

Science informatique

CM3

Antonio E. Porreca

aeporreca.org/scienceinfo

Retour sur le tri par insertion

Efficacité

```
def trier_par_insertion(A) :  
    n = len(A)  
    for i in range(1, n):  
        x = A[i]  
        j = i  
        while j > 0 and x < A[j - 1]:  
            A[j] = A[j - 1]  
            j = j - 1  
        A[j] = x
```

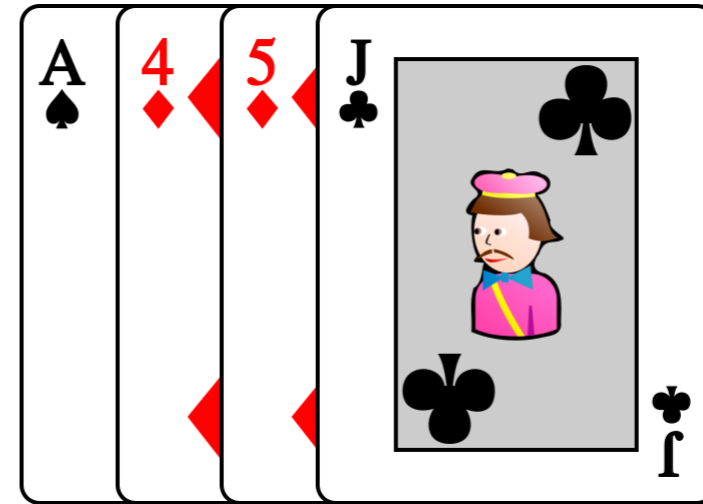
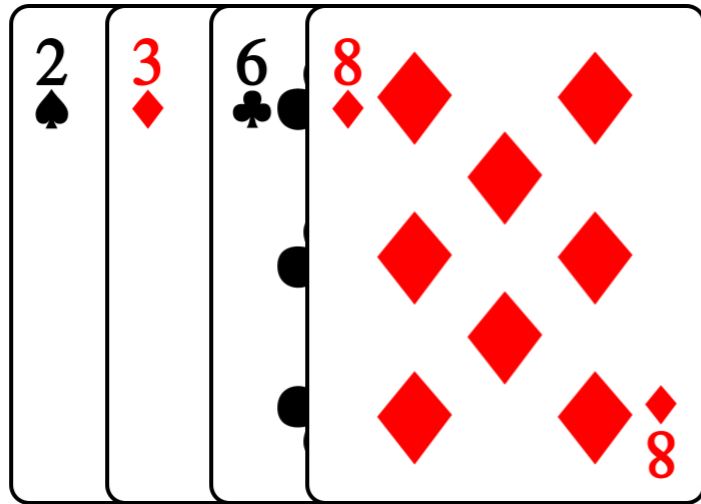
Efficacité

```
def trier_par_insertion(A) :  
    n = len(A)  
    for i in range(1, n):  
        x = A[i]  
        j = i  
        while j > 0 and x < A[j - 1]:  
            A[j] = A[j - 1]  
            j = j - 1  
        A[j] = x
```

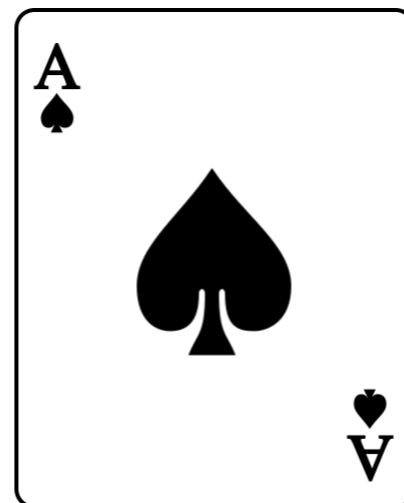
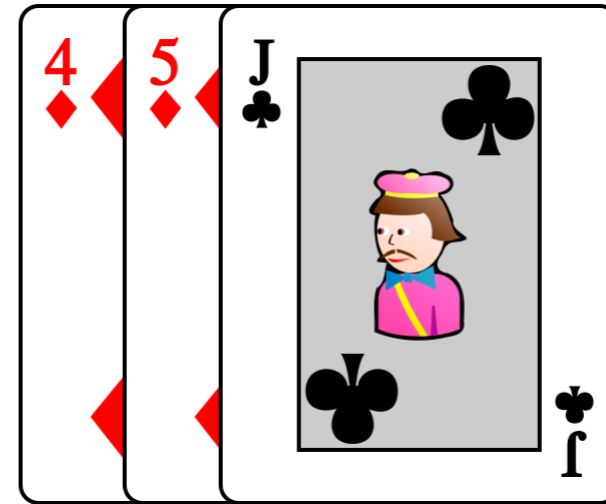
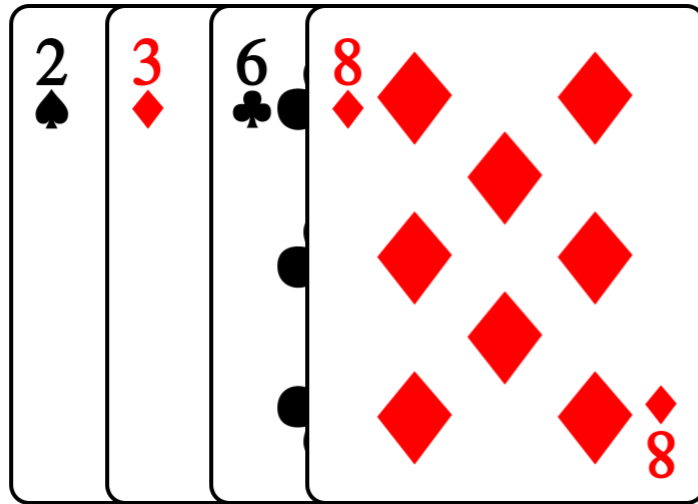
- $O(n)$ opérations dans le meilleur des cas
- $O(n^2)$ opérations dans le pire des cas

**Est-ce qu'on peut
faire mieux que ça ?**

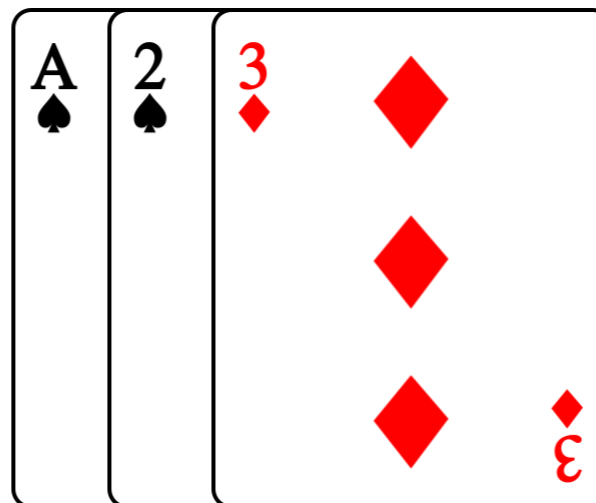
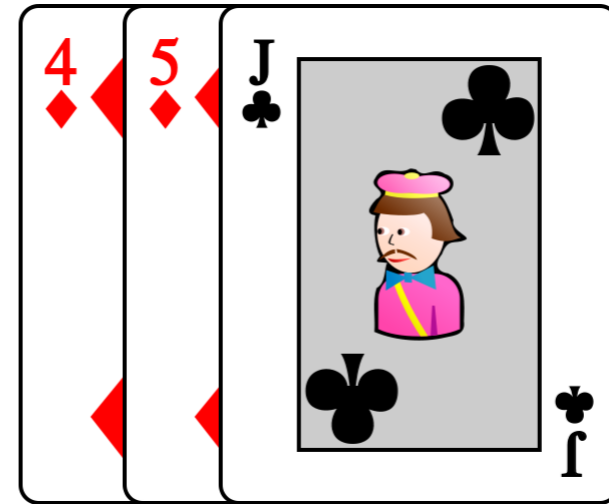
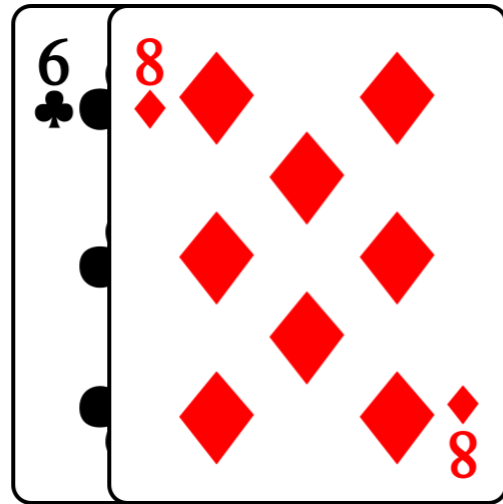
Fusion de tableaux triés



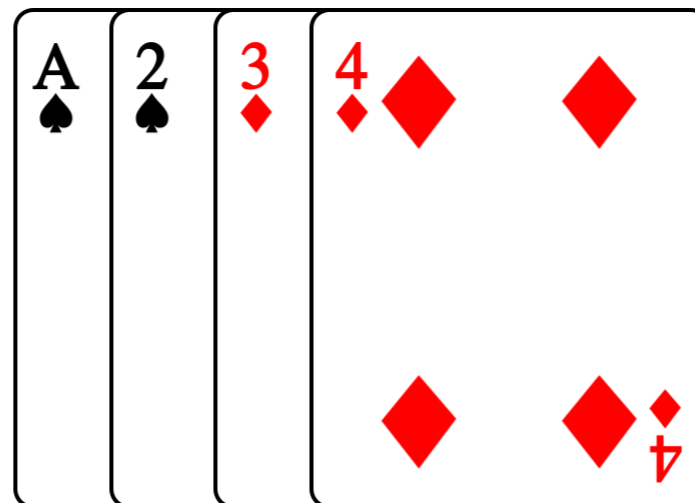
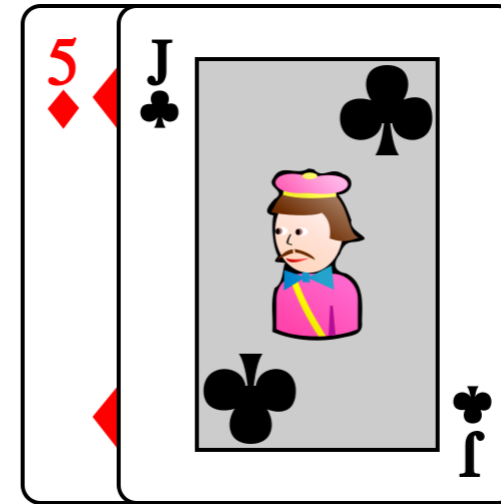
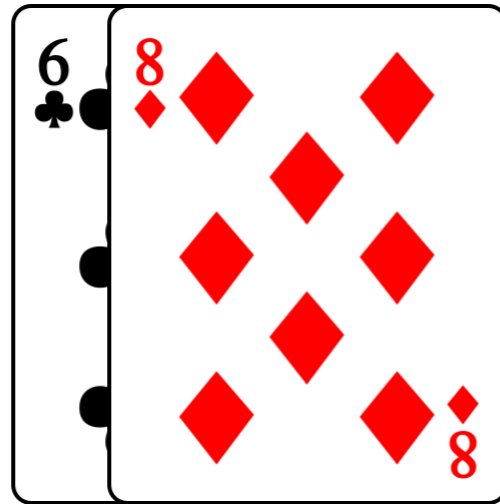
Fusion de tableaux triés



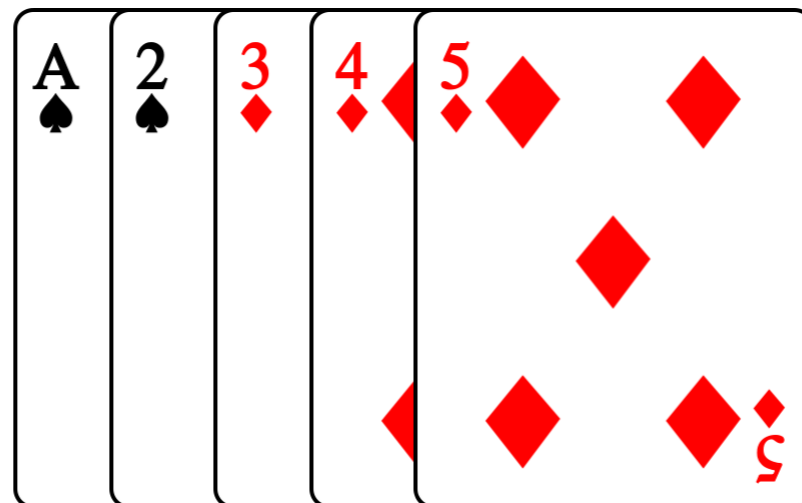
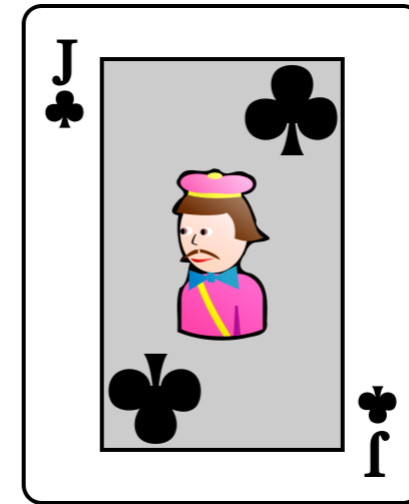
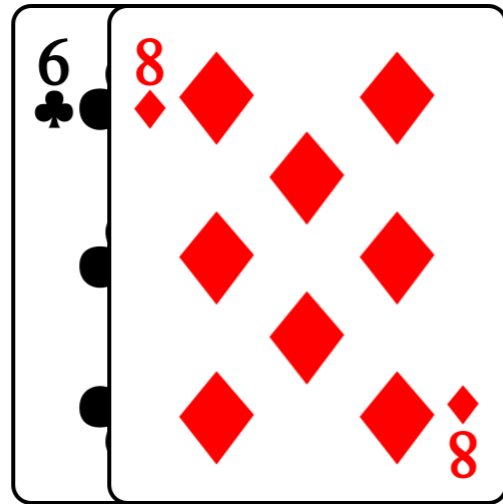
Fusion de tableaux triés



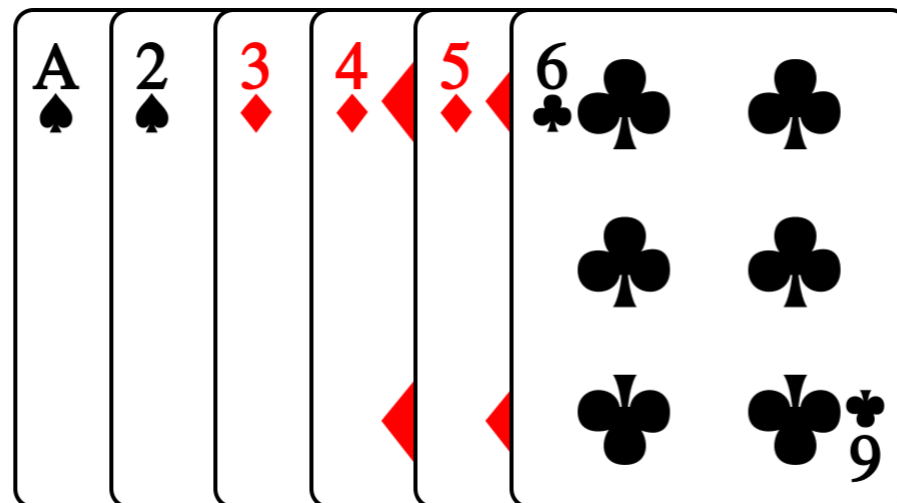
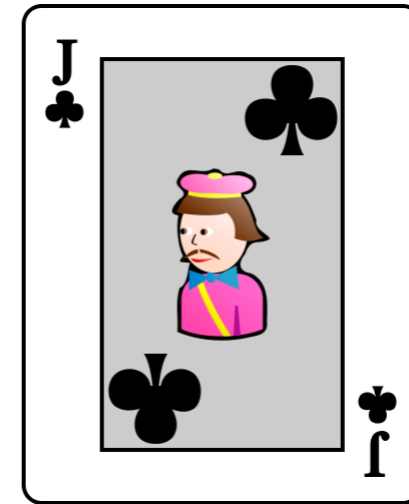
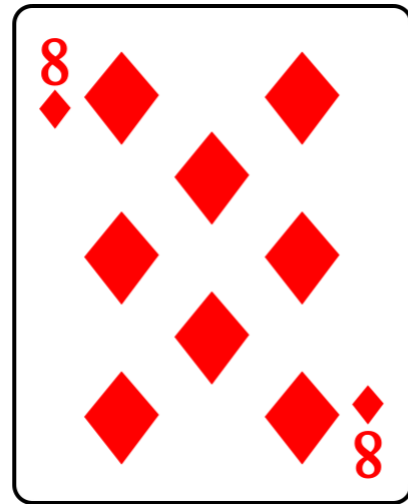
Fusion de tableaux triés



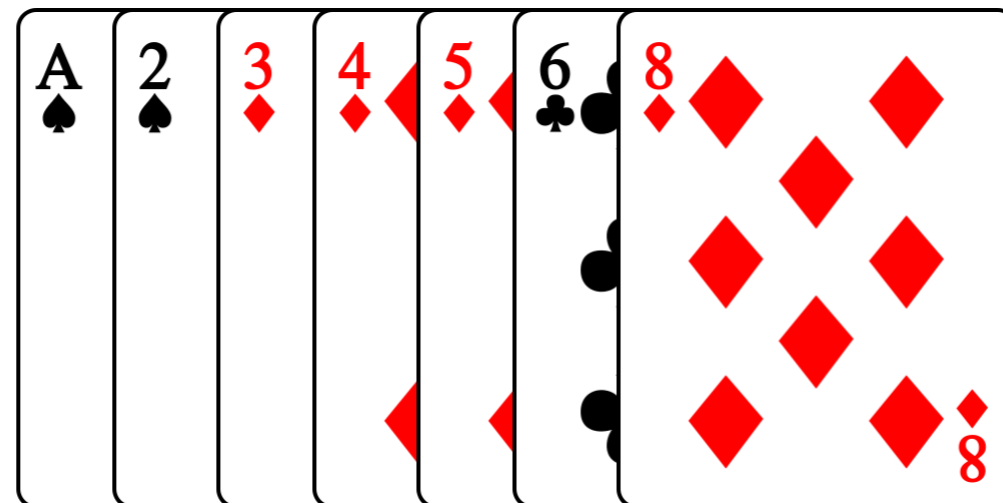
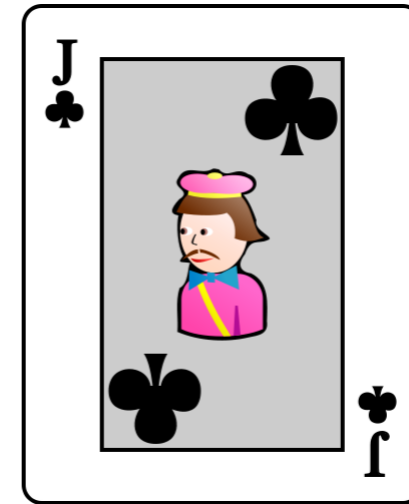
Fusion de tableaux triés



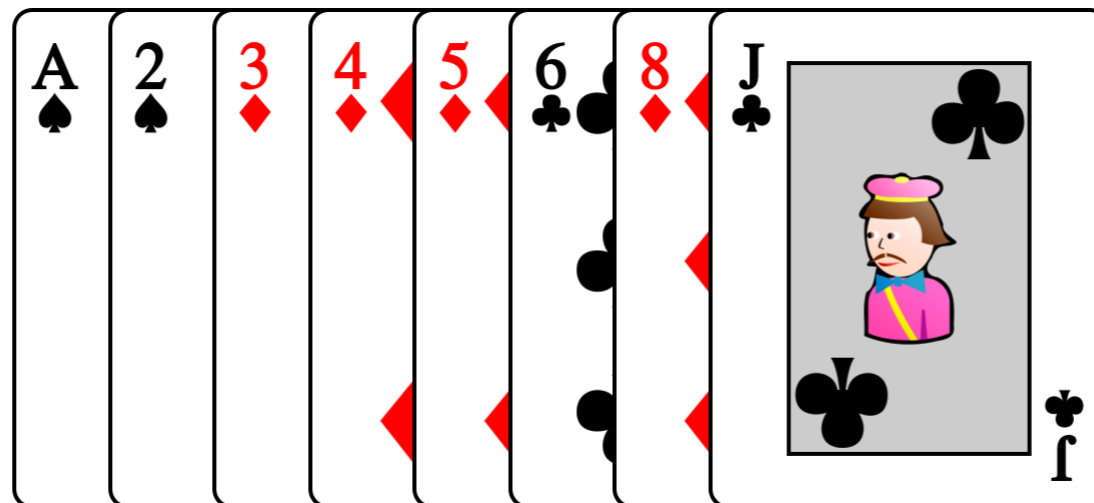
Fusion de tableaux triés



Fusion de tableaux triés

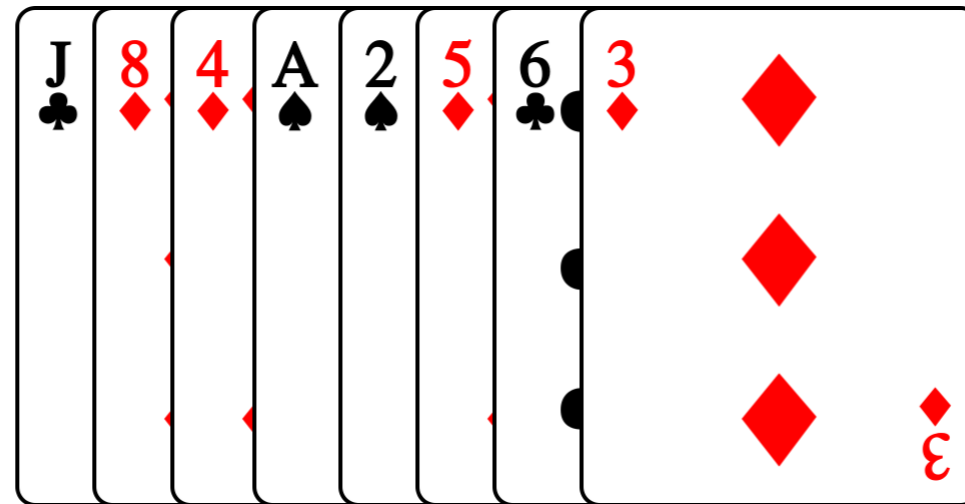


Fusion de tableaux triés

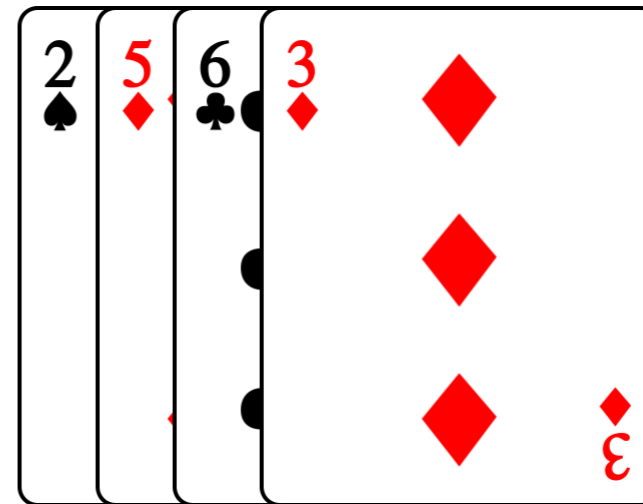
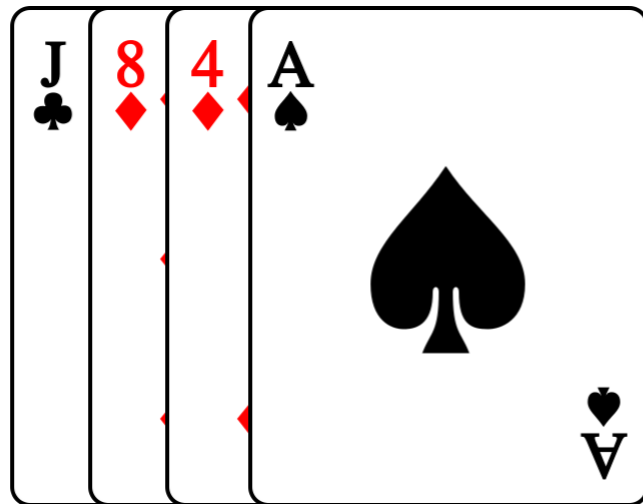


**Diviser pour
mieux régner !**

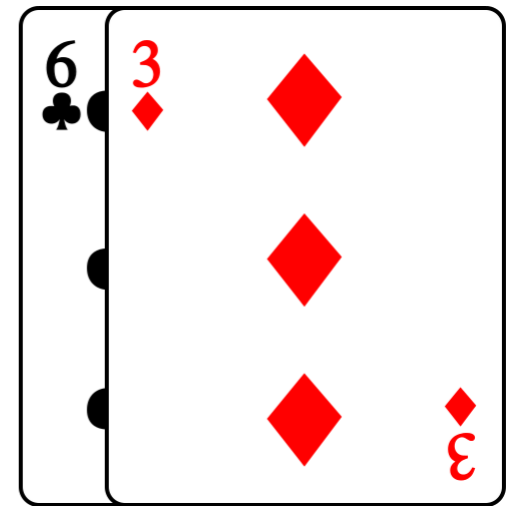
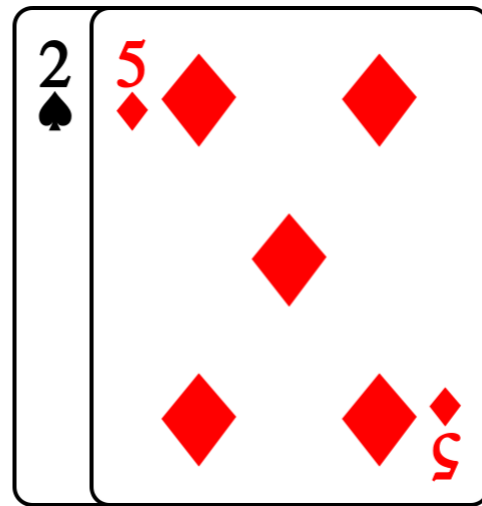
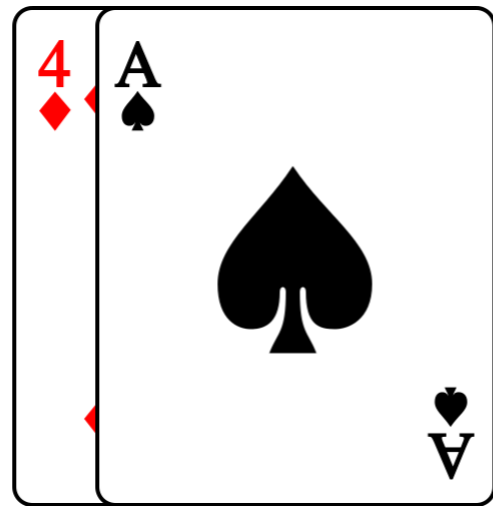
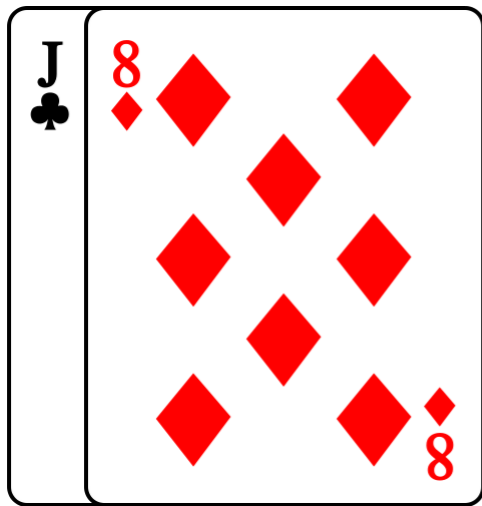
Tri fusion



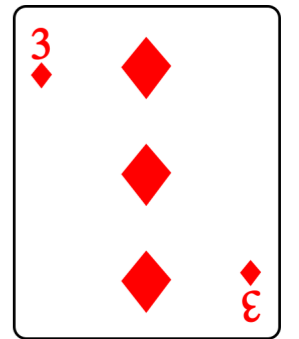
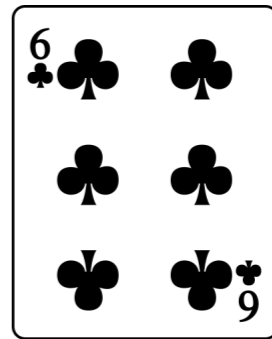
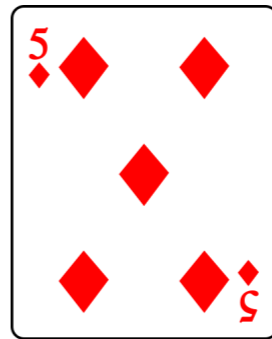
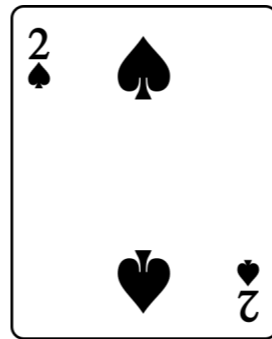
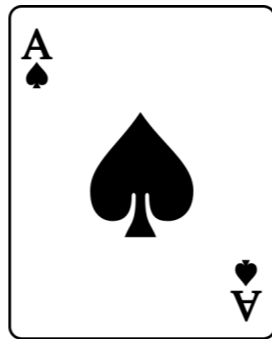
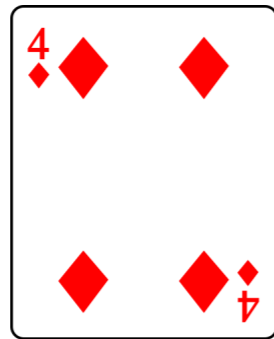
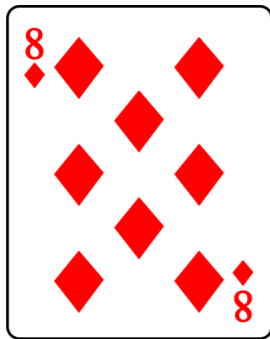
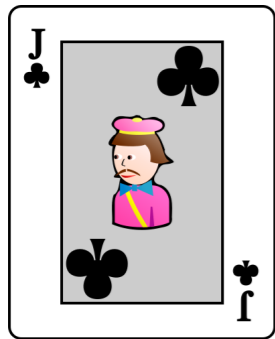
Diviser



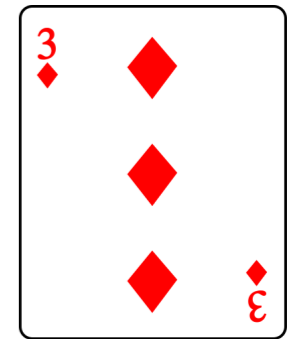
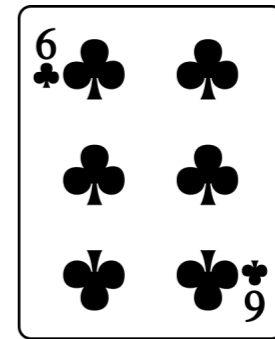
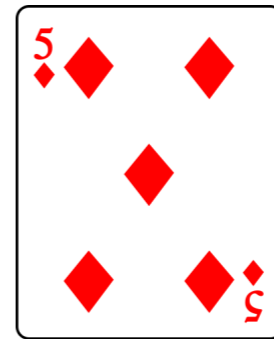
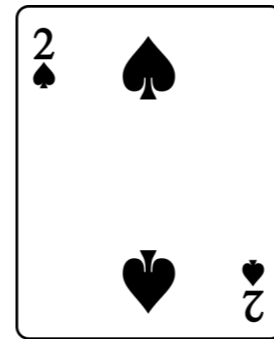
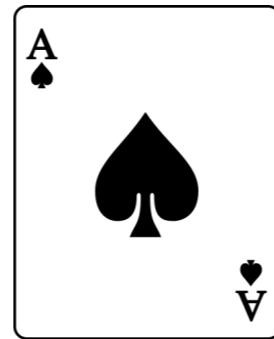
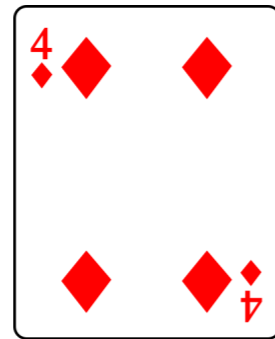
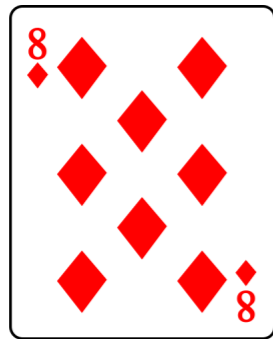
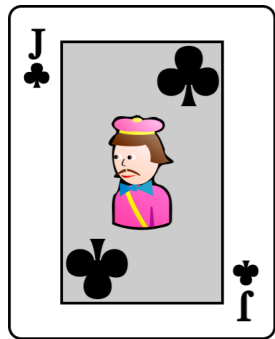
Diviser



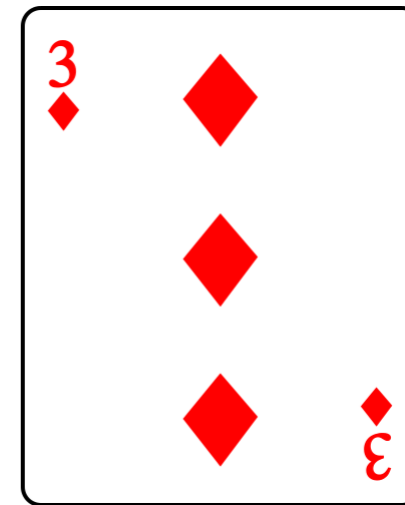
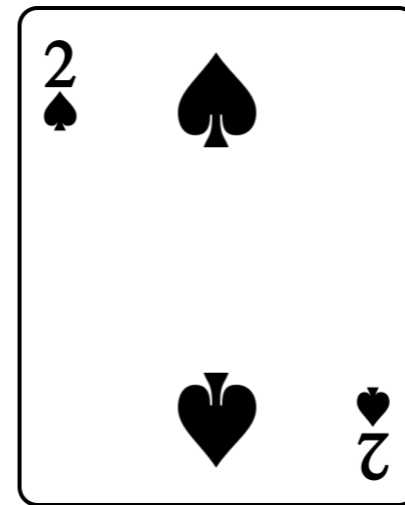
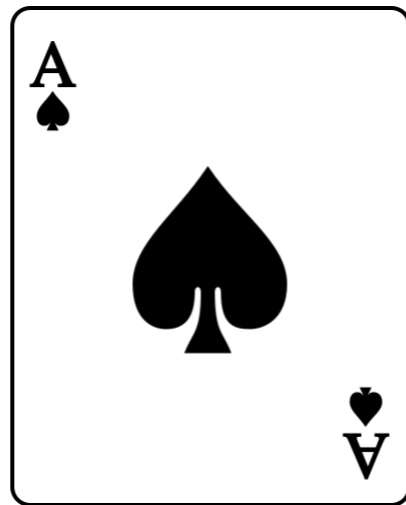
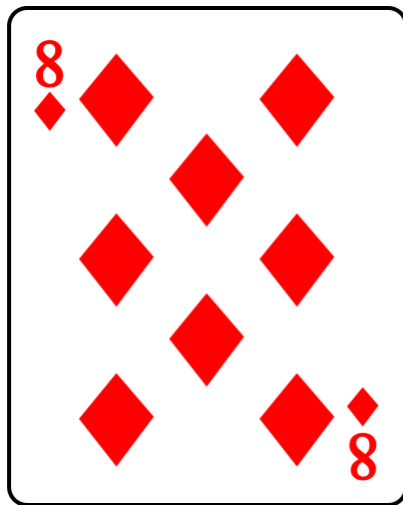
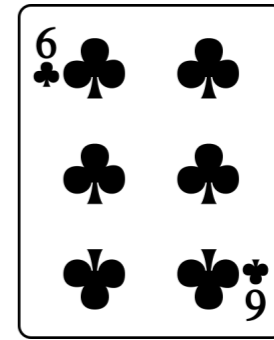
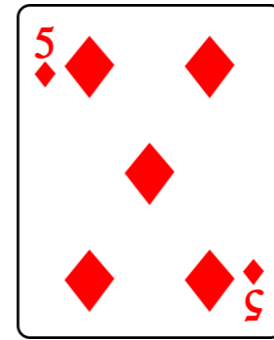
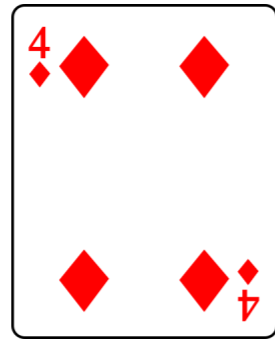
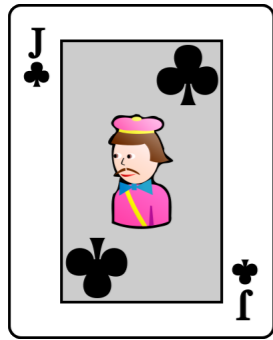
Diviser



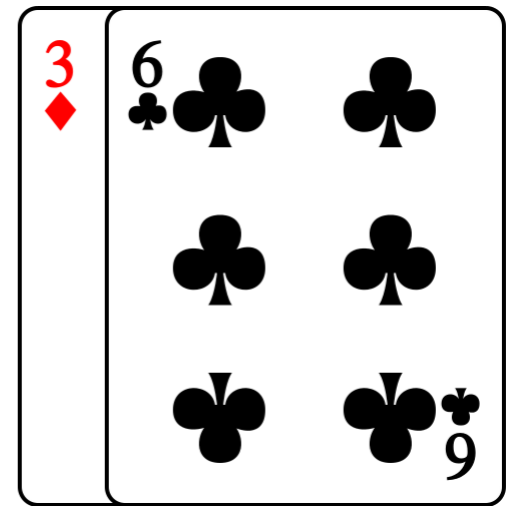
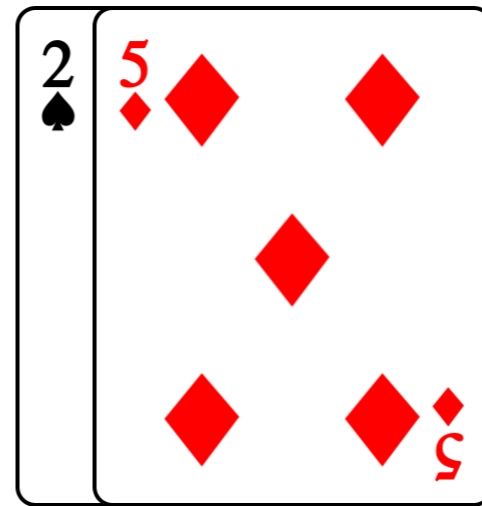
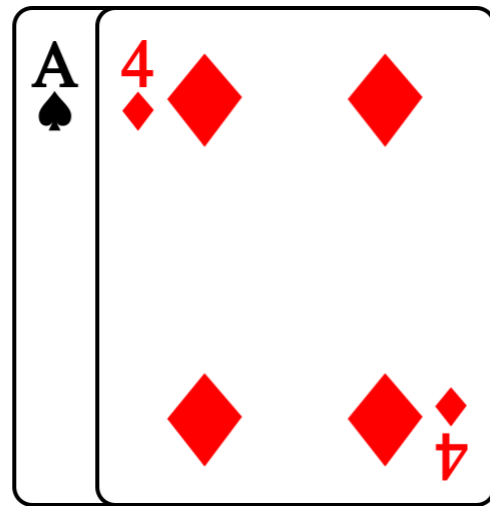
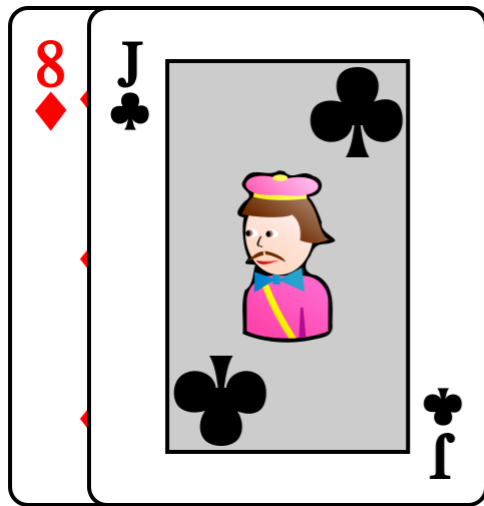
Tous les jeux sont triés !



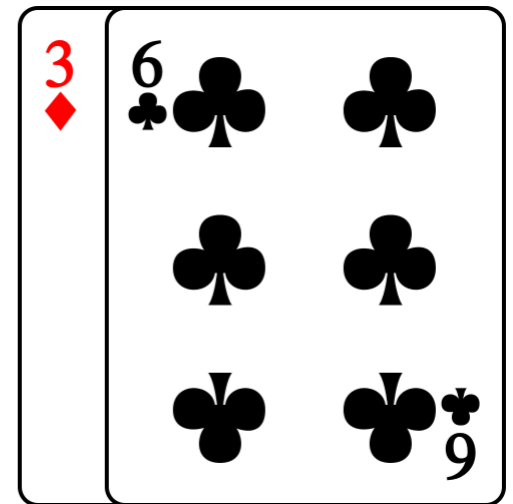
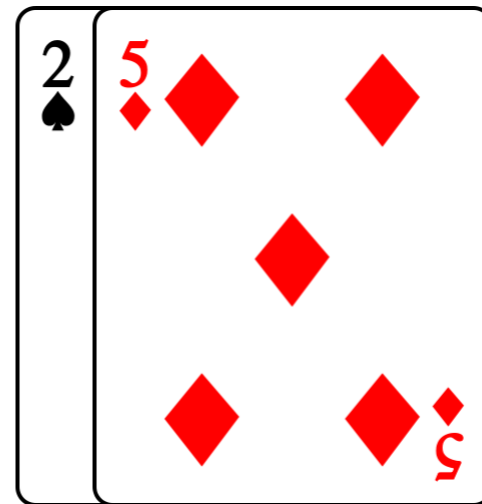
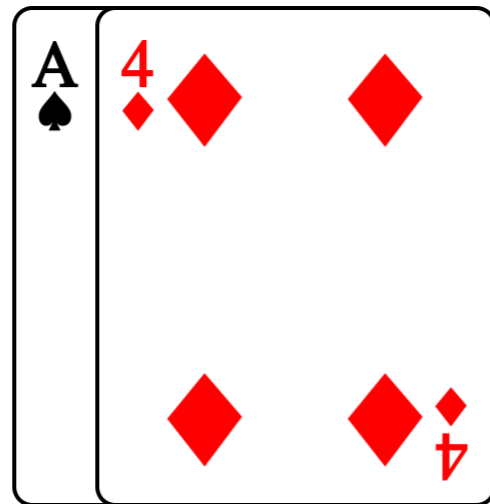
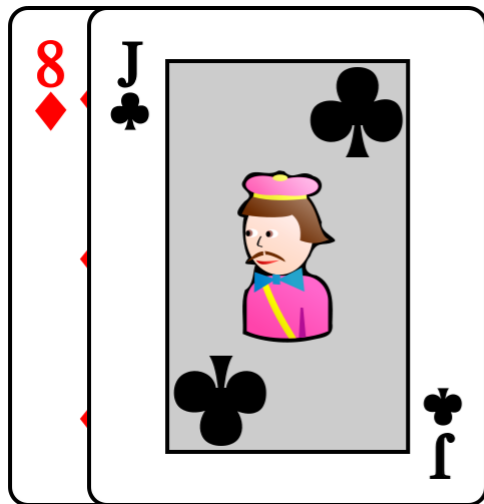
Fusionner



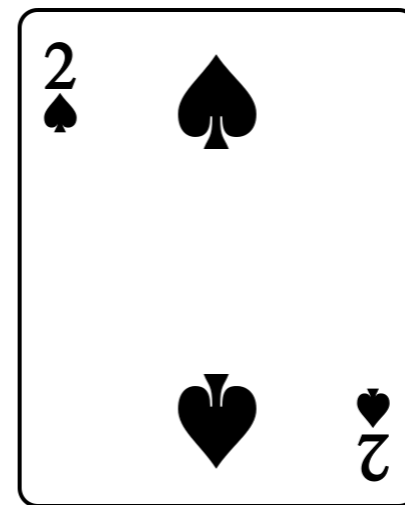
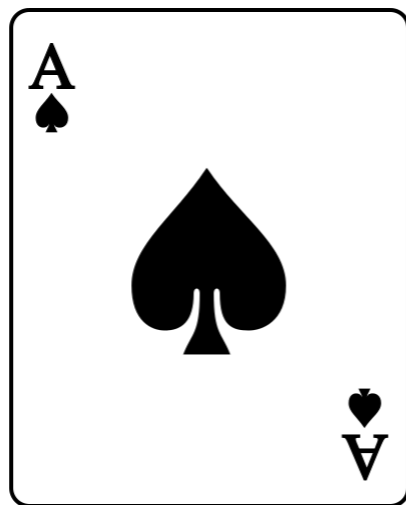
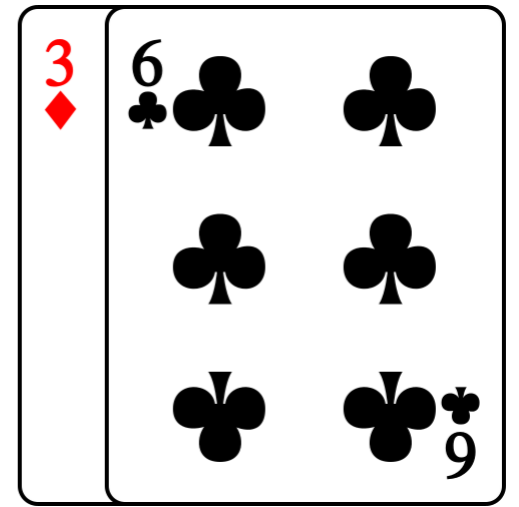
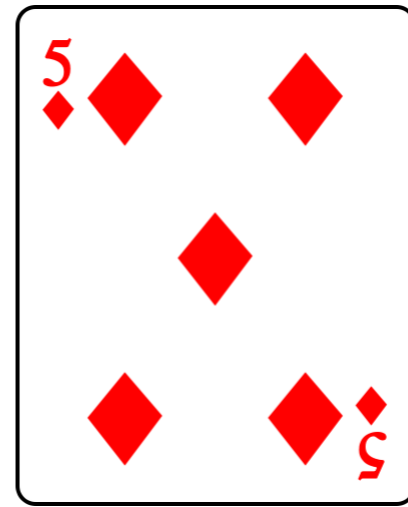
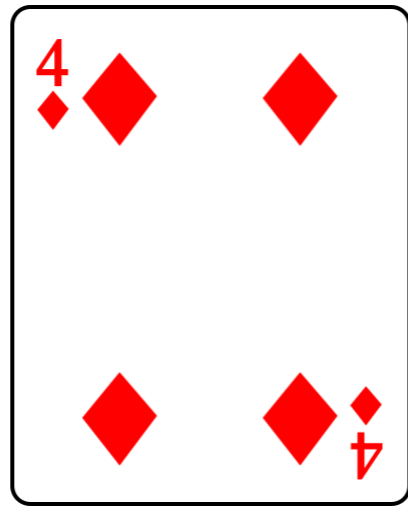
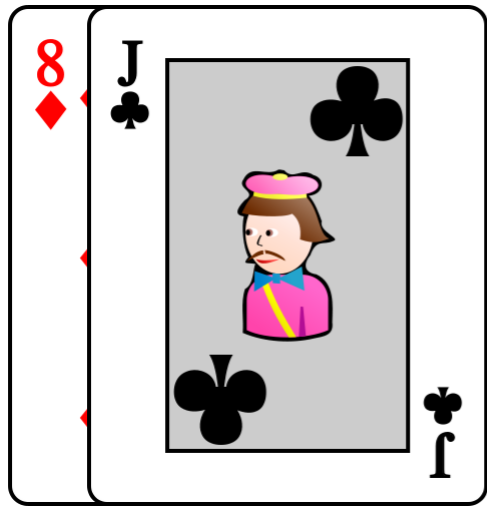
Fusionner



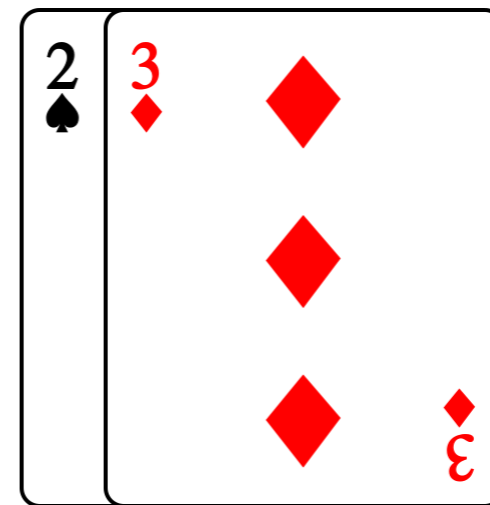
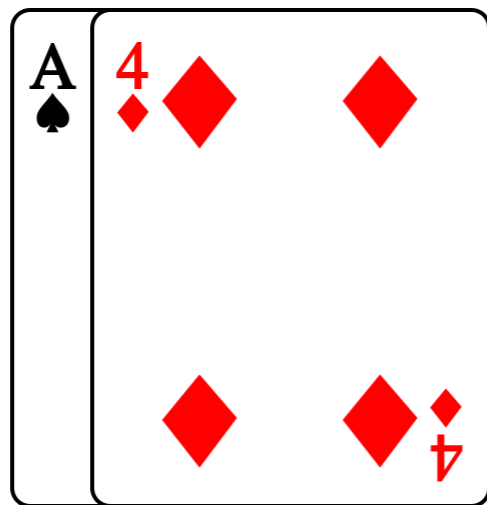
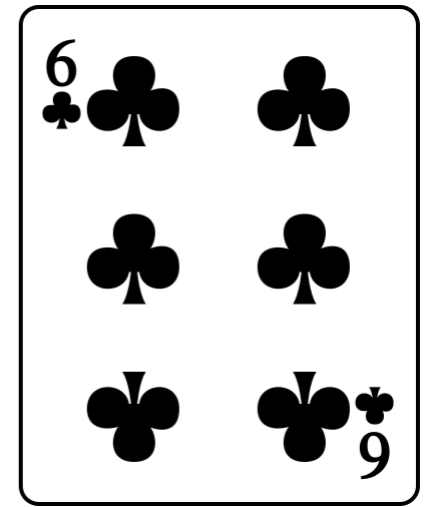
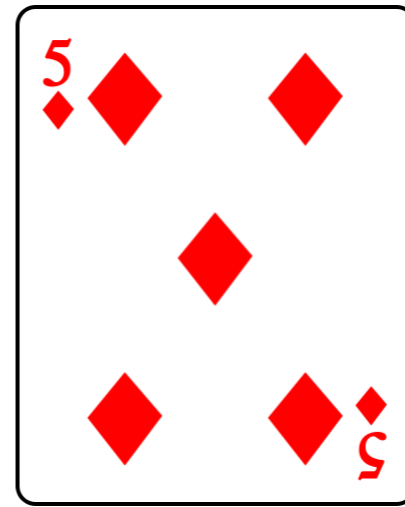
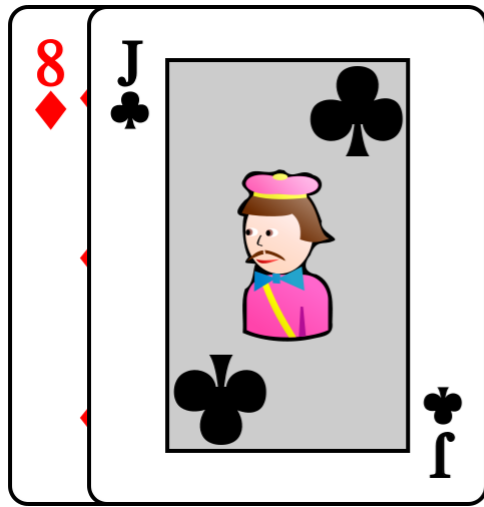
Tous les jeux sont triés !



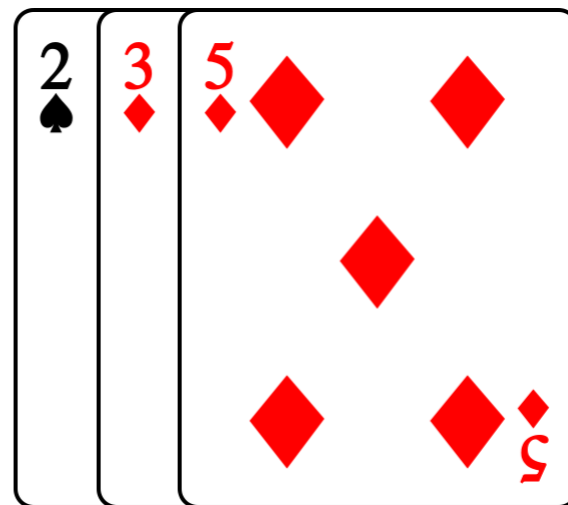
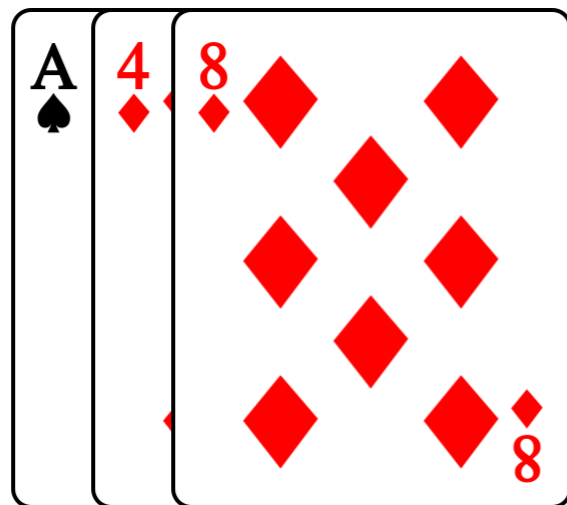
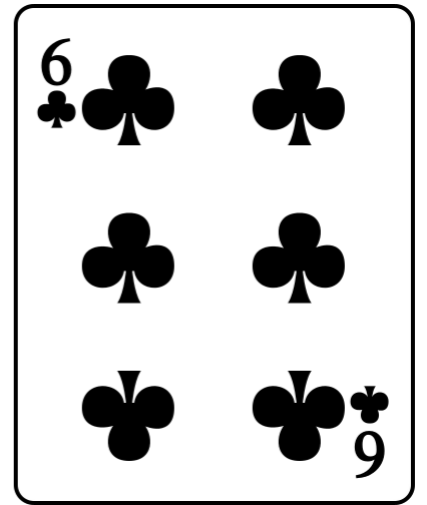
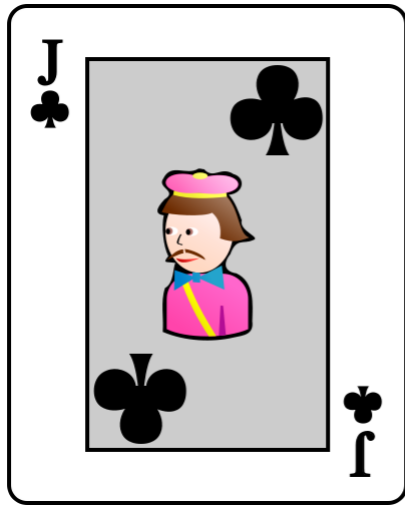
Fusionner



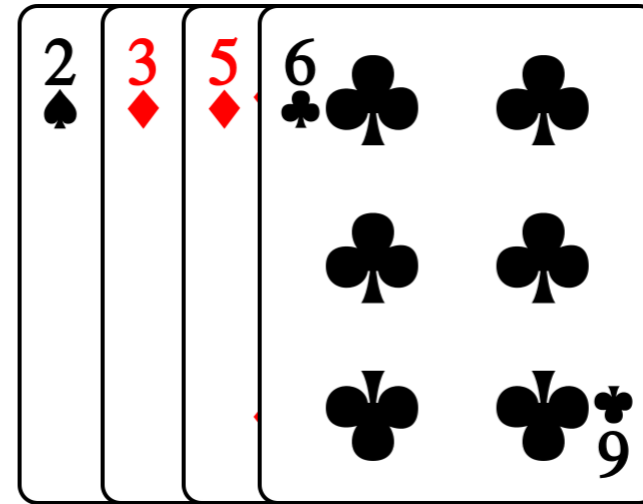
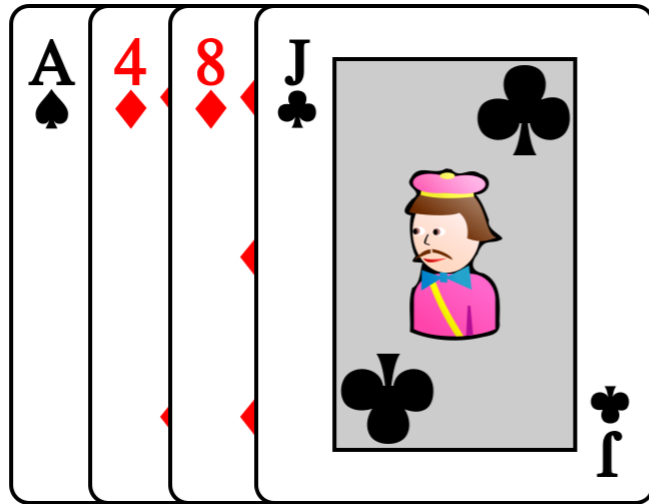
Fusionner



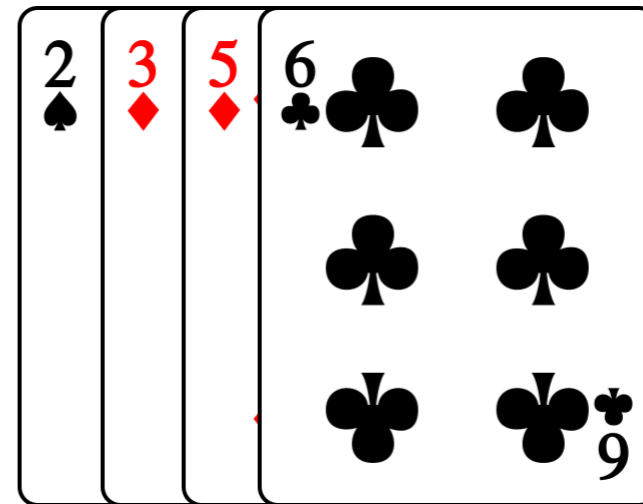
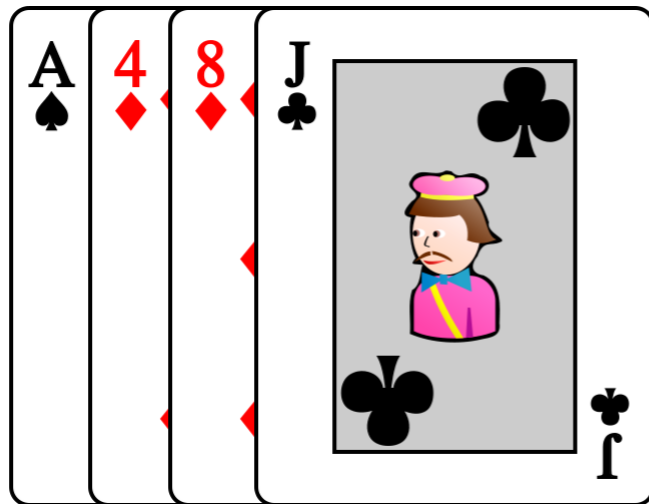
Fusionner



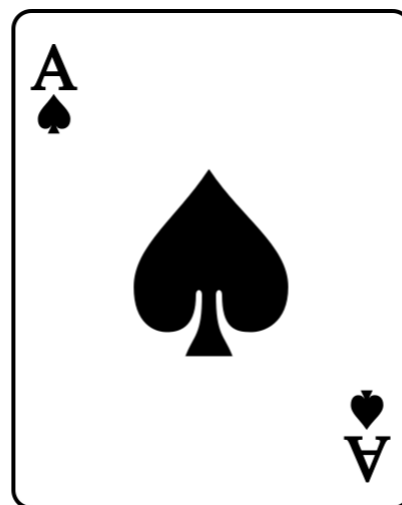
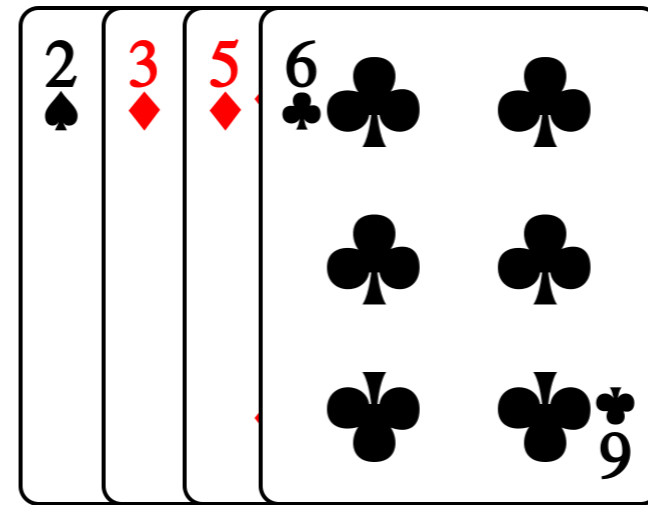
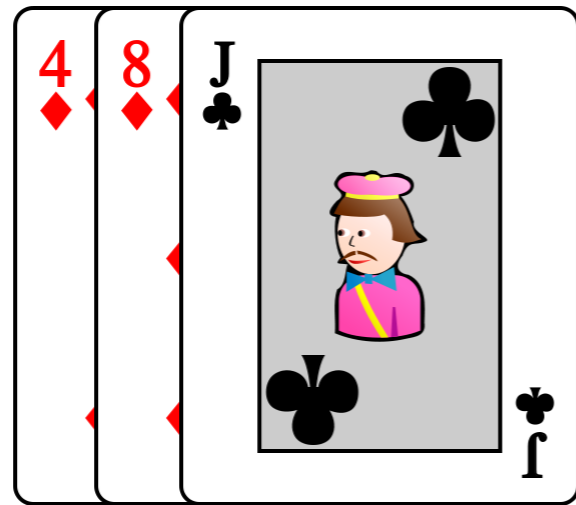
Fusionner



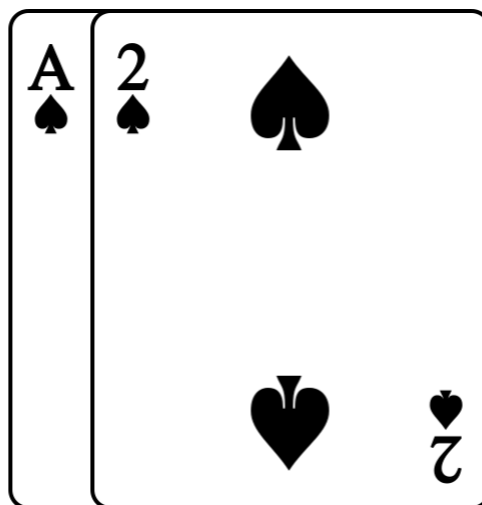
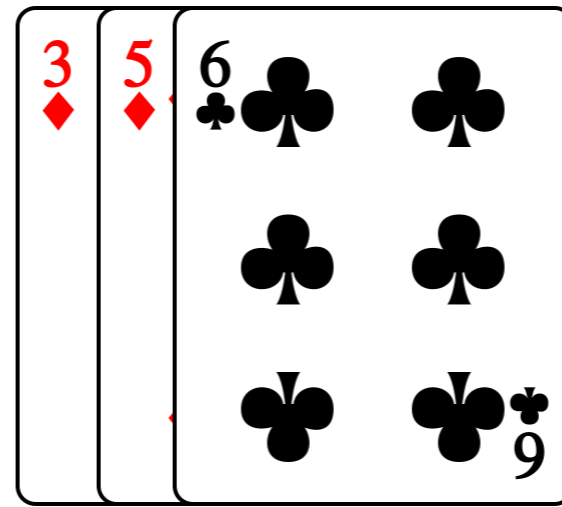
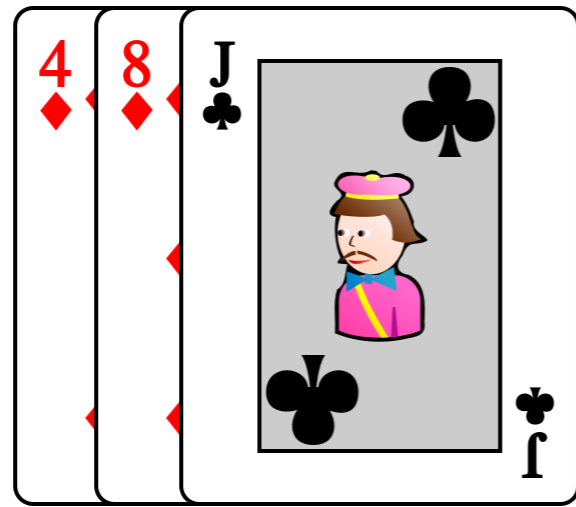
Tous les jeux sont triés !



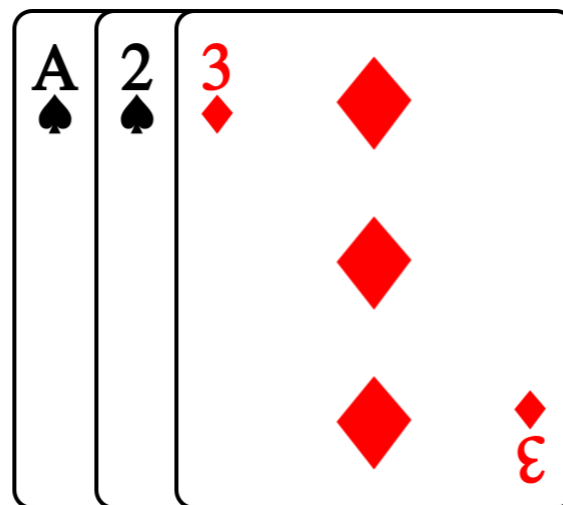
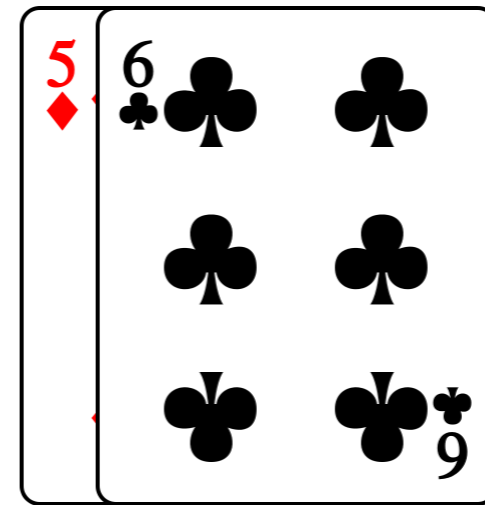
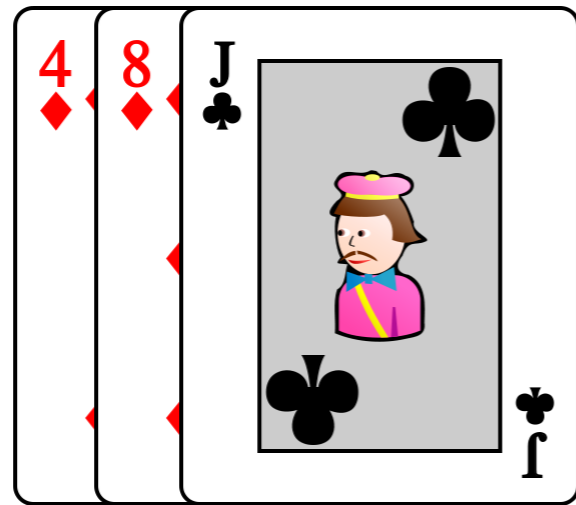
Fusionner



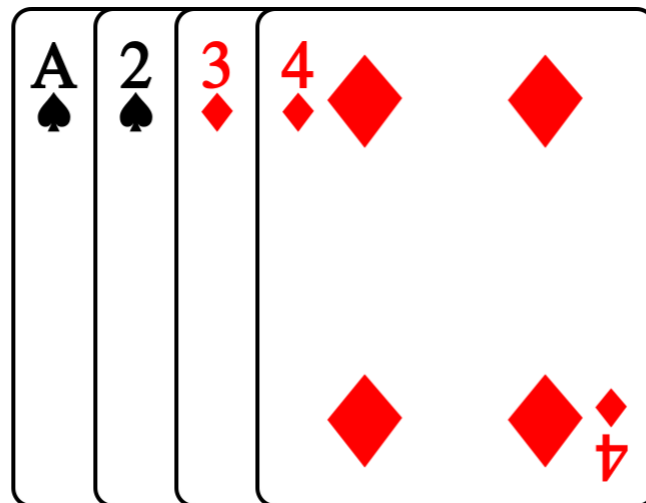
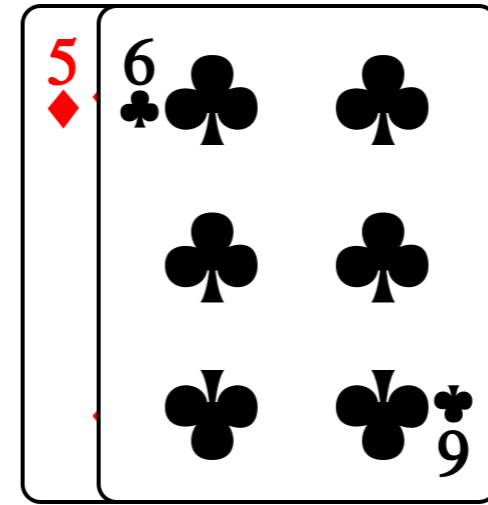
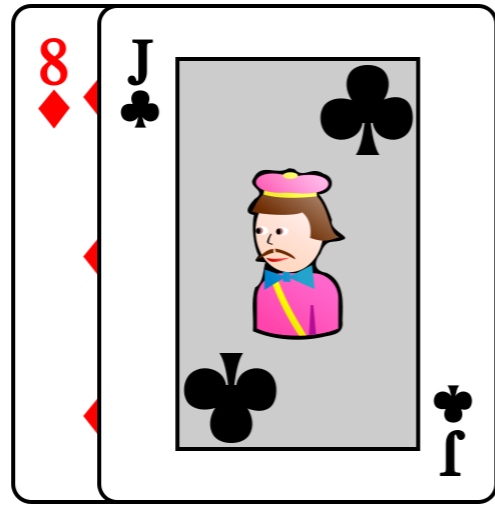
Fusionner



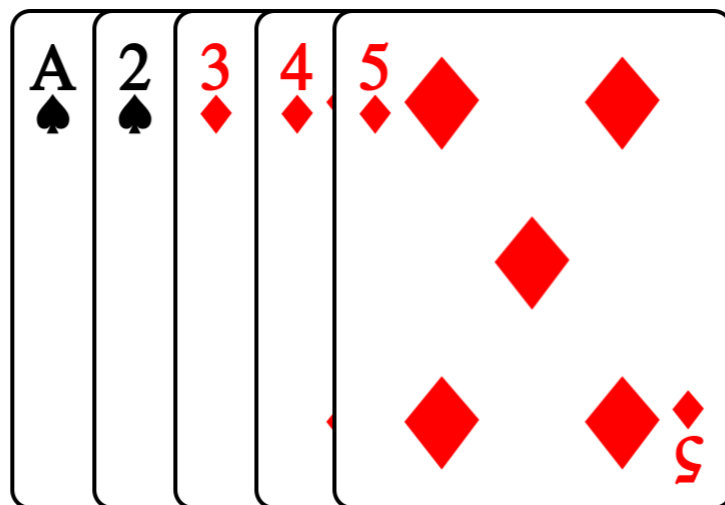
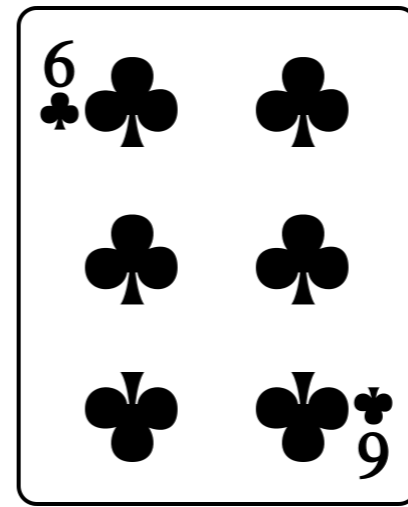
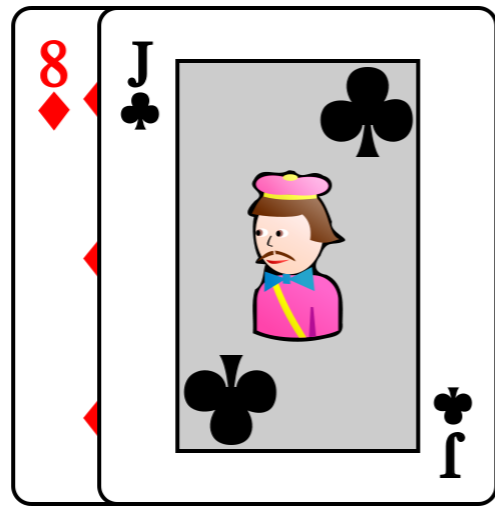
Fusionner



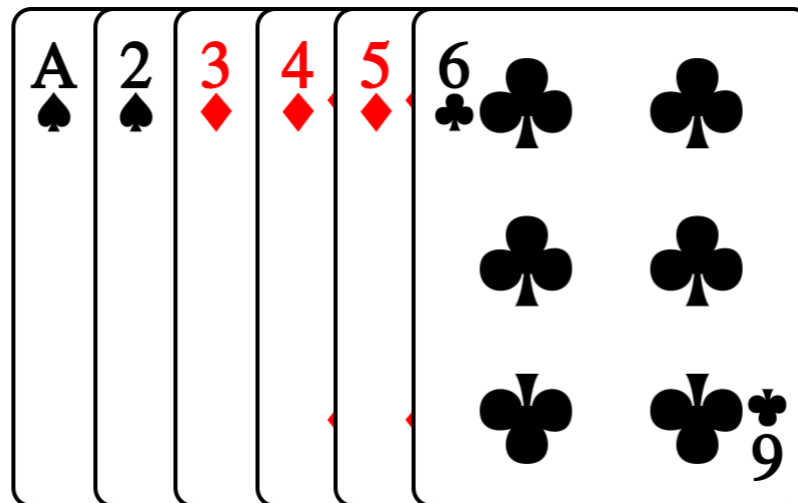
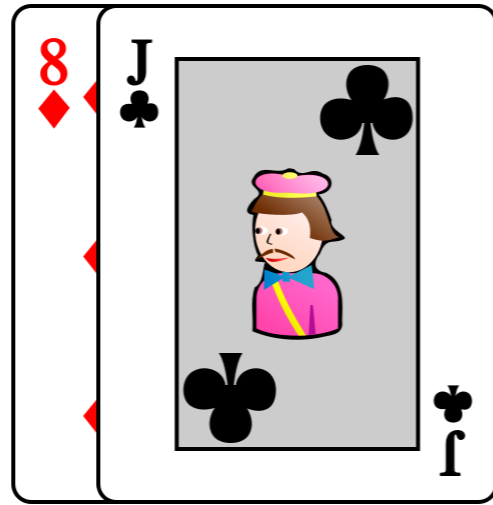
Fusionner



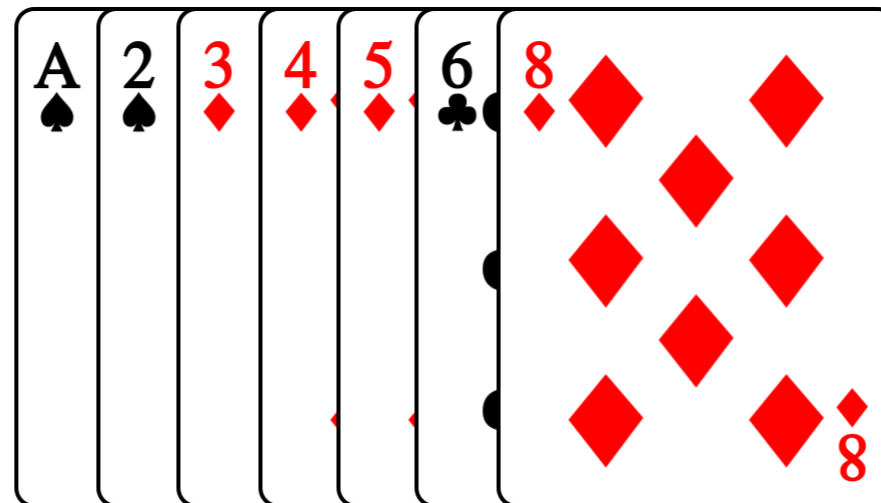
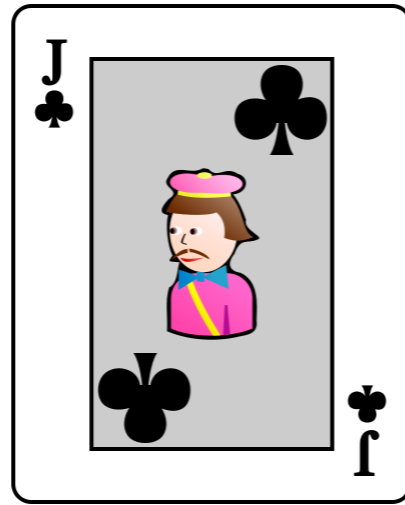
Fusionner



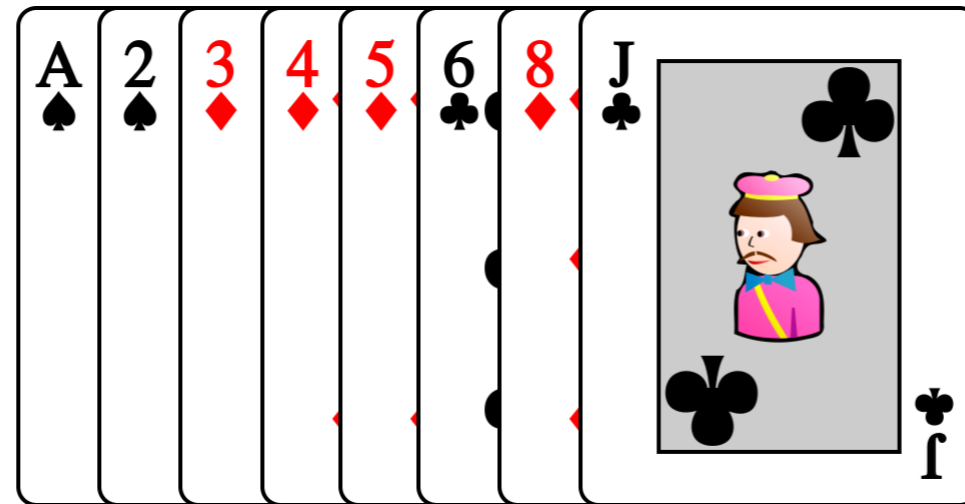
Fusionner



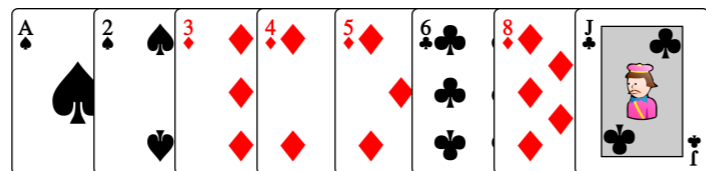
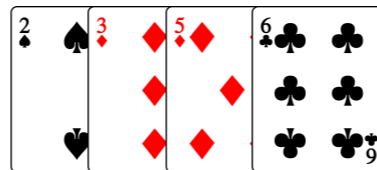
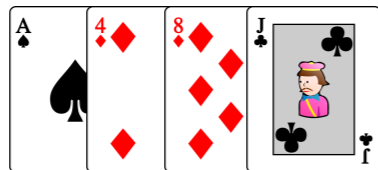
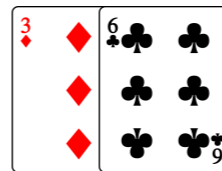
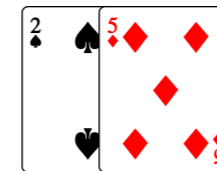
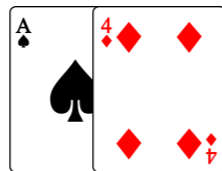
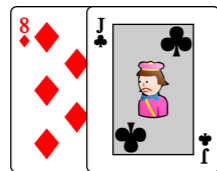
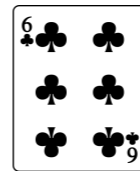
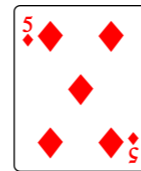
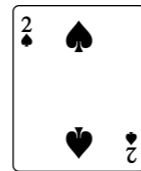
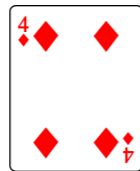
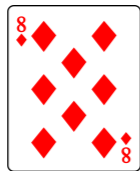
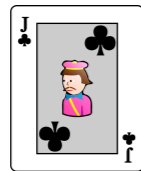
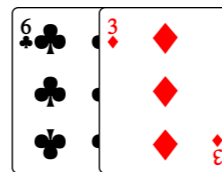
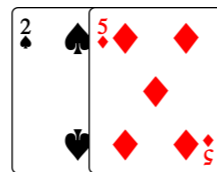
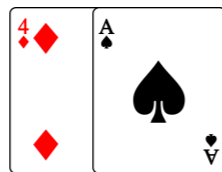
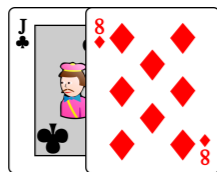
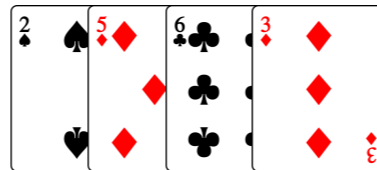
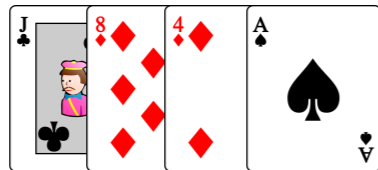
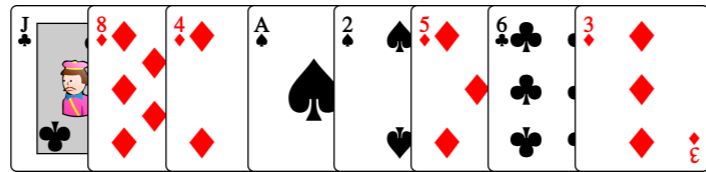
Fusionner



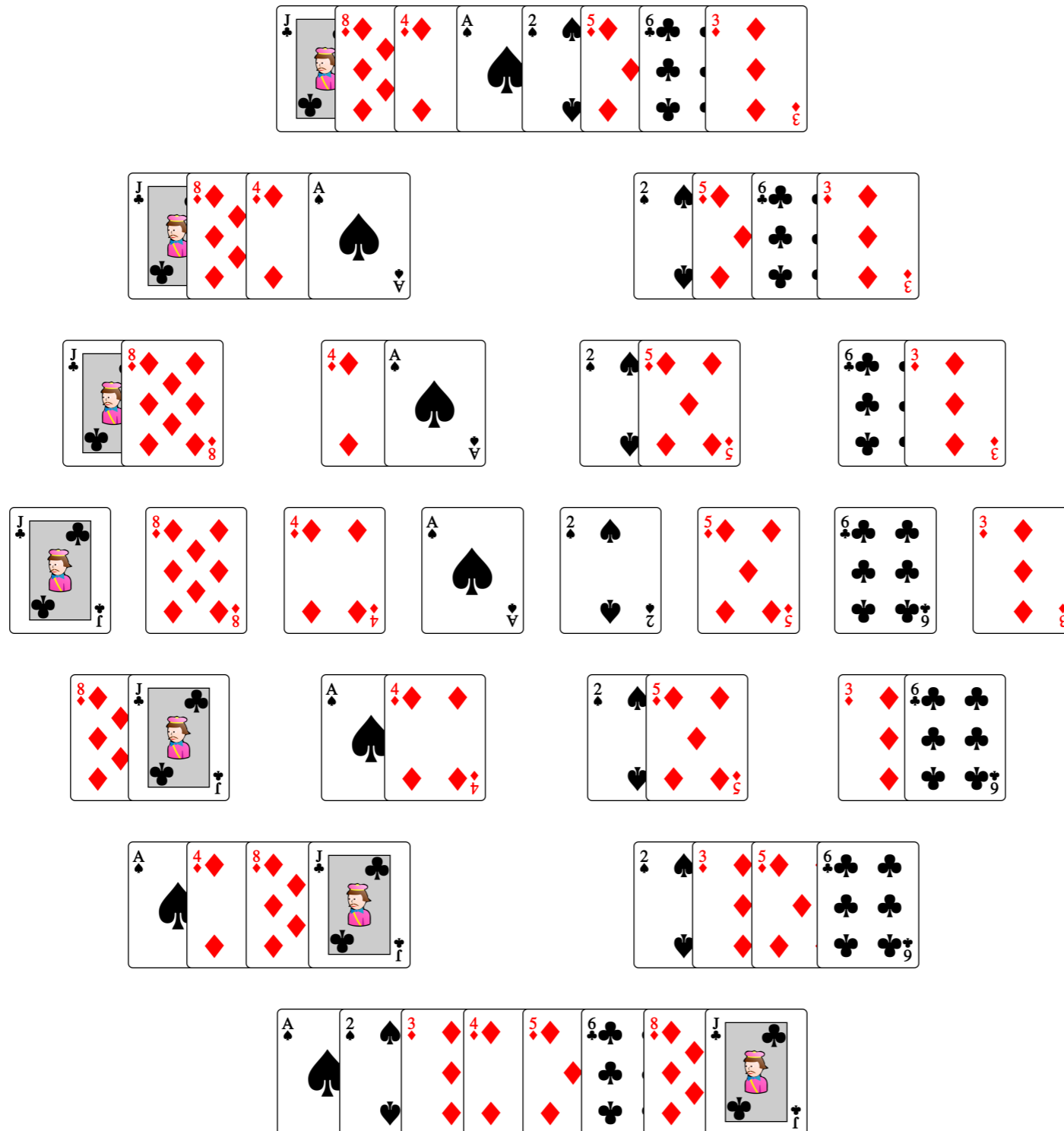
Le jeu est trié !



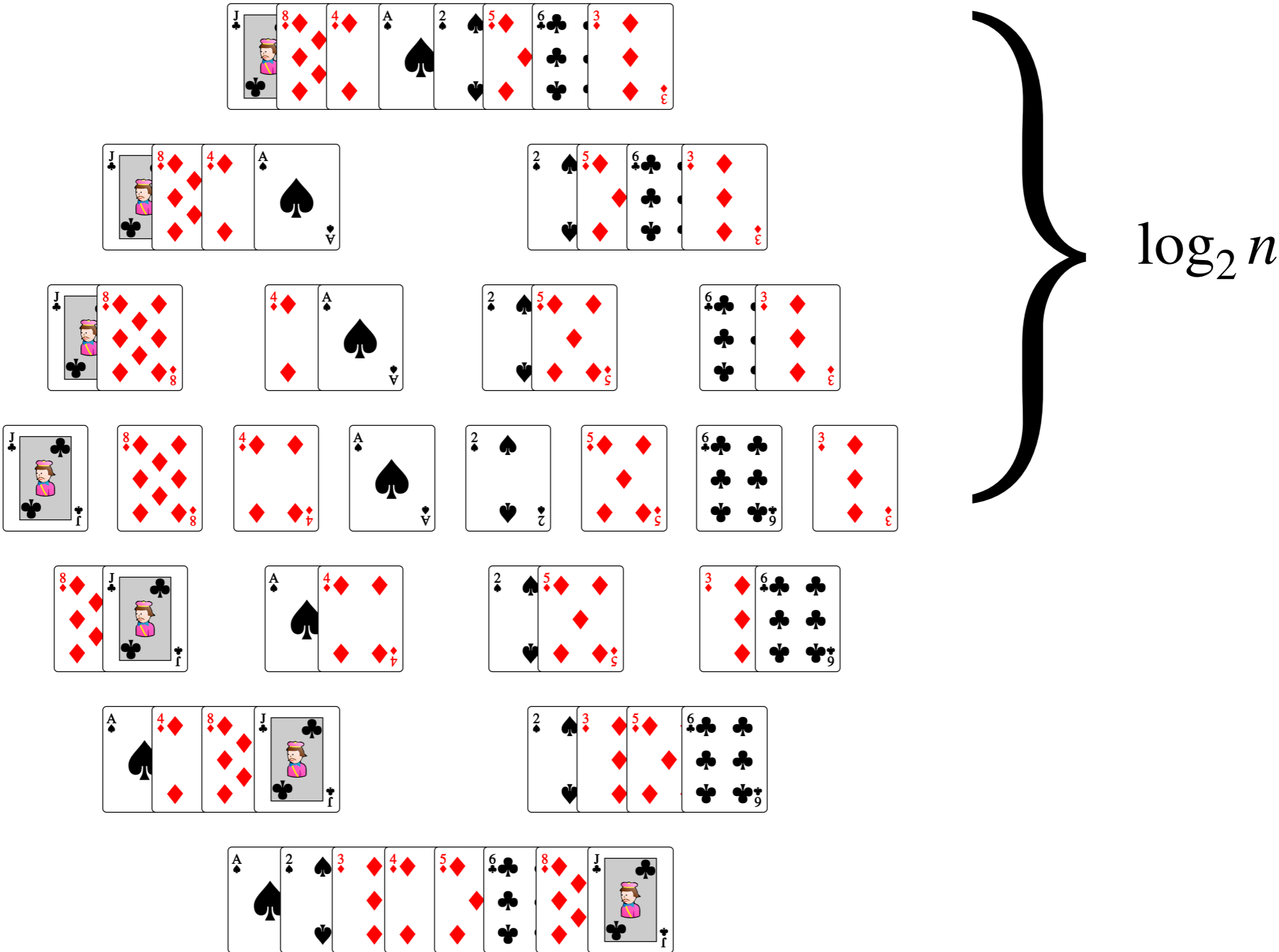
Efficacité du tri fusion



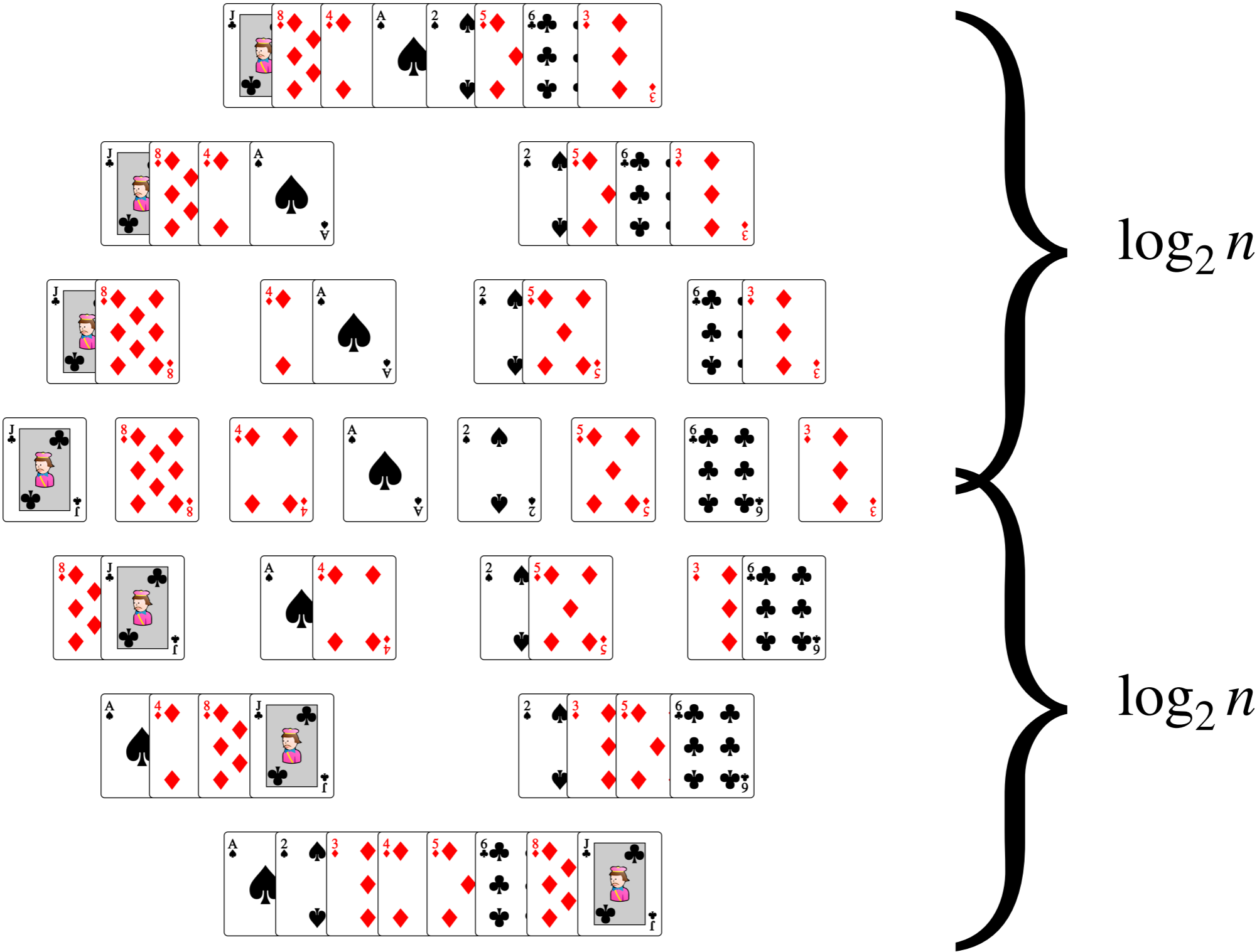
Efficacité du tri fusion



Efficacité du tri fusion

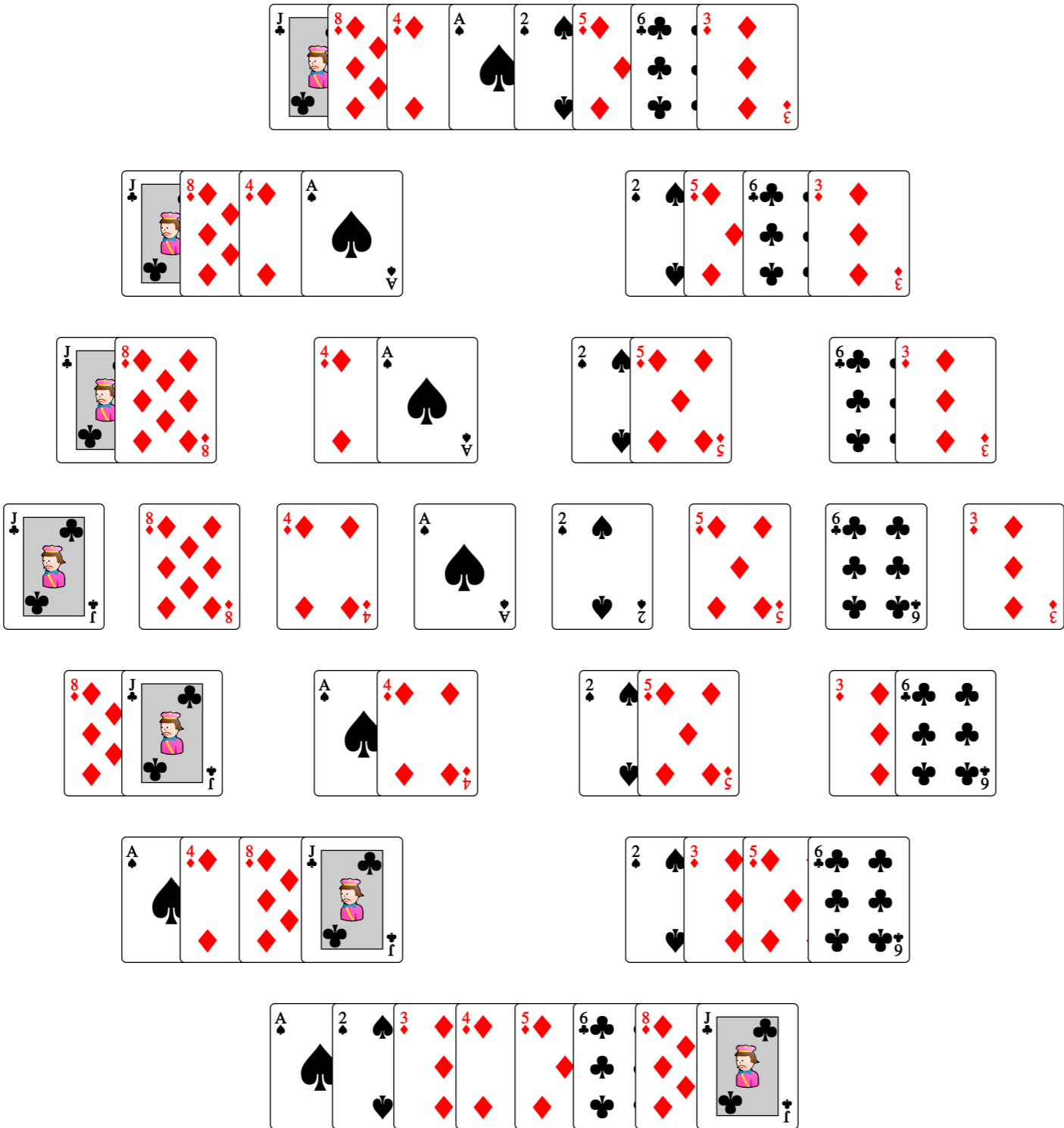


Efficacité du tri fusion



Efficacité du tri fusion

n

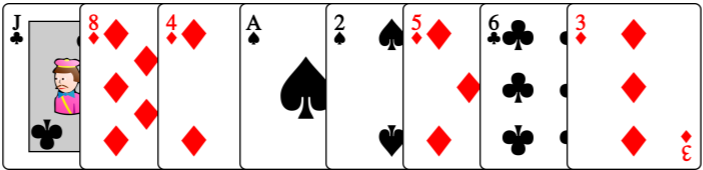


$\log_2 n$

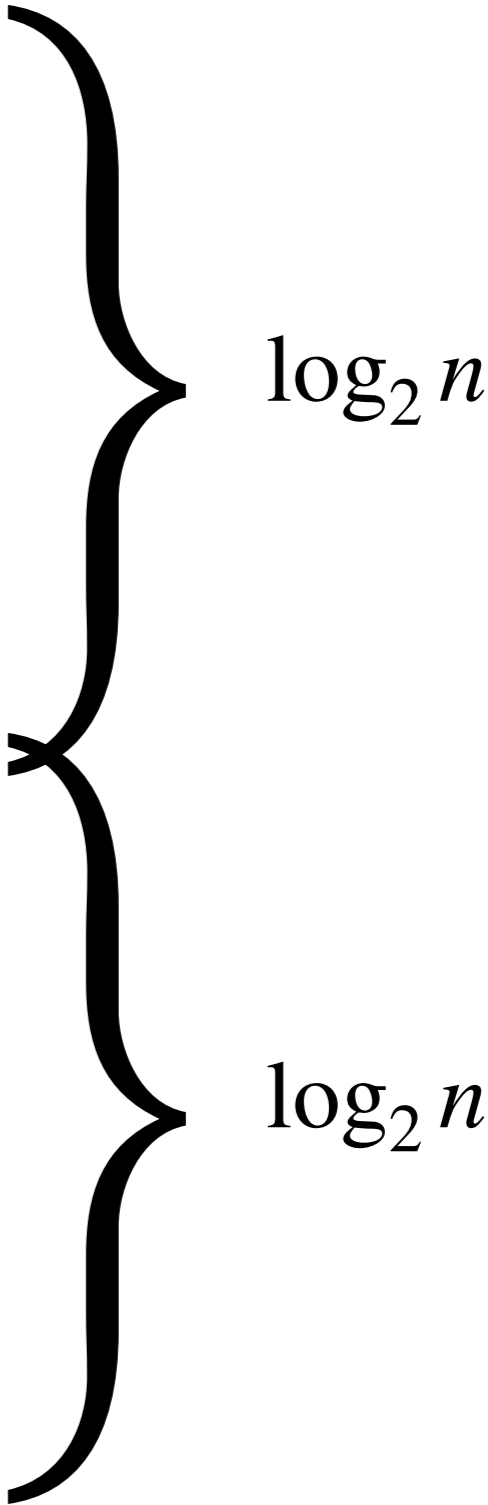
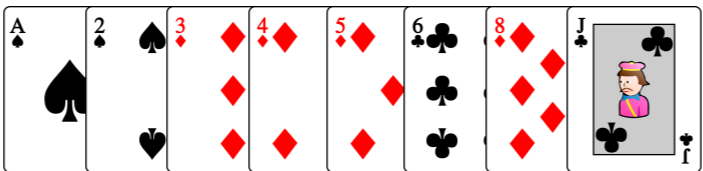
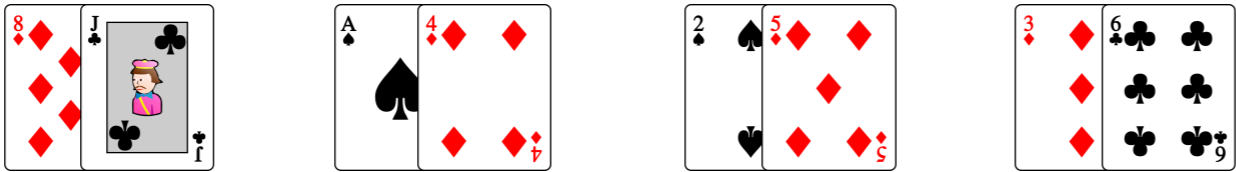
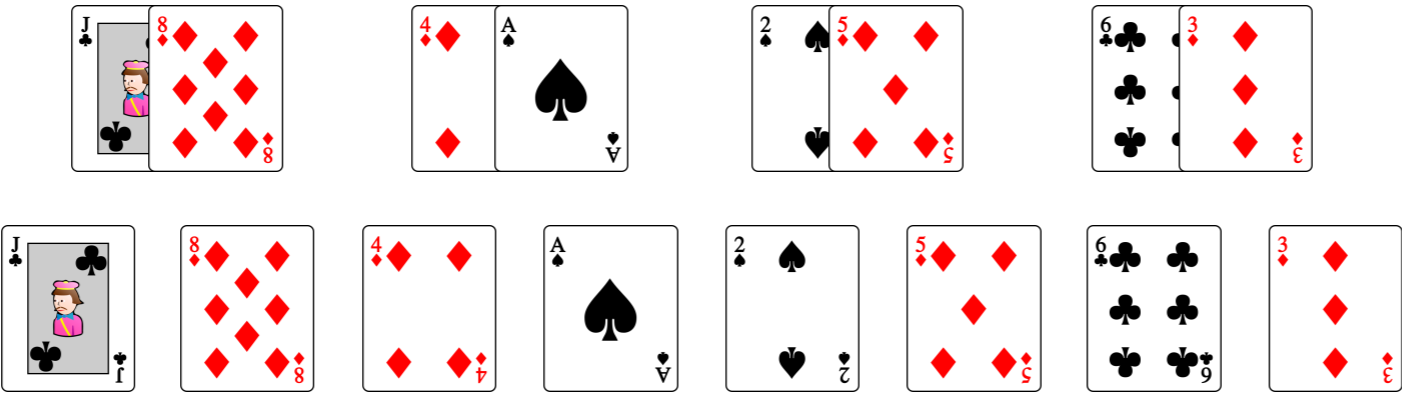
$\log_2 n$

Efficacité du tri fusion

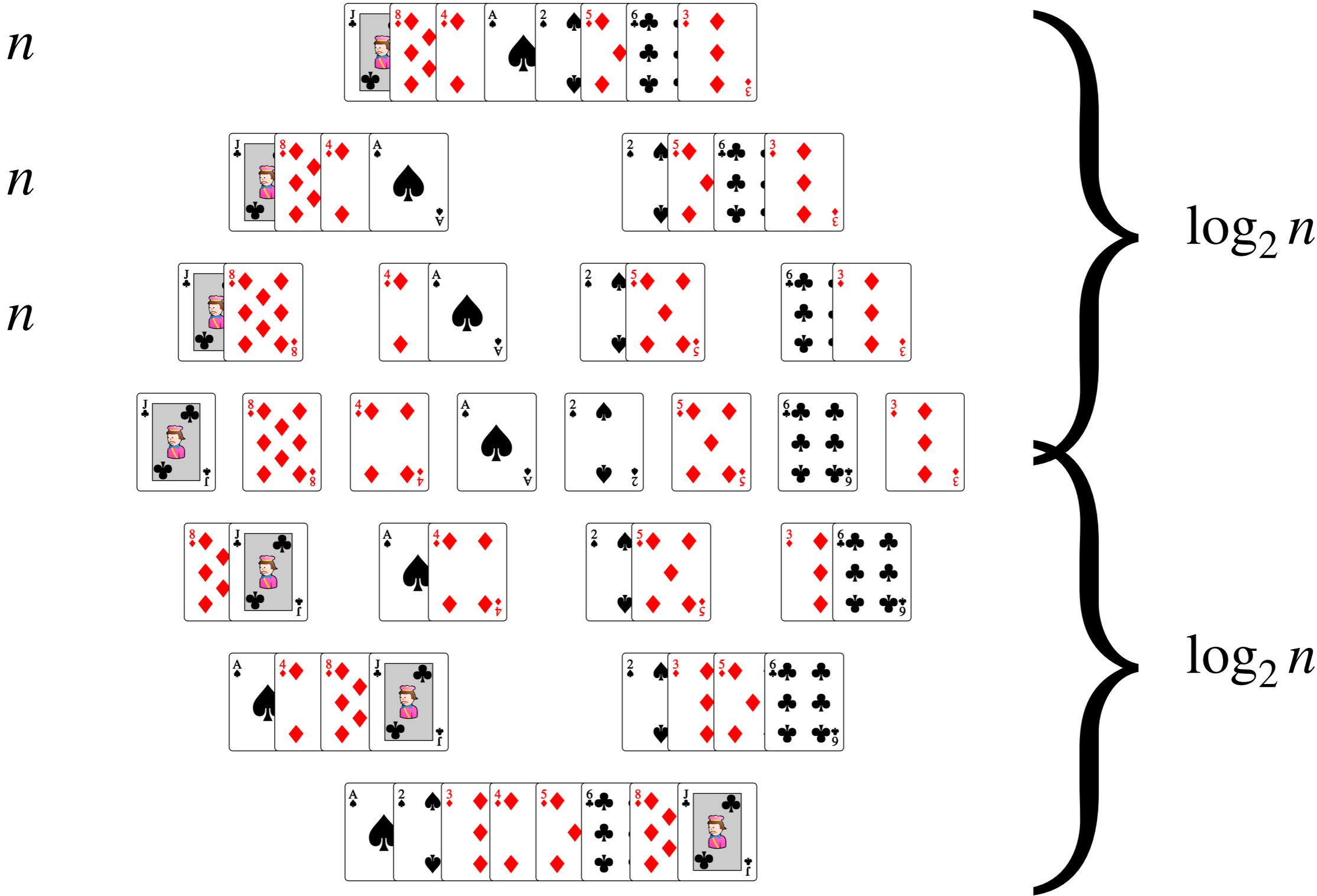
n



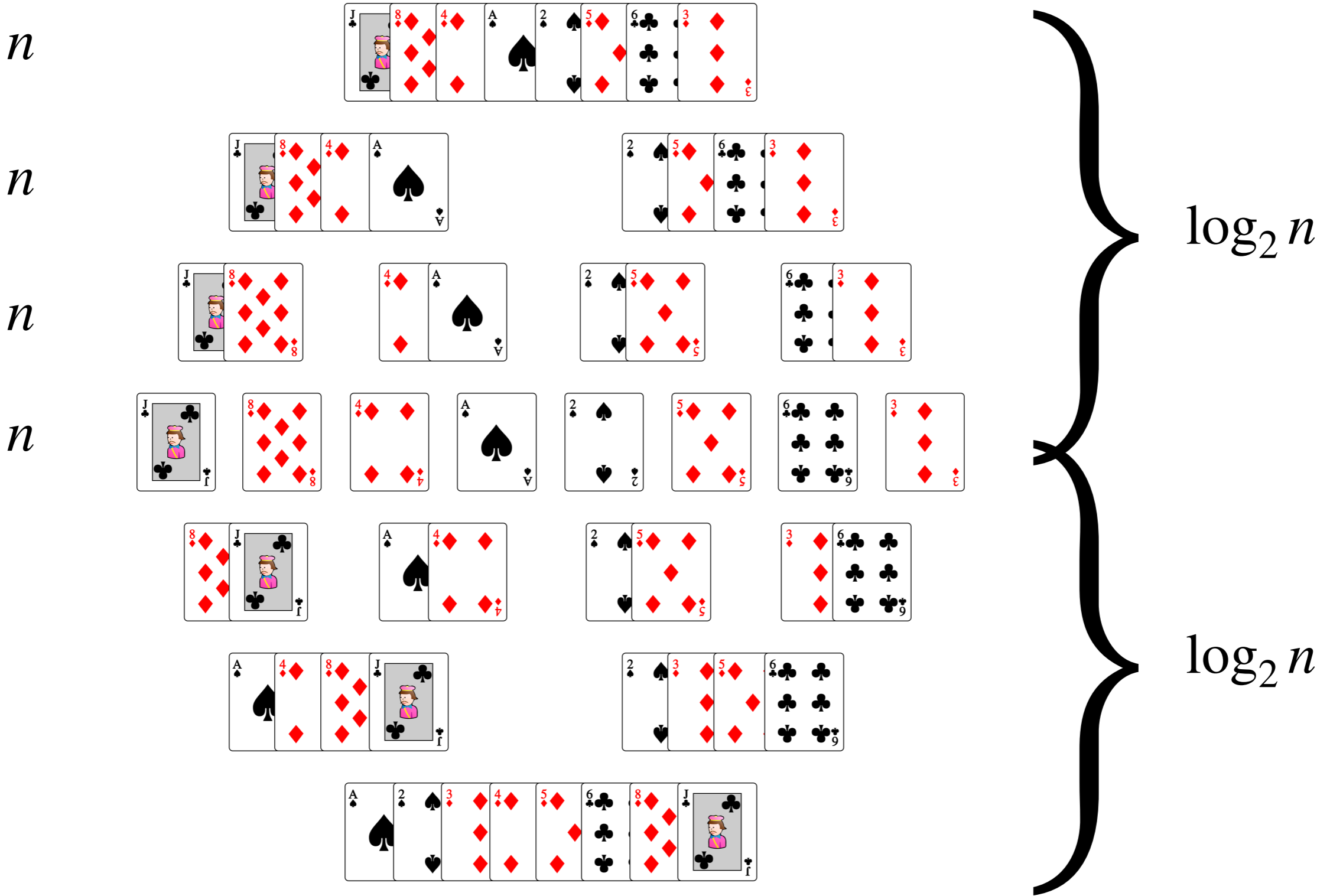
n



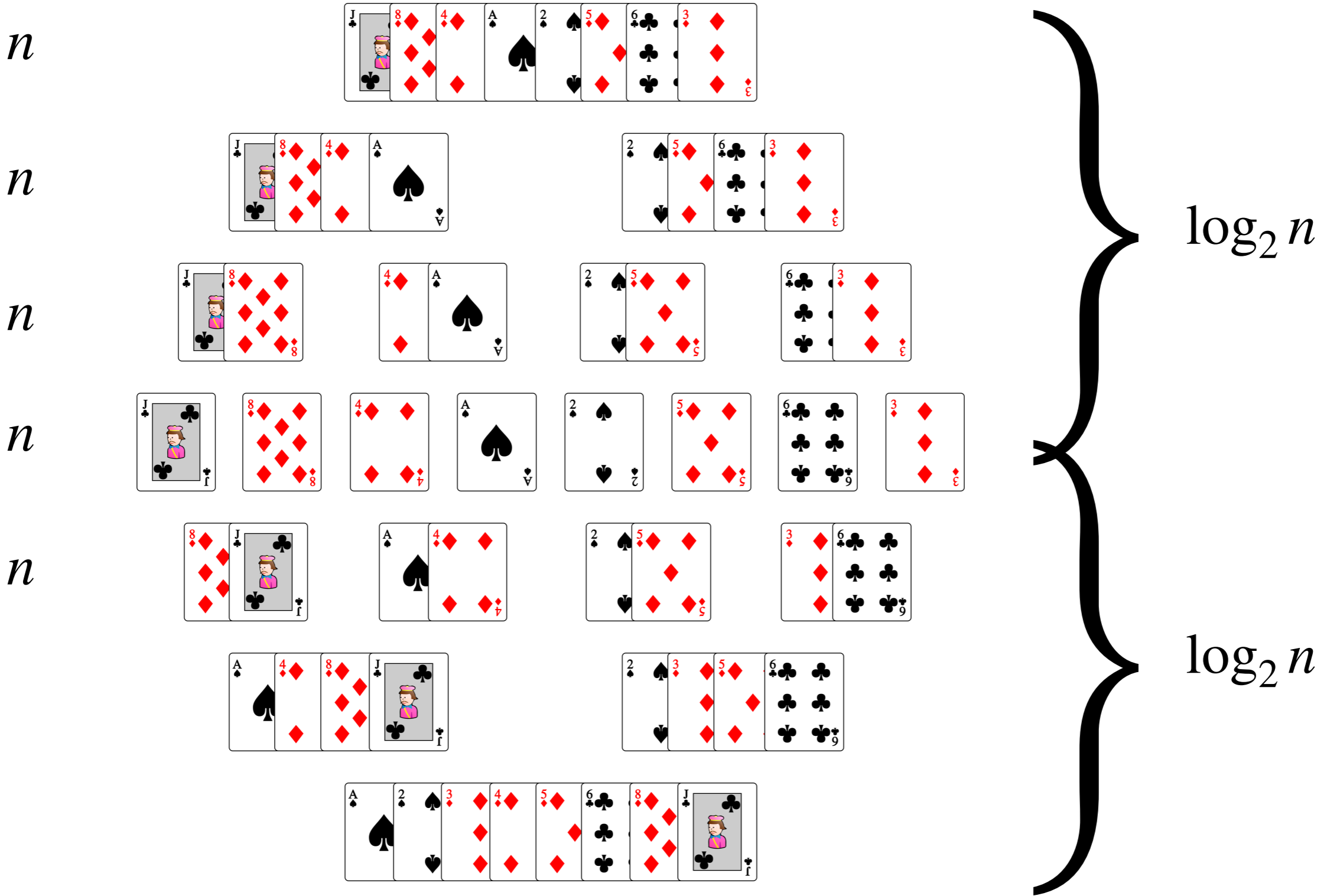
Efficacité du tri fusion



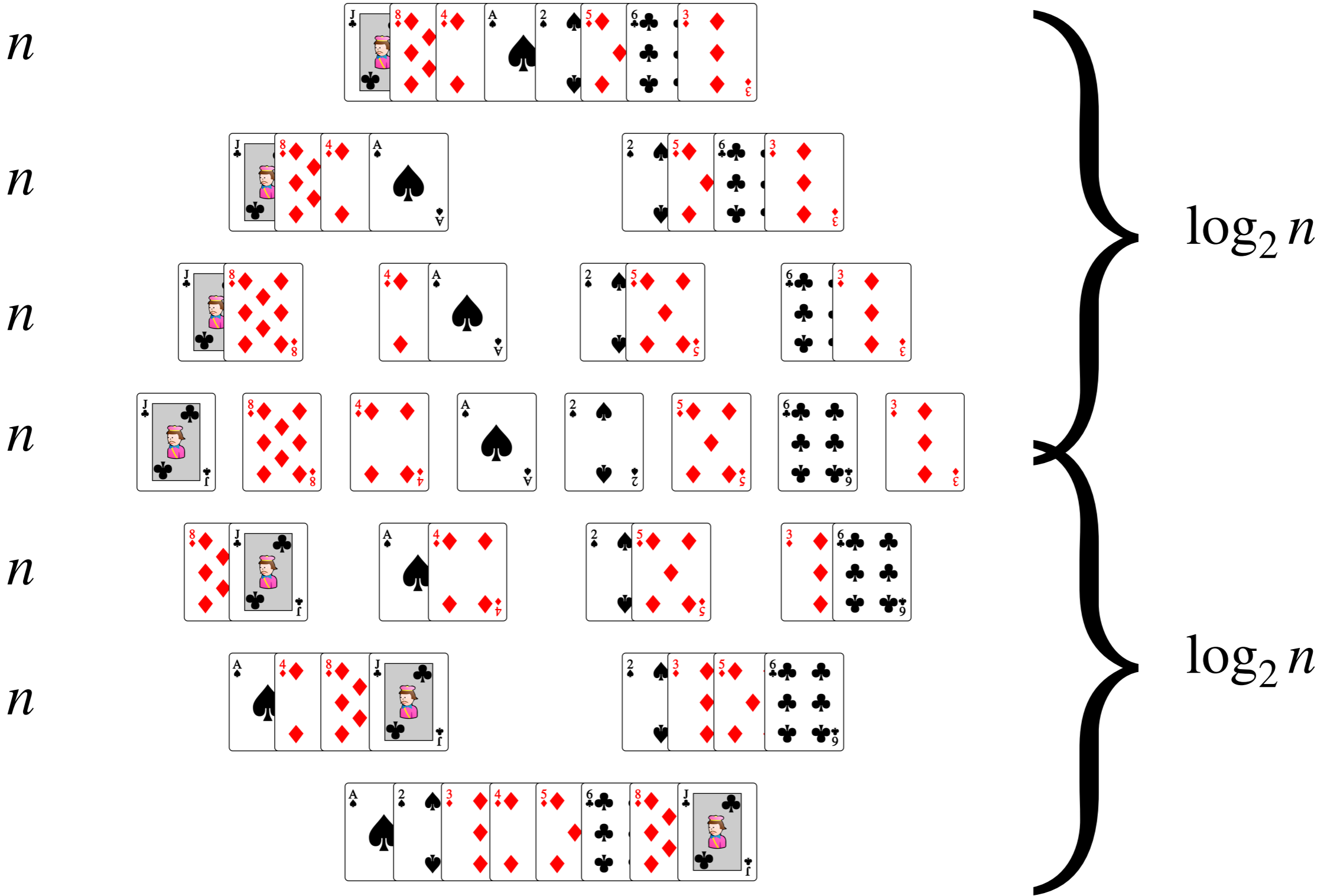
Efficacité du tri fusion



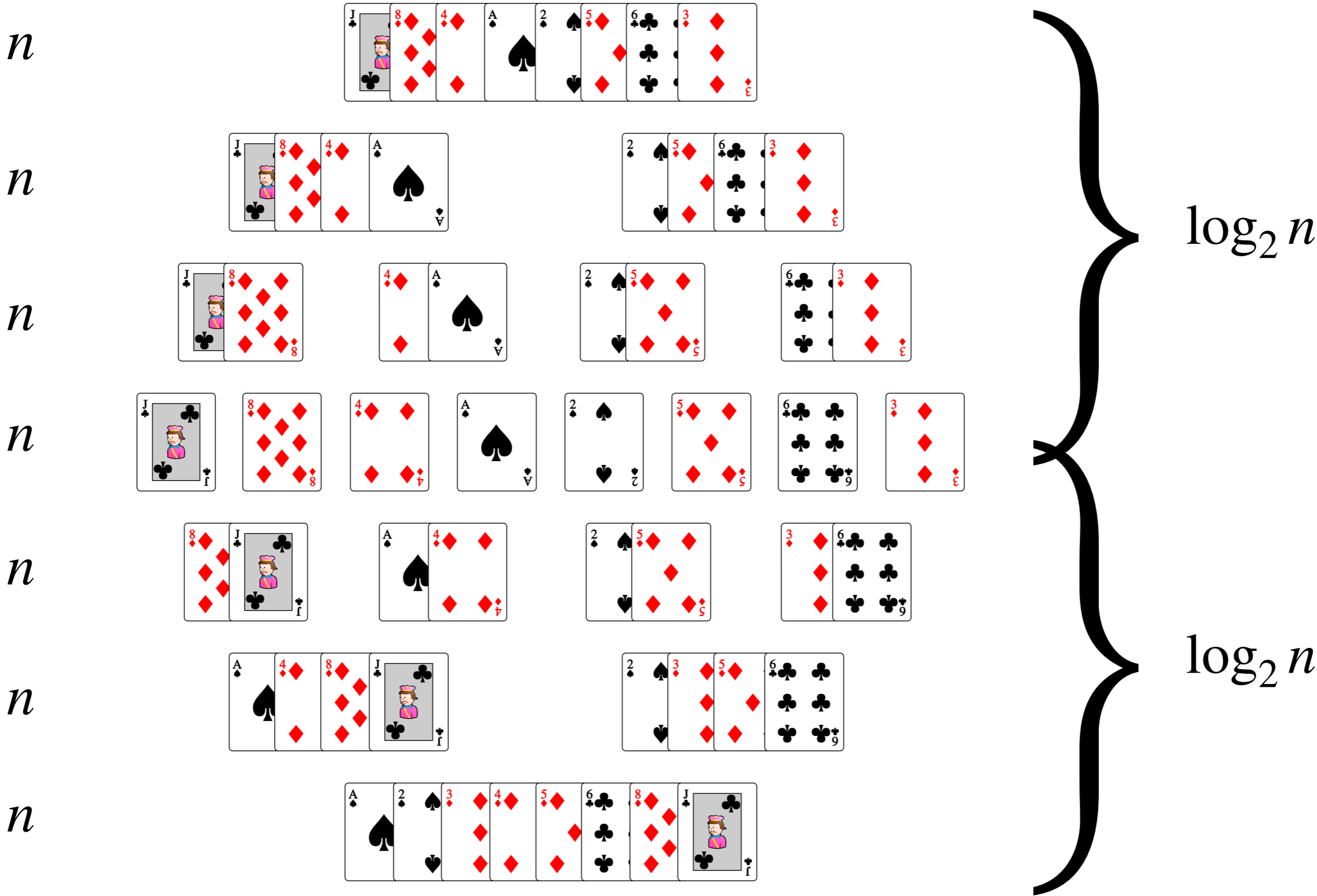
Efficacité du tri fusion



Efficacité du tri fusion



Efficacité du tri fusion



**La complexité
du tri fusion est**

$$O(n \log_2 n)$$

Tri fusion

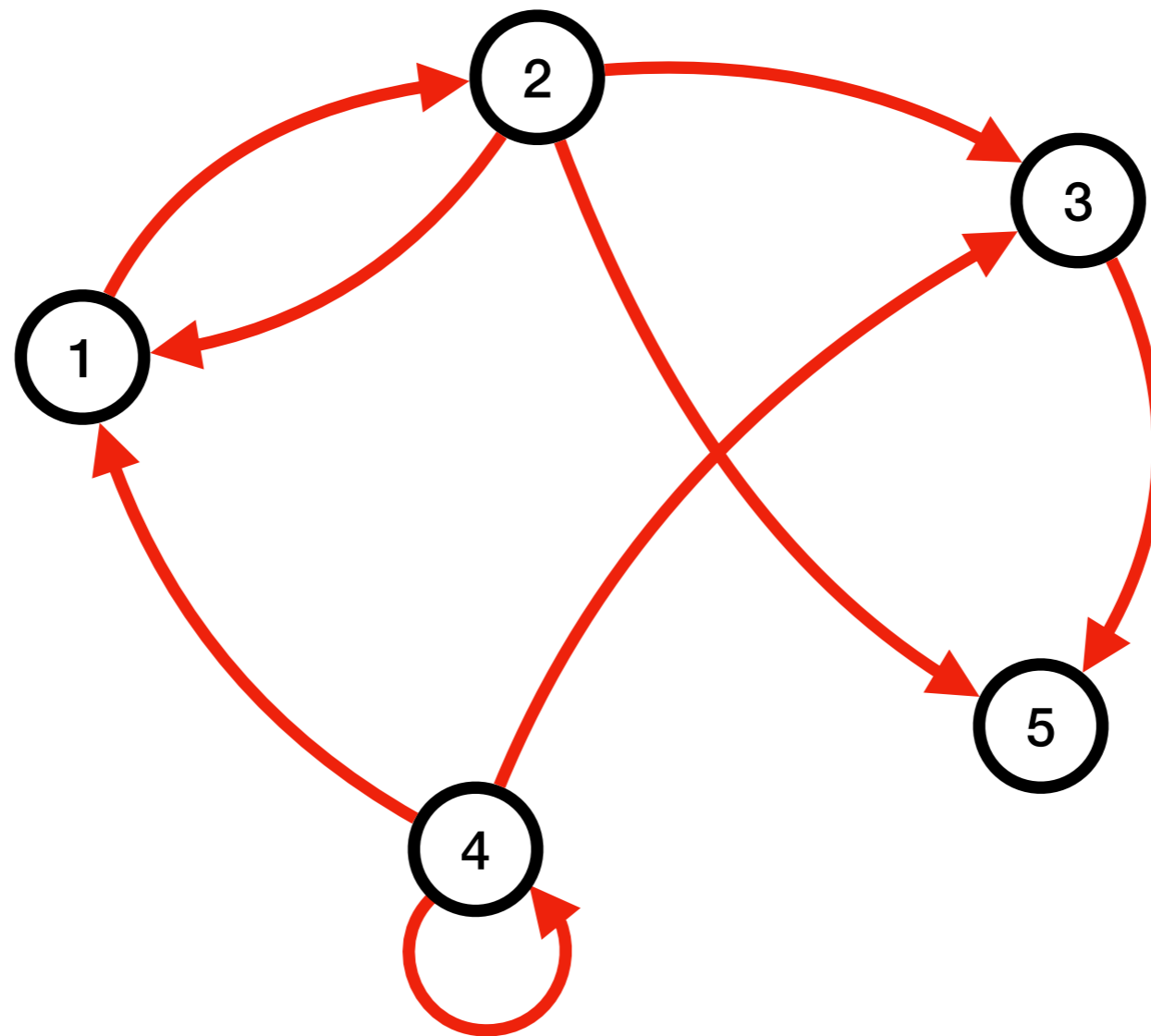
```
def tri_fusion(A):  
    n = len(A)  
    if n > 1:  
        m = n // 2  
        B = tri_fusion(A[0:m])  
        C = tri_fusion(A[m:n])  
        return fusionner(B, C)  
    else:  
        return A
```

Exercice 1 du TD3

Graphes

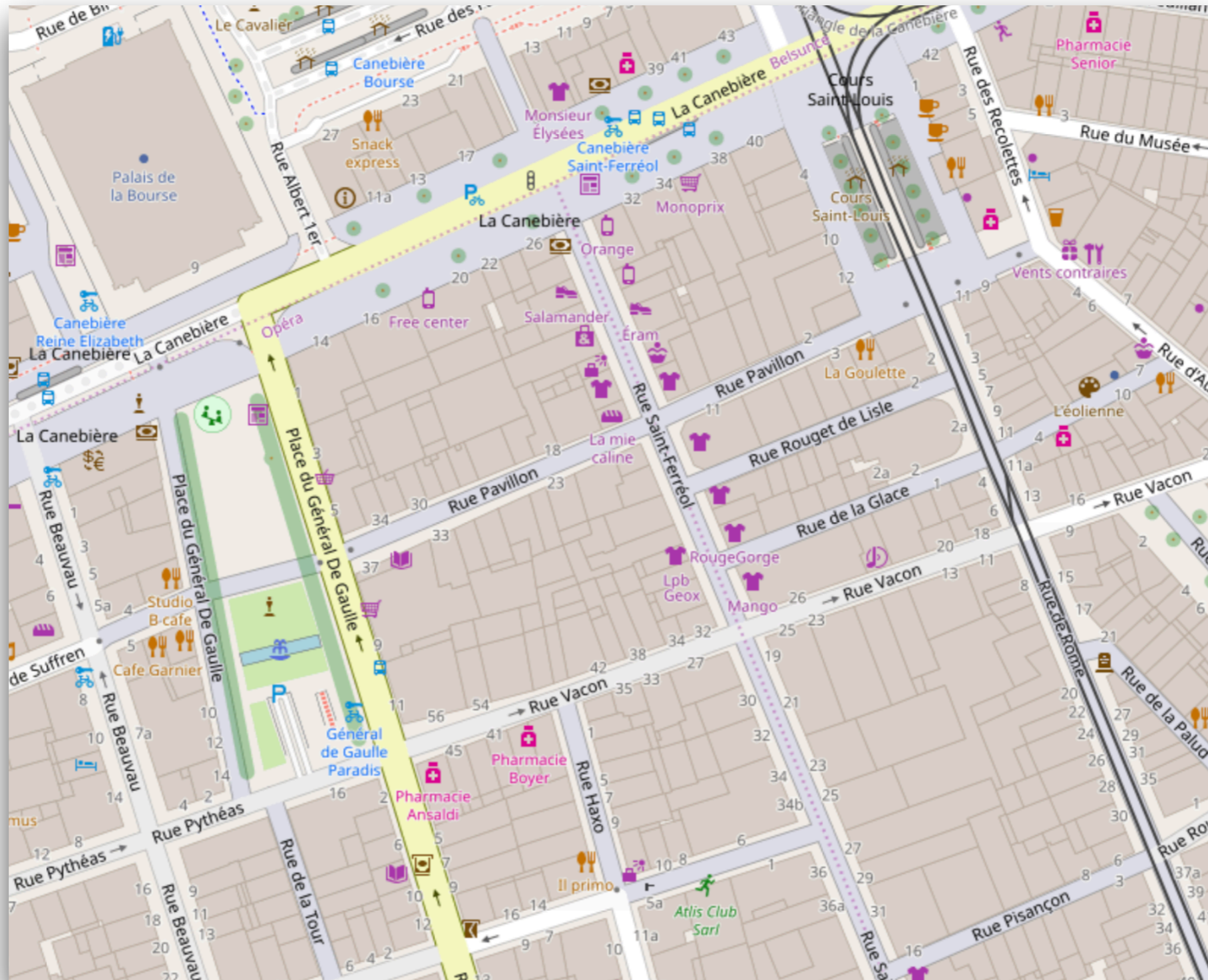
Graphes (orientés)

sommets
(nœuds,
points)



arcs

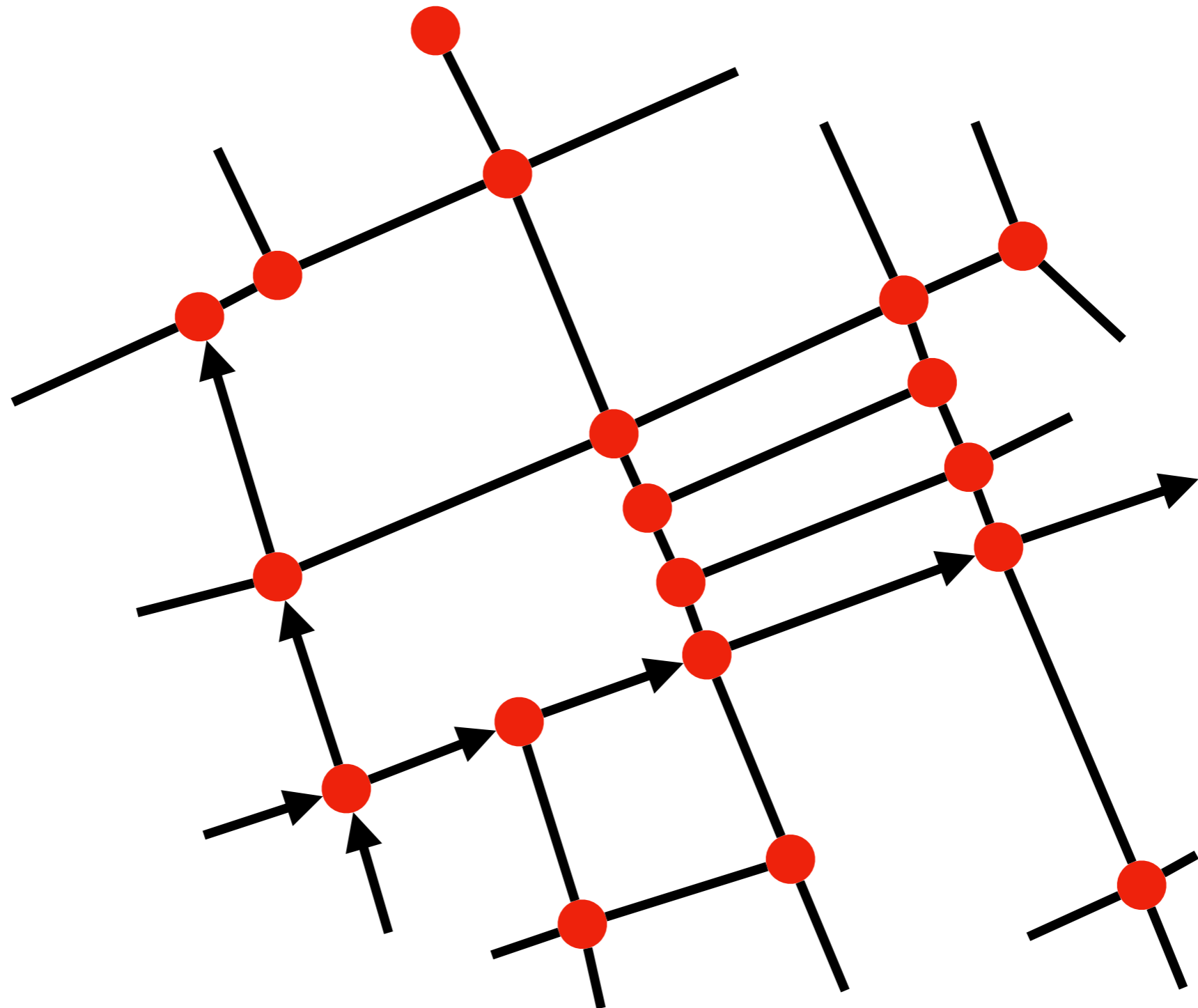
Plan d'une ville



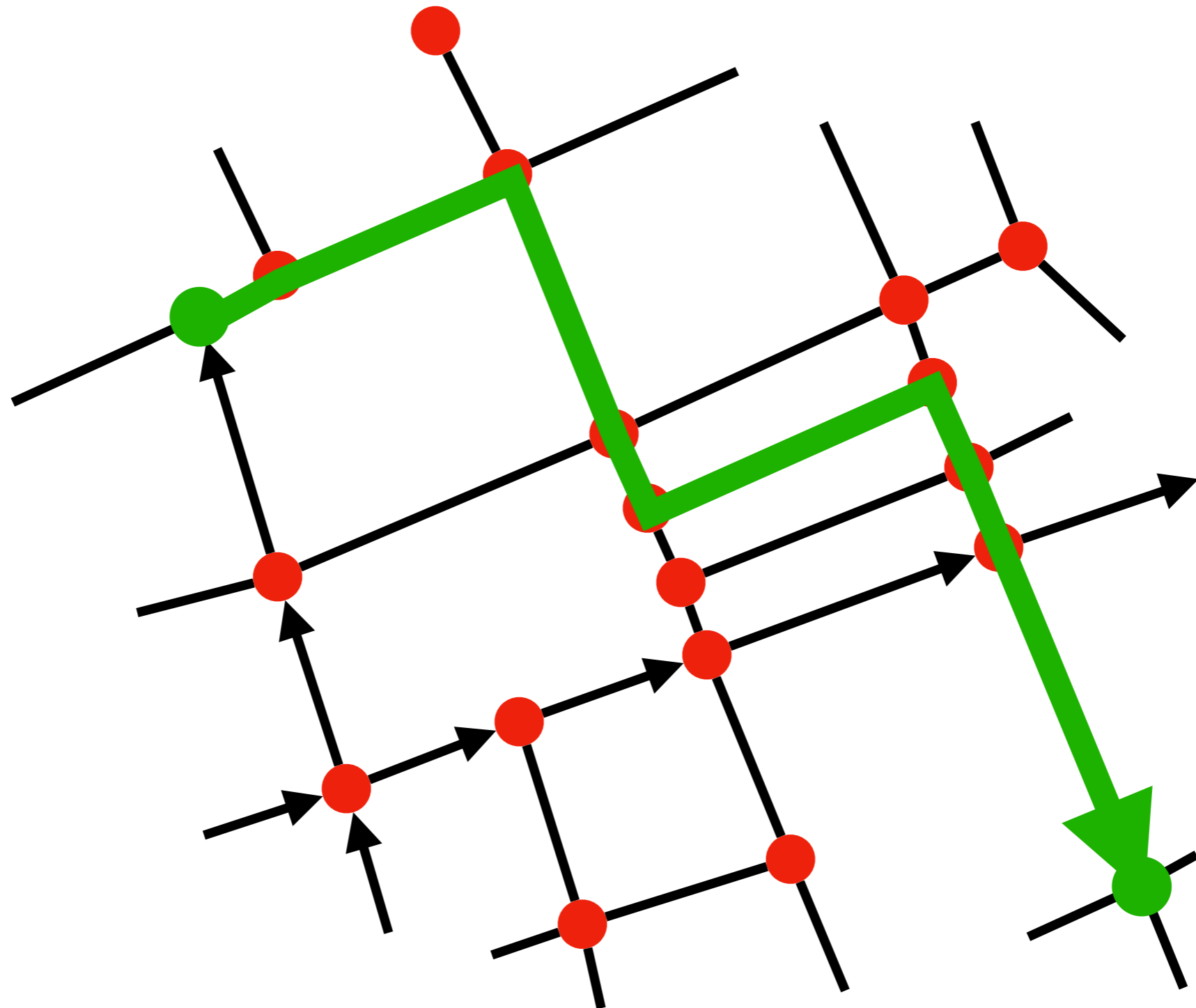
Plan d'une ville



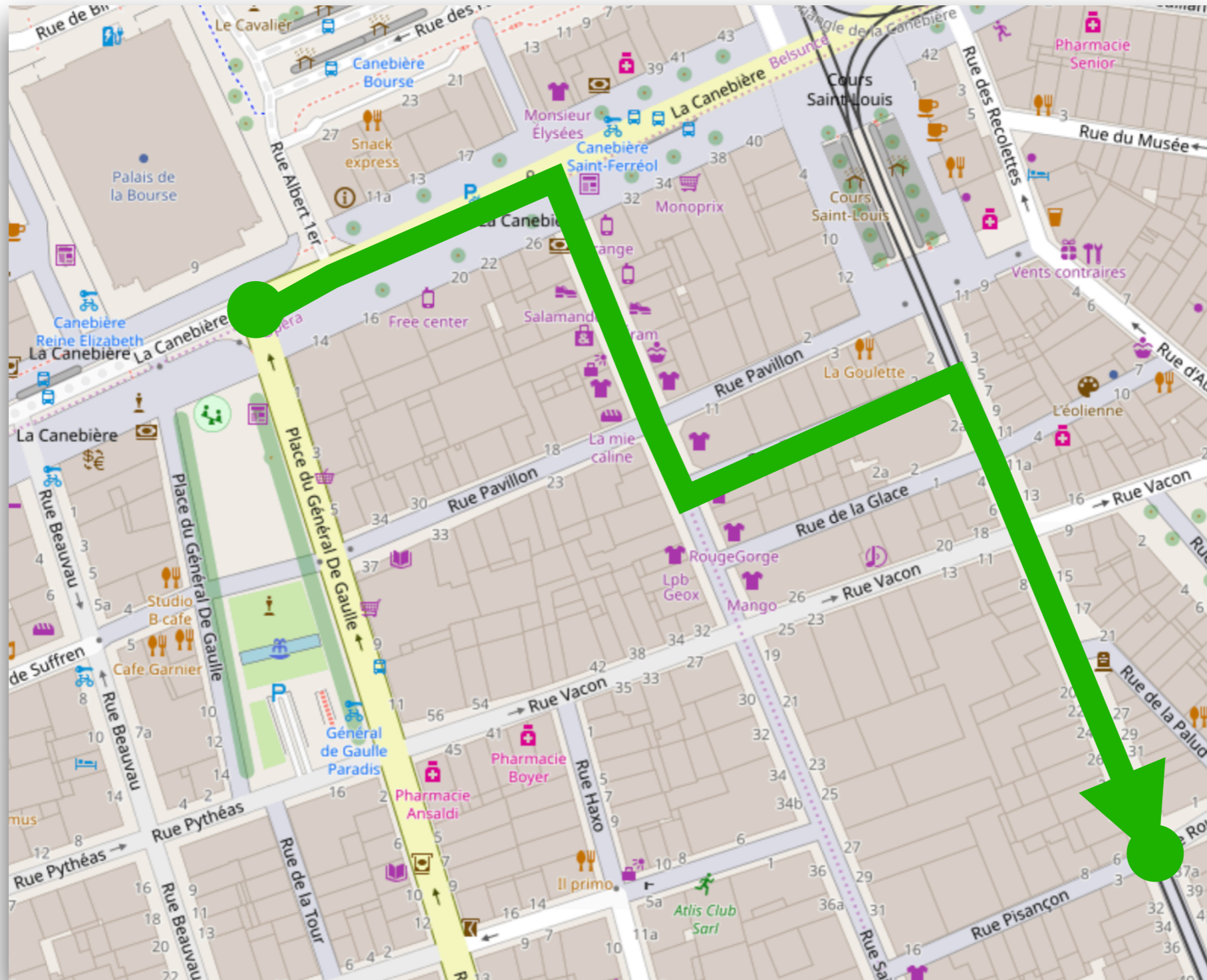
Plan d'une ville



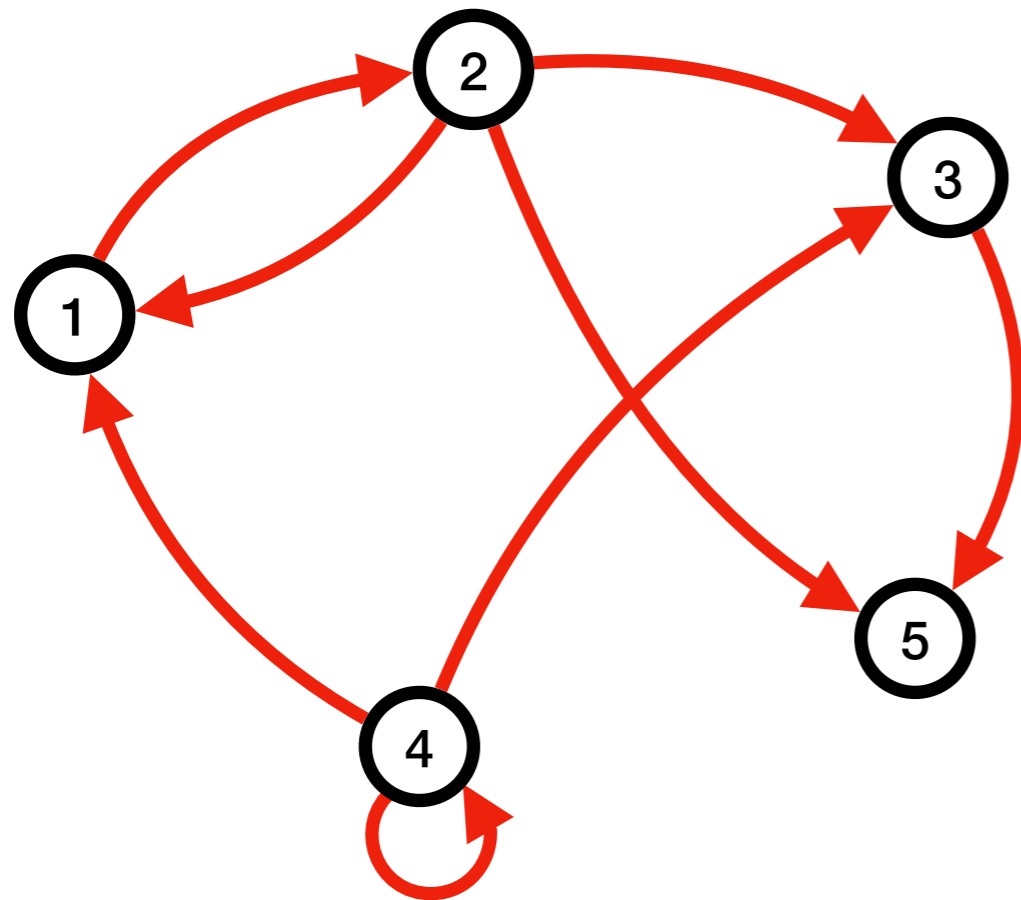
Plan d'une ville



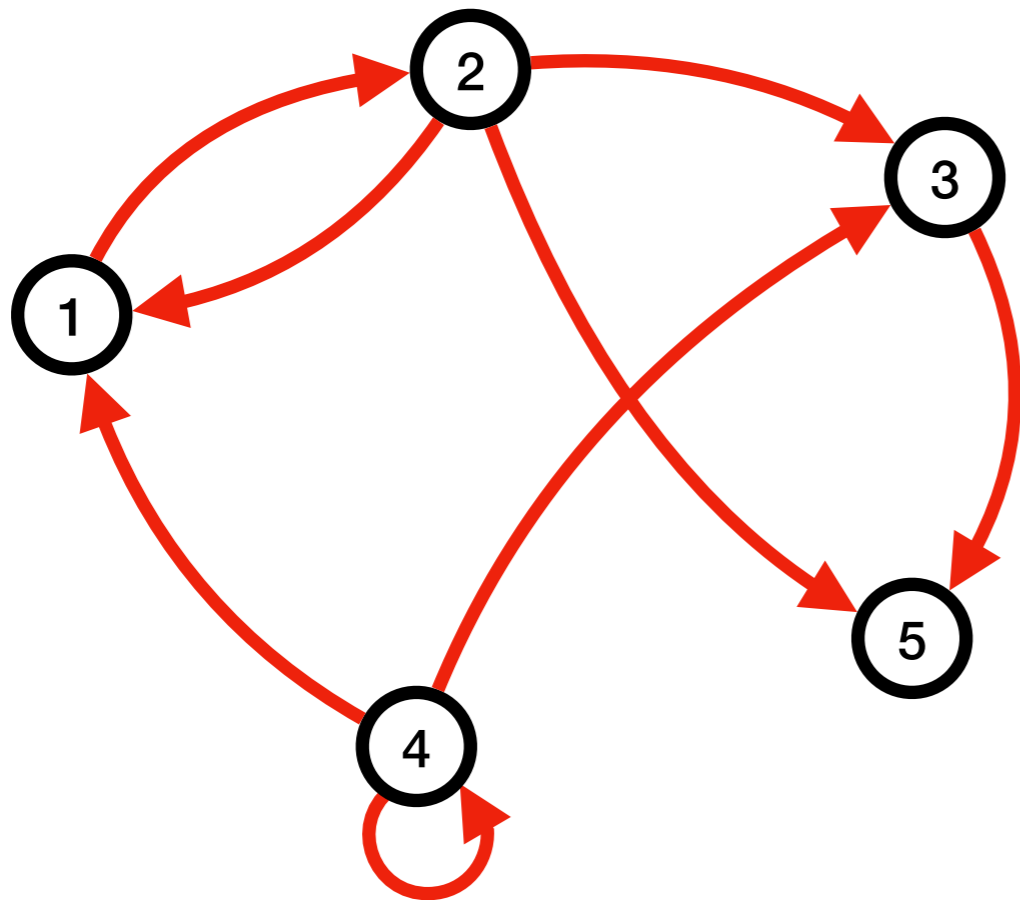
Plan d'une ville



Matrices d'adjacence

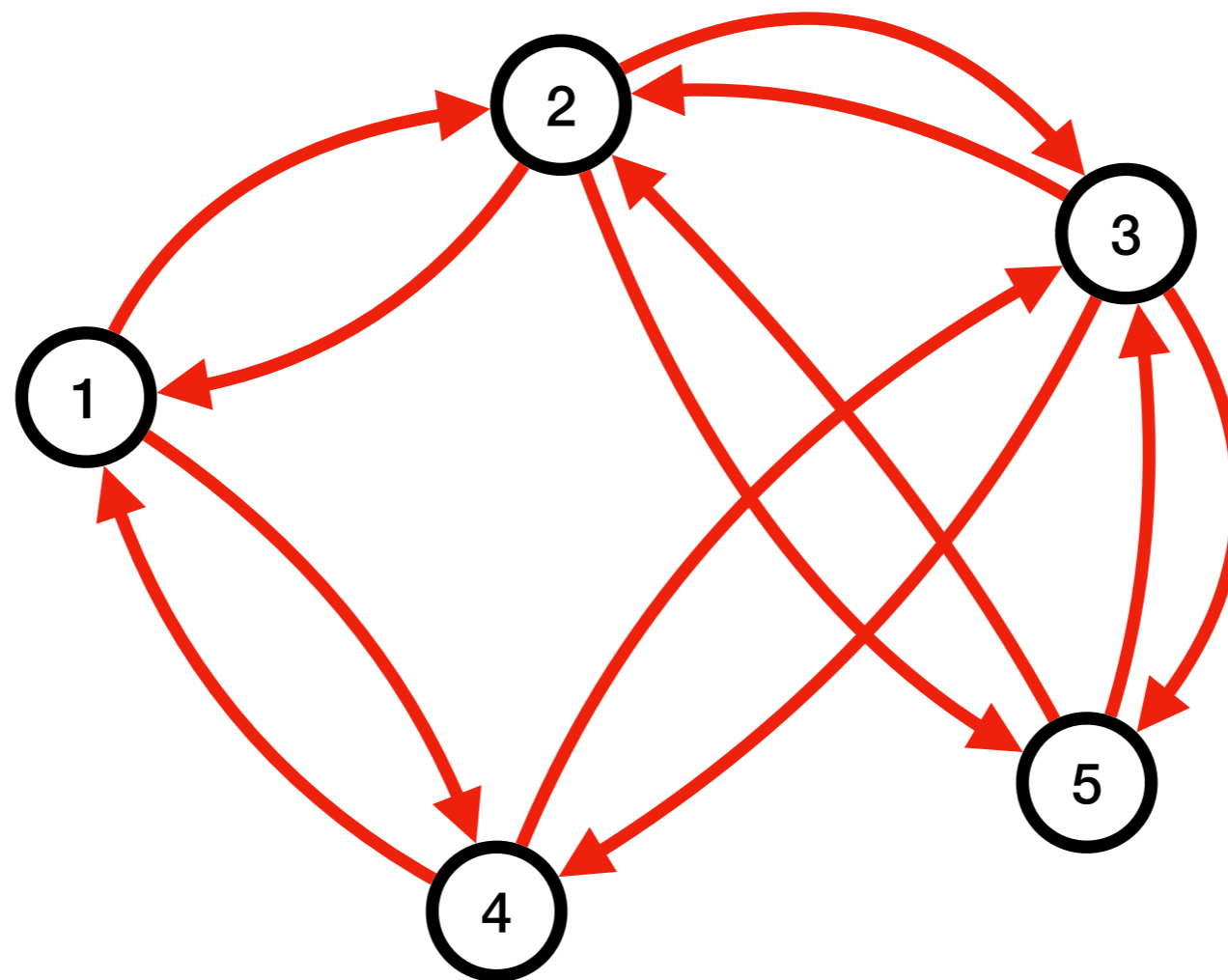


Matrices d'adjacence

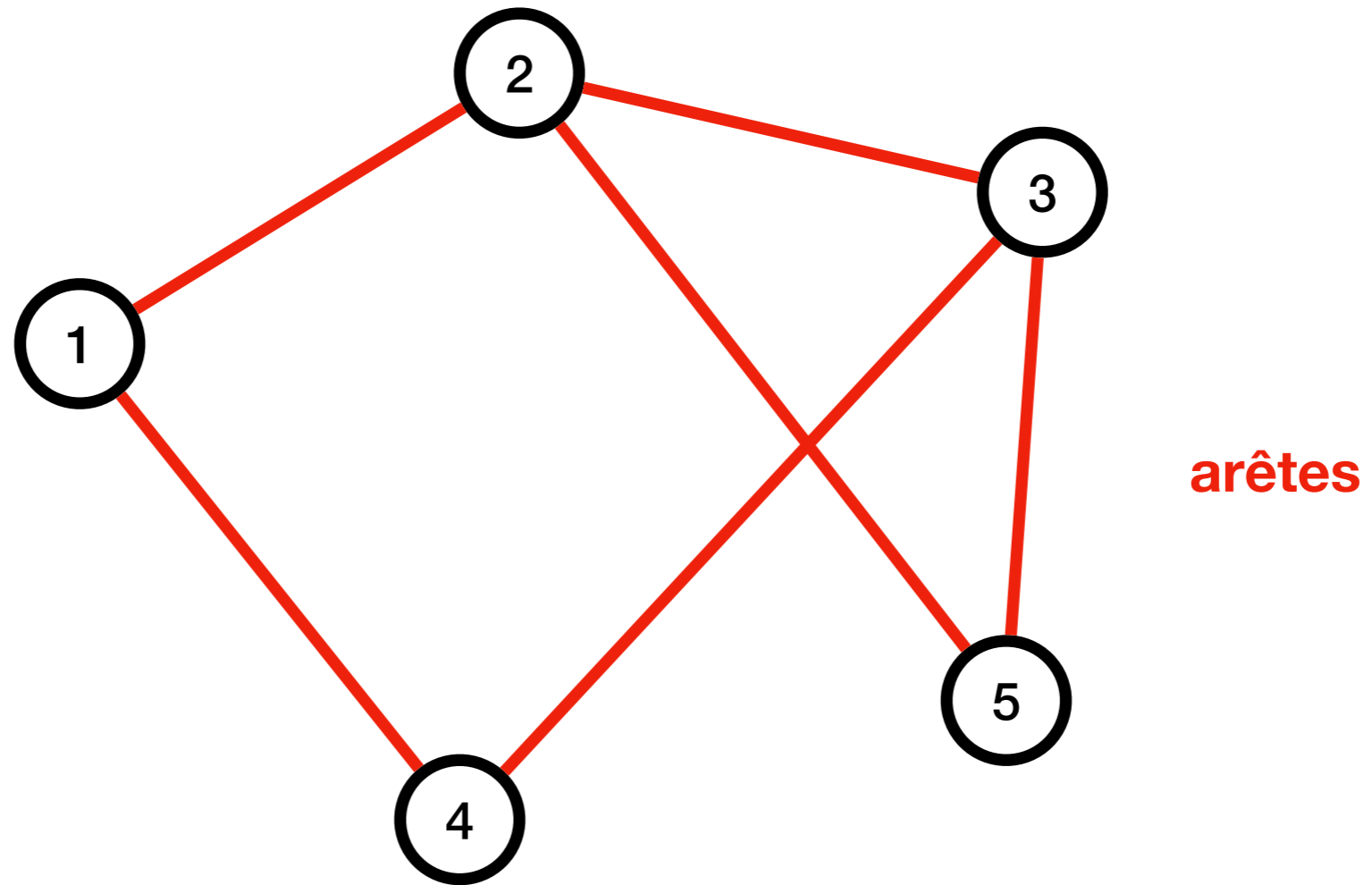


	1	2	3	4	5
1	0	1	0	0	0
2	1	0	1	0	1
3	0	0	0	0	1
4	1	0	1	1	0
5	0	0	0	0	0

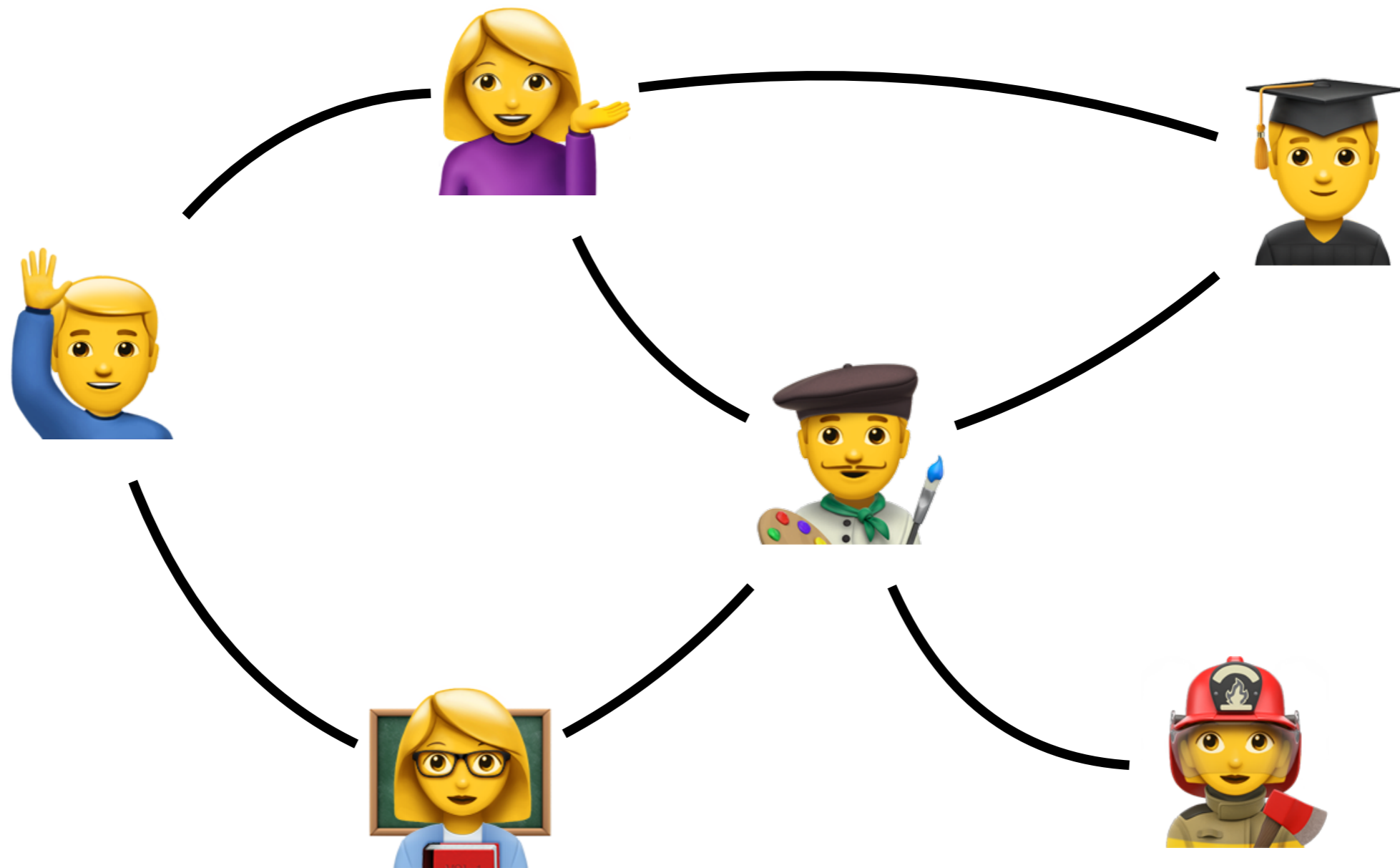
Graphes non orientés



Graphes non orientés



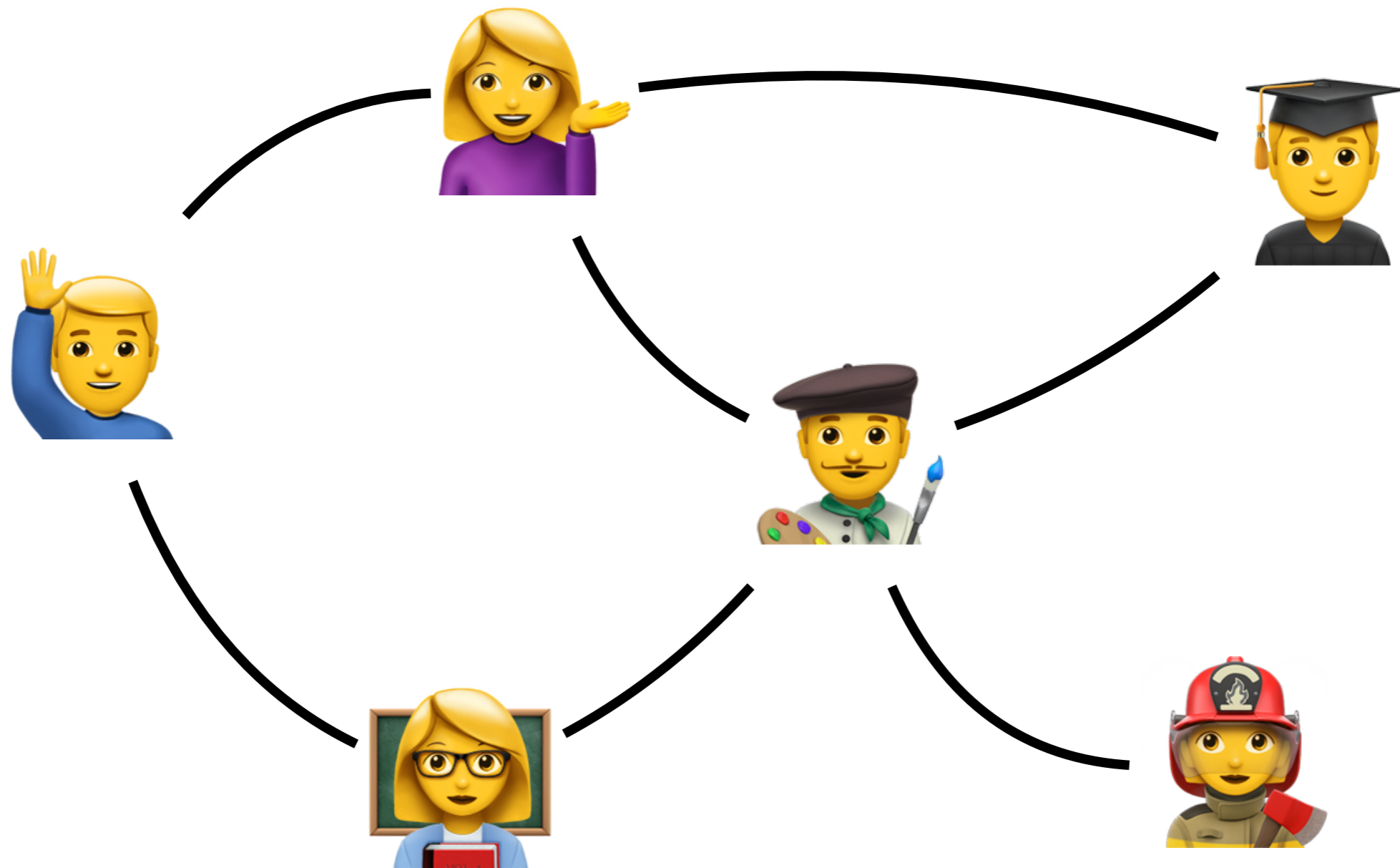
Réseau social (symétrique)



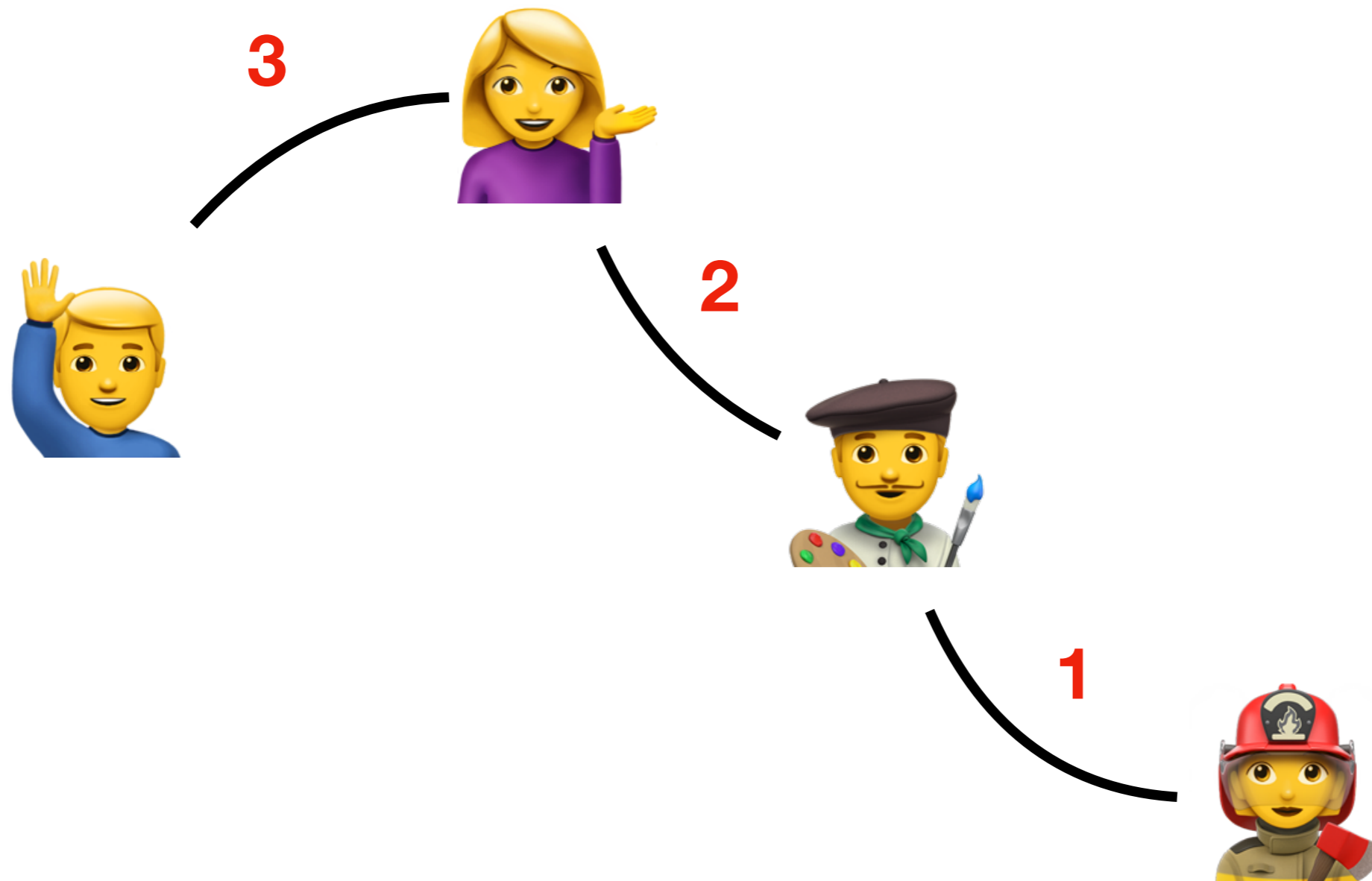
**Each person in the world (at least among the
1.59 billion people active on Facebook)
is connected to every other person
by an average of three and a half other people.**

<https://research.fb.com/three-and-a-half-degrees-of-separation/>

Diamètre d'un graphe



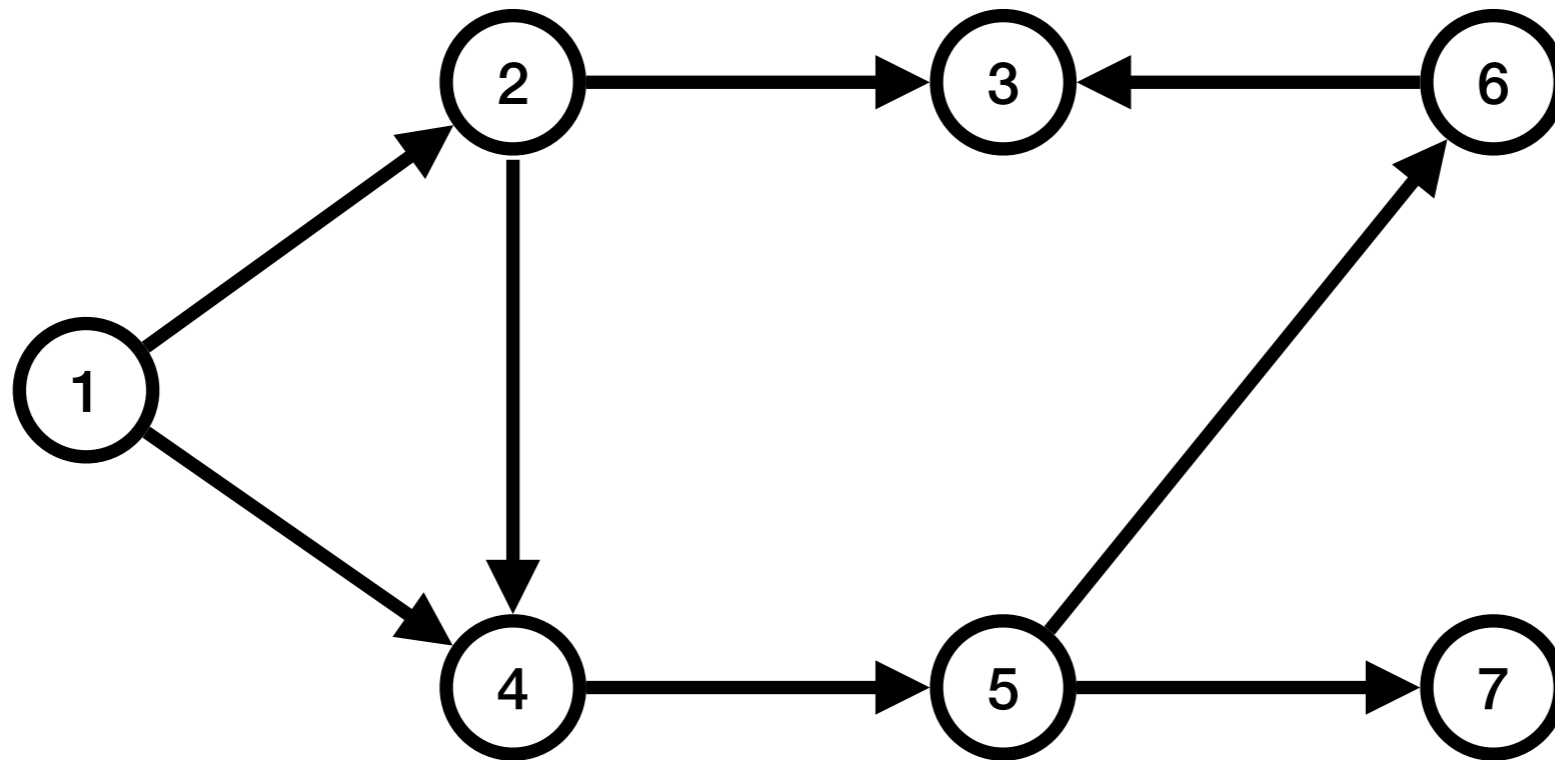
Diamètre d'un graphe



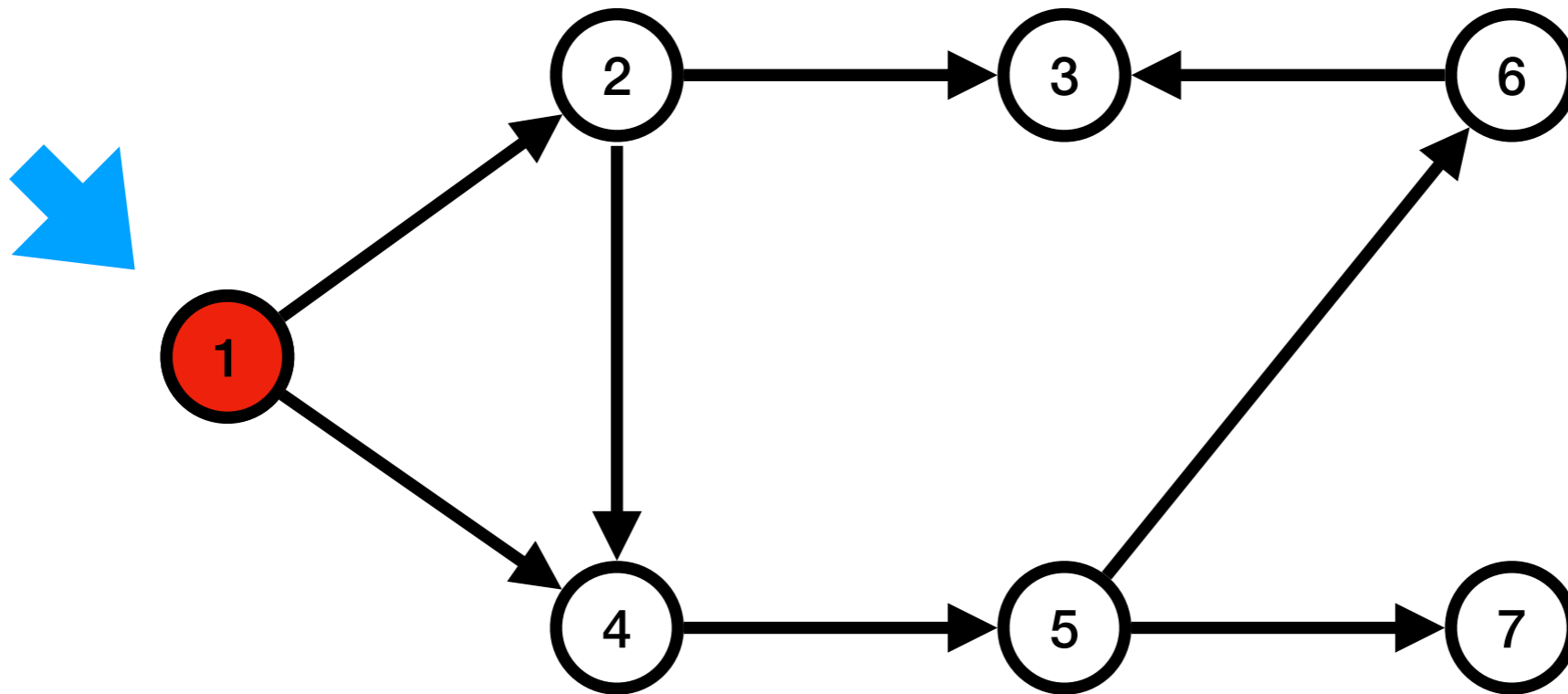
Exercice 3 du TD3

Parcours de graphes

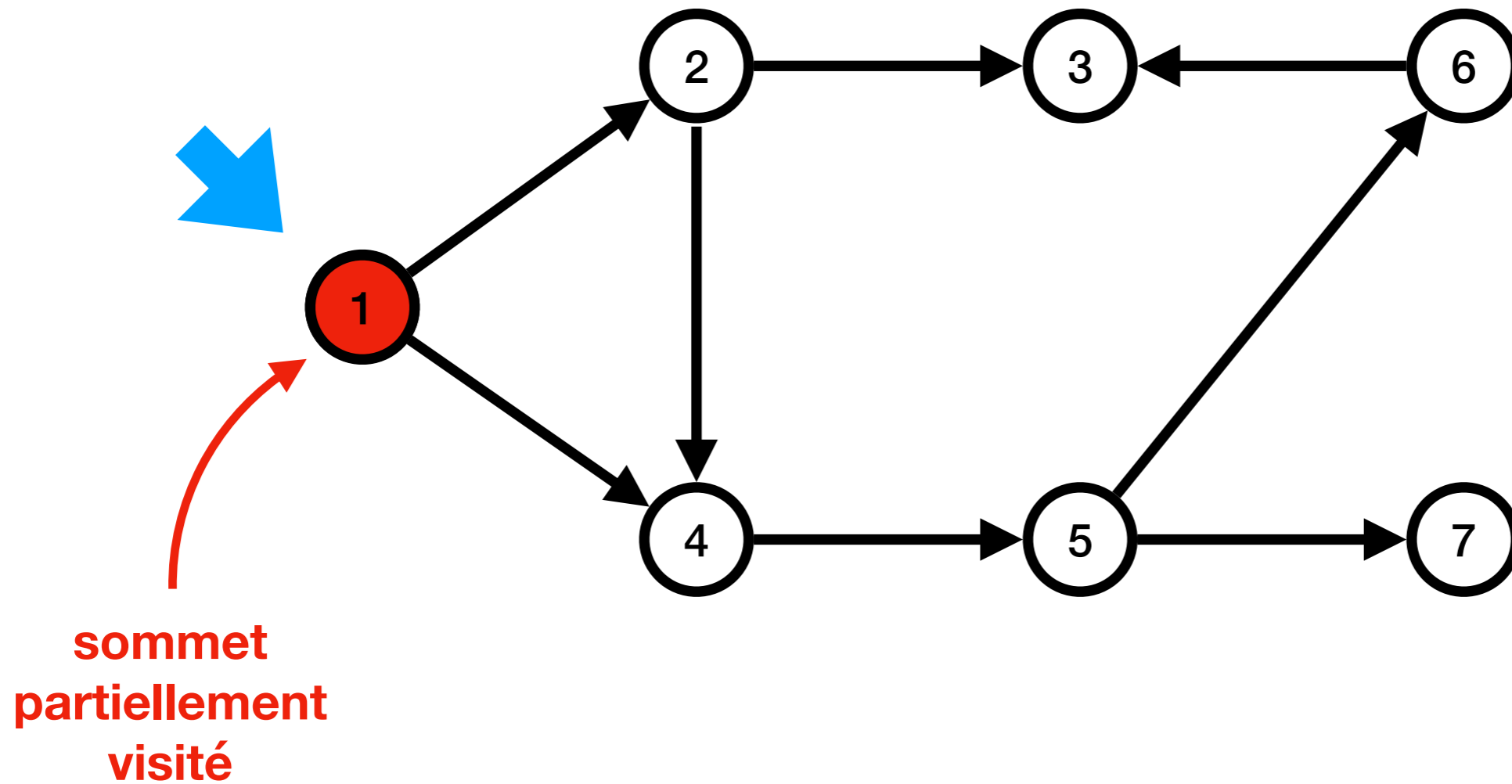
Parcours en largeur



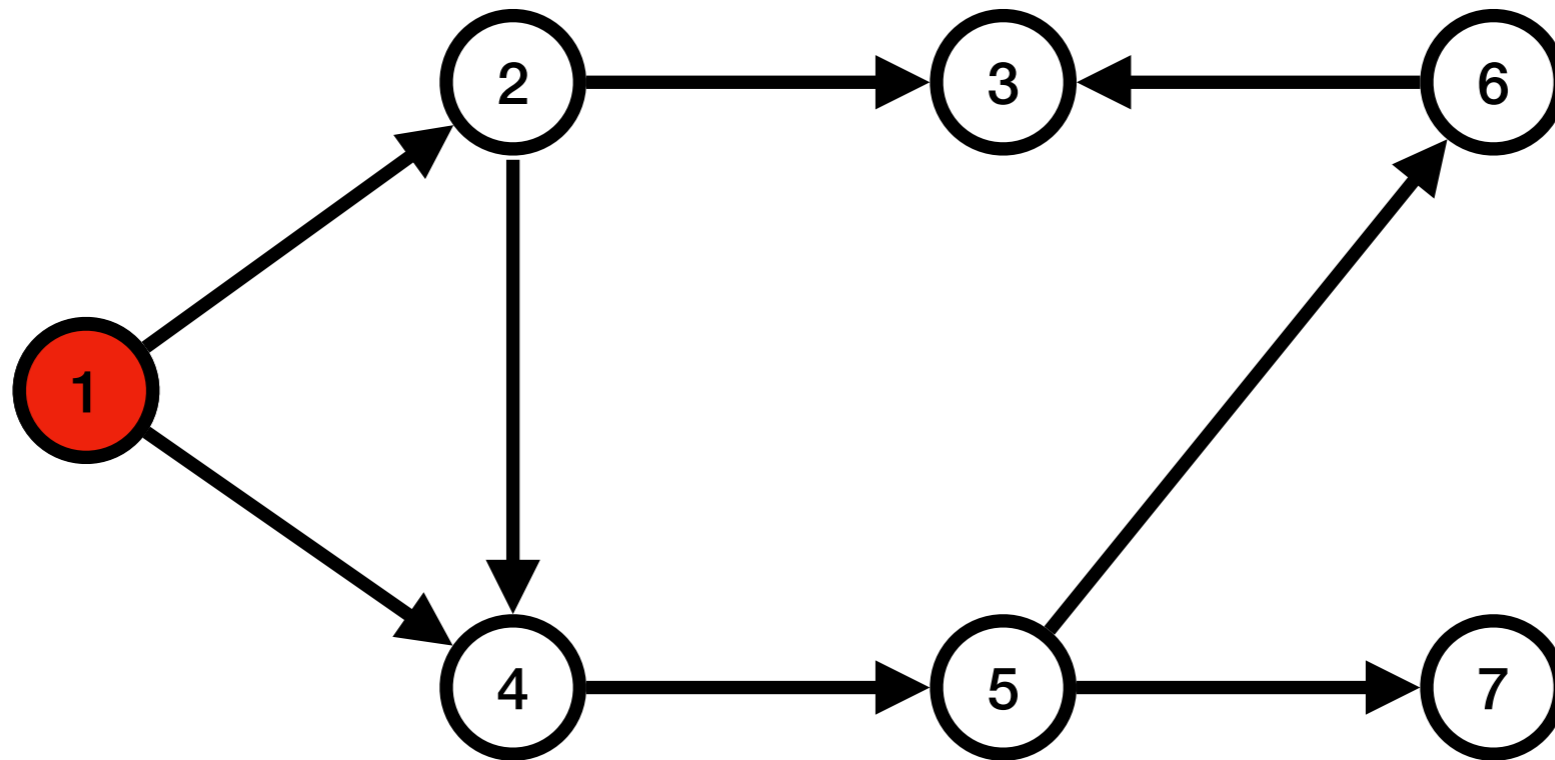
Parcours en largeur



Parcours en largeur

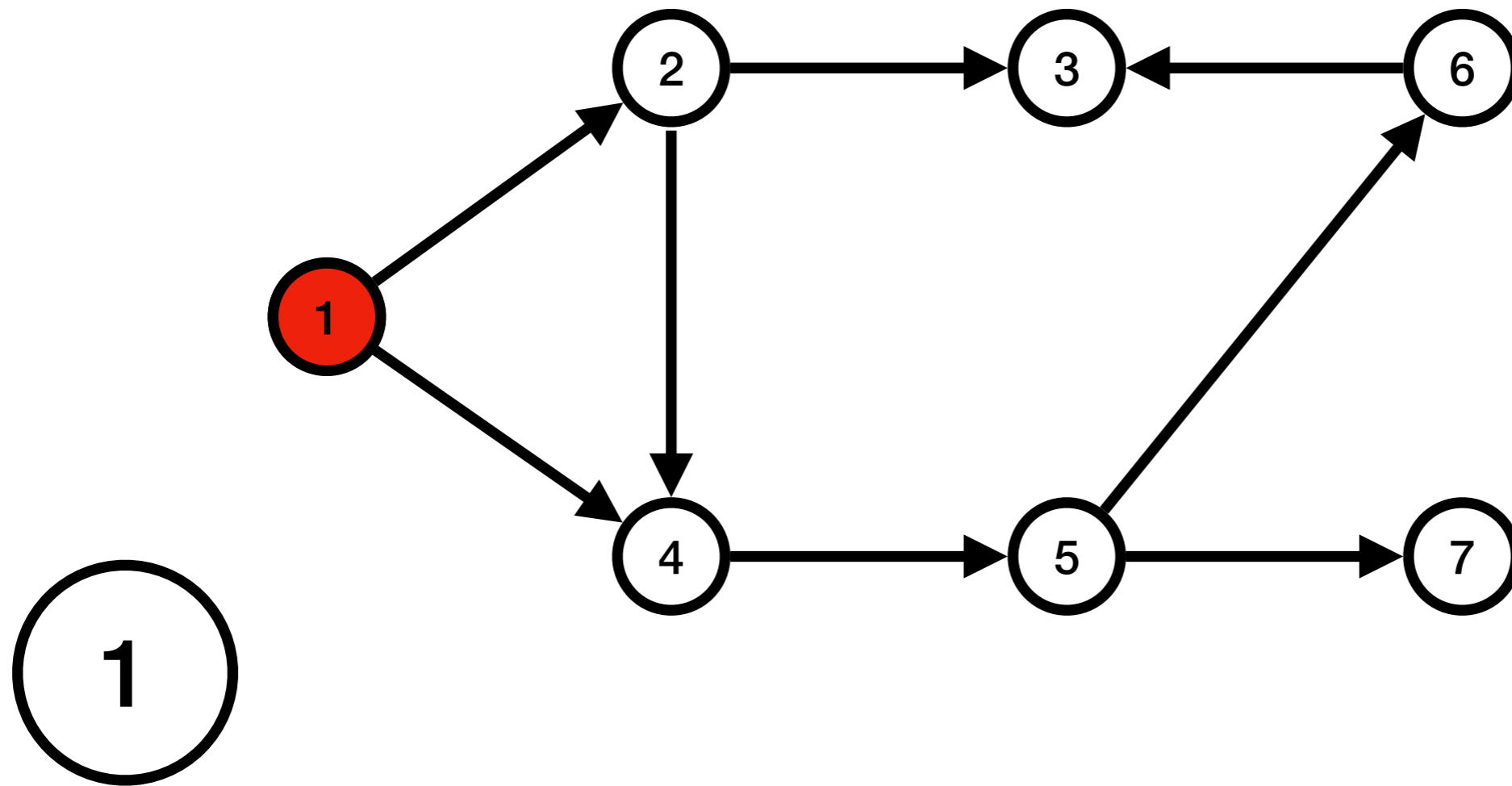


Parcours en largeur



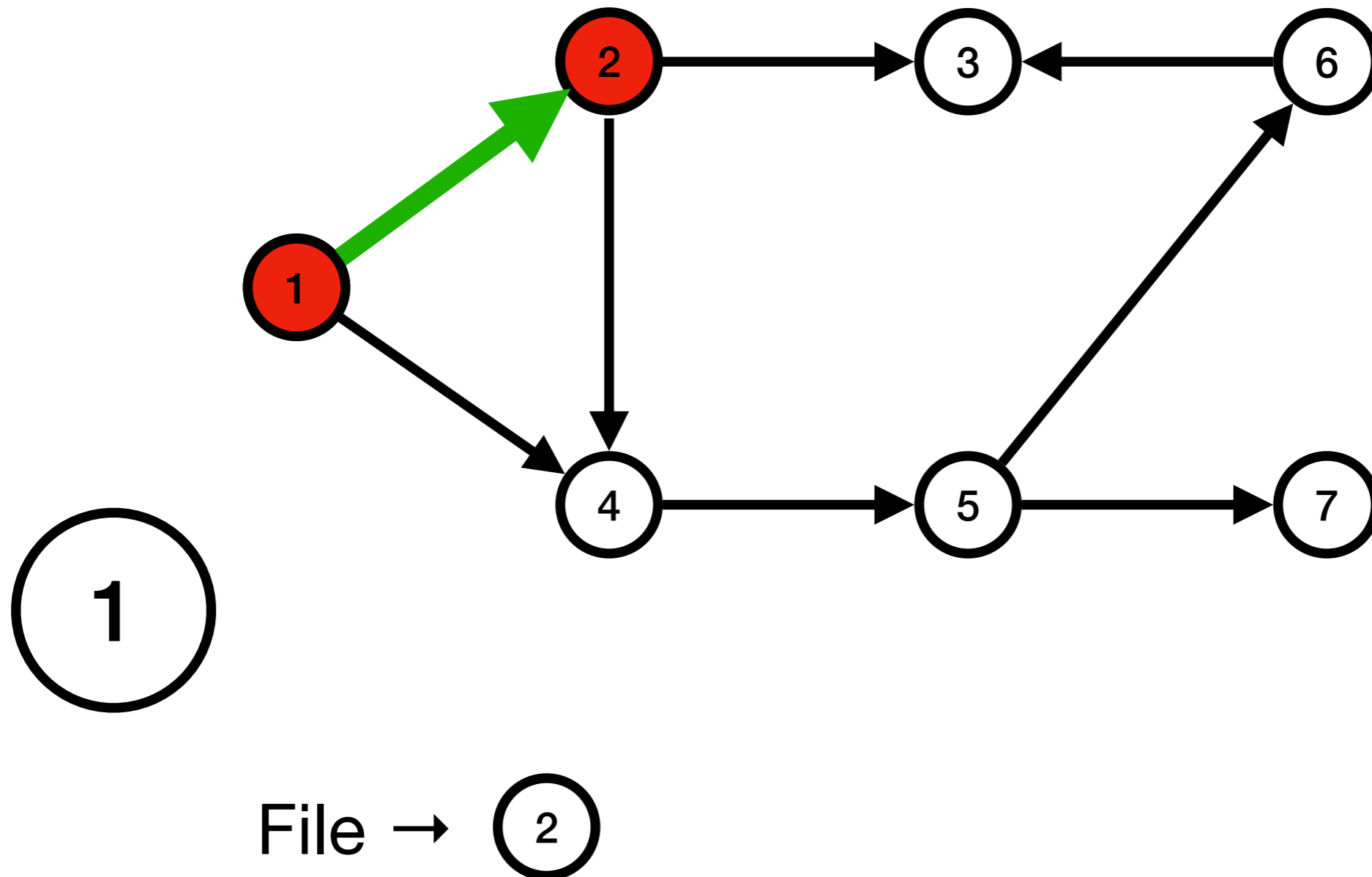
File → **1**

Parcours en largeur

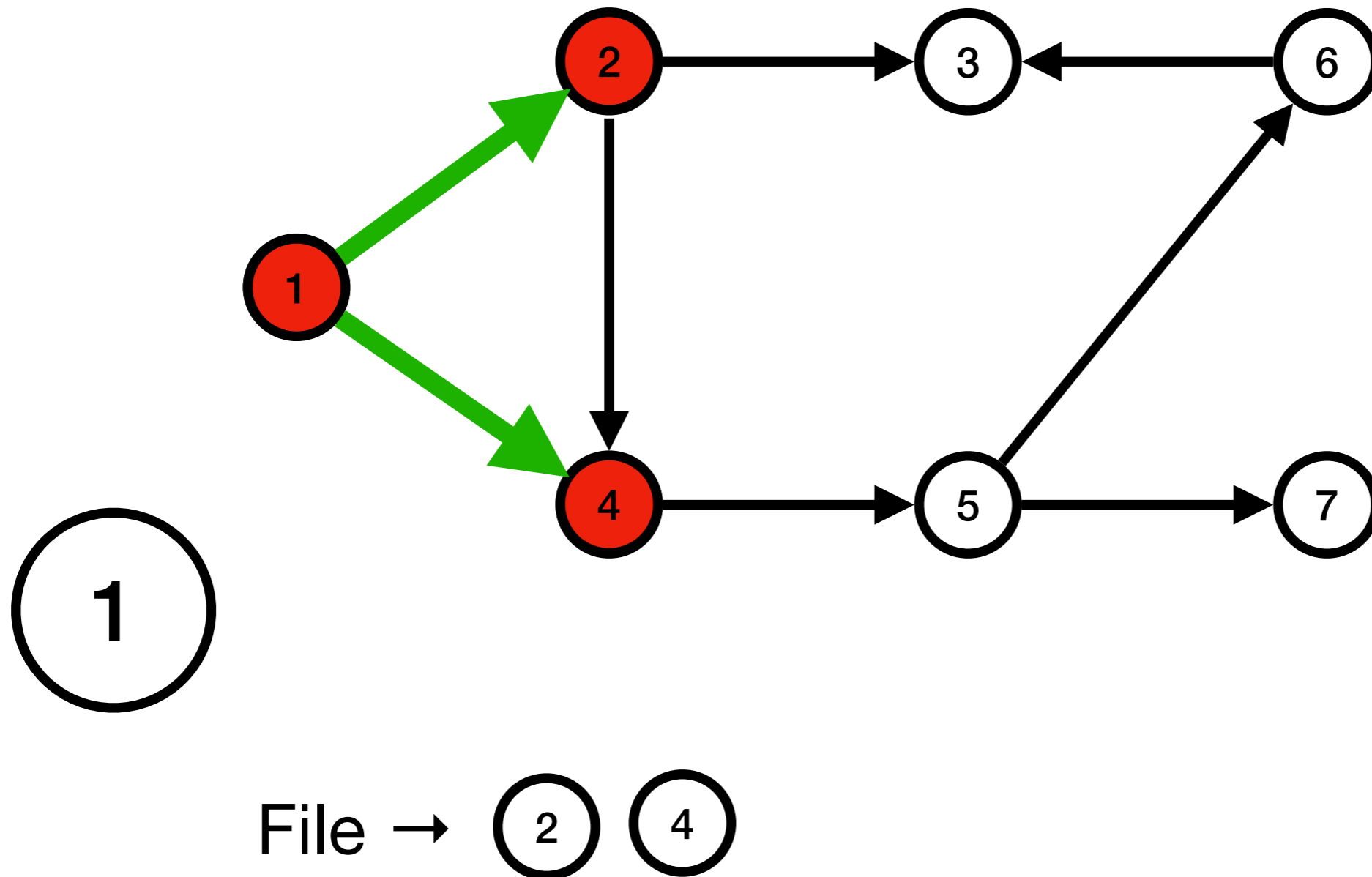


File →

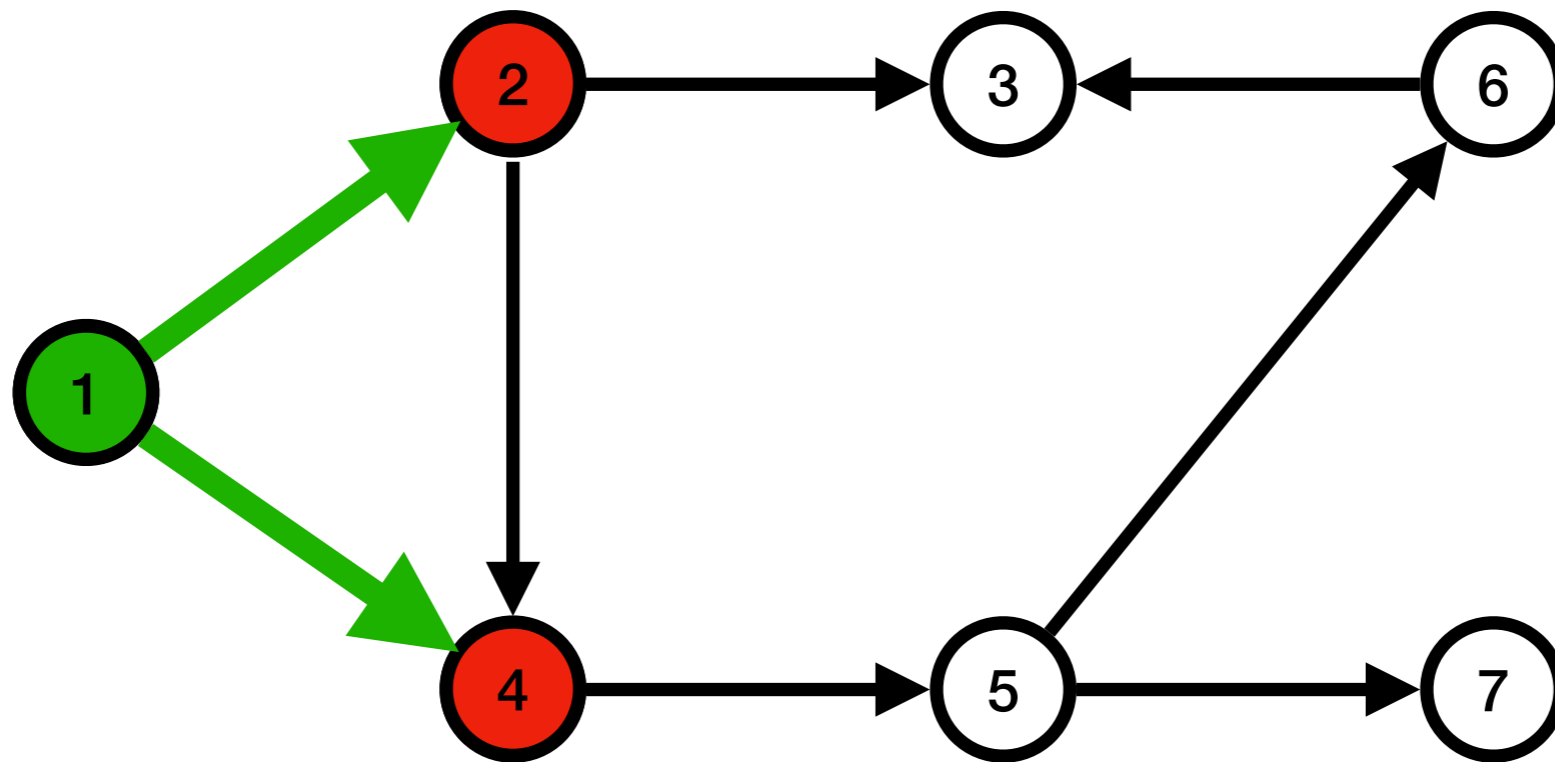
Parcours en largeur



Parcours en largeur

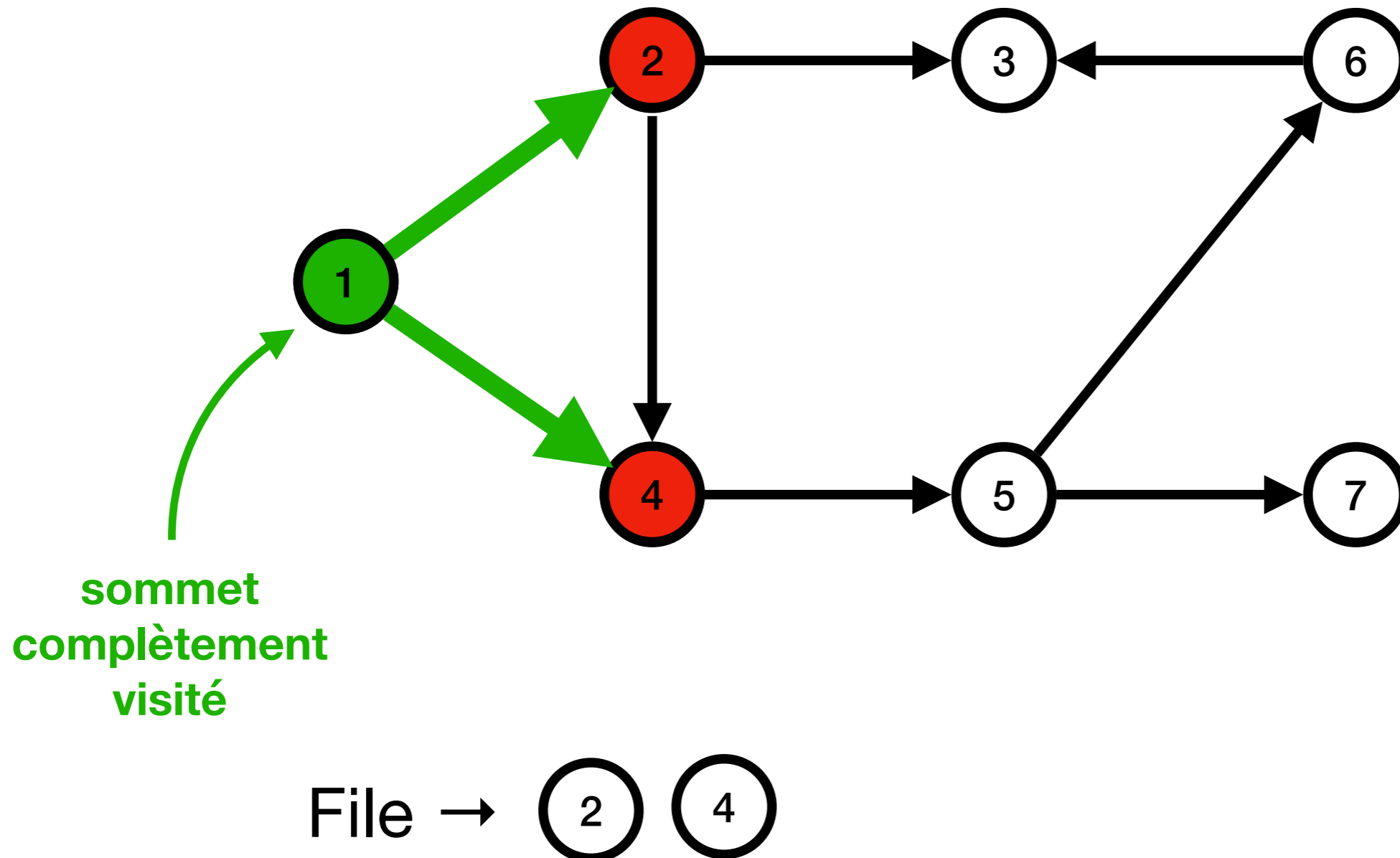


Parcours en largeur

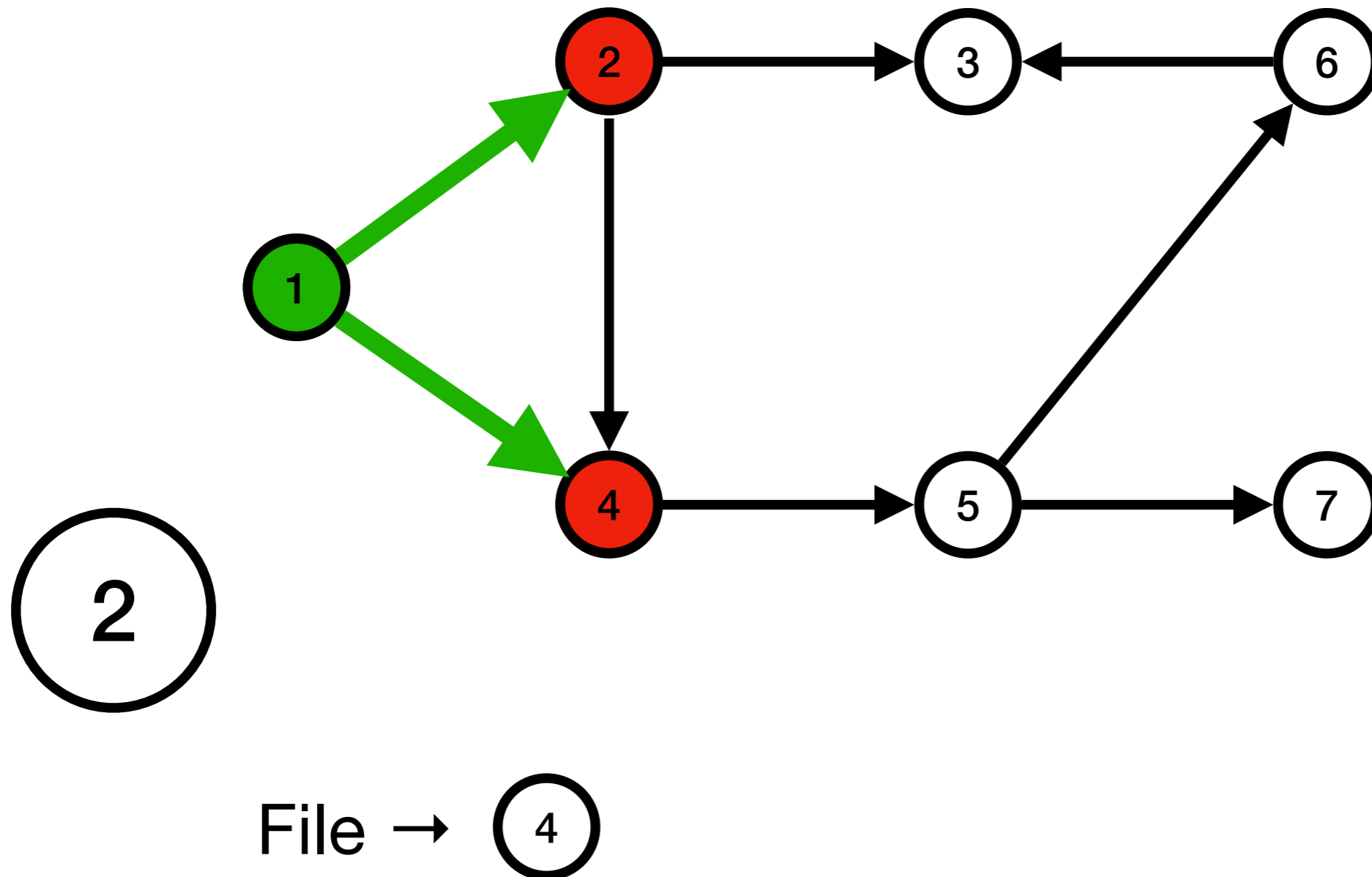


File → (2) (4)

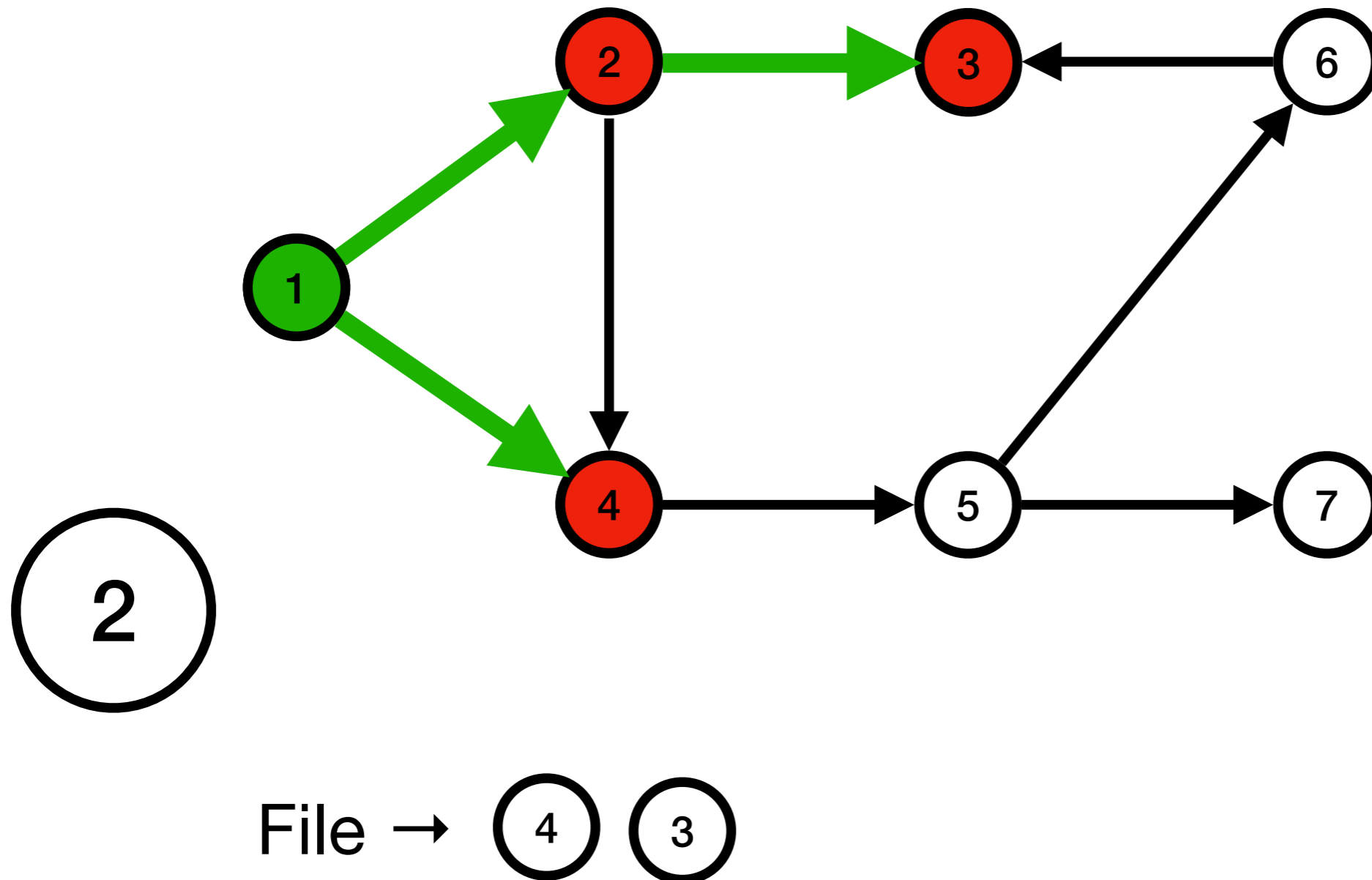
Parcours en largeur



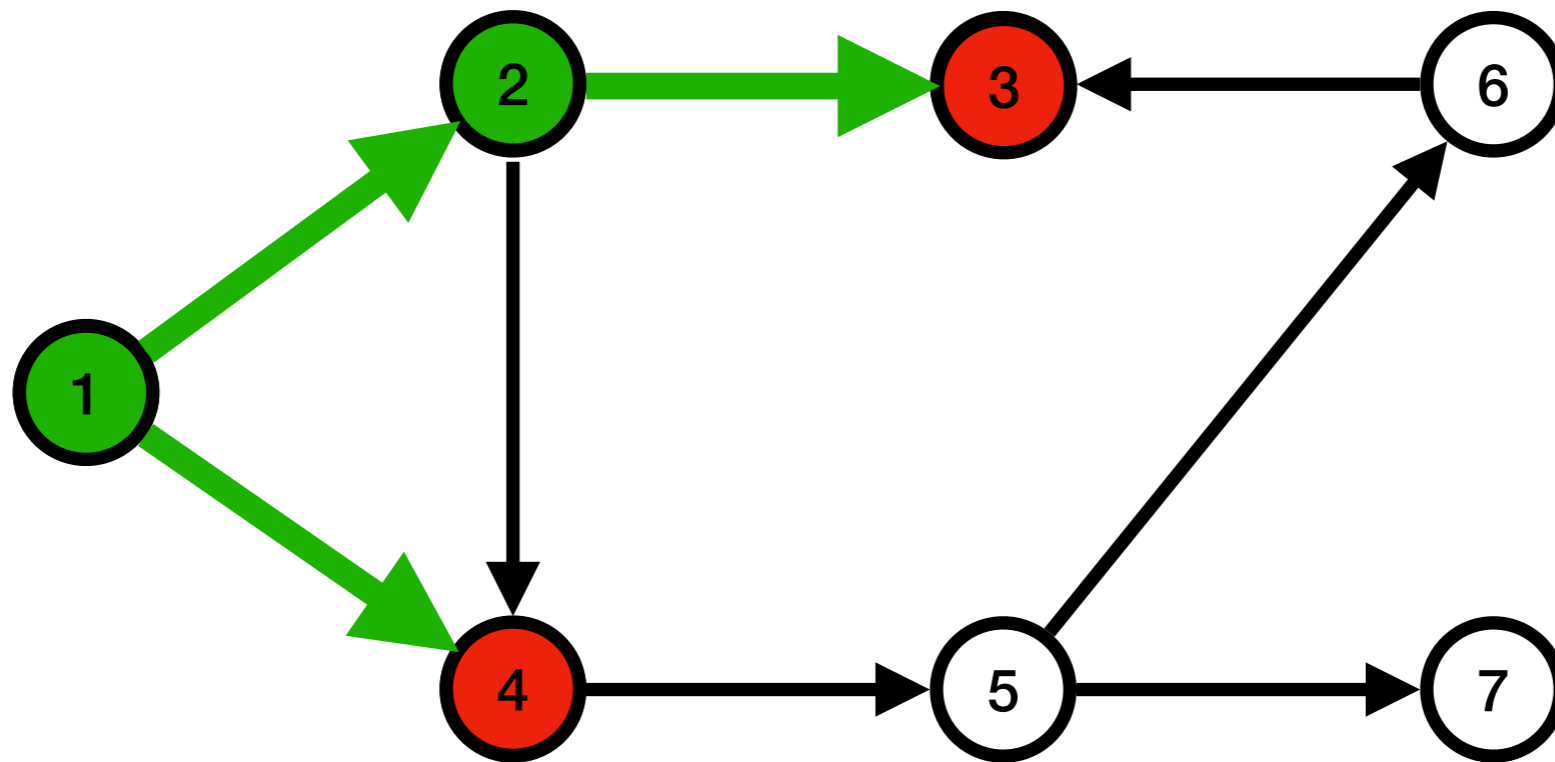
Parcours en largeur



Parcours en largeur

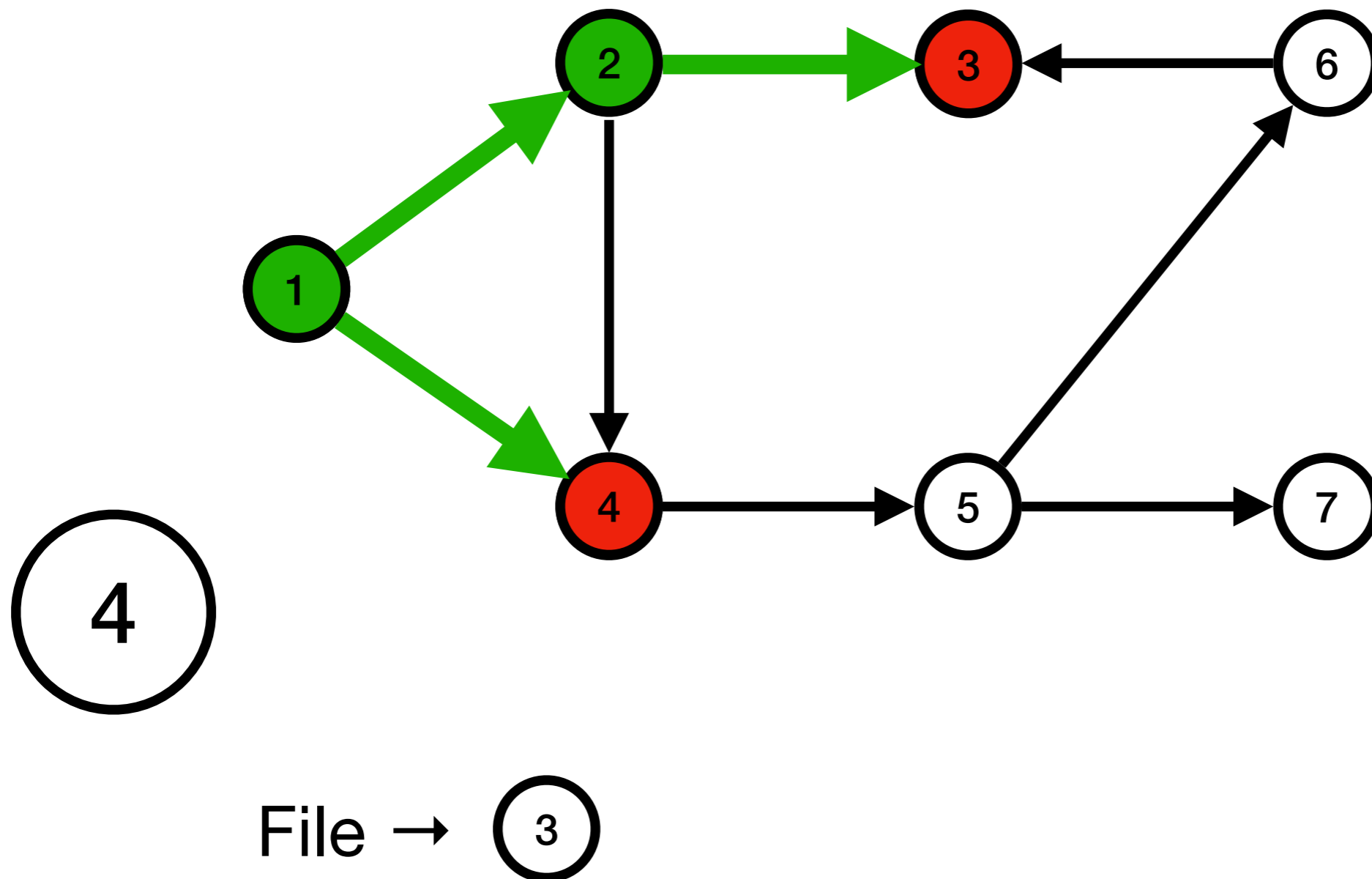


Parcours en largeur

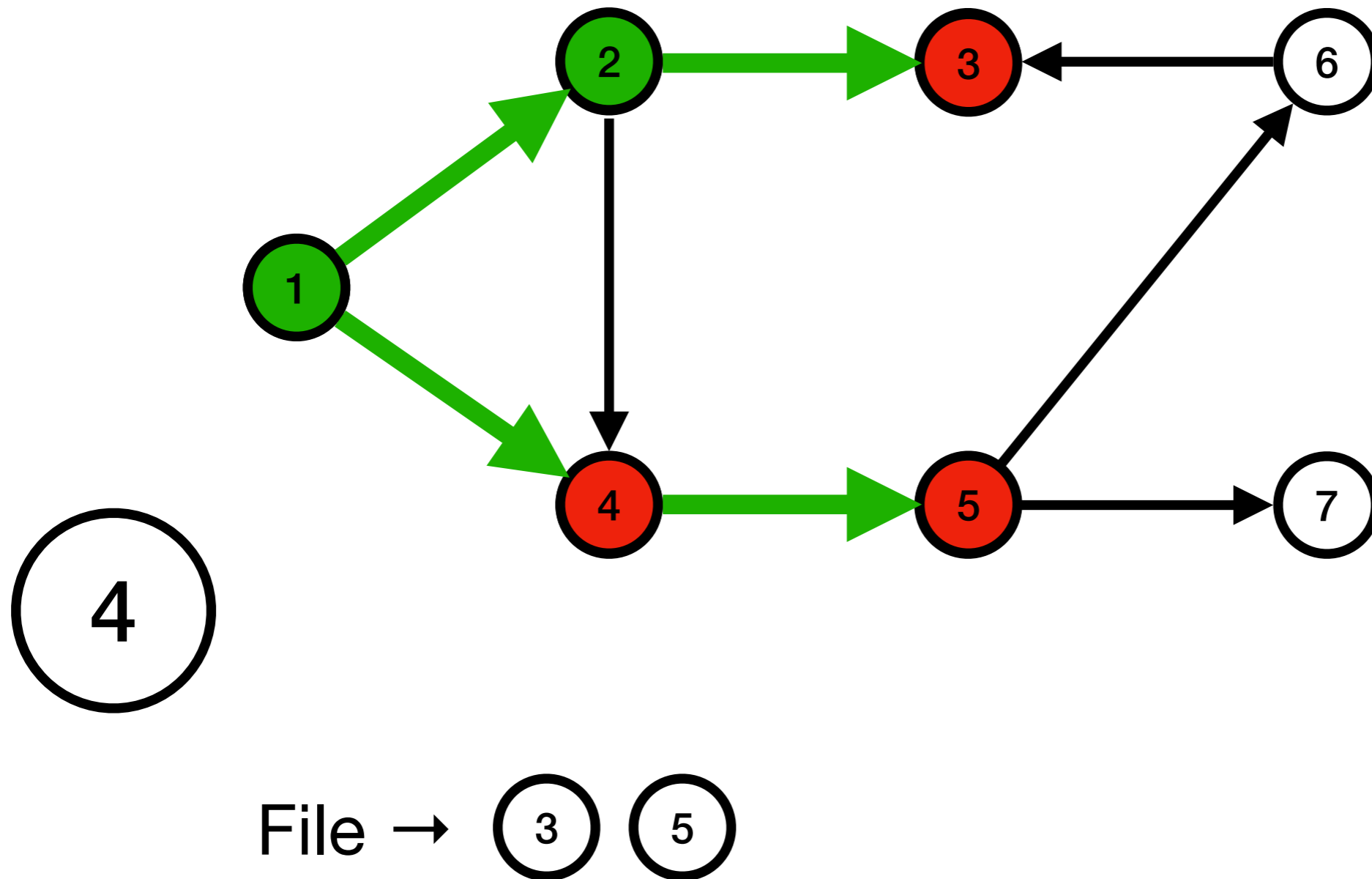


File → (4) (3)

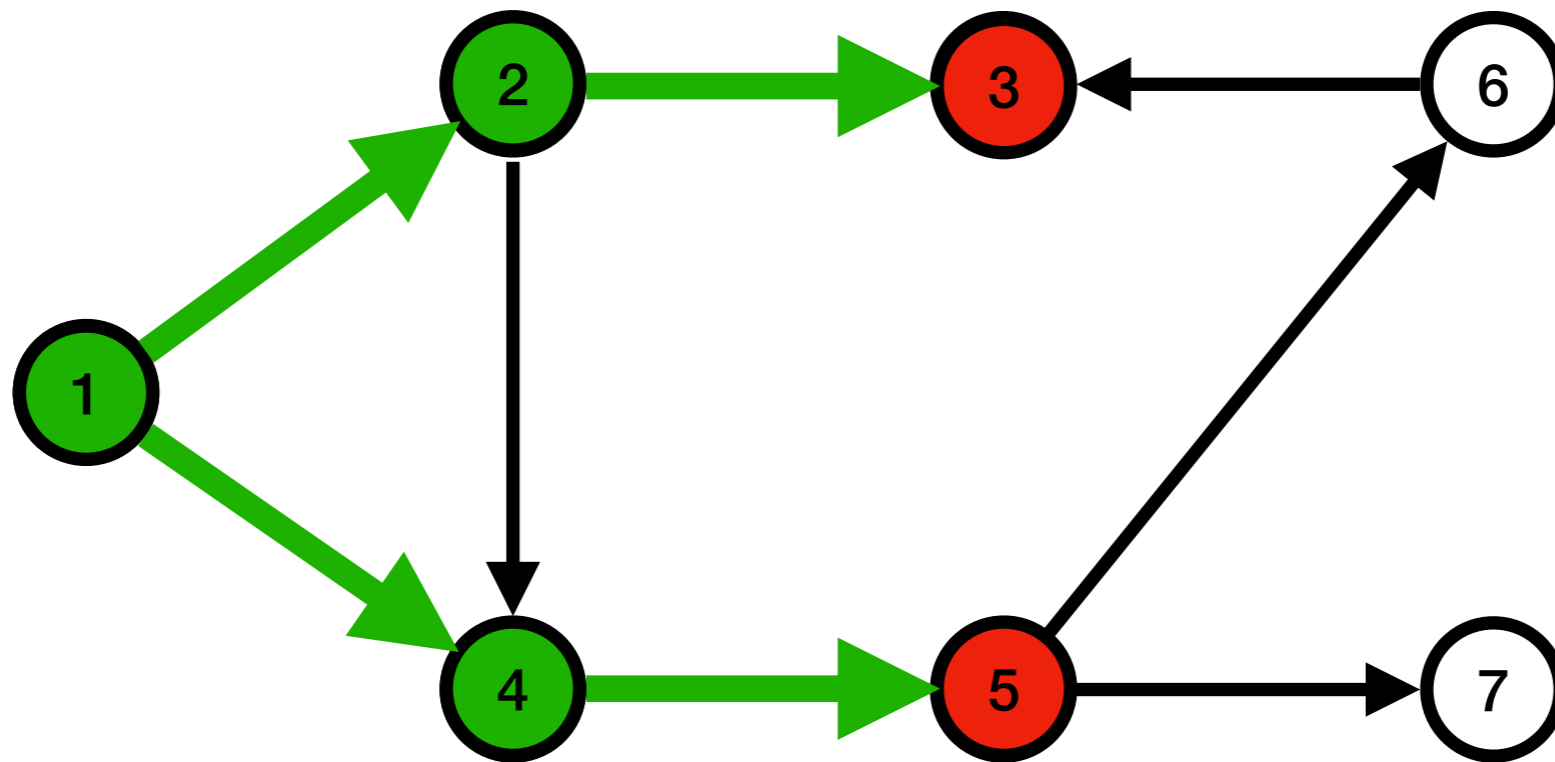
Parcours en largeur



Parcours en largeur

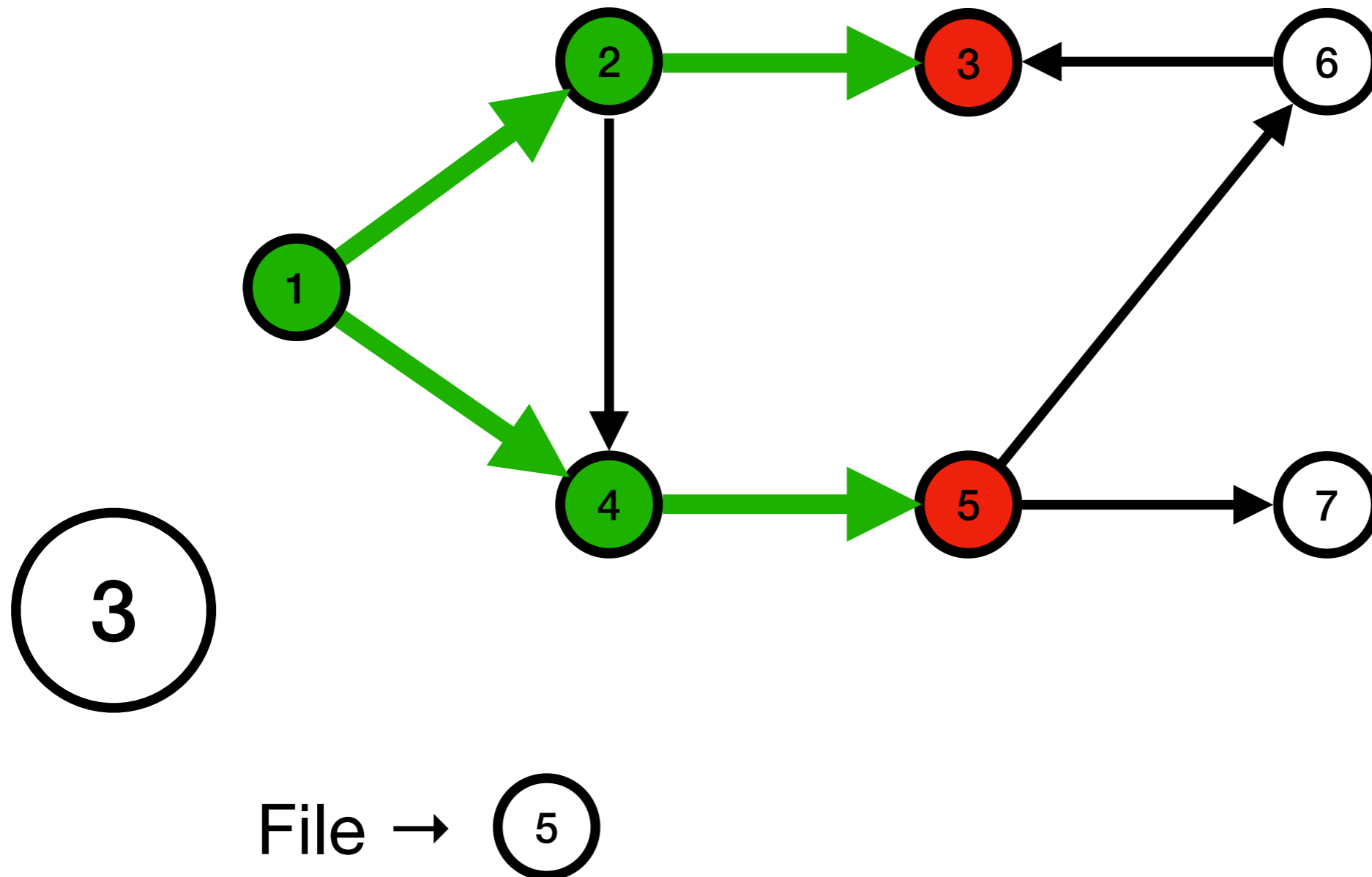


Parcours en largeur

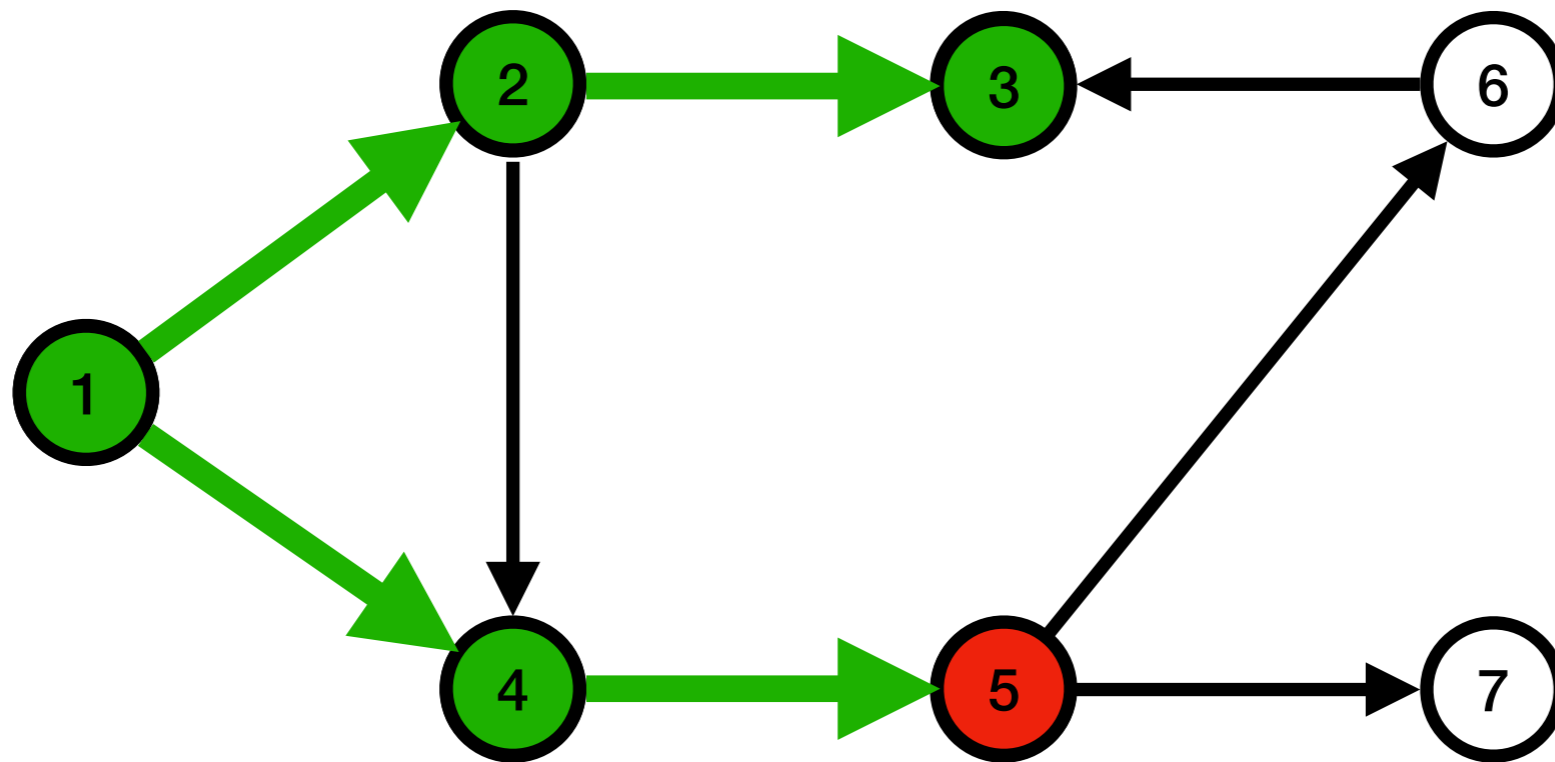


File → (3) (5)

Parcours en largeur

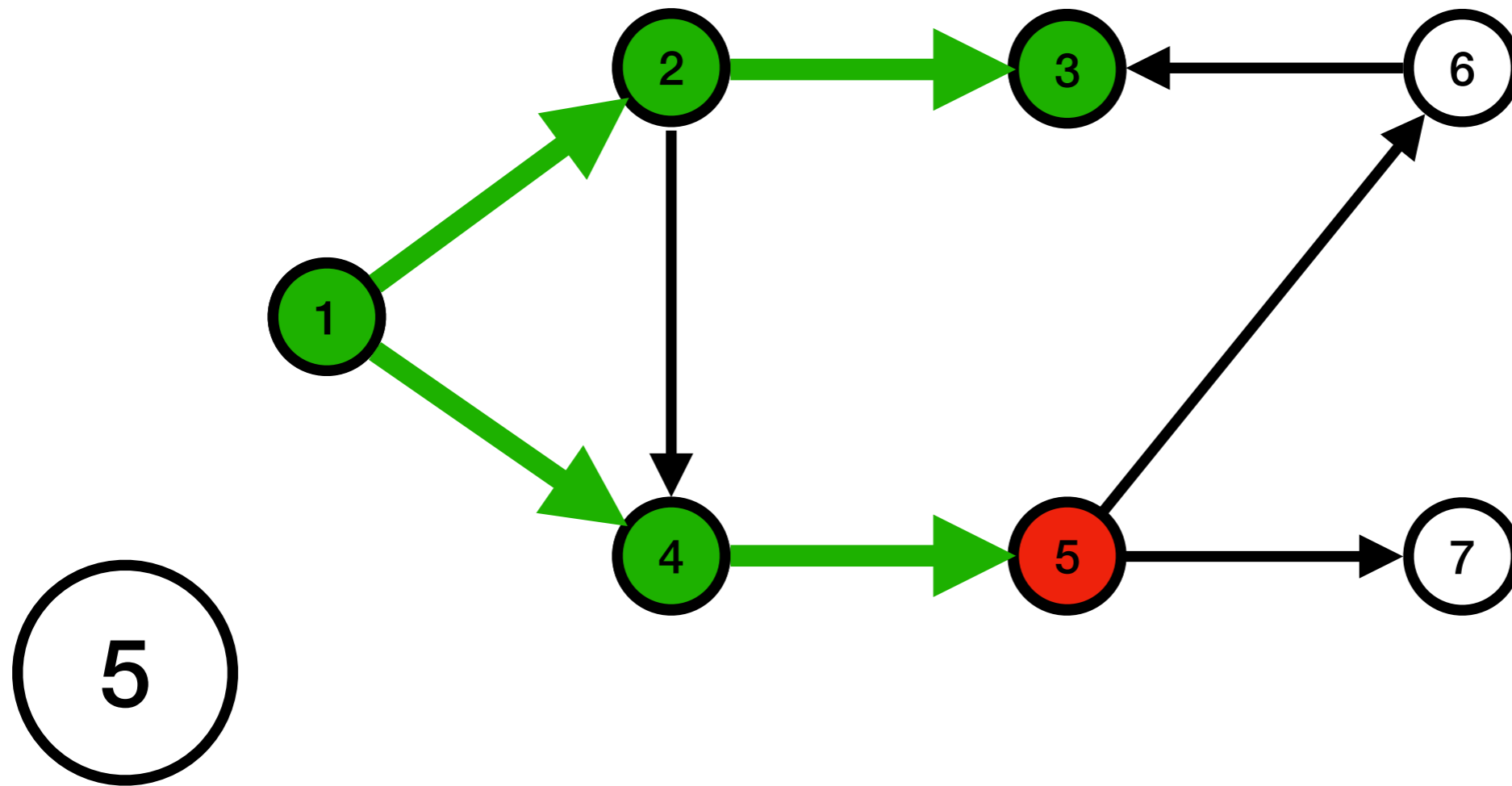


Parcours en largeur



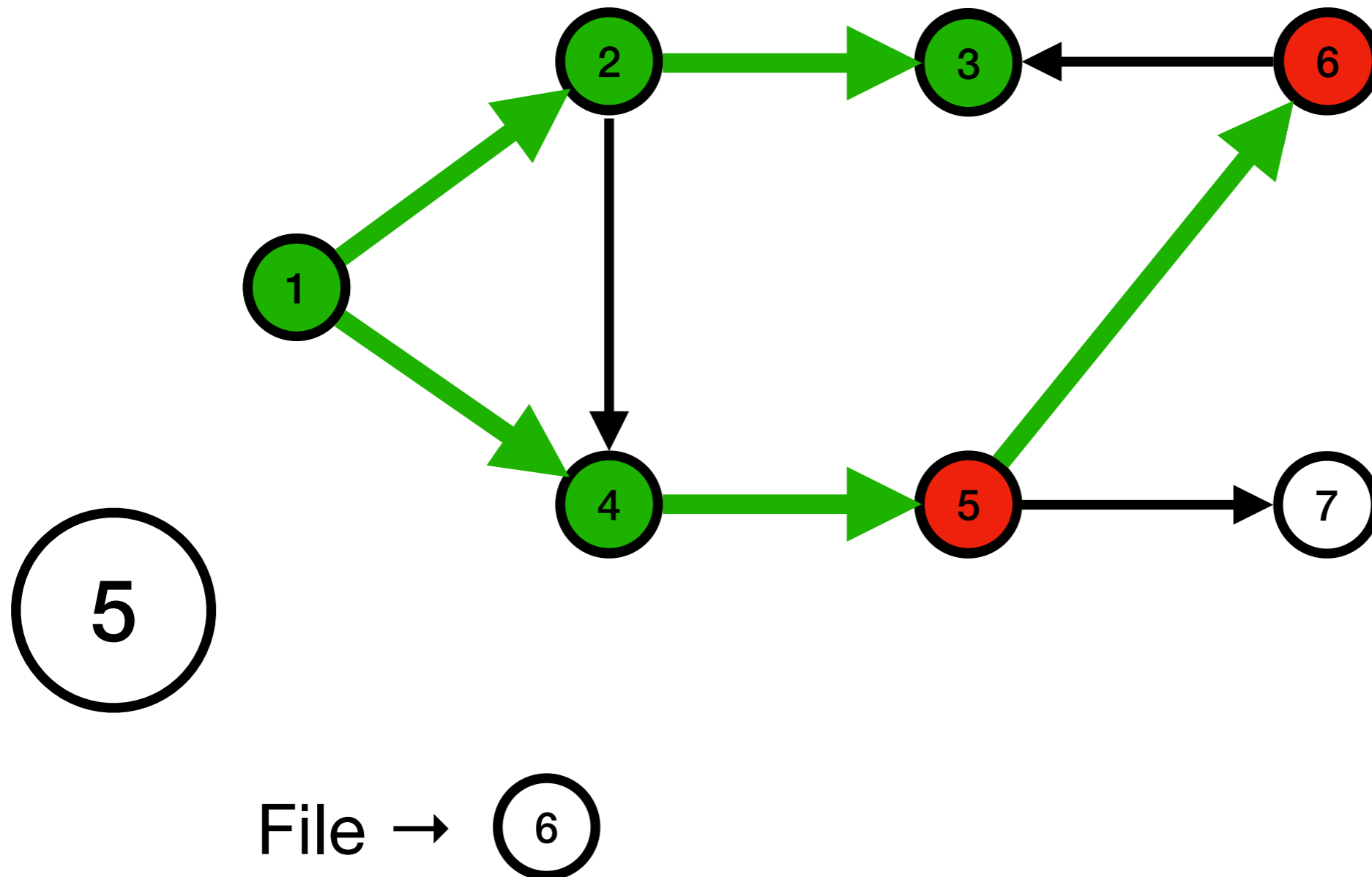
File → (5)

Parcours en largeur

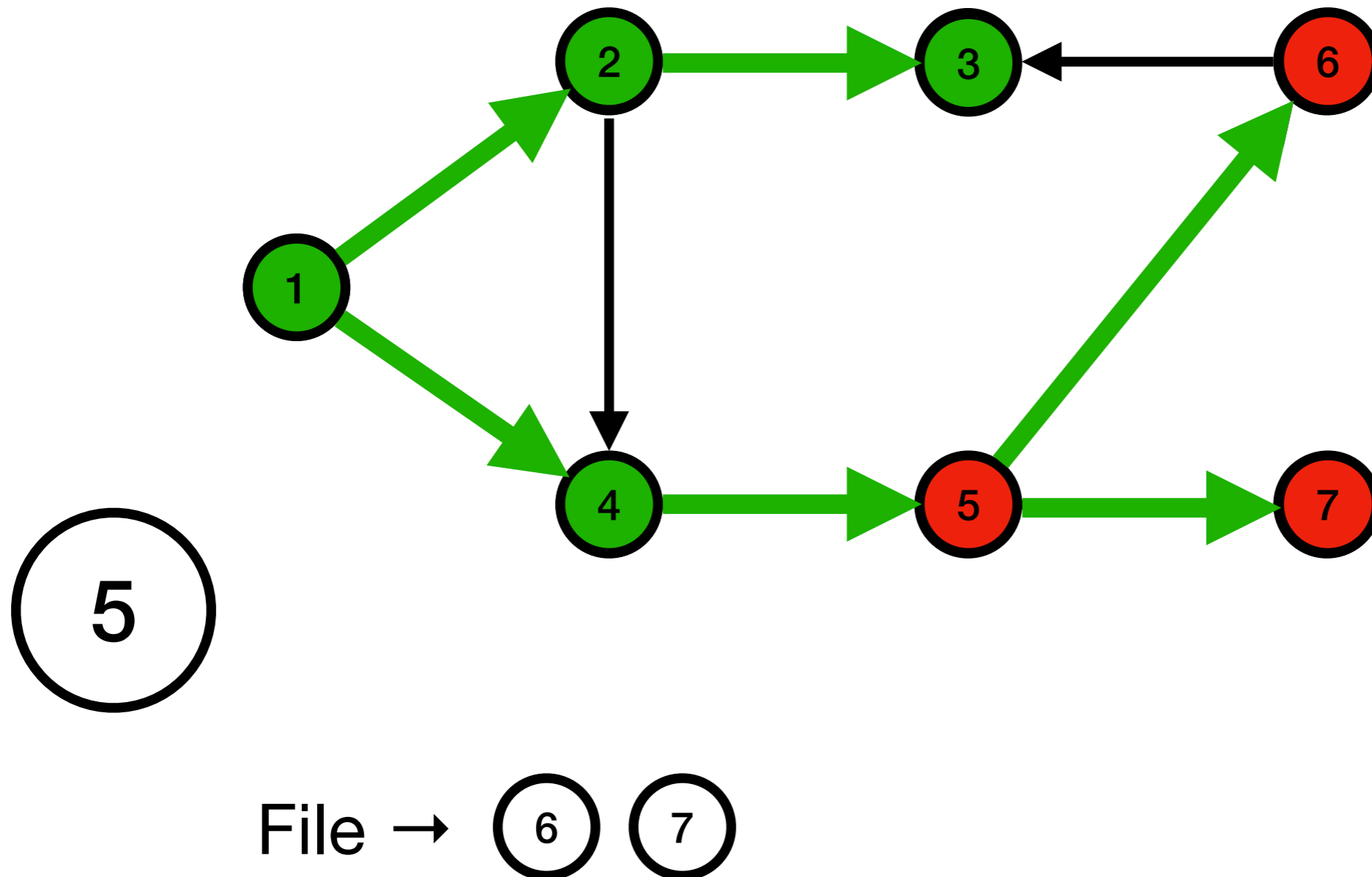


File →

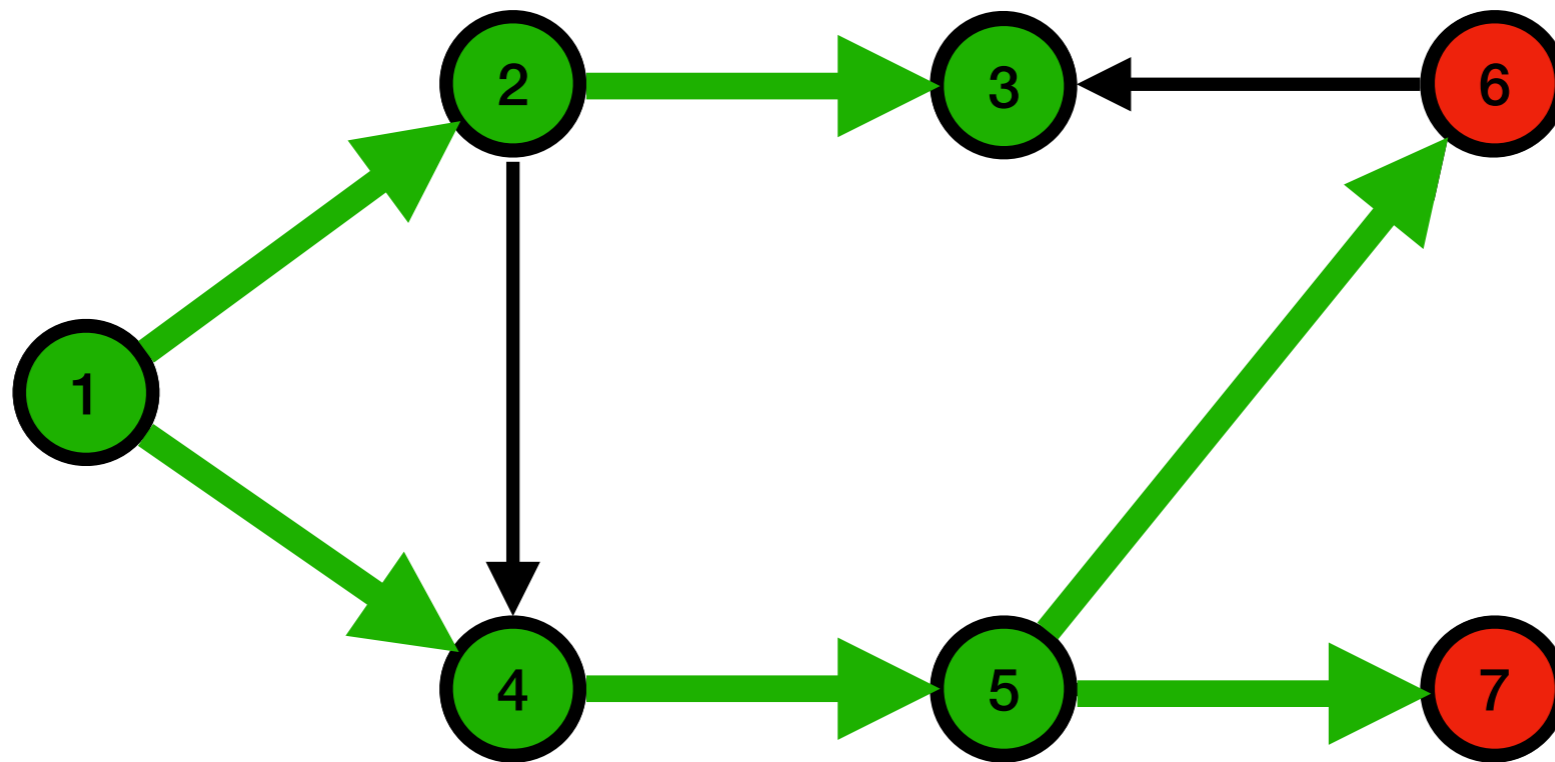
Parcours en largeur



Parcours en largeur

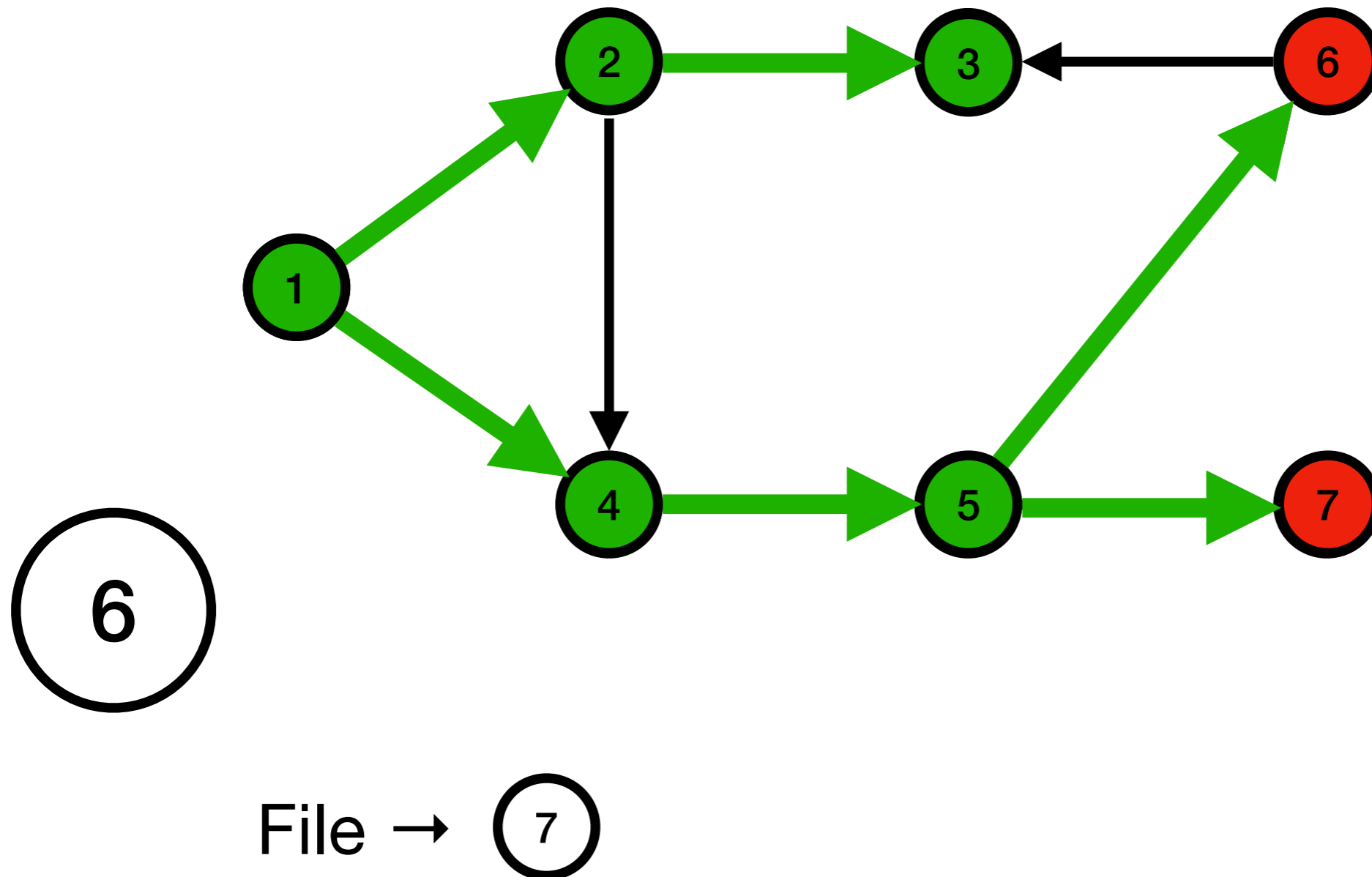


Parcours en largeur

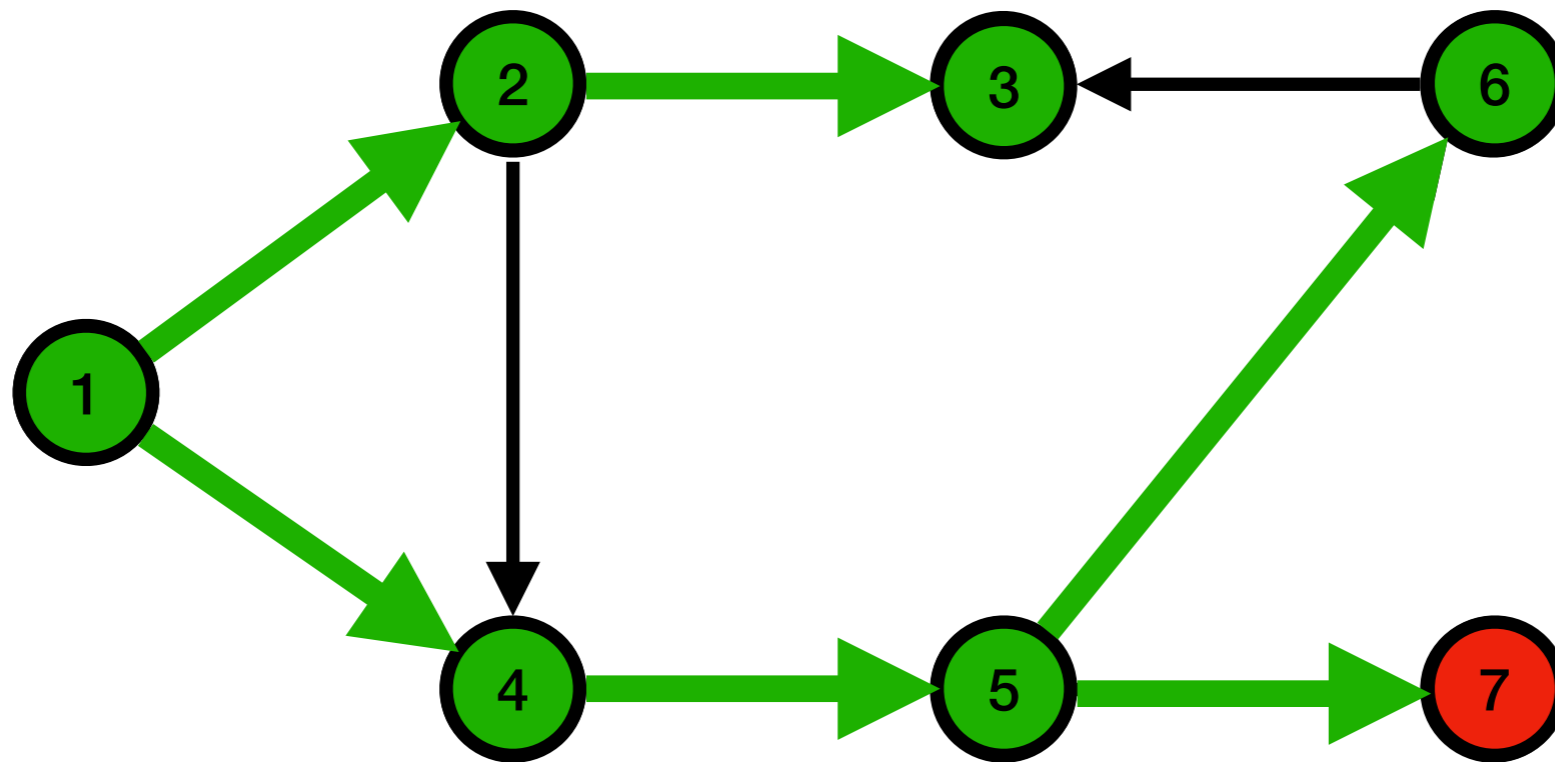


File → (6) (7)

Parcours en largeur

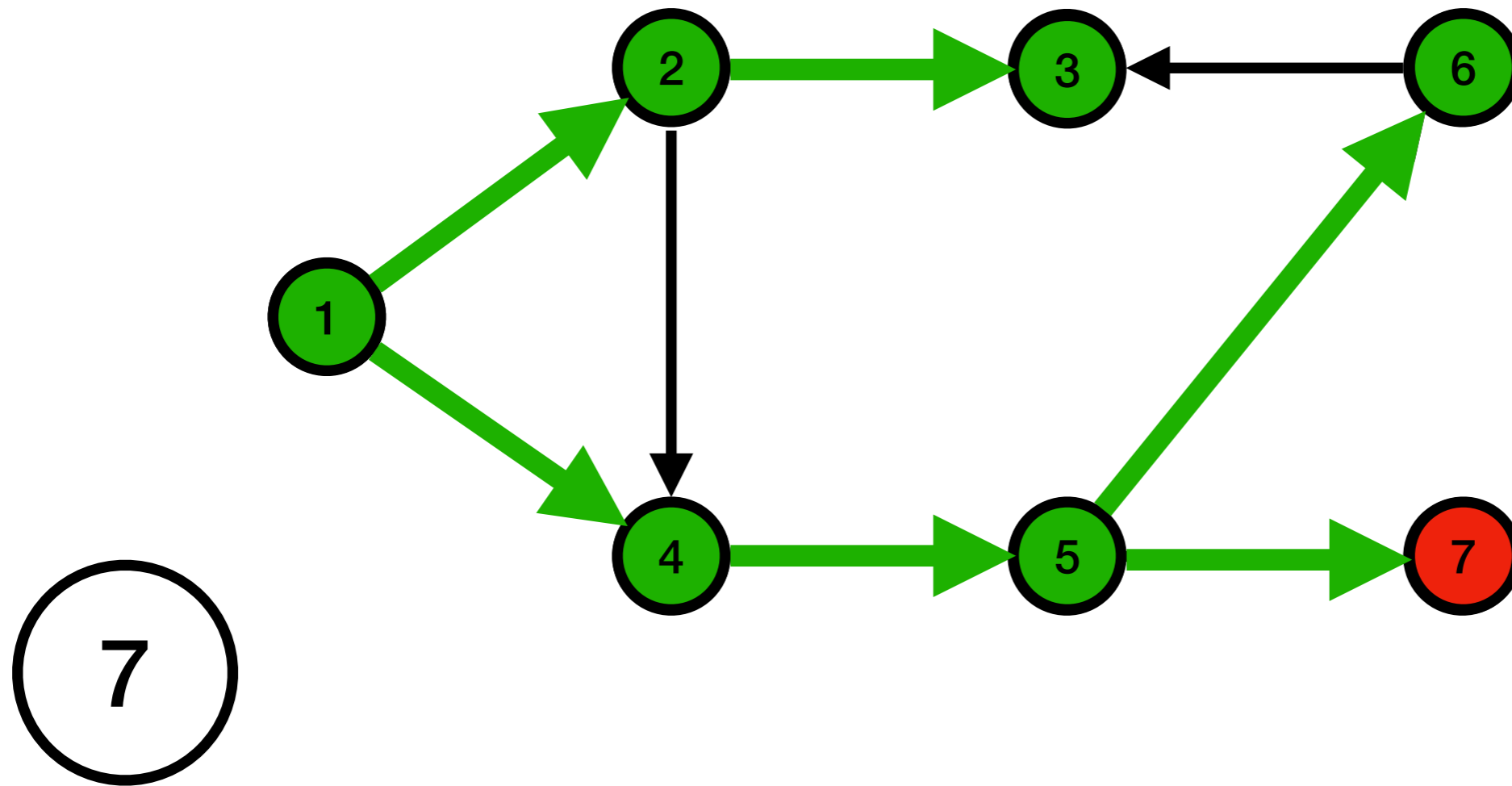


Parcours en largeur



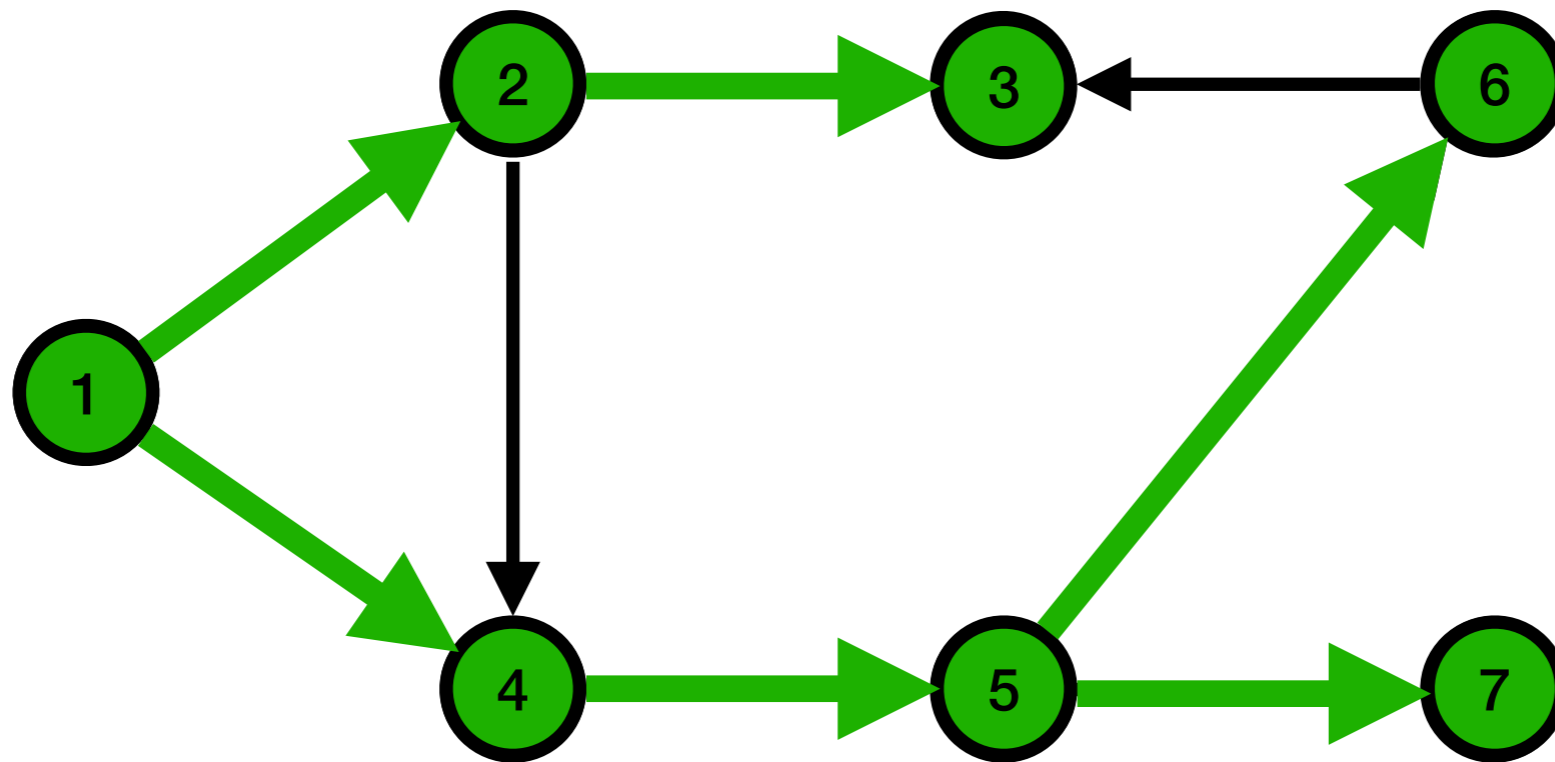
File → (7)

Parcours en largeur



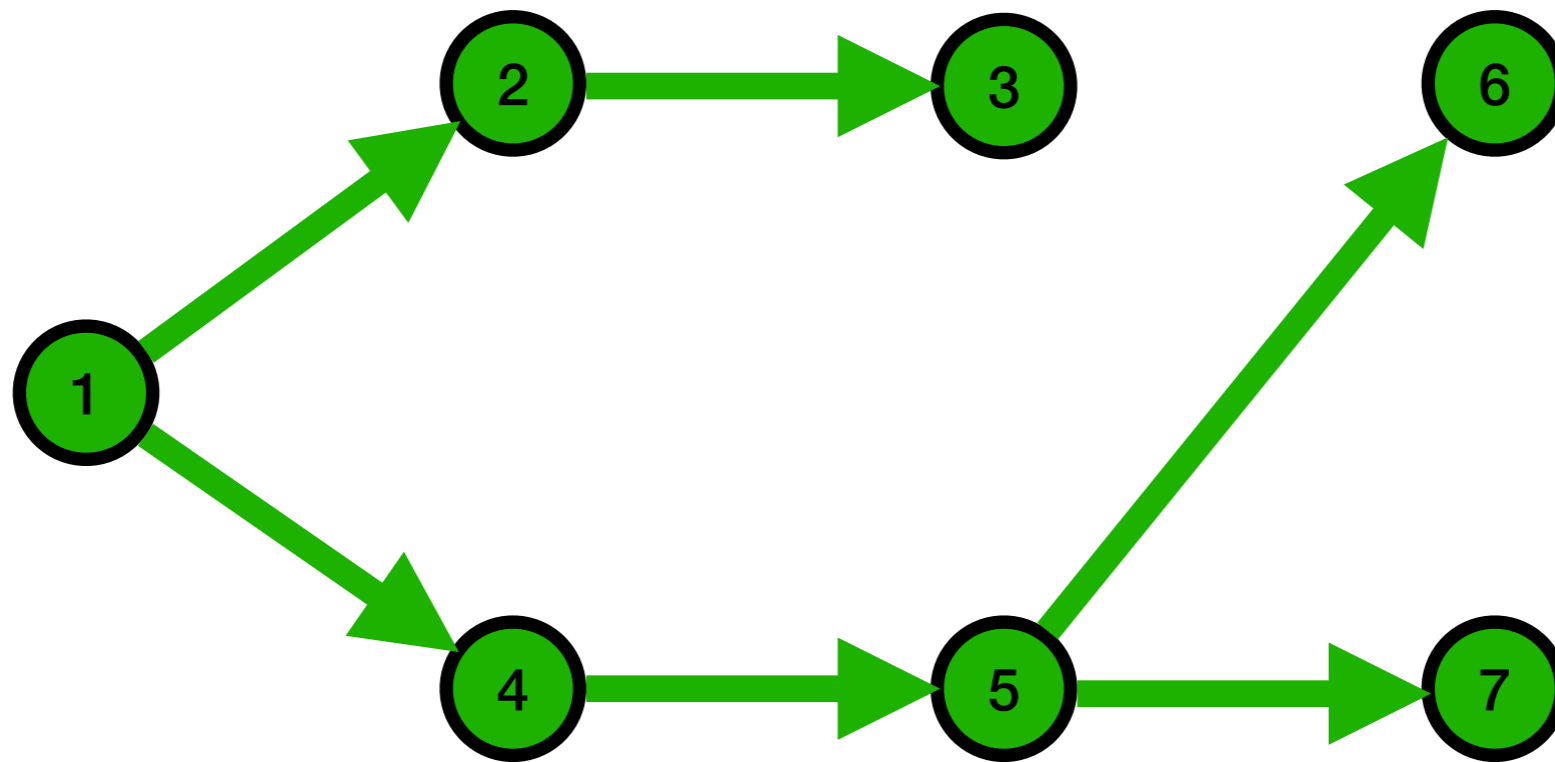
File →

Parcours en largeur



File →

Parcours en largeur



Structure de données « file »

$$F := \emptyset$$

Structure de données « file »

$F := \emptyset$

$F \rightarrow$

Structure de données « file »

enfiler(F, 1)

F →

Structure de données « file »

enfiler(F, 1)

F → 1

Structure de données « file »

enfiler(F, 2)

F → 1 2

Structure de données « file »

$x := \text{défiler}(F)$

$F \rightarrow 2$

Structure de données « file »

enfiler(F, 3)

F → 2 3

Structure de données « file »

enfiler(F, 4)

F → 2 3 4

Structure de données « file »

enfiler(F, 5)

F → 2 3 4 5

Structure de données « file »

$x := \text{défiler}(F)$

$F \rightarrow 3 \quad 4 \quad 5$

Structure de données « file »

$x := \text{défiler}(F)$

$F \rightarrow 4 \ 5$

Structure de données « file »

$x := \text{défiler}(F)$

$F \rightarrow 5$

Structure de données « file »

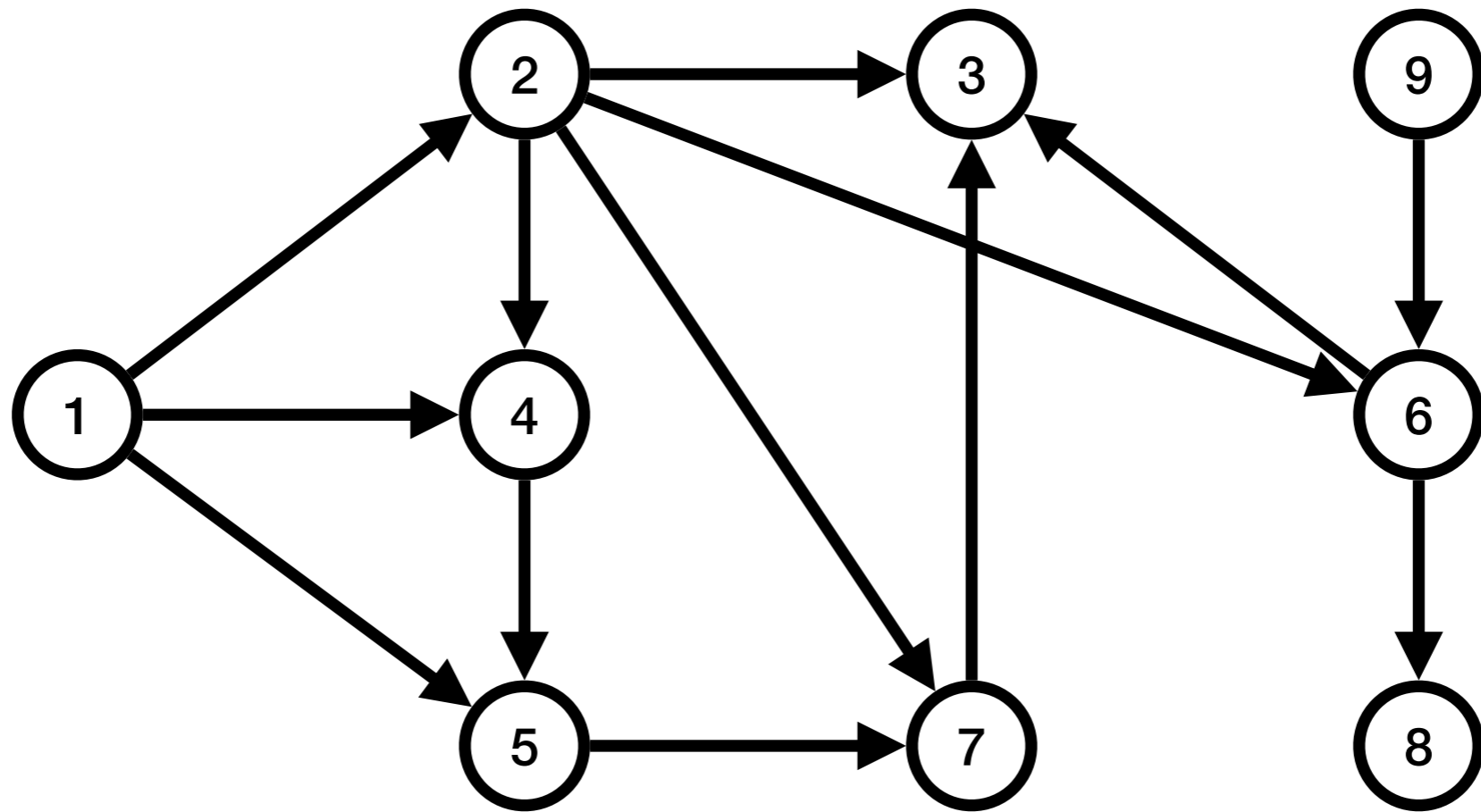
$x := \text{défiler}(F)$

$F \rightarrow$

Structure de données « file »

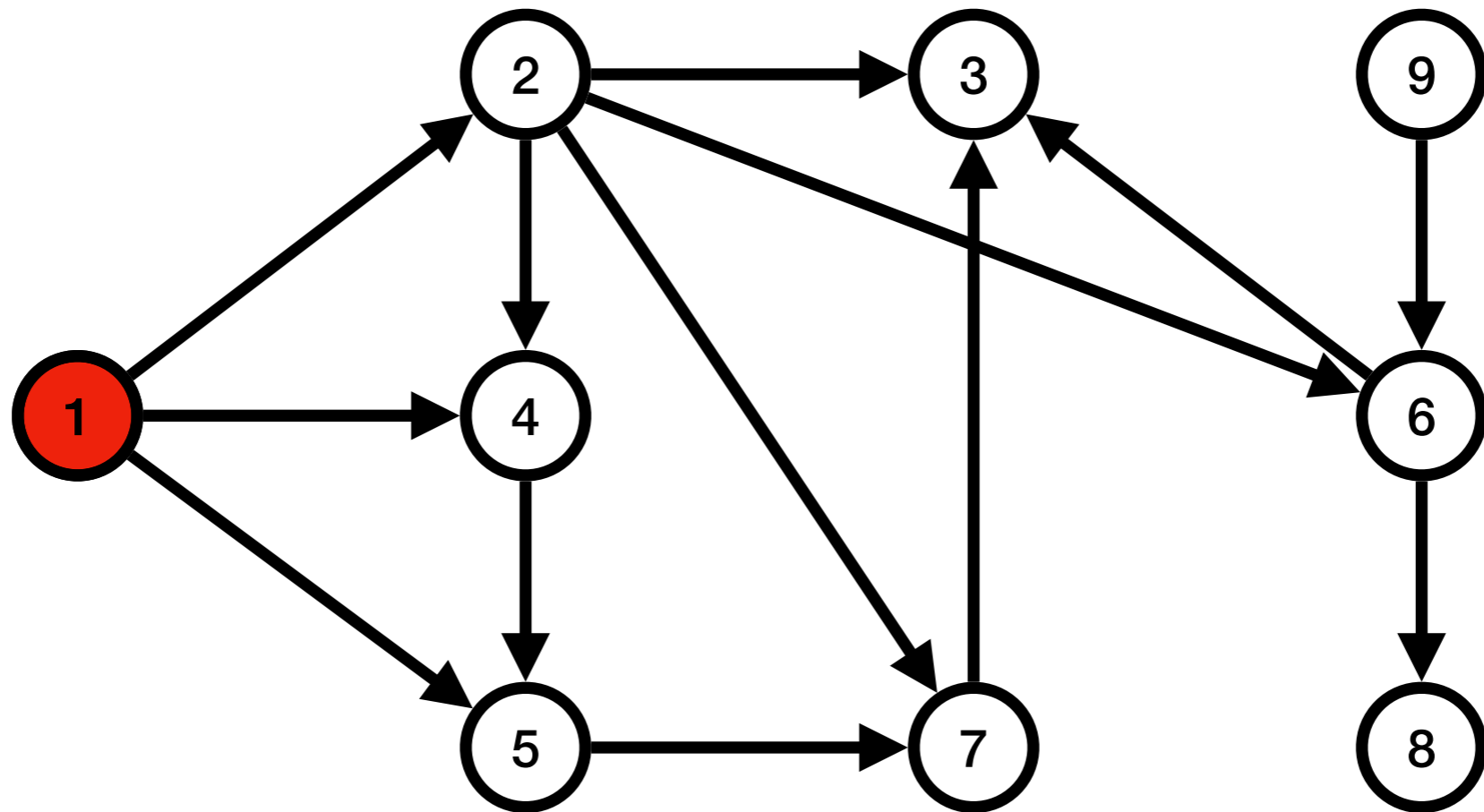
F →

Un autre exemple



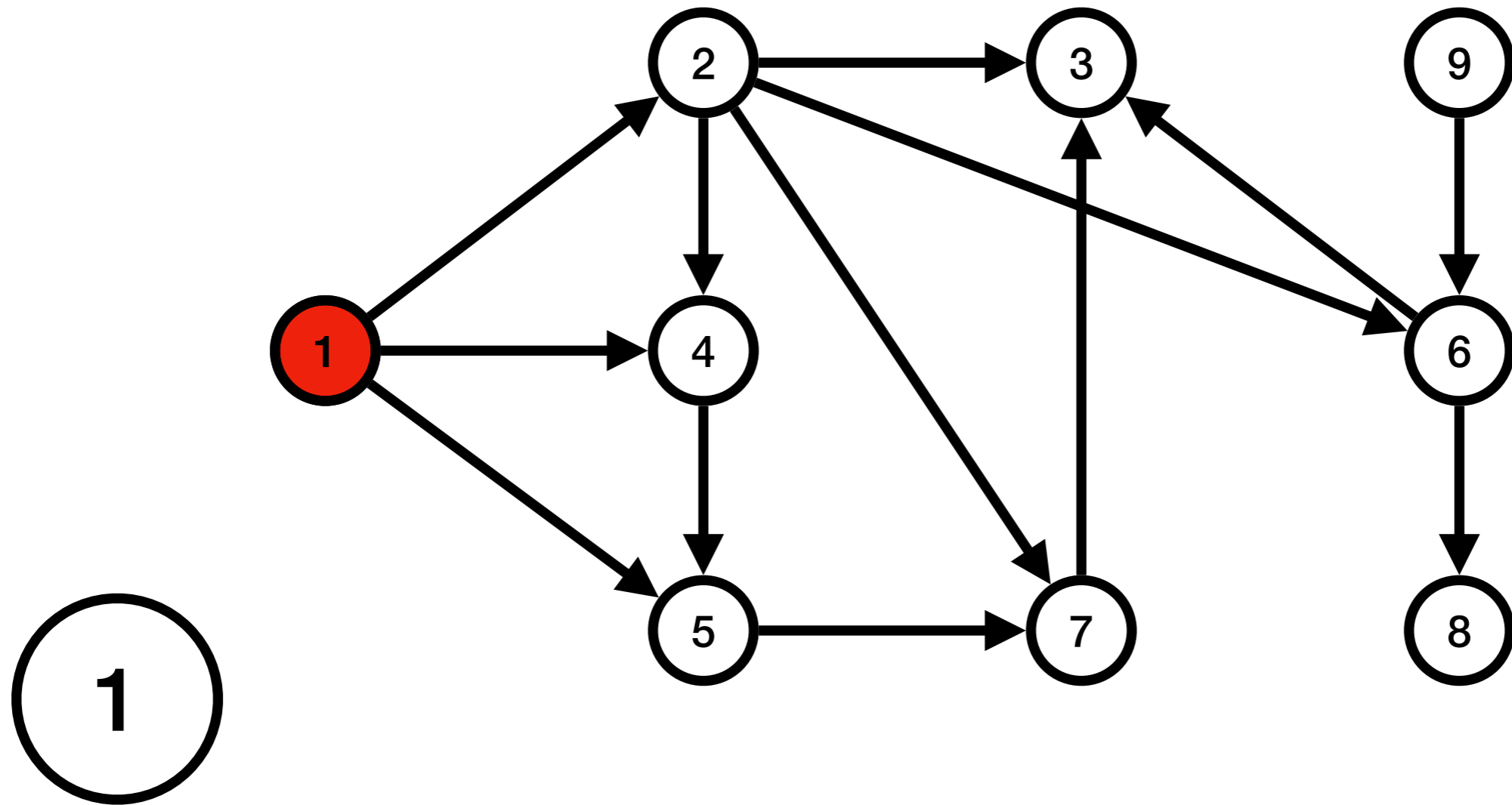
File →

Un autre exemple



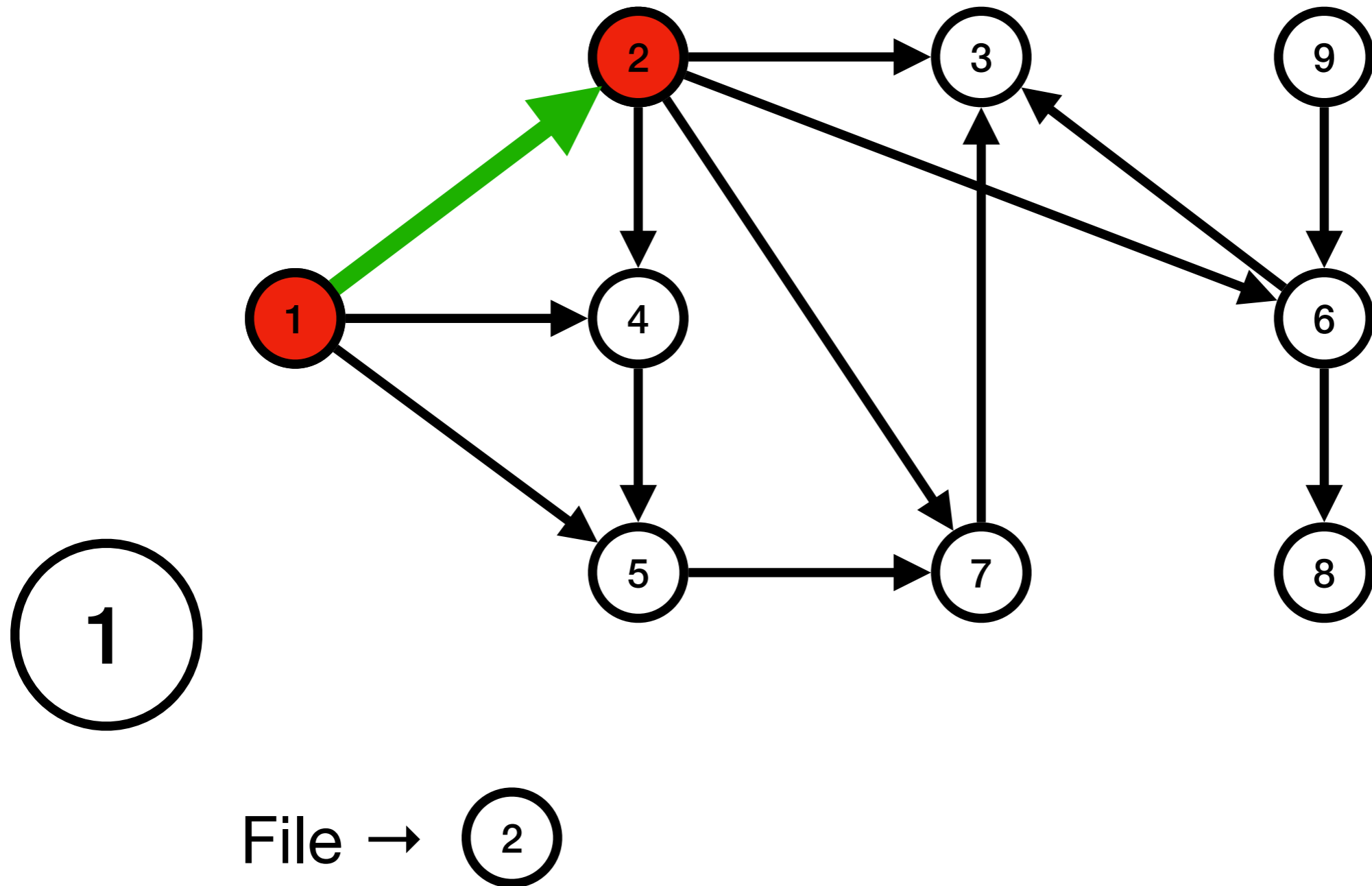
File → **1**

Un autre exemple

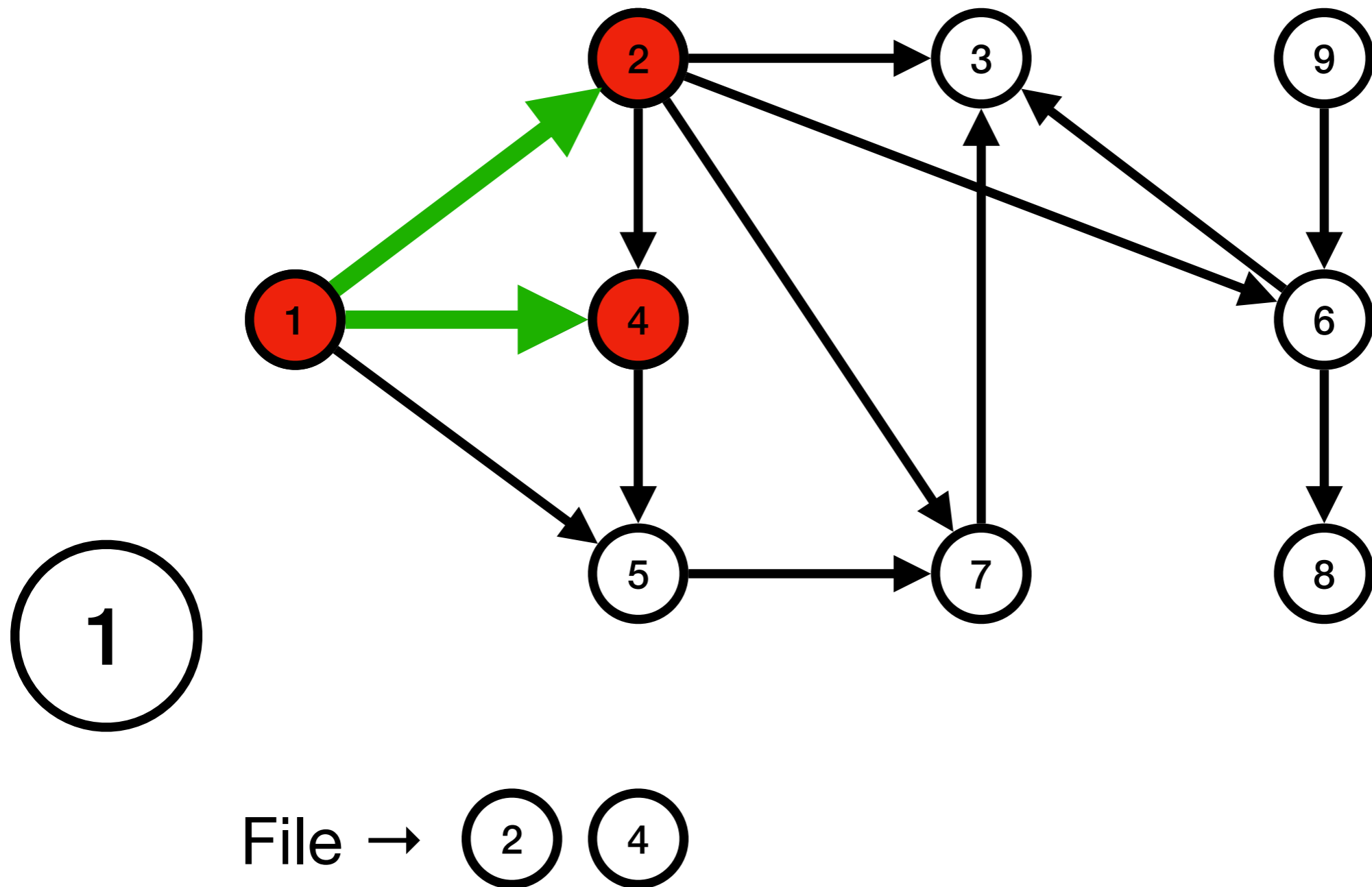


File →

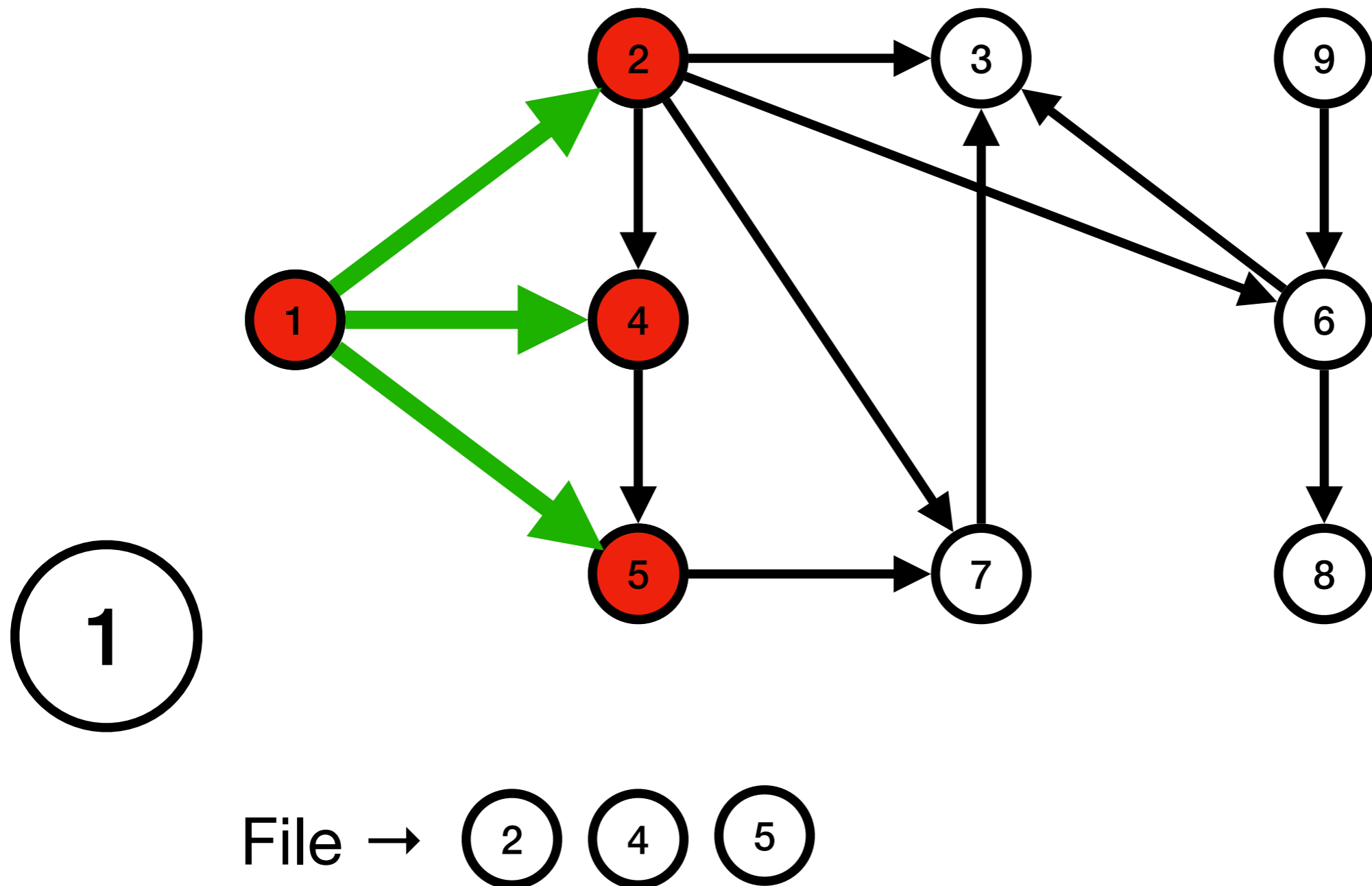
Un autre exemple



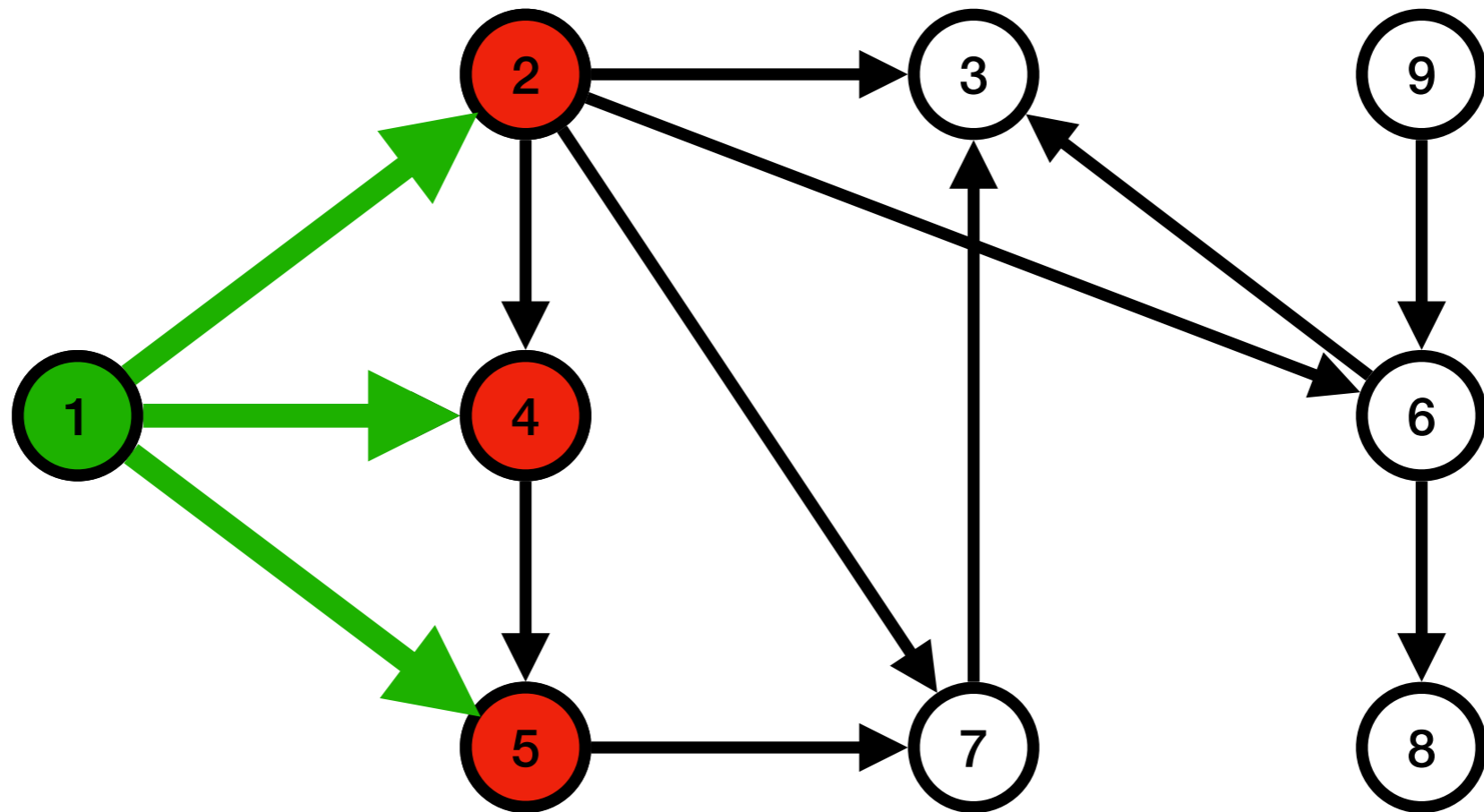
Un autre exemple



Un autre exemple

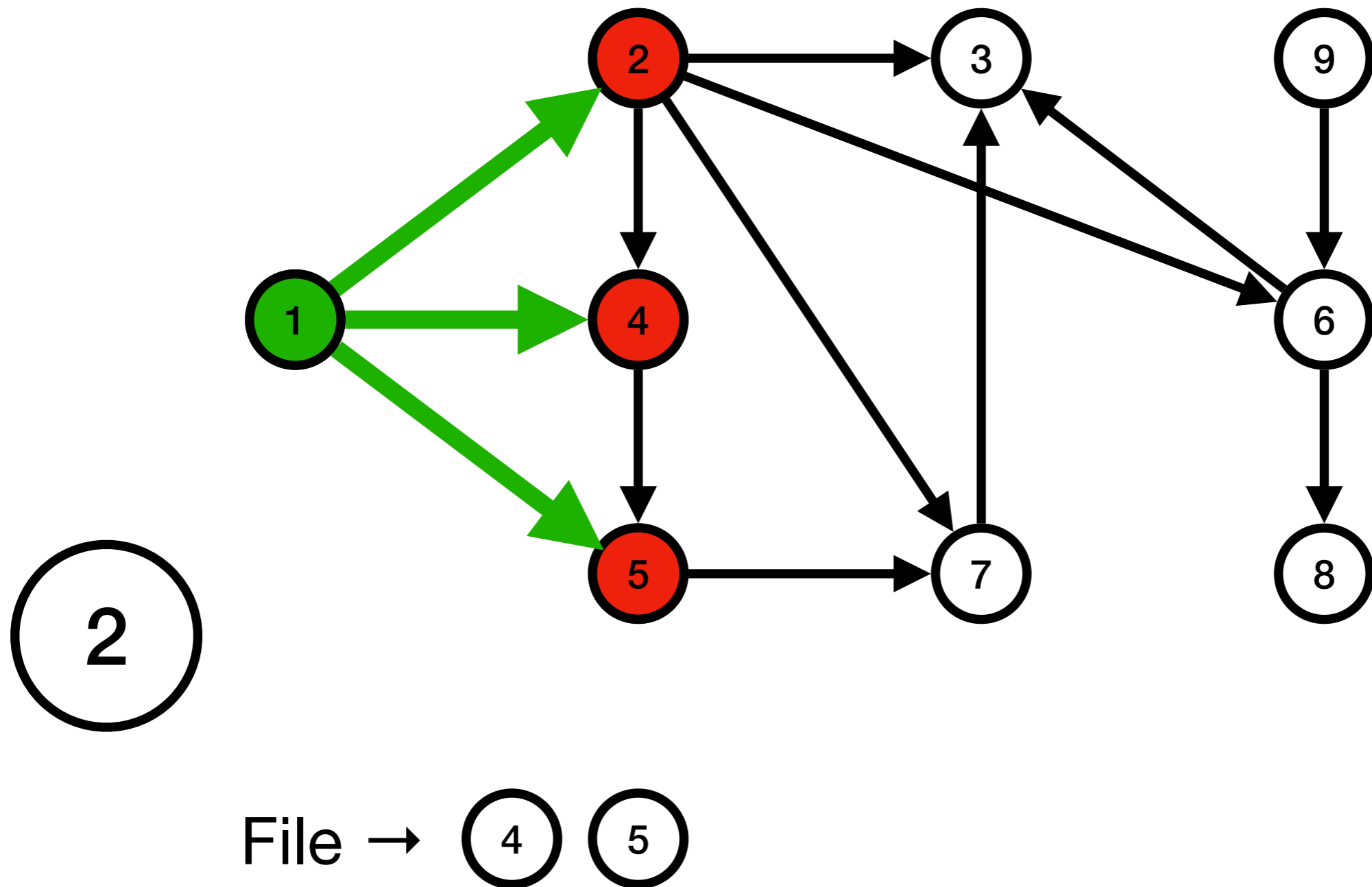


Un autre exemple

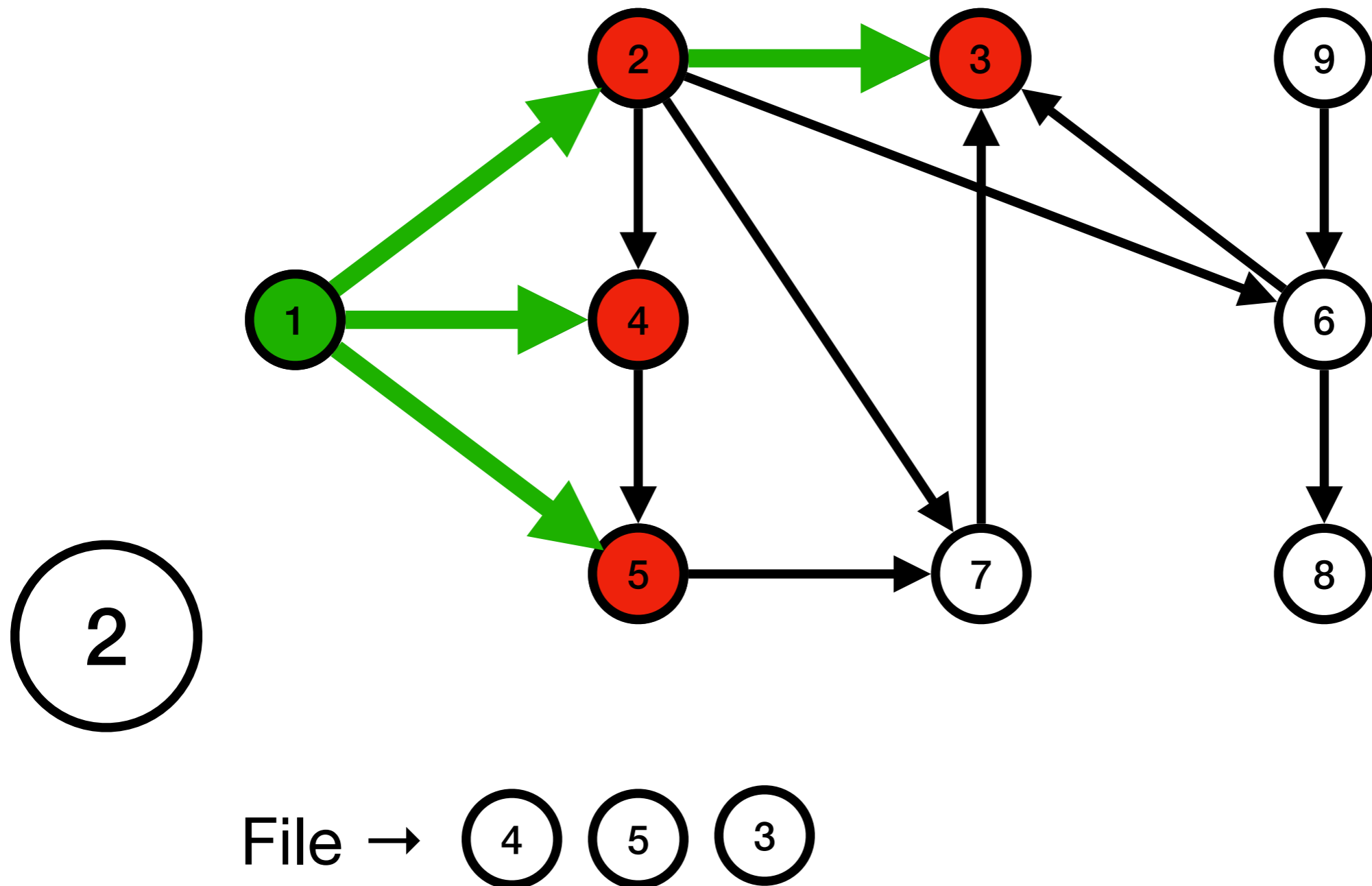


File → (2) (4) (5)

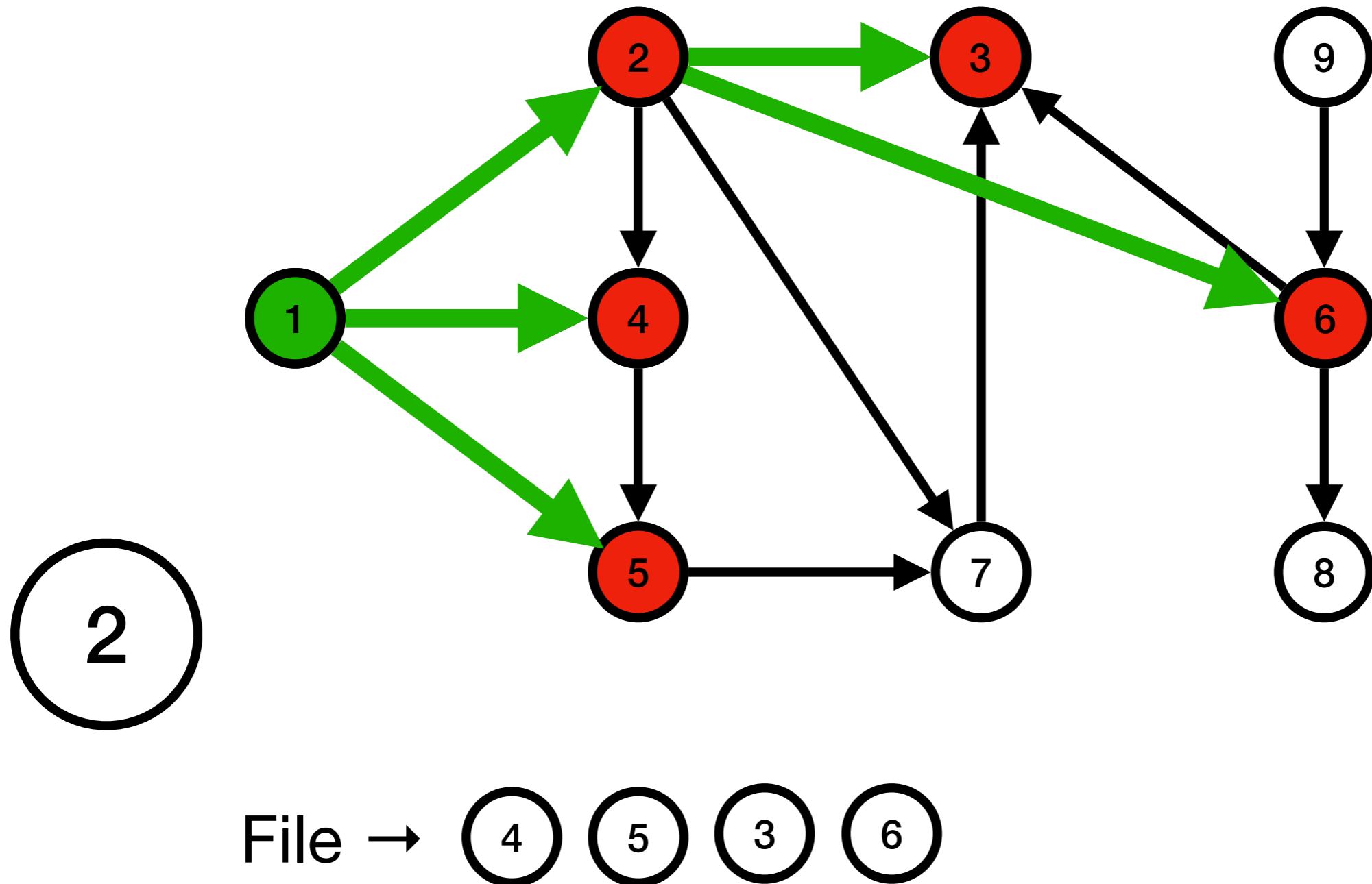
Un autre exemple



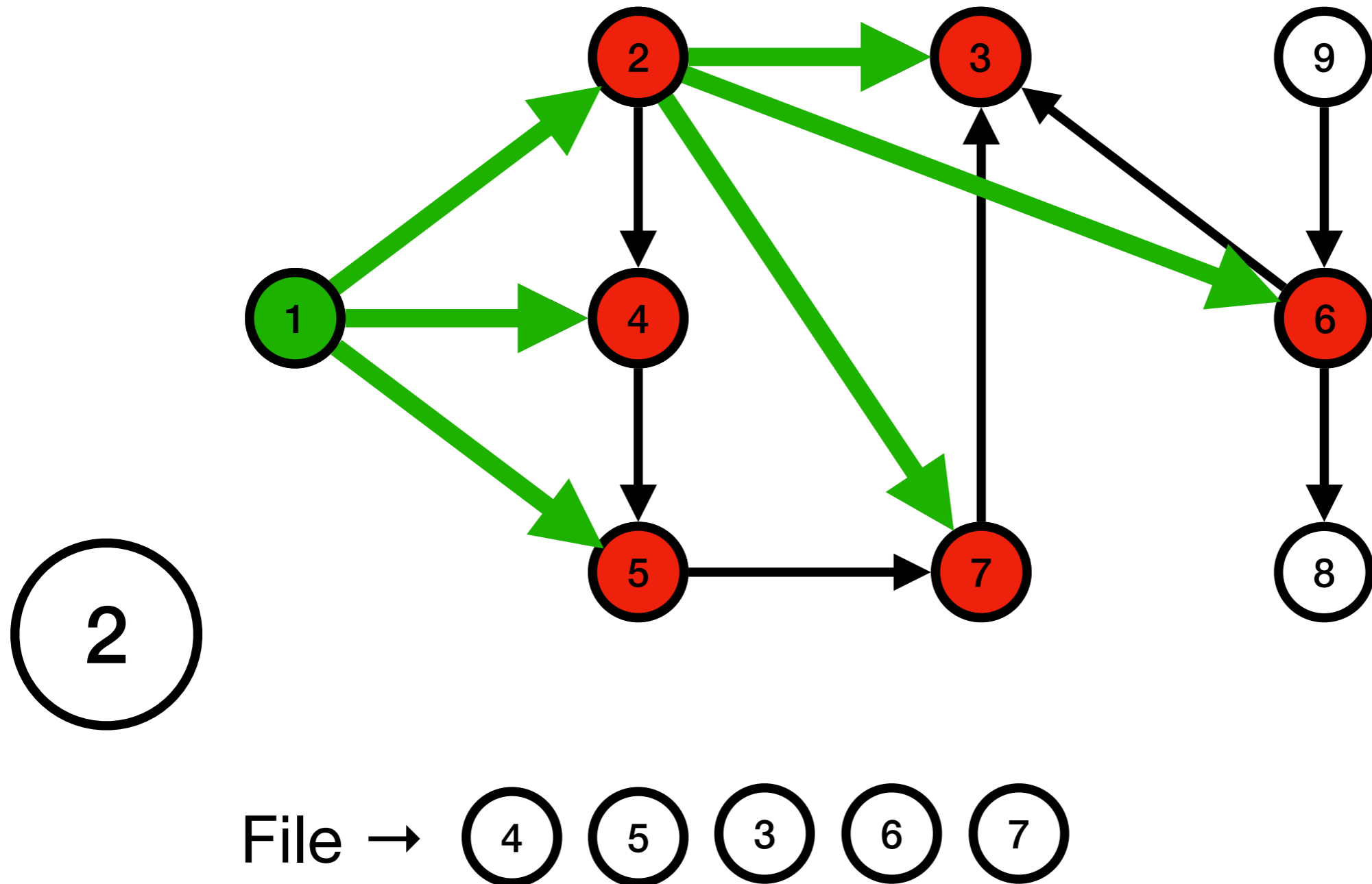
Un autre exemple



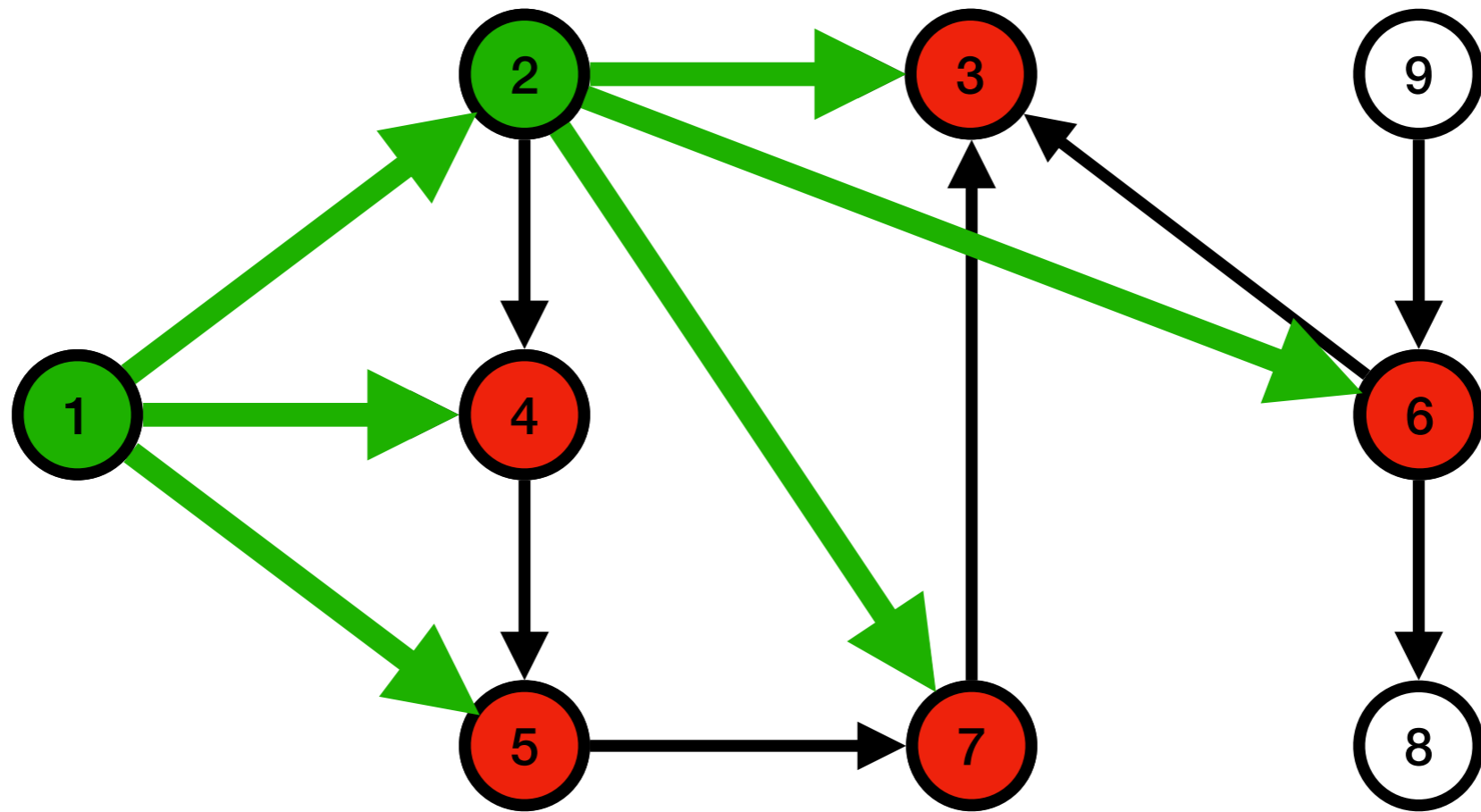
Un autre exemple



Un autre exemple

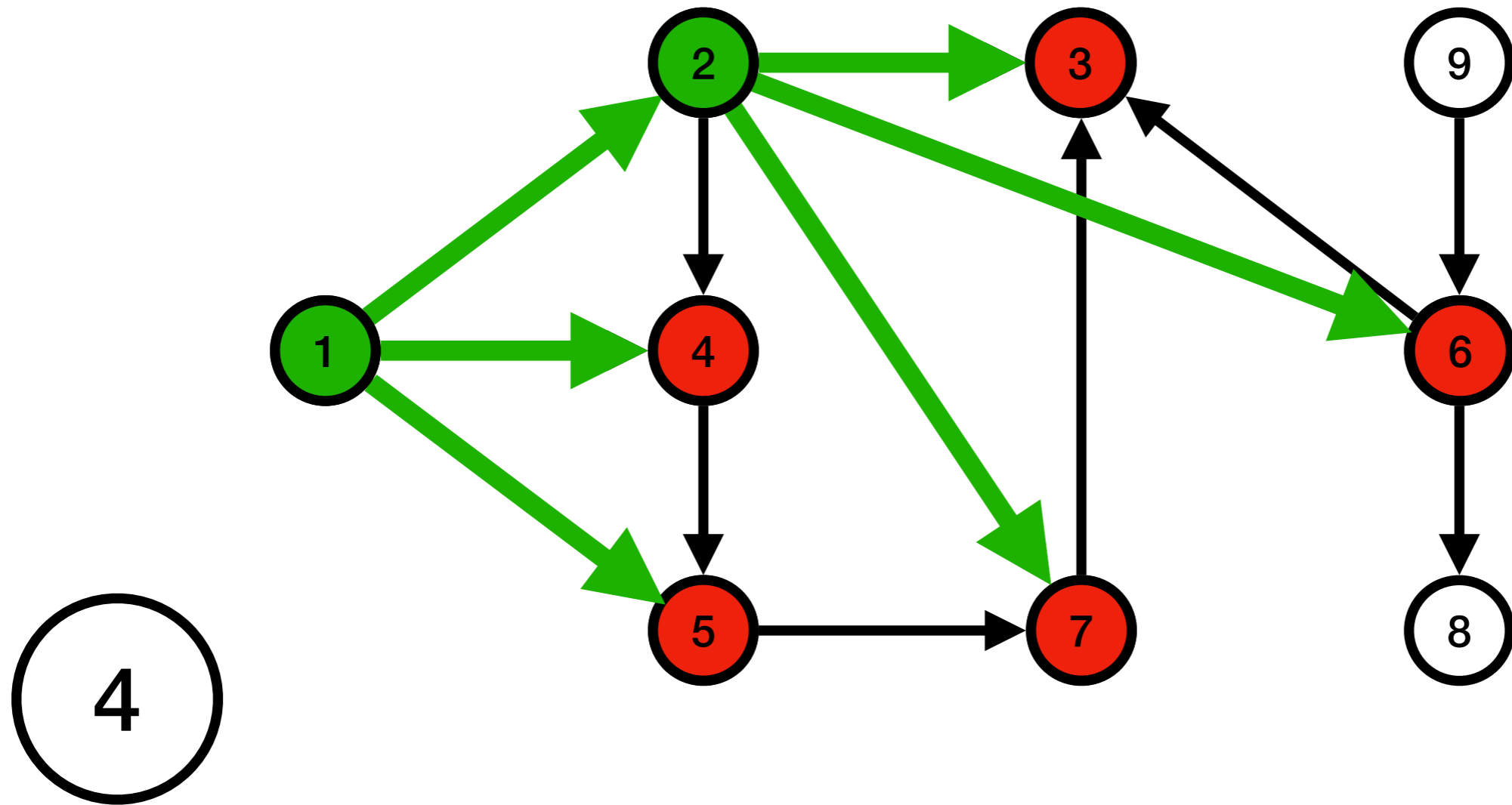


Un autre exemple



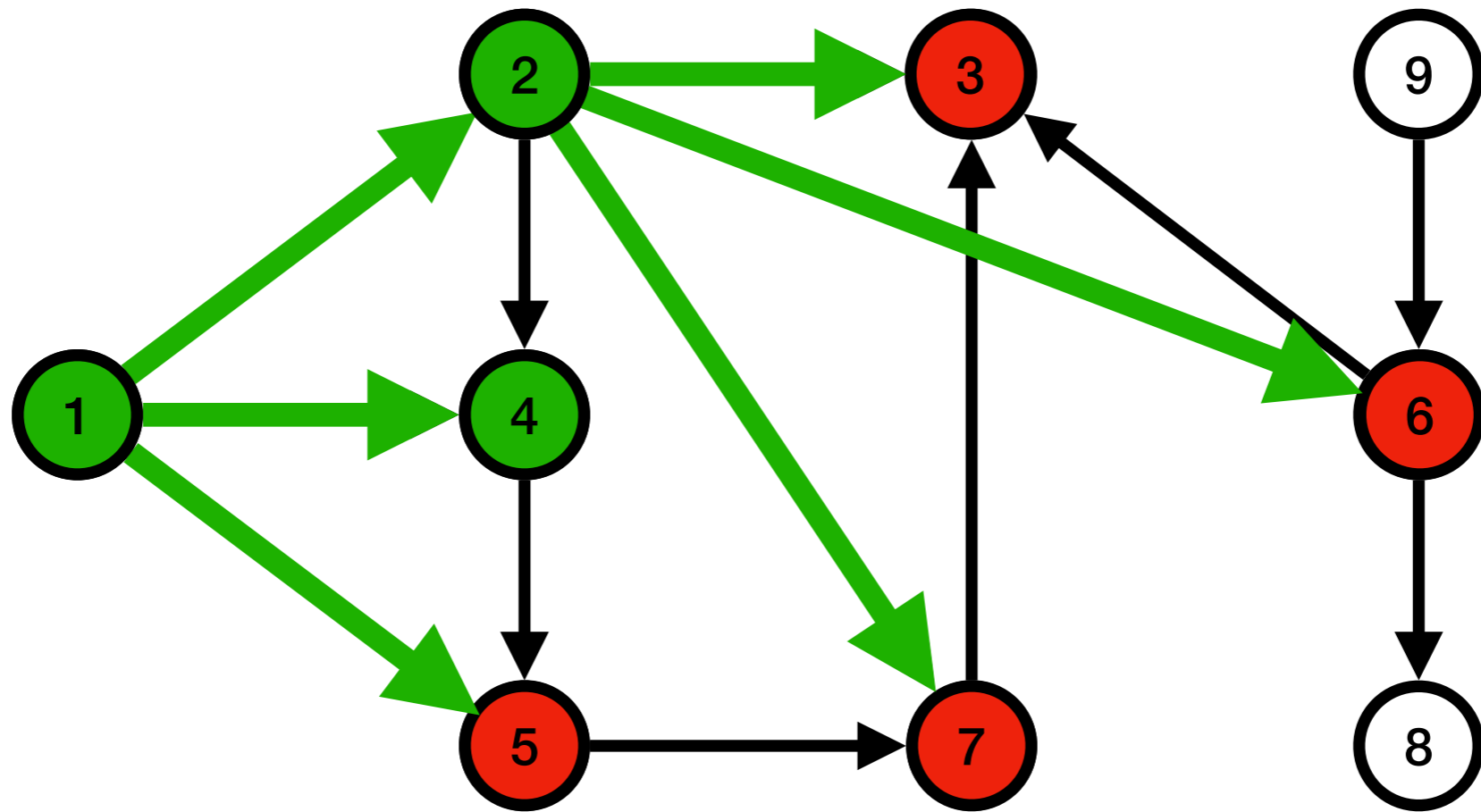
File → (4) (5) (3) (6) (7)

Un autre exemple



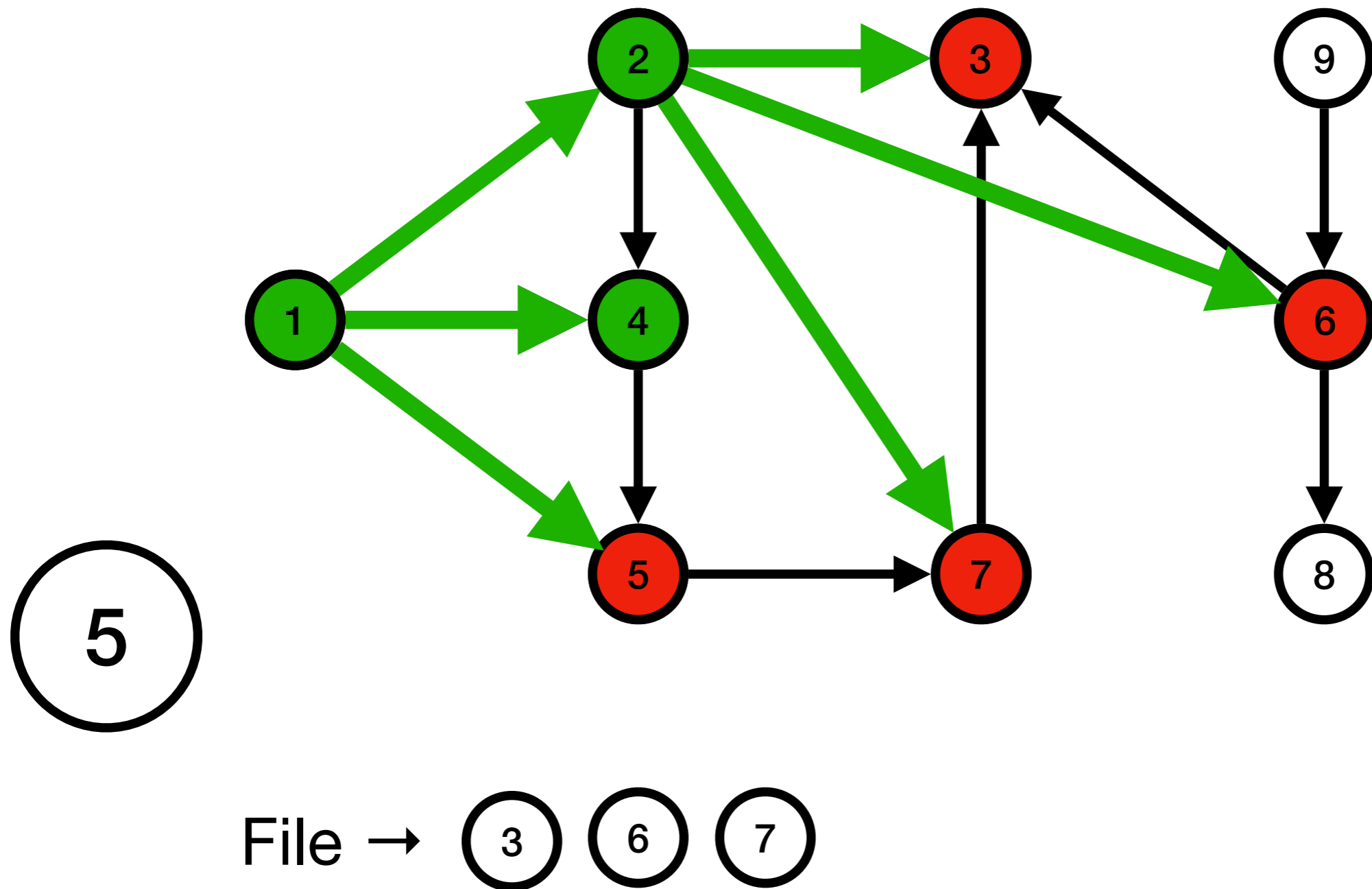
File → 5 3 6 7

Un autre exemple

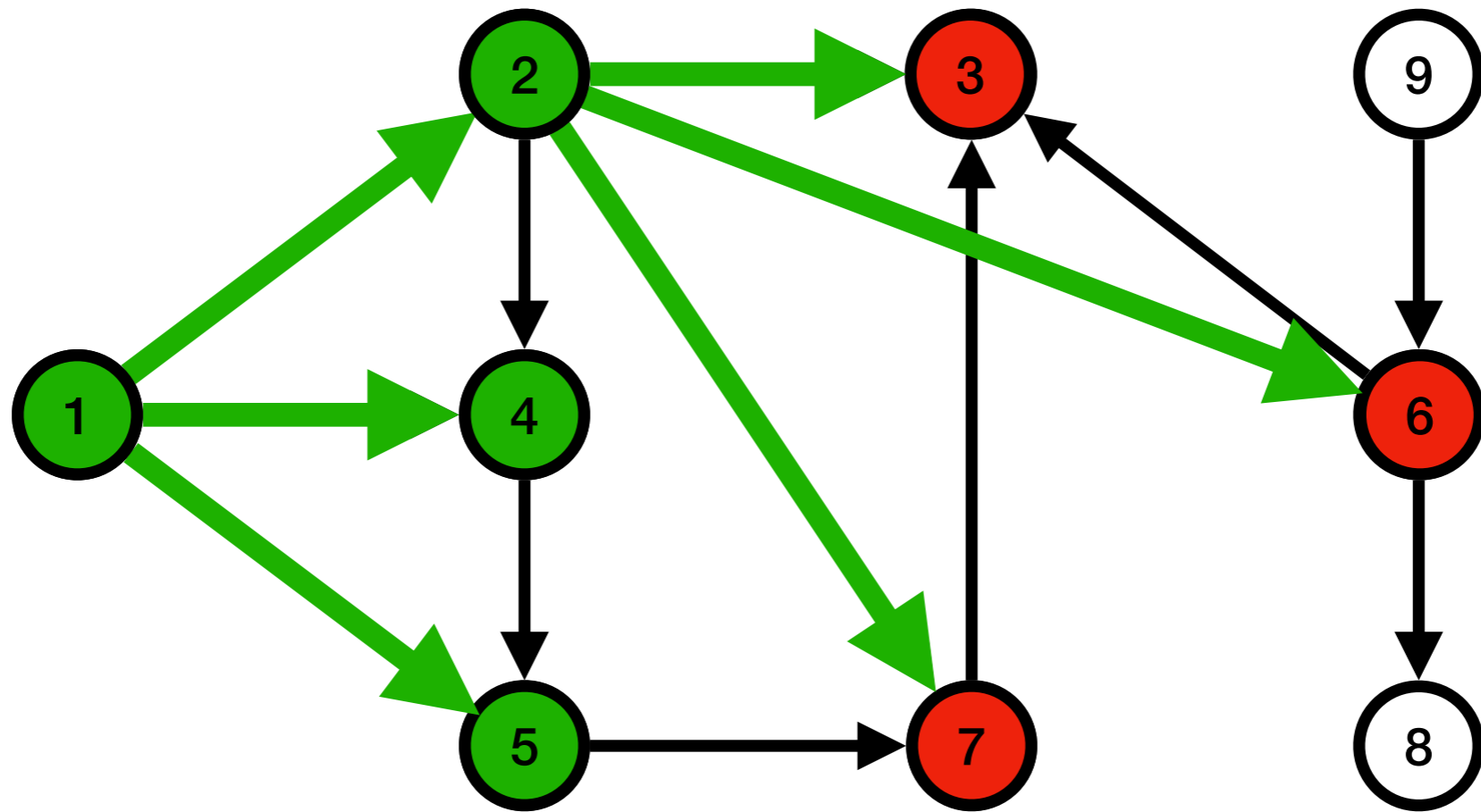


File → (5) (3) (6) (7)

Un autre exemple

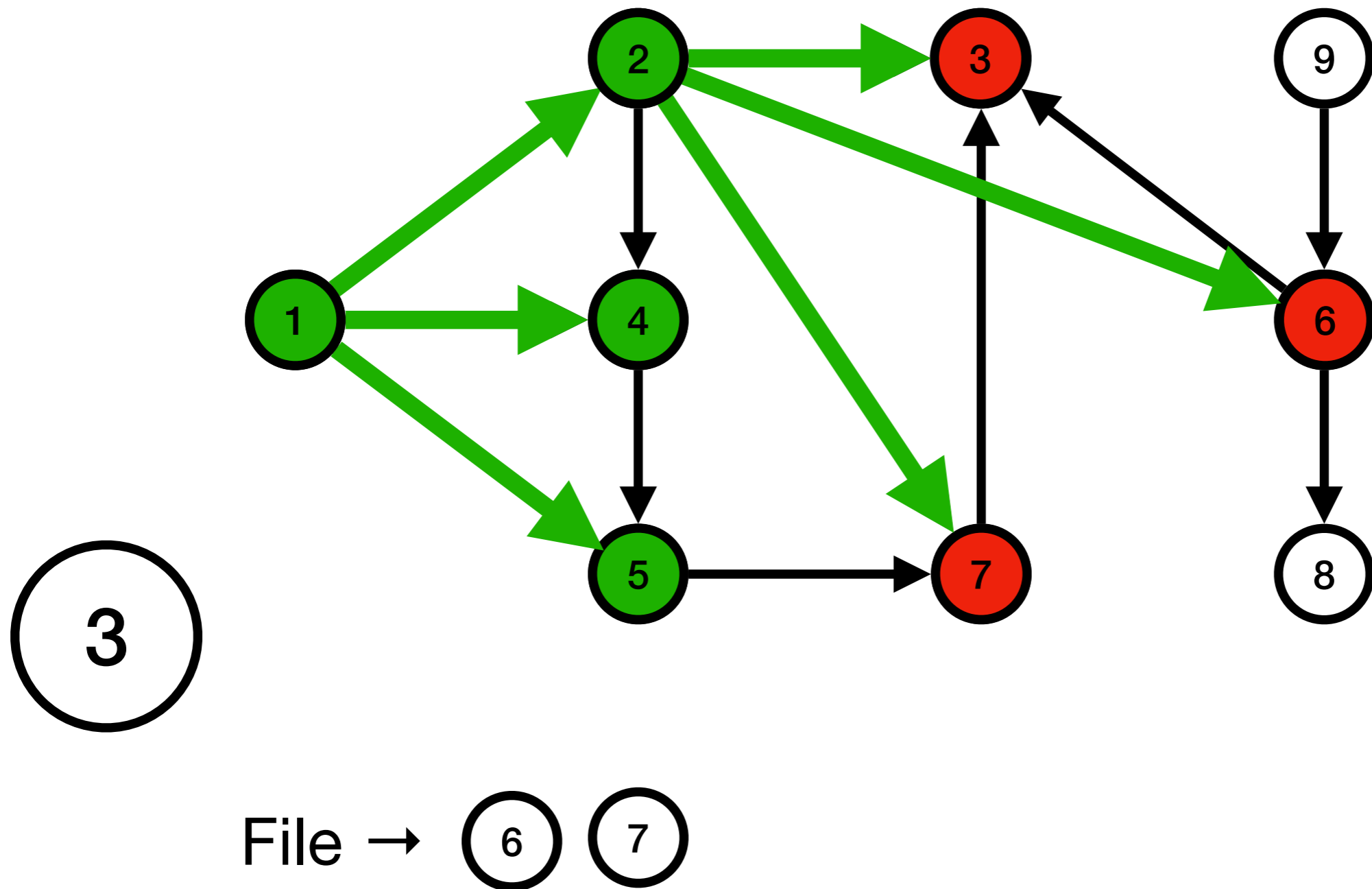


Un autre exemple

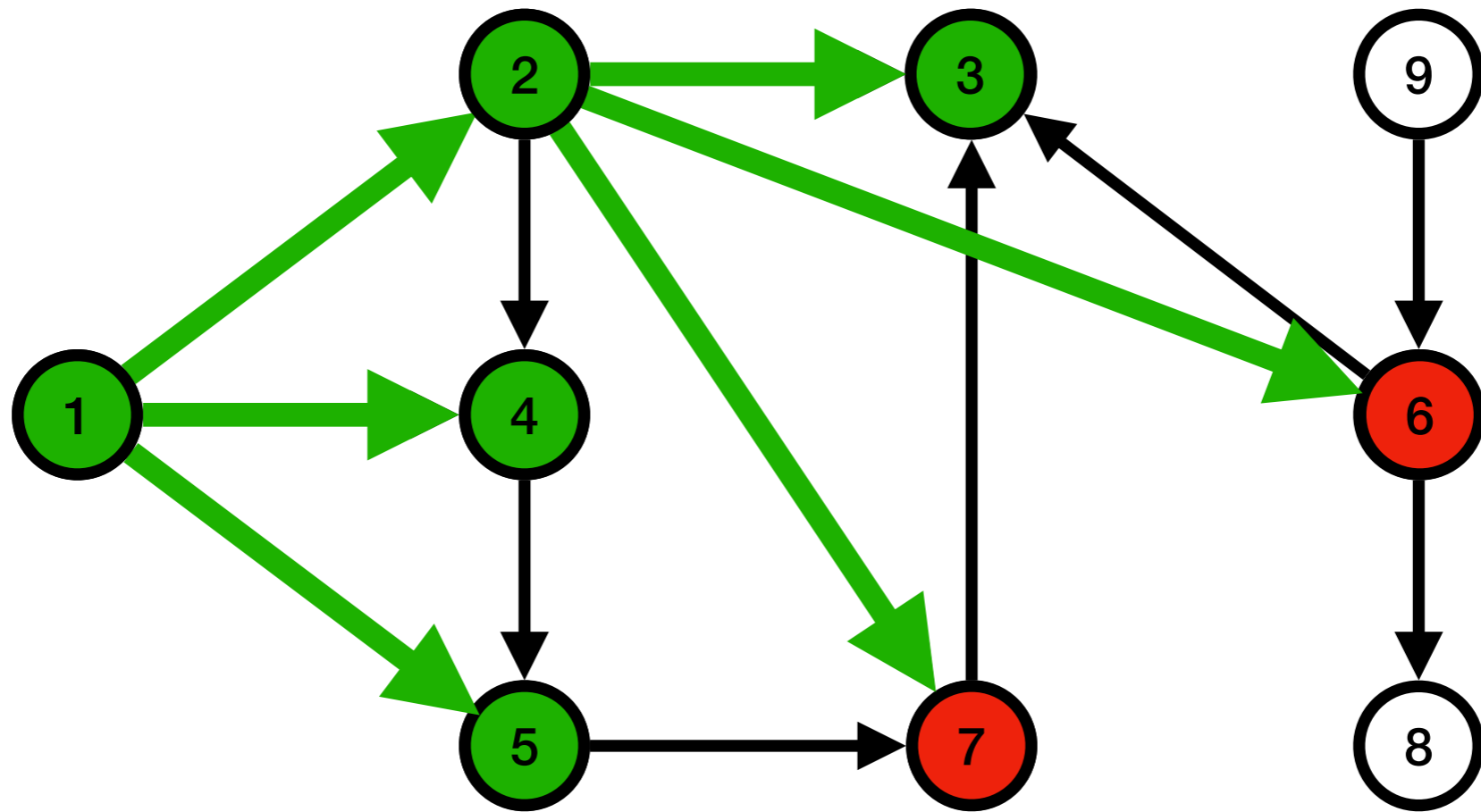


File → (3) (6) (7)

Un autre exemple

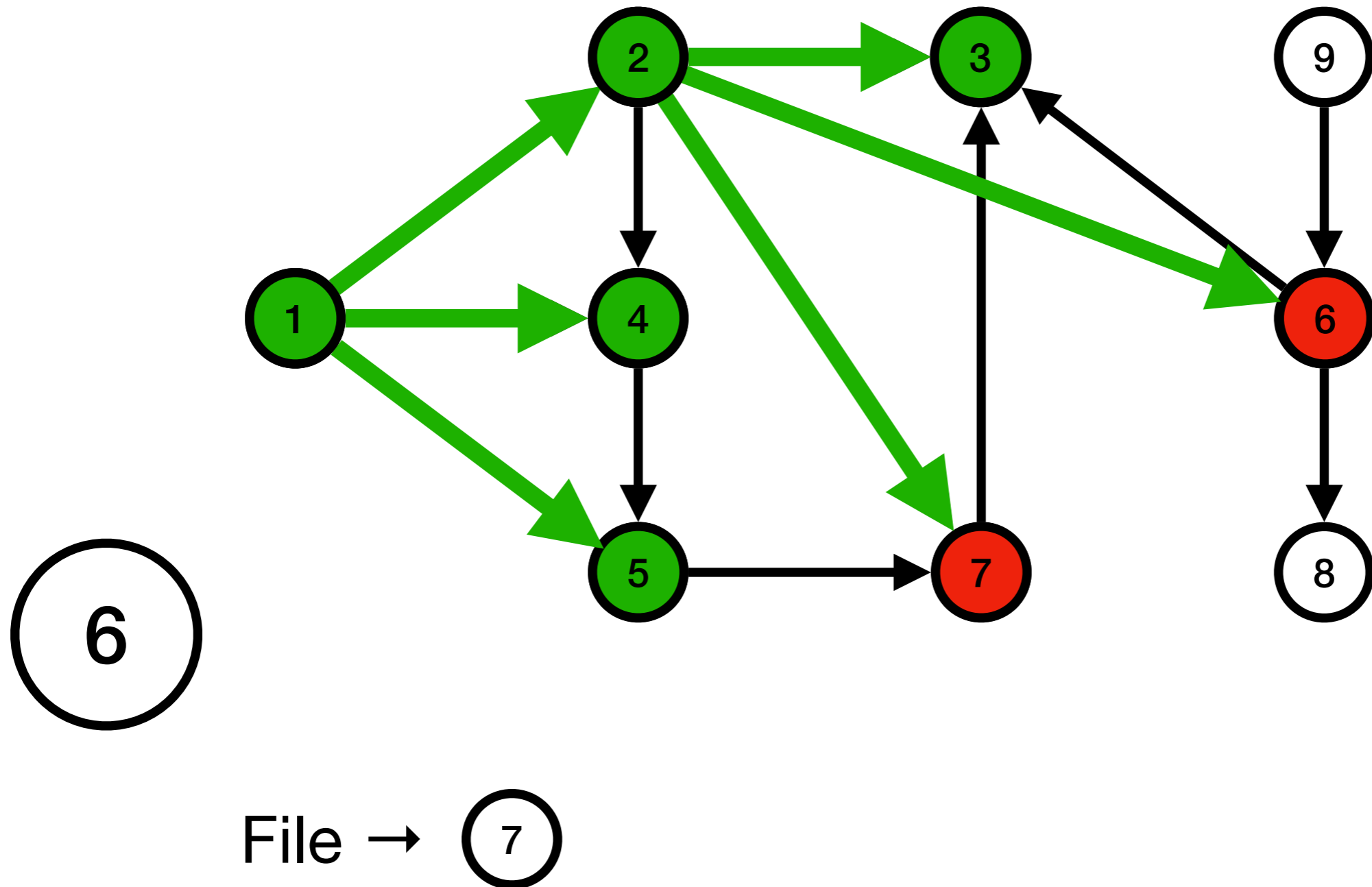


Un autre exemple

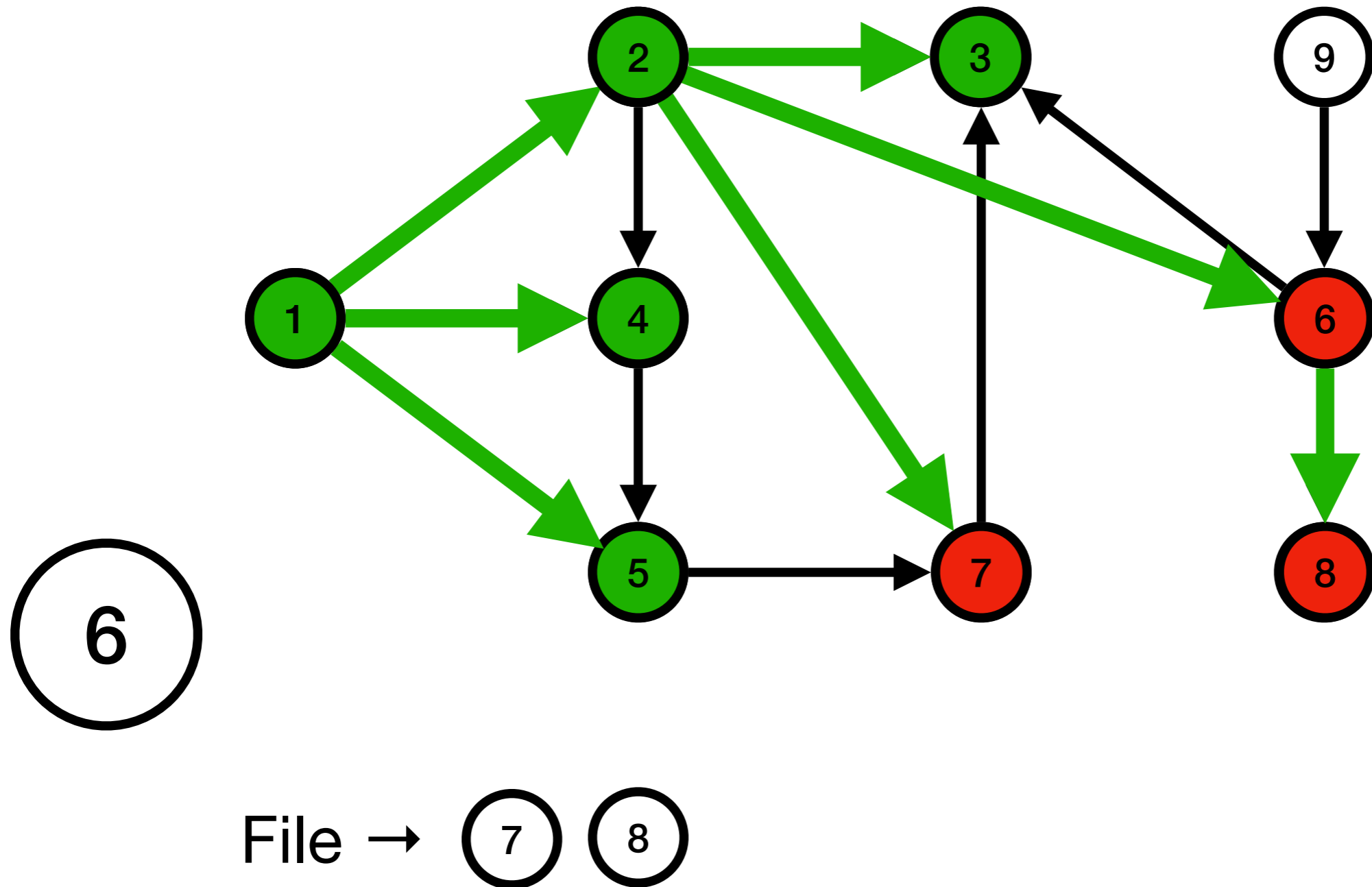


File → (6) (7)

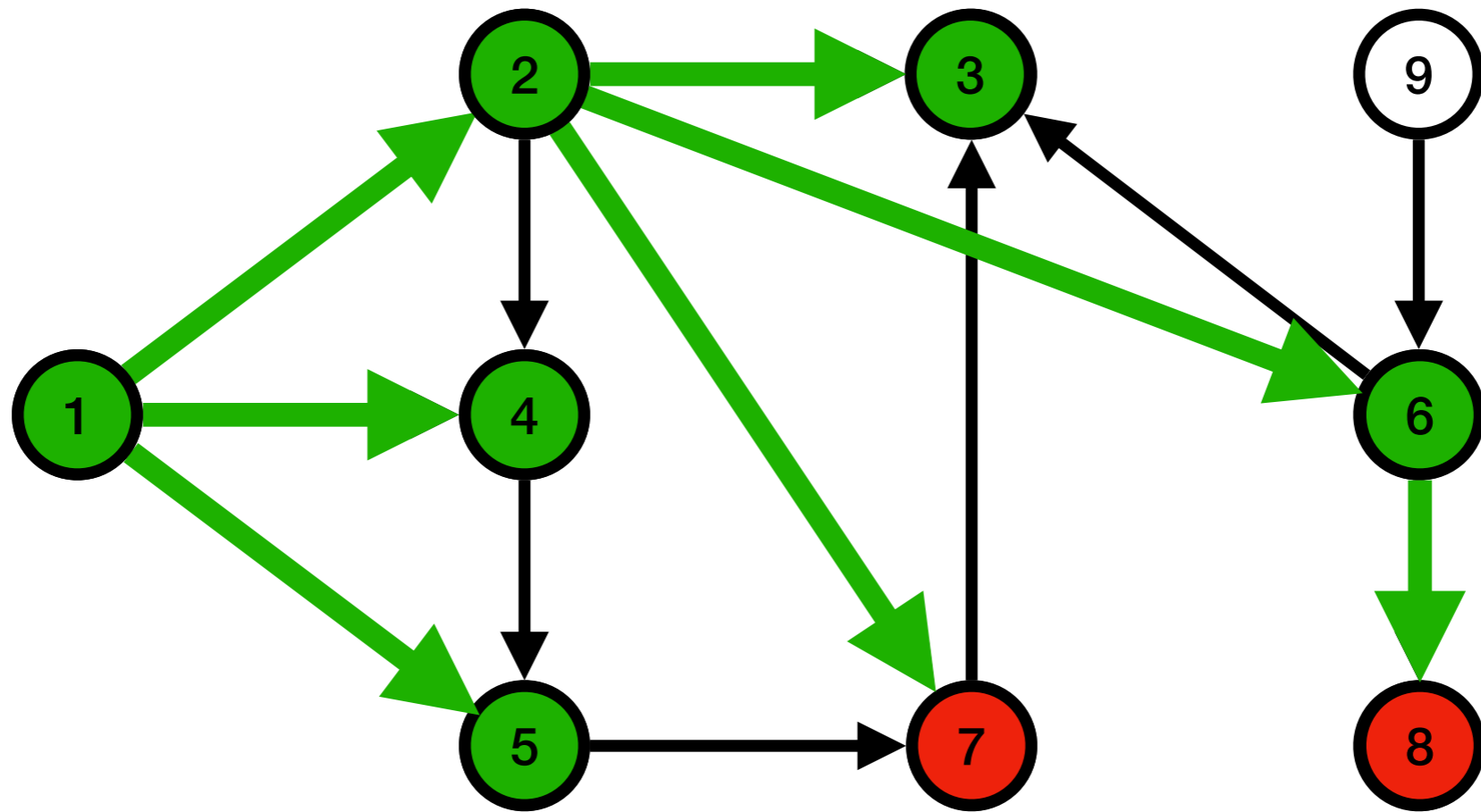
Un autre exemple



Un autre exemple

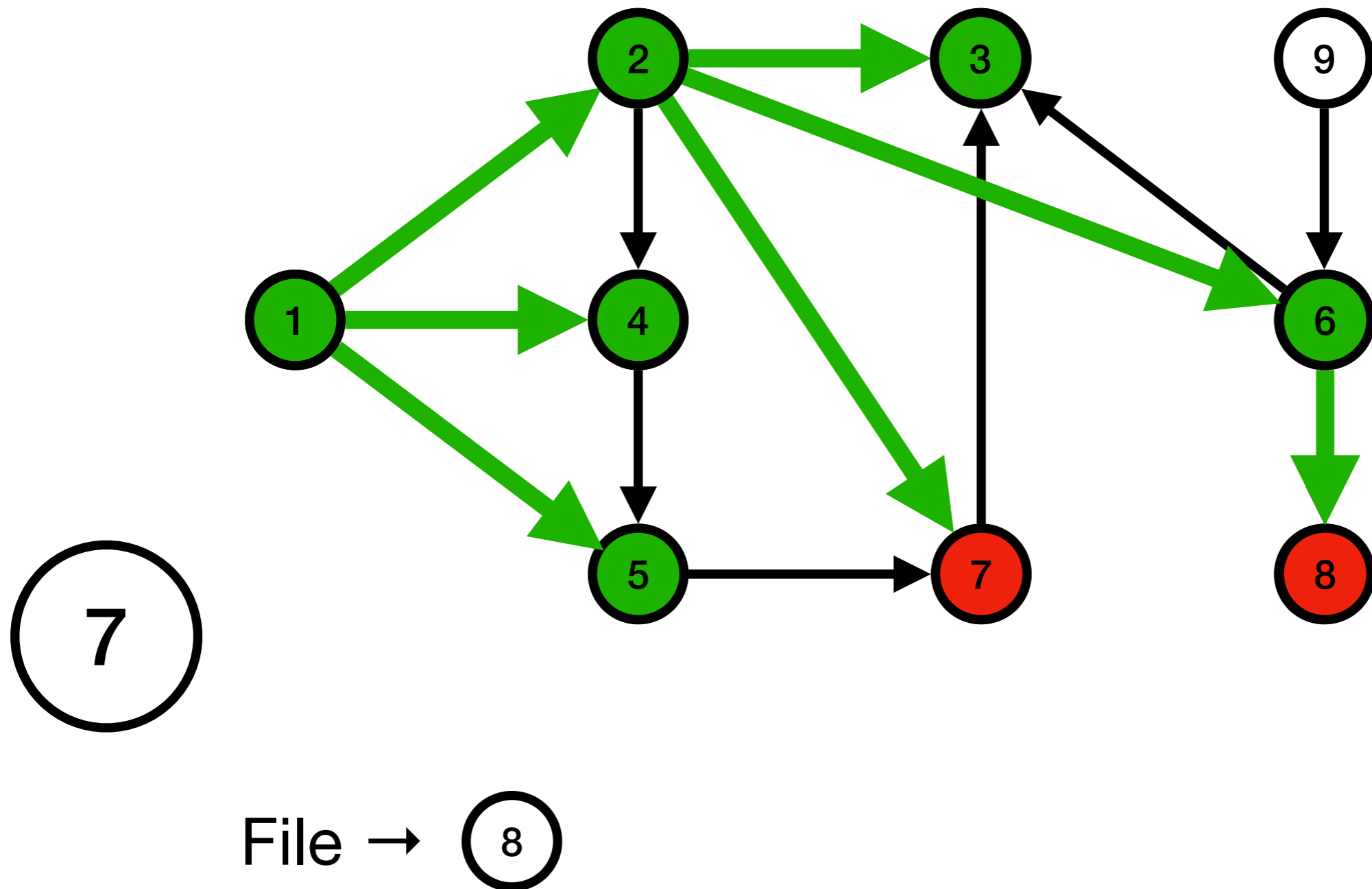


Un autre exemple

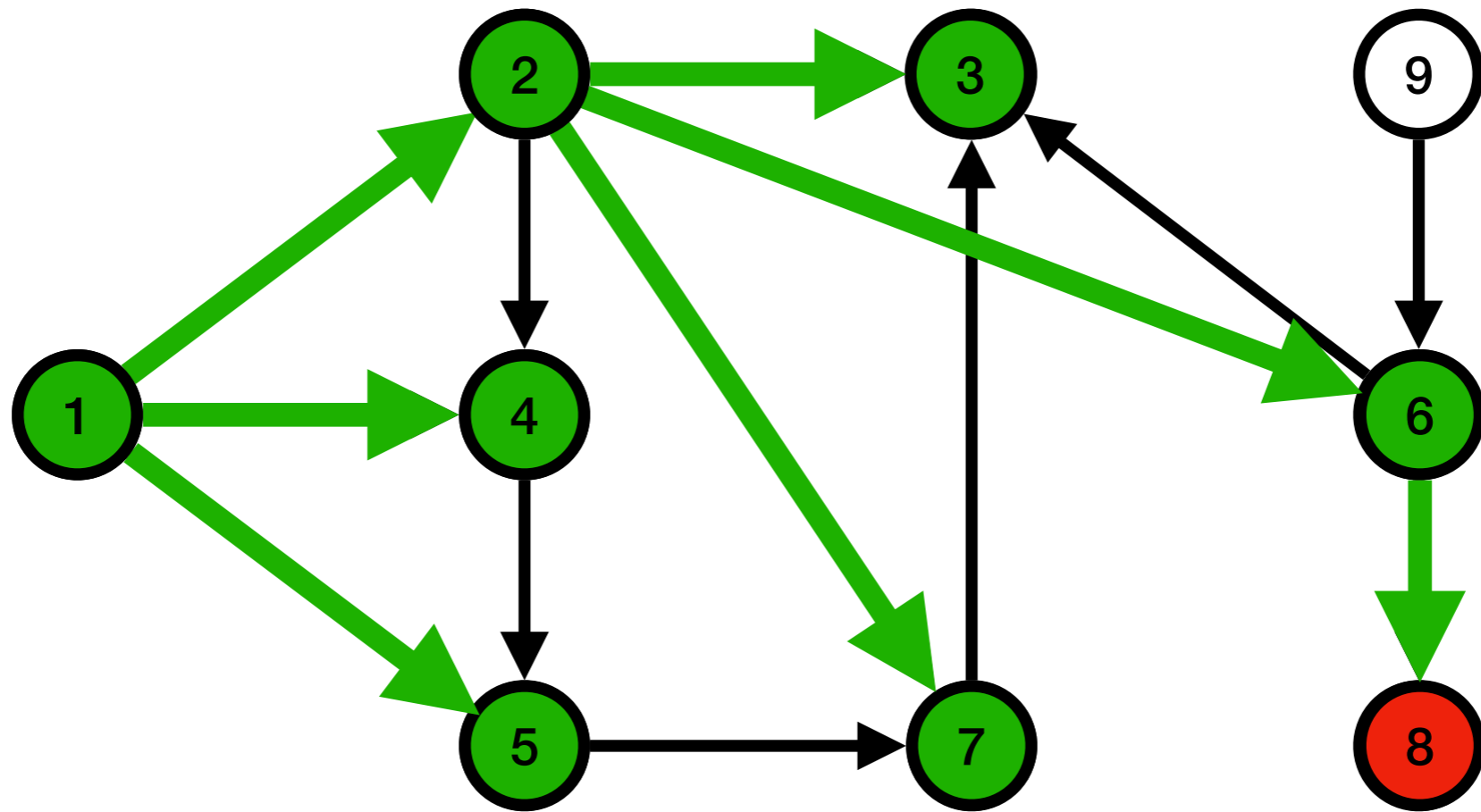



File → (7) (8)

Un autre exemple

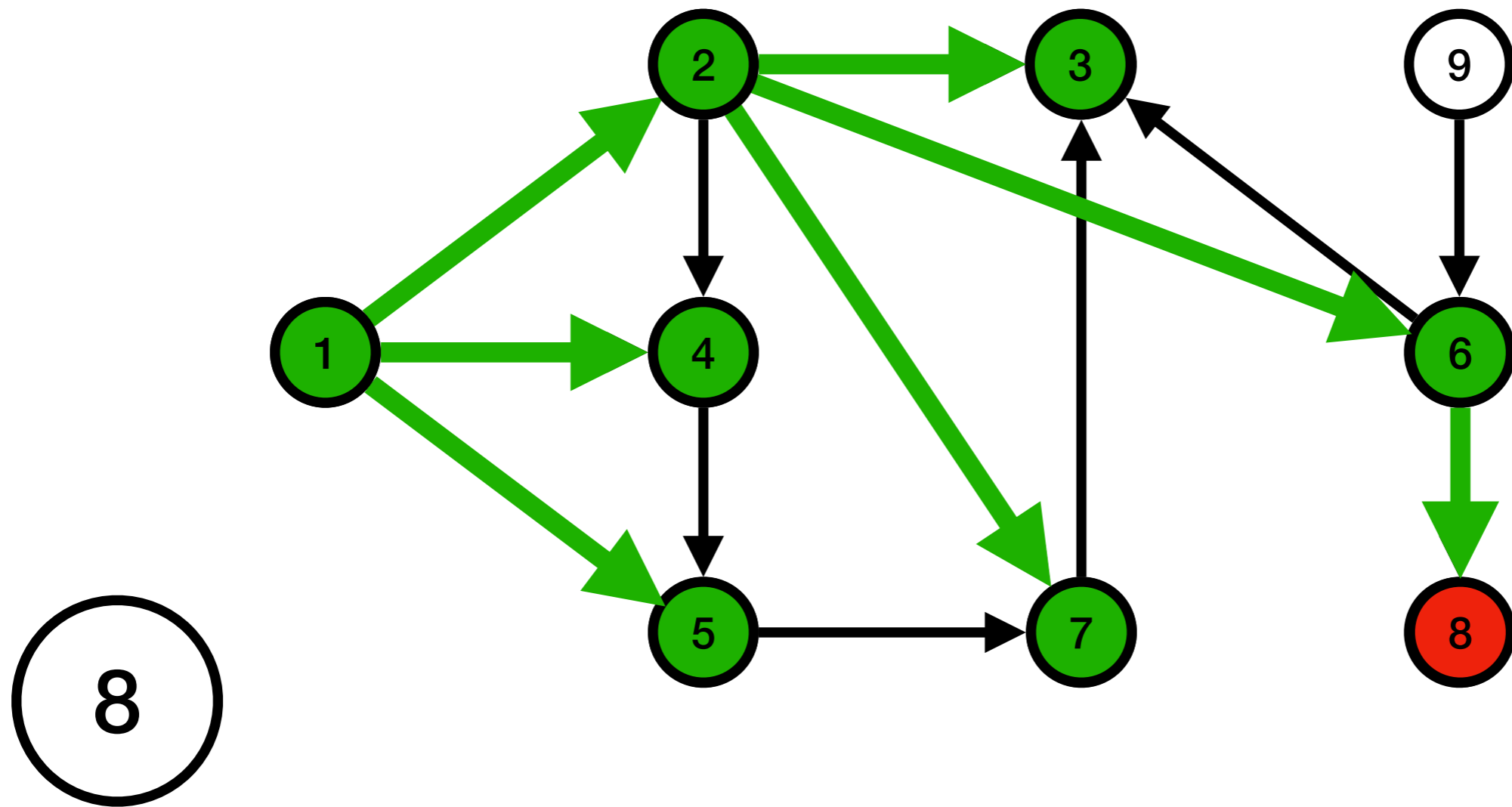


Un autre exemple



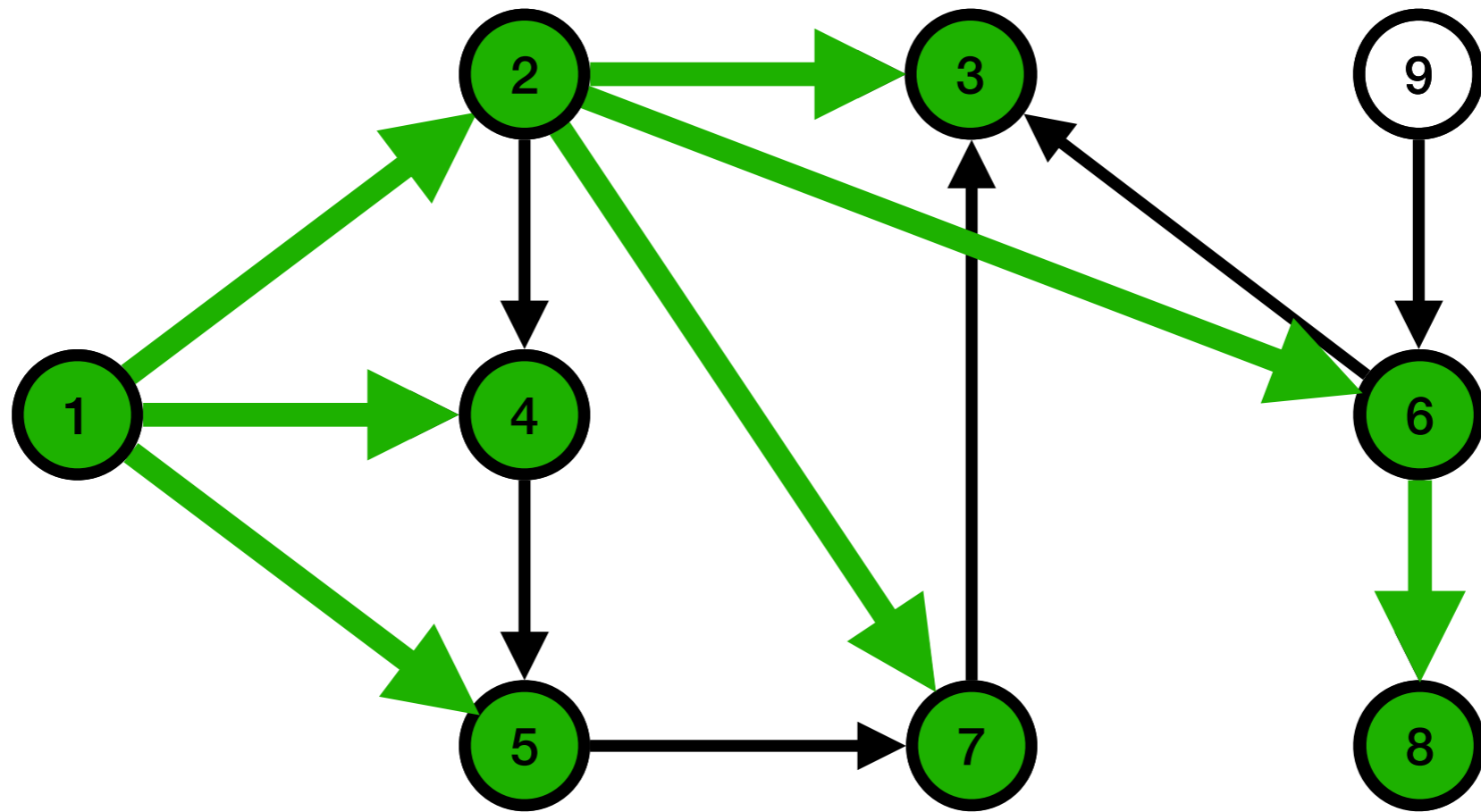
File → 

Un autre exemple



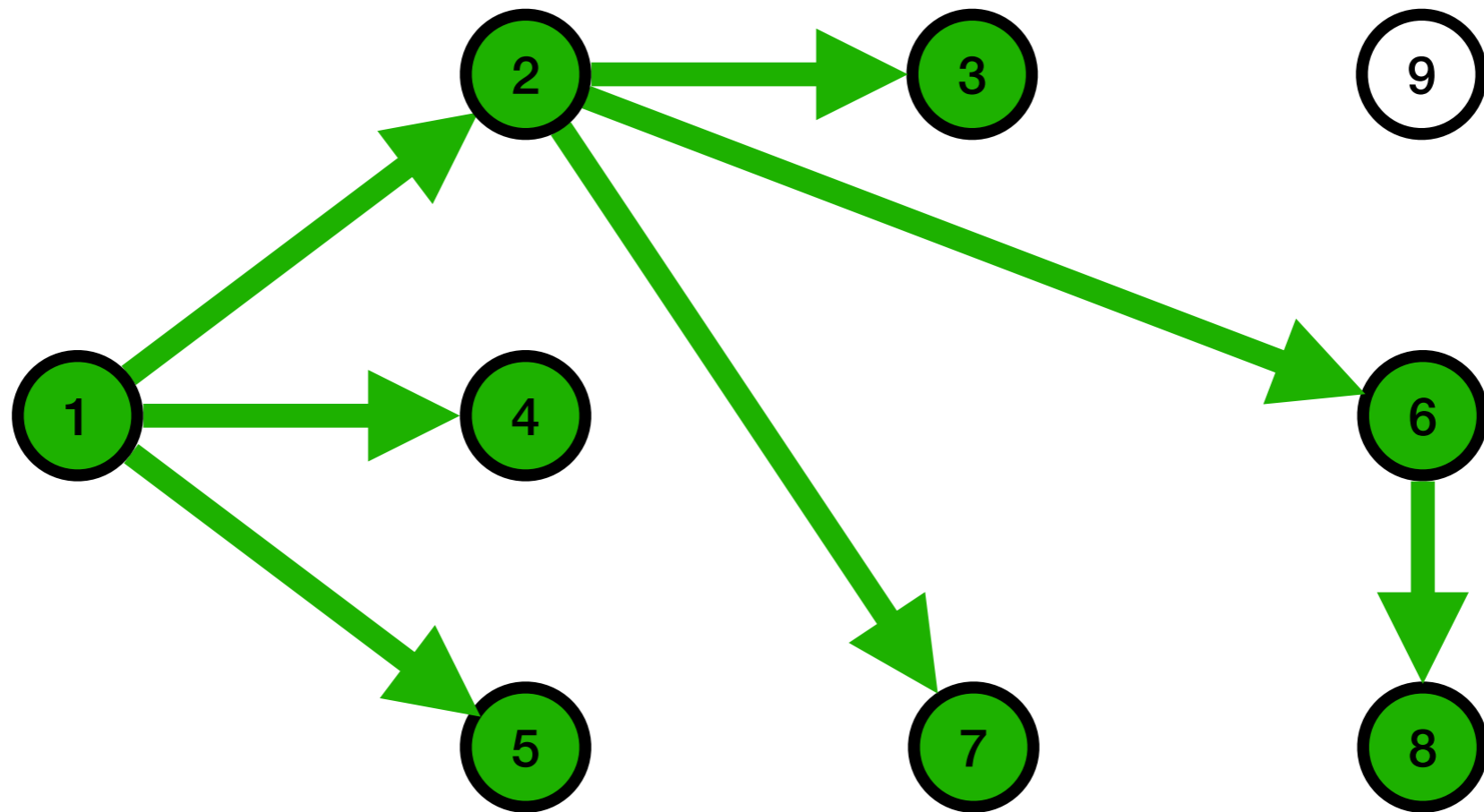
File →

Un autre exemple



File →

Un autre exemple



Un autre exemple

