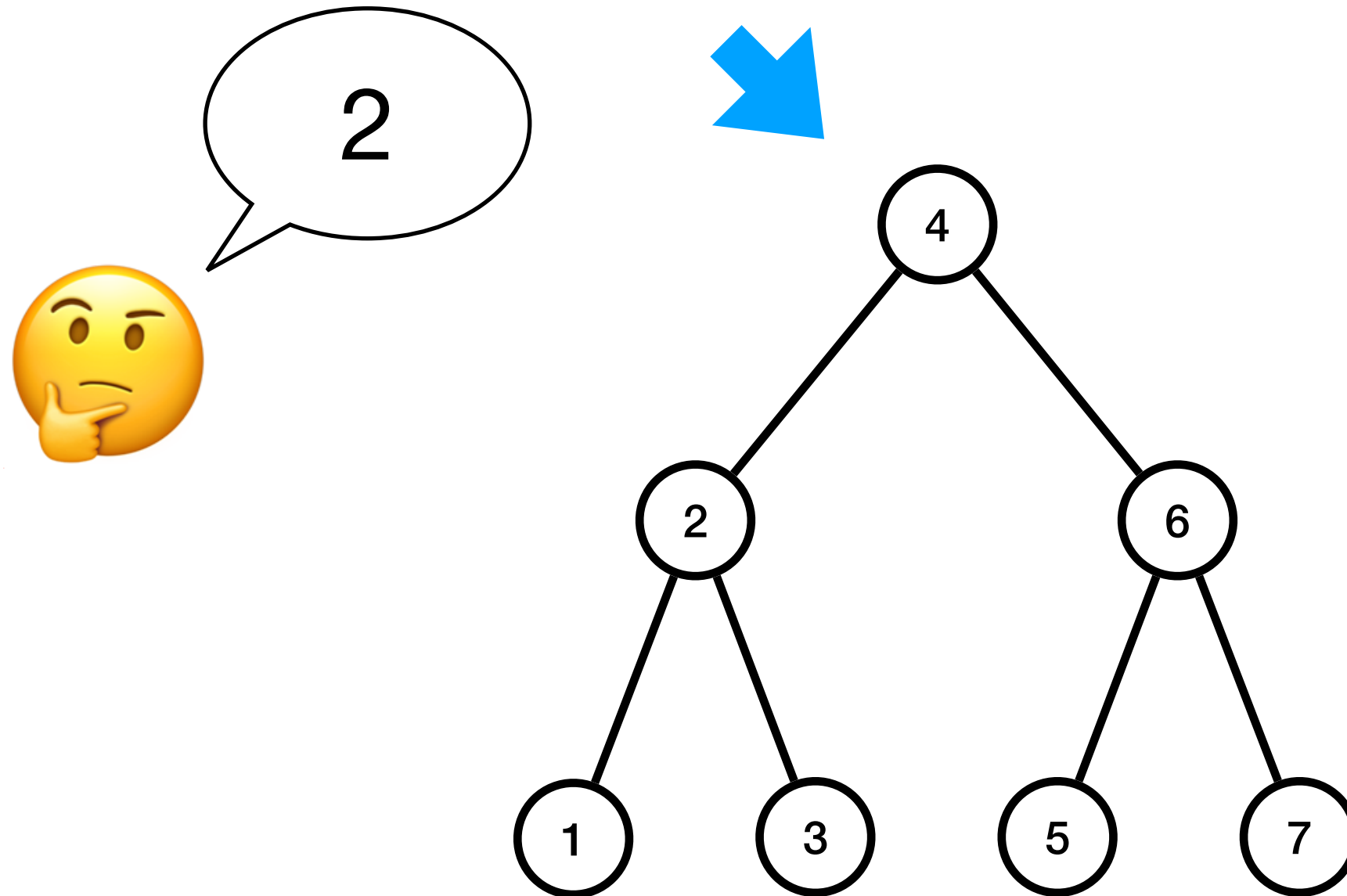
Recherche dans un ABR



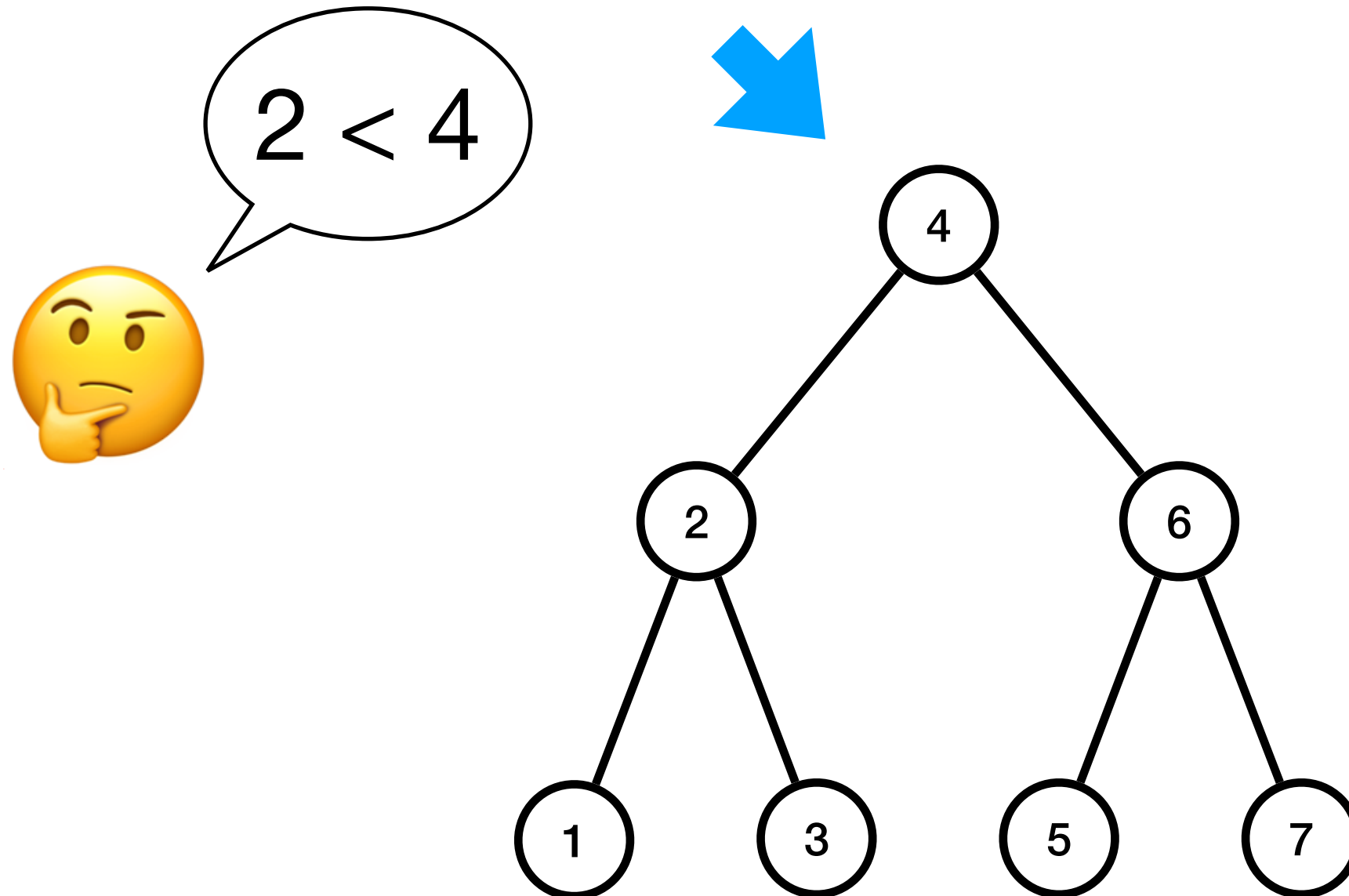
Recherche dans un ABR



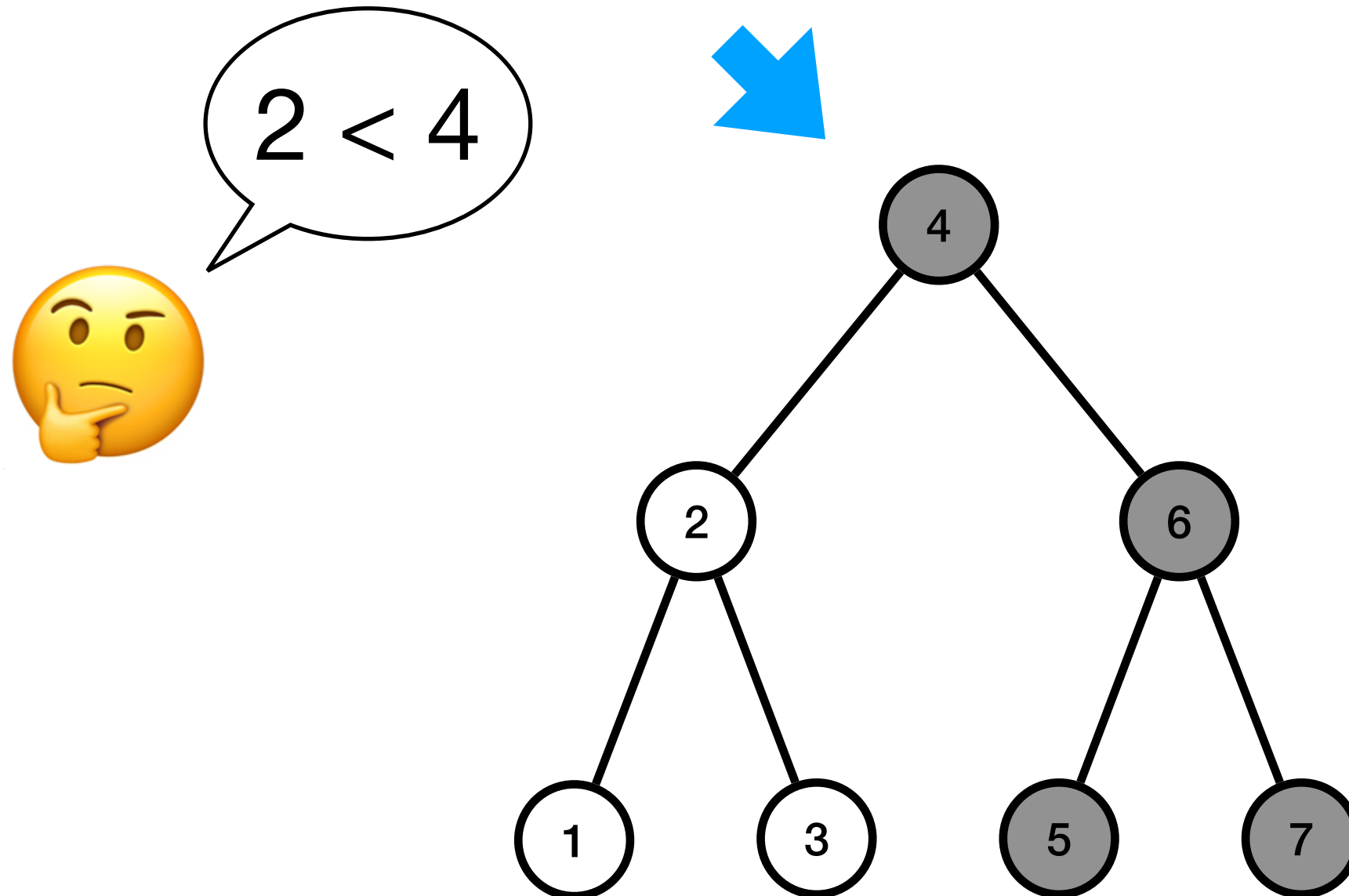
Recherche dans un ABR



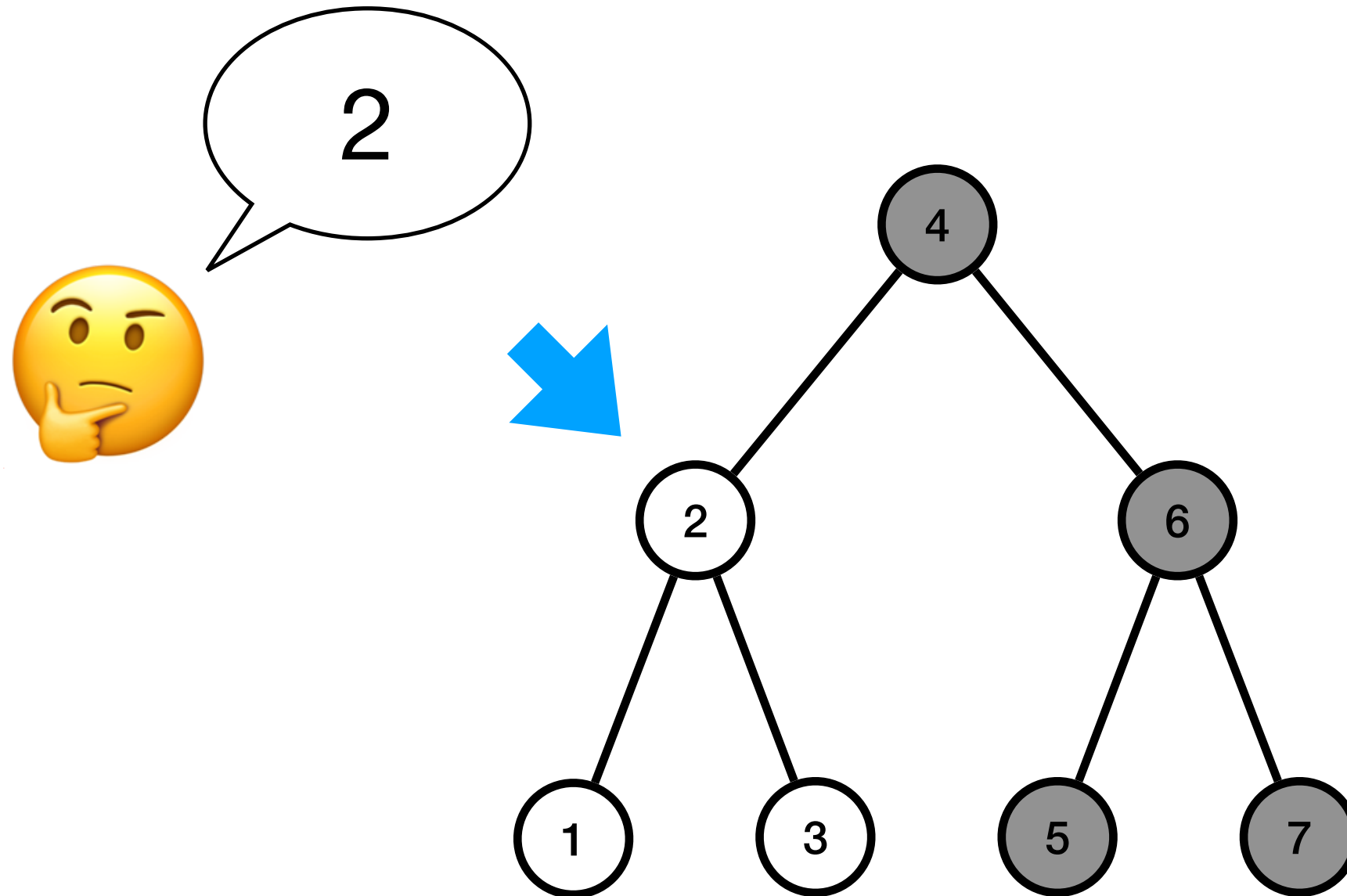
Recherche dans un ABR



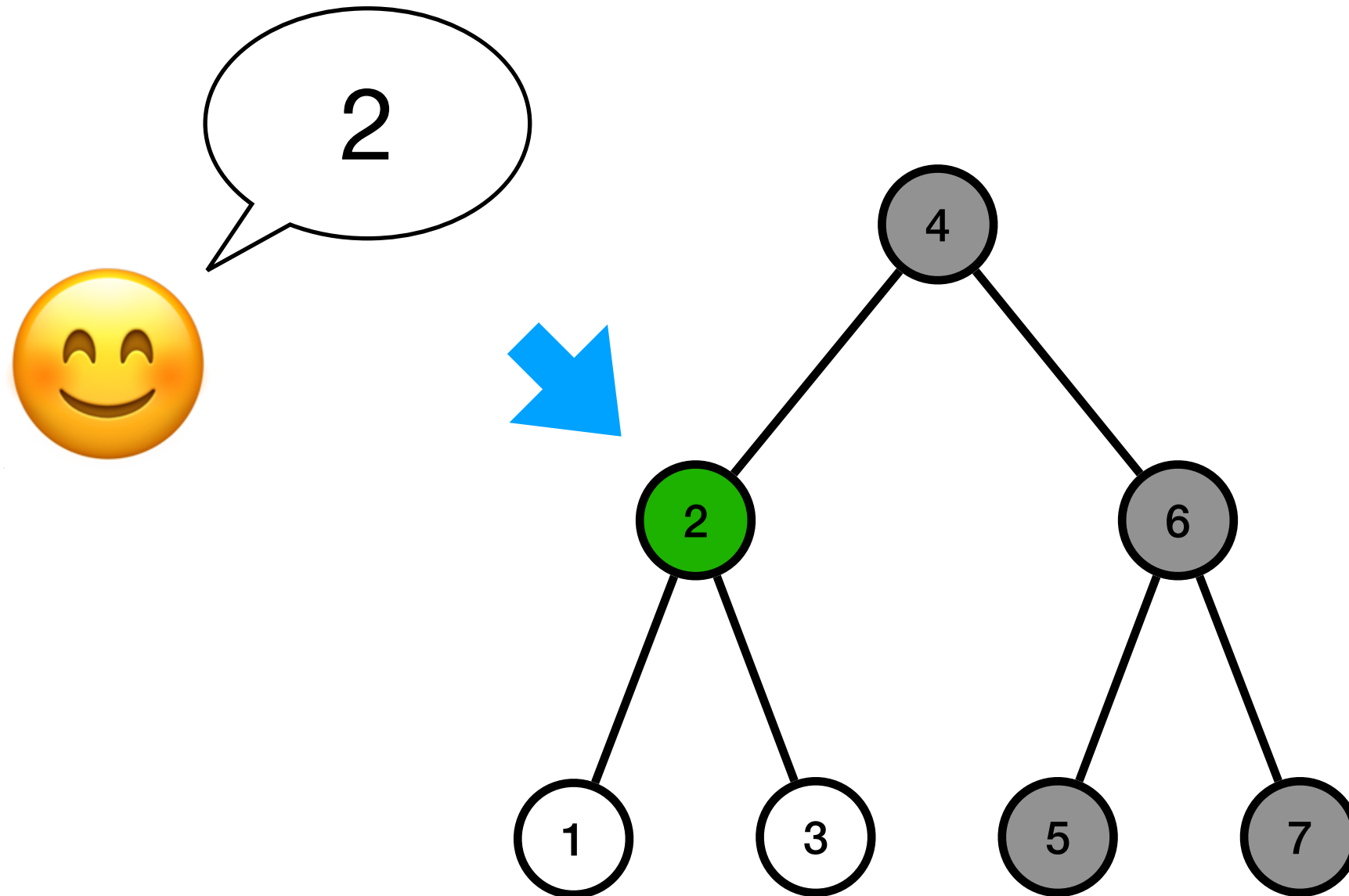
Recherche dans un ABR



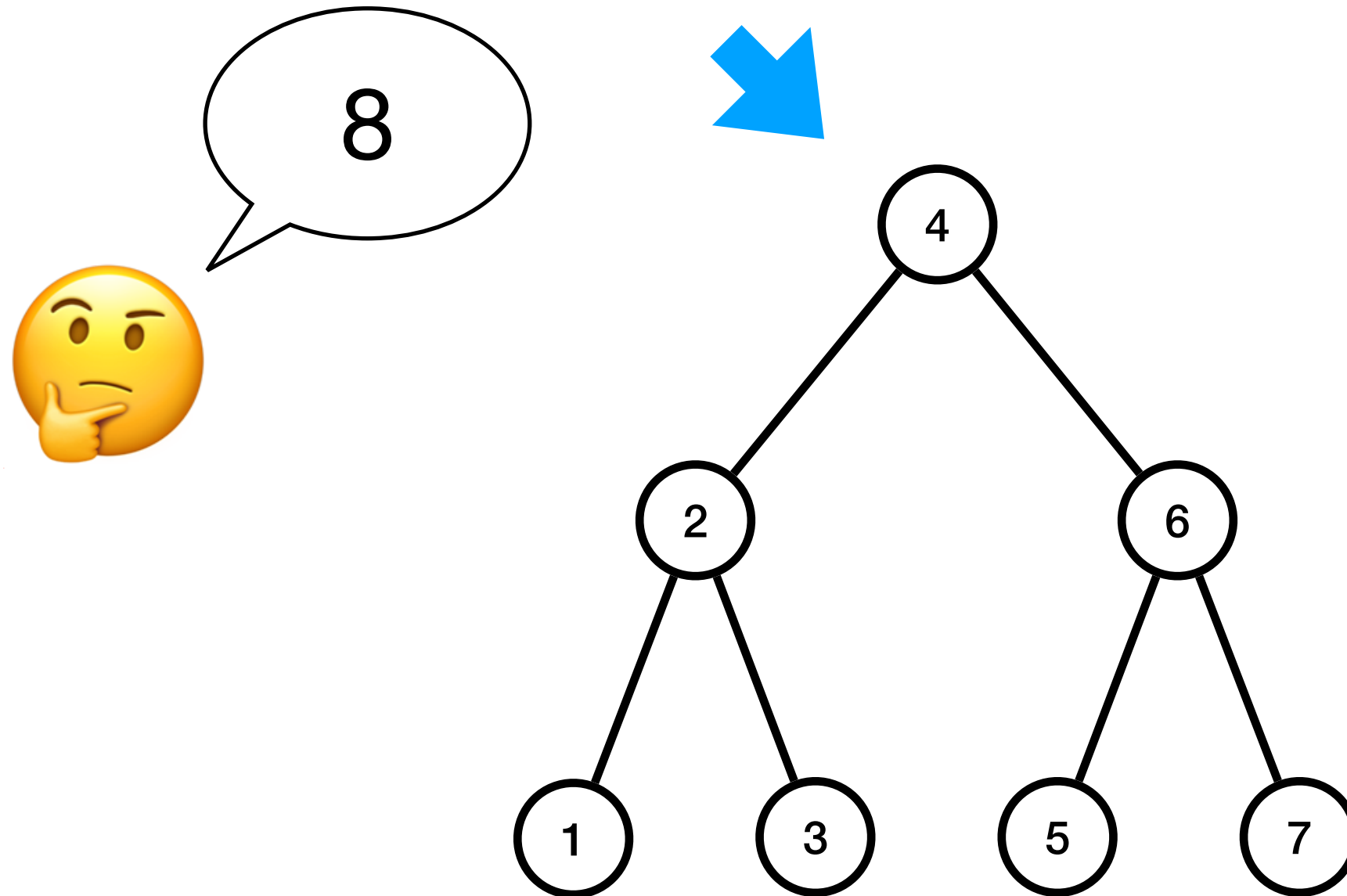
Recherche dans un ABR



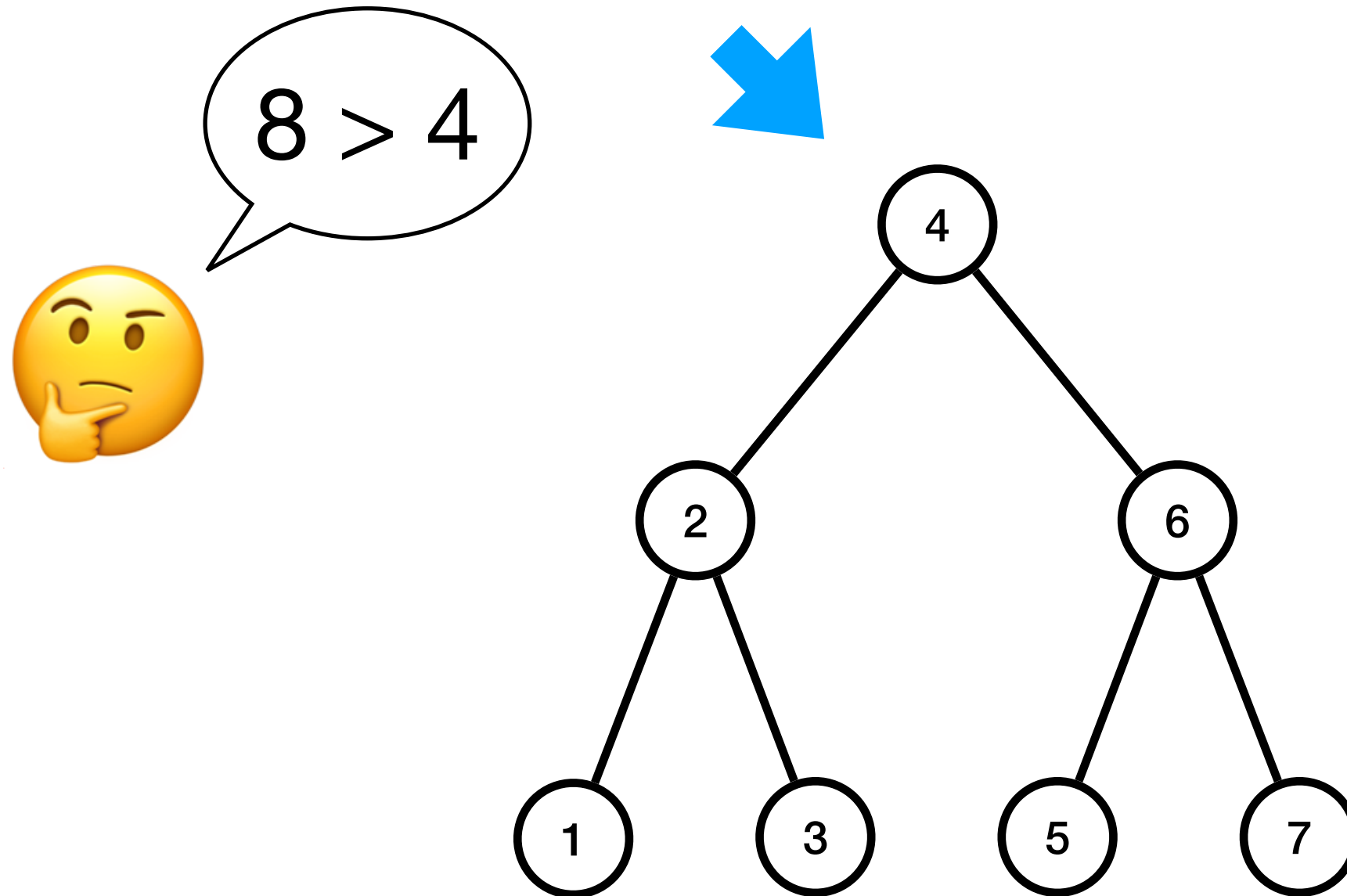
Recherche dans un ABR



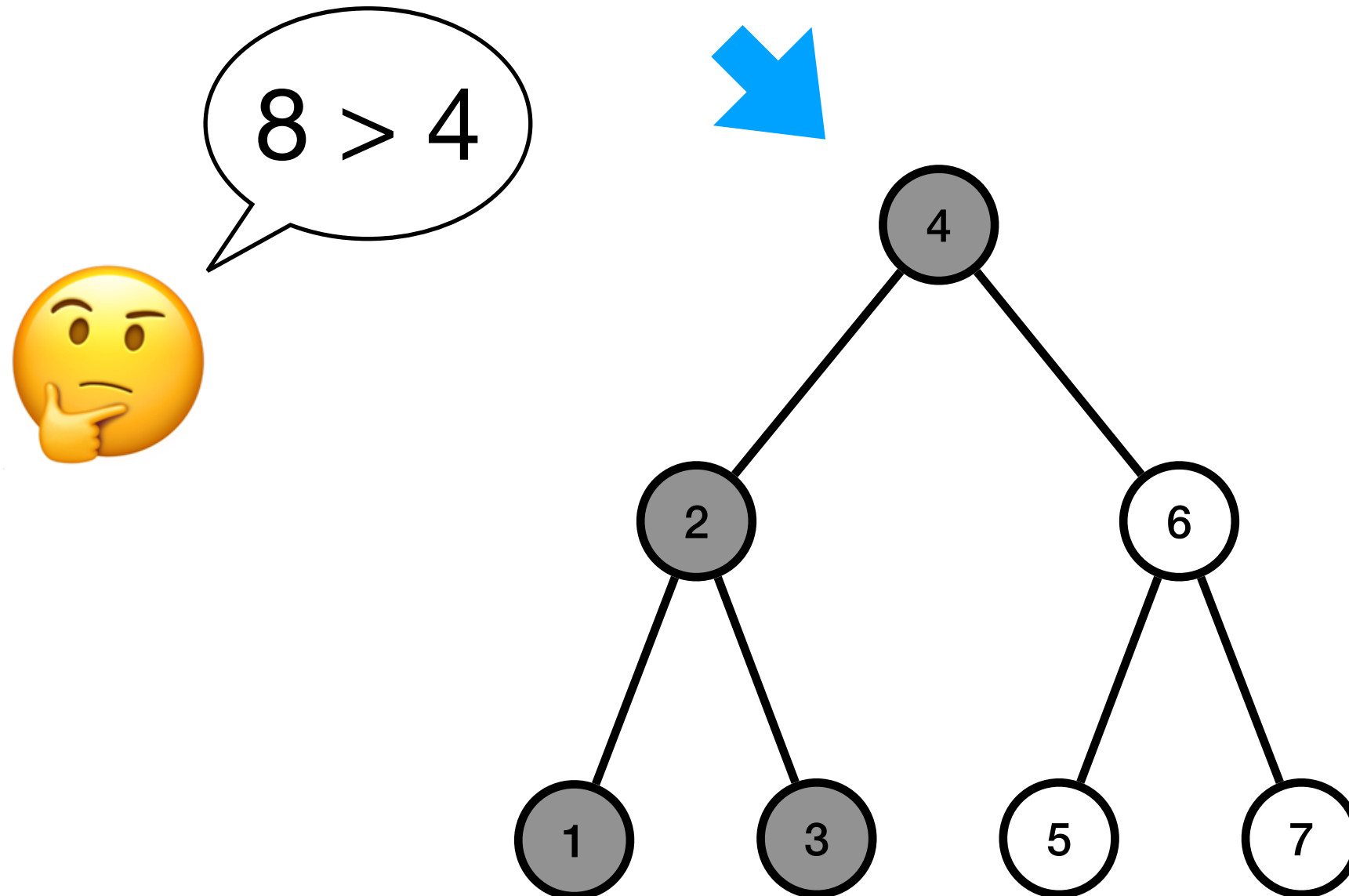
Recherche dans un ABR



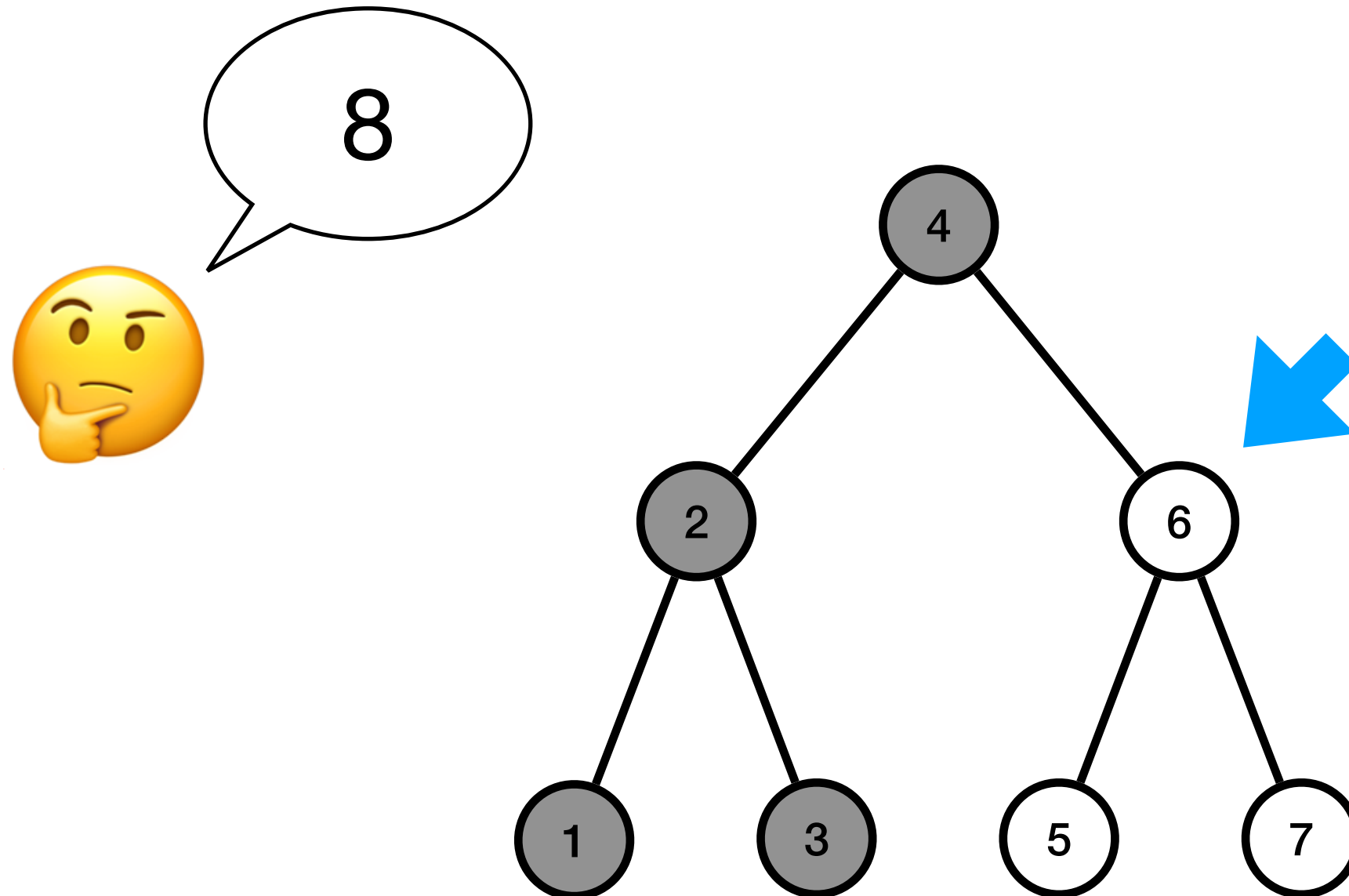
Recherche dans un ABR



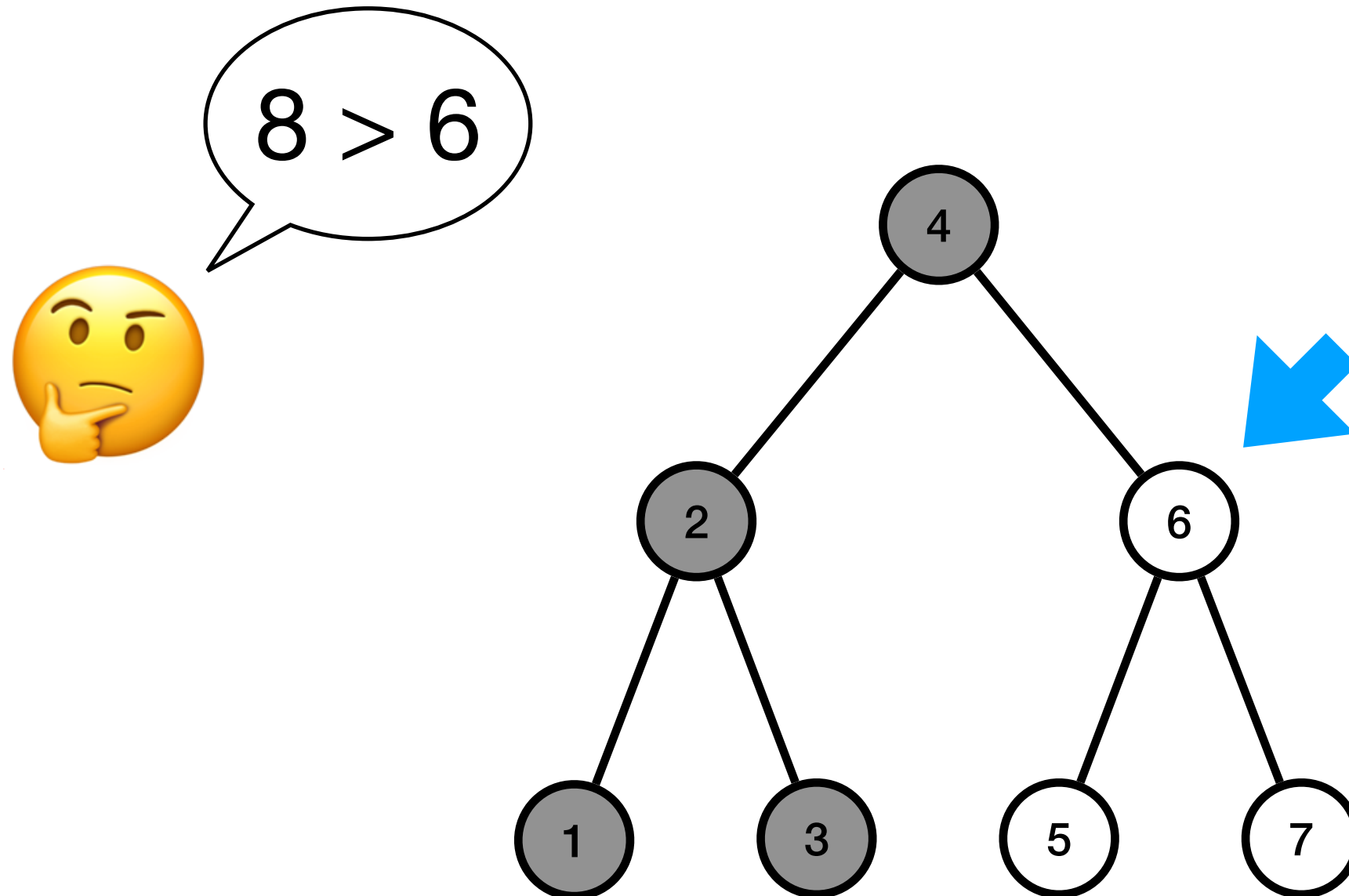
Recherche dans un ABR



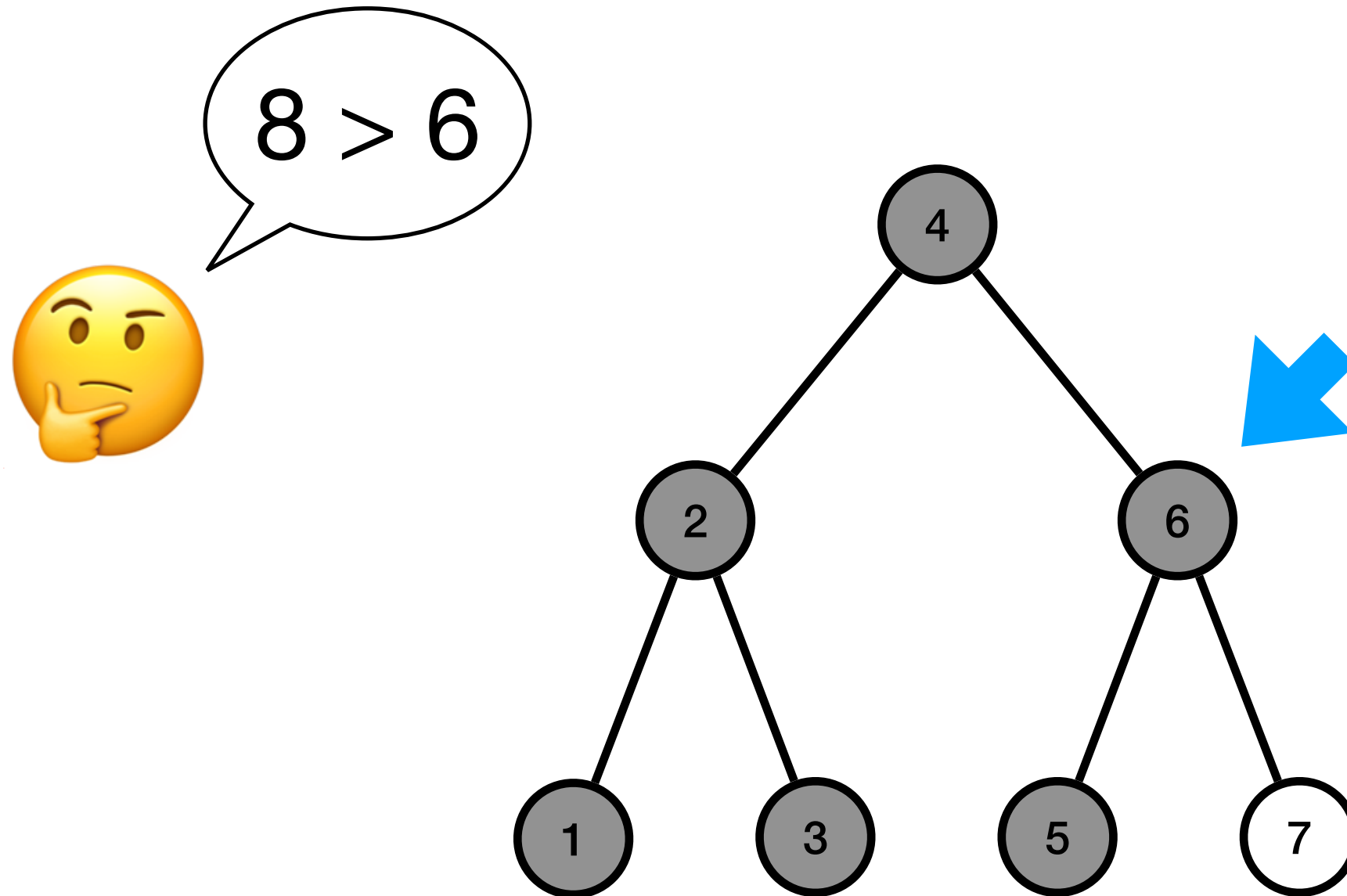
Recherche dans un ABR



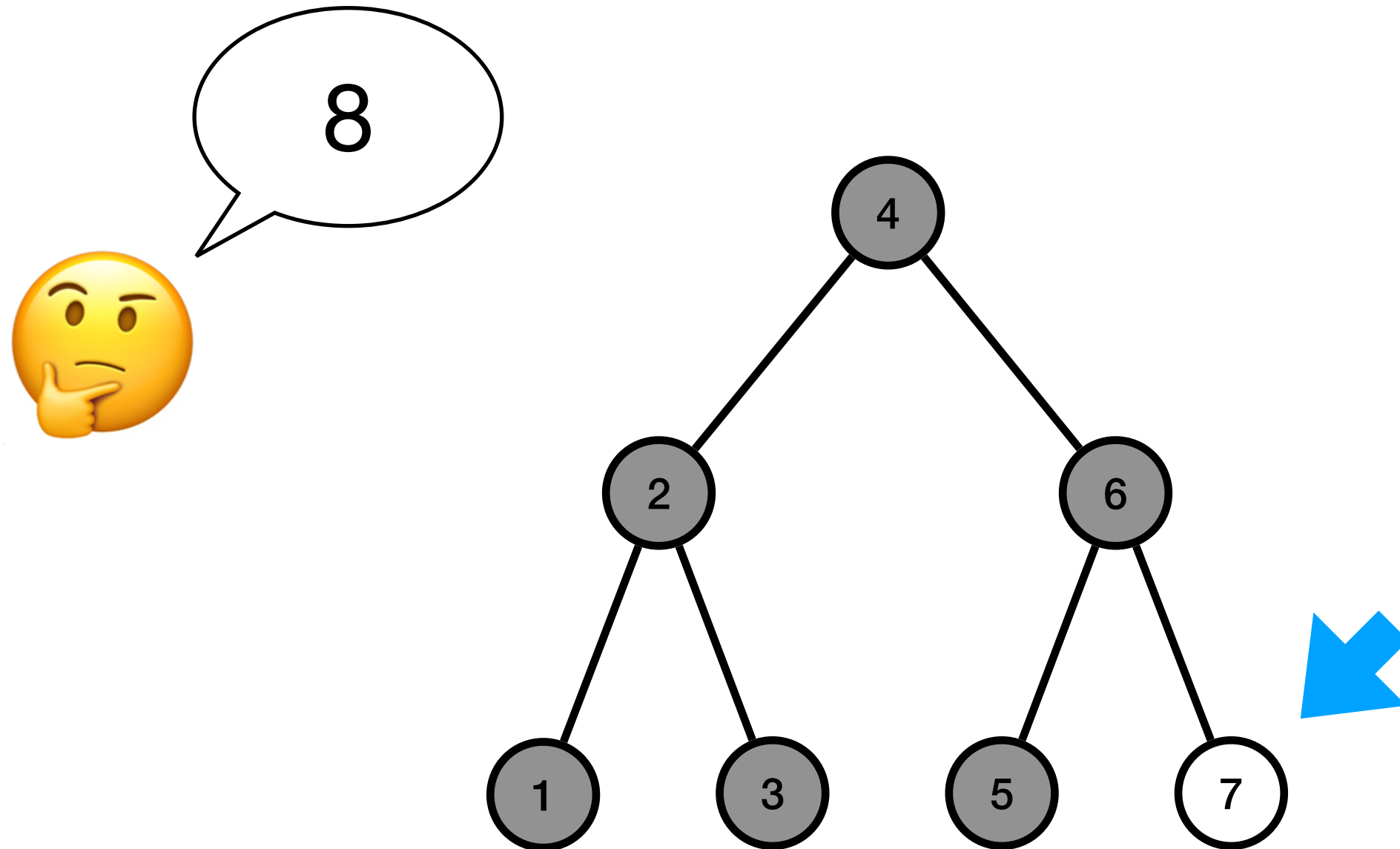
Recherche dans un ABR



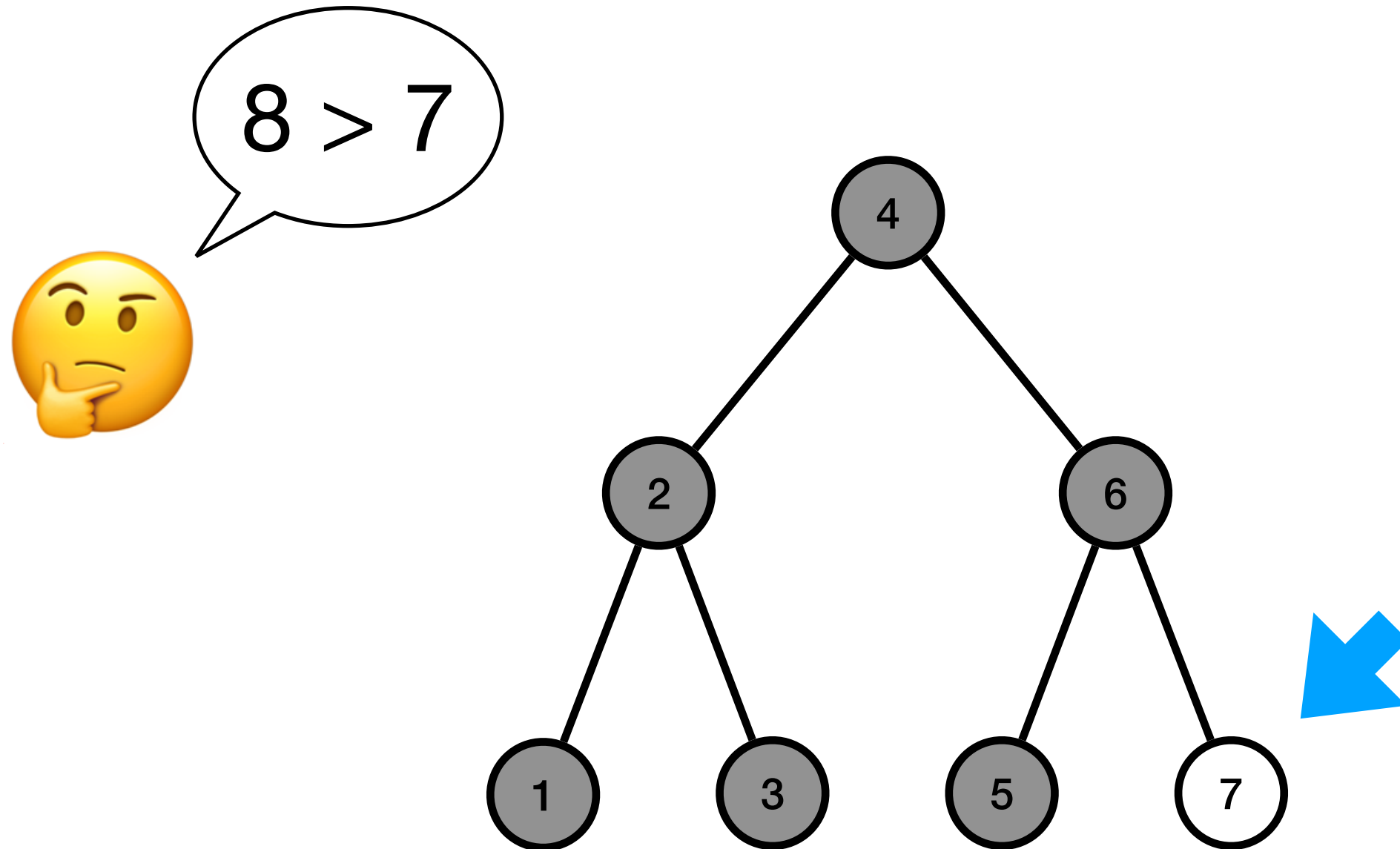
Recherche dans un ABR



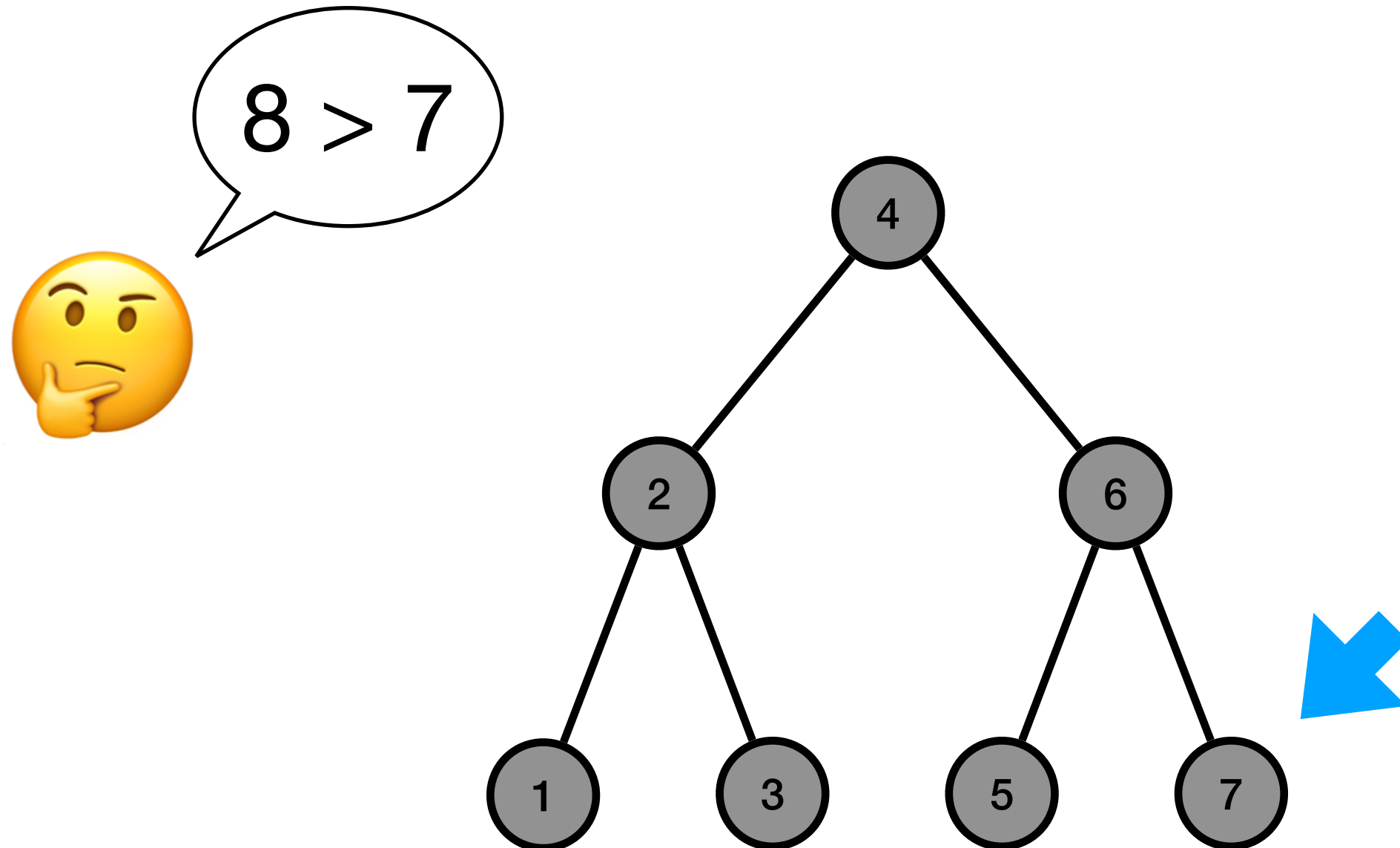
Recherche dans un ABR



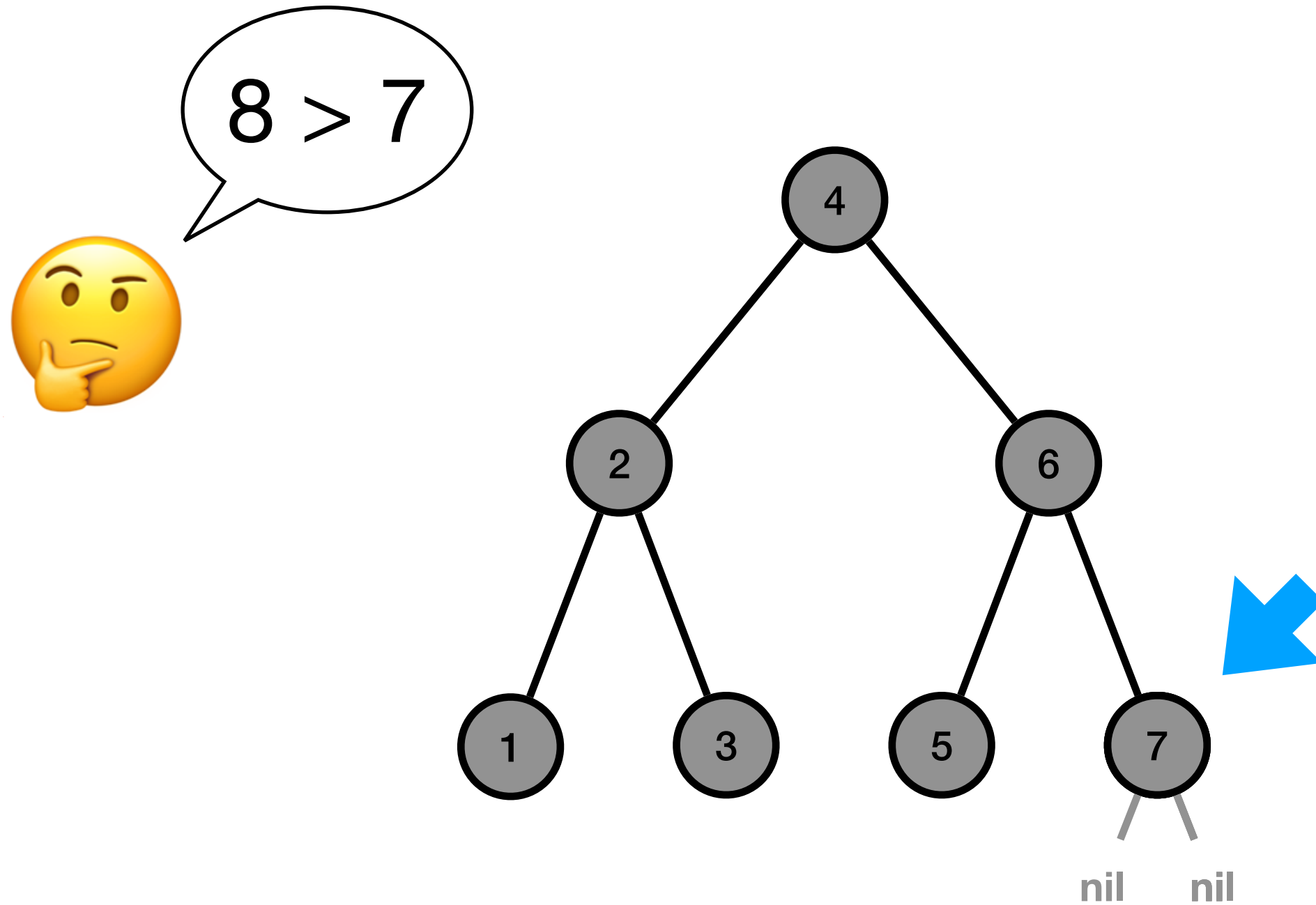
Recherche dans un ABR



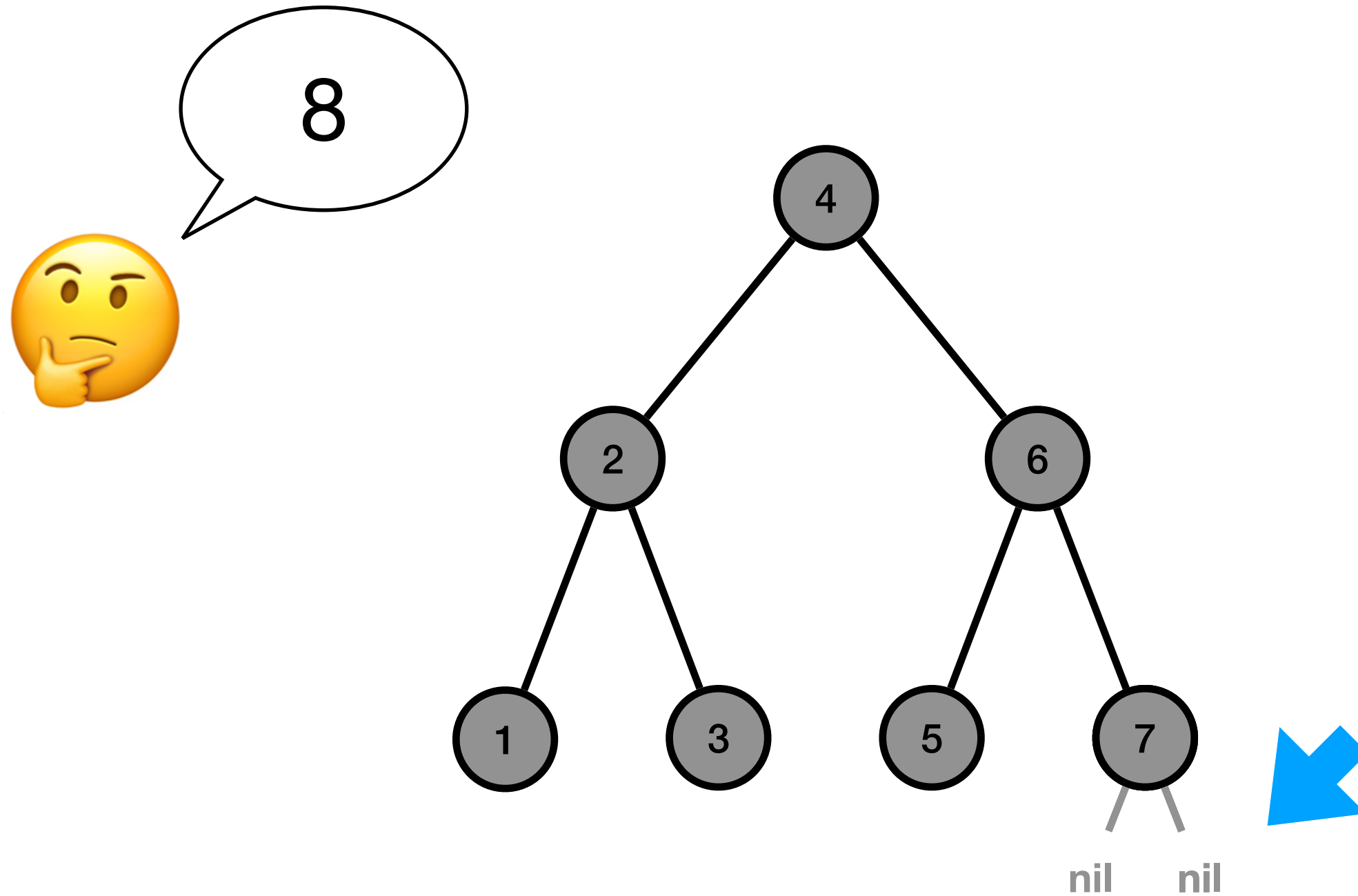
Recherche dans un ABR



Recherche dans un ABR



Recherche dans un ABR



Recherche dans un ABR

