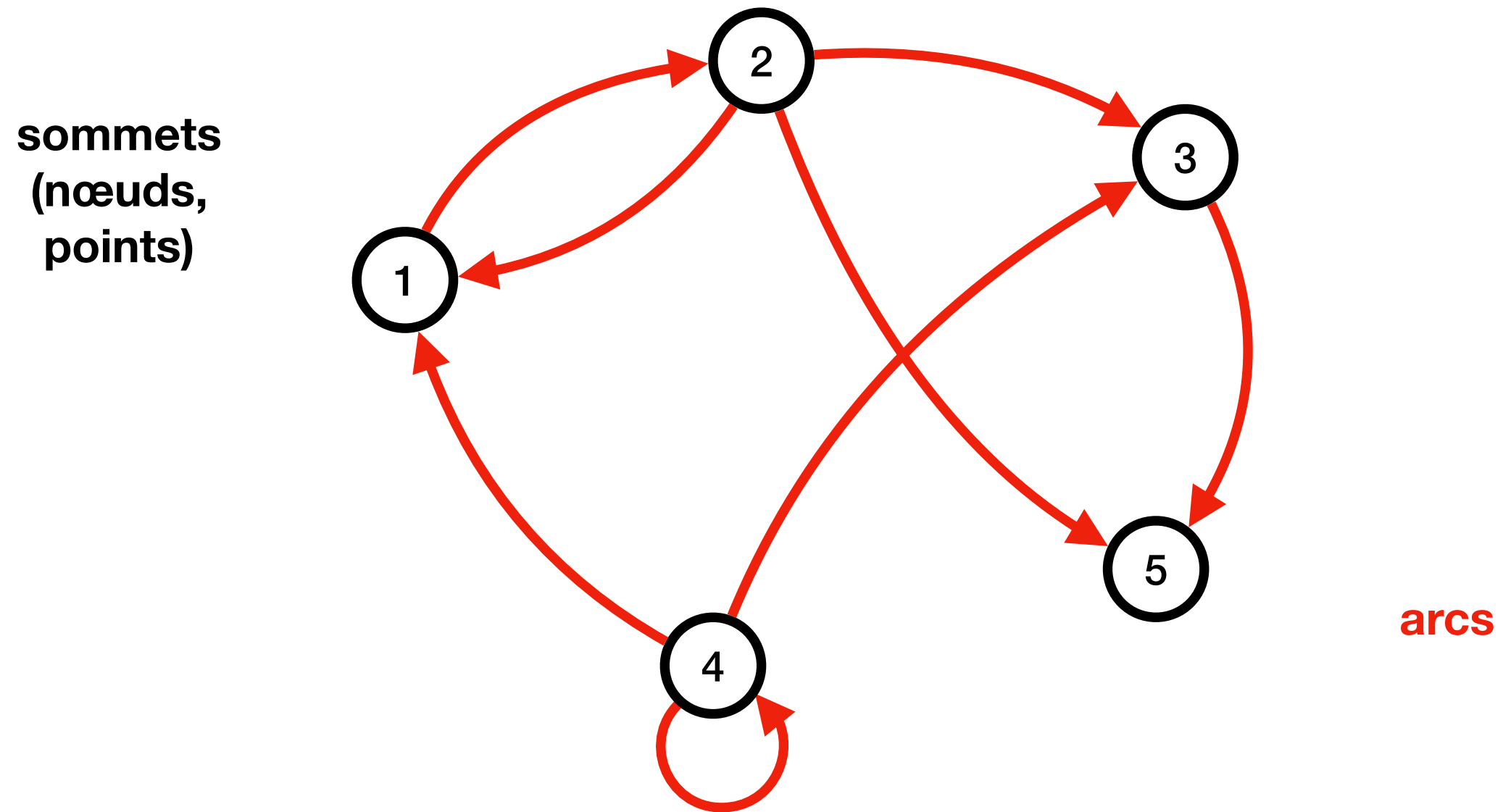


Introduction à l'informatique CM8

Antonio E. Porreca
aeporreca.org/introinfo

Graphes

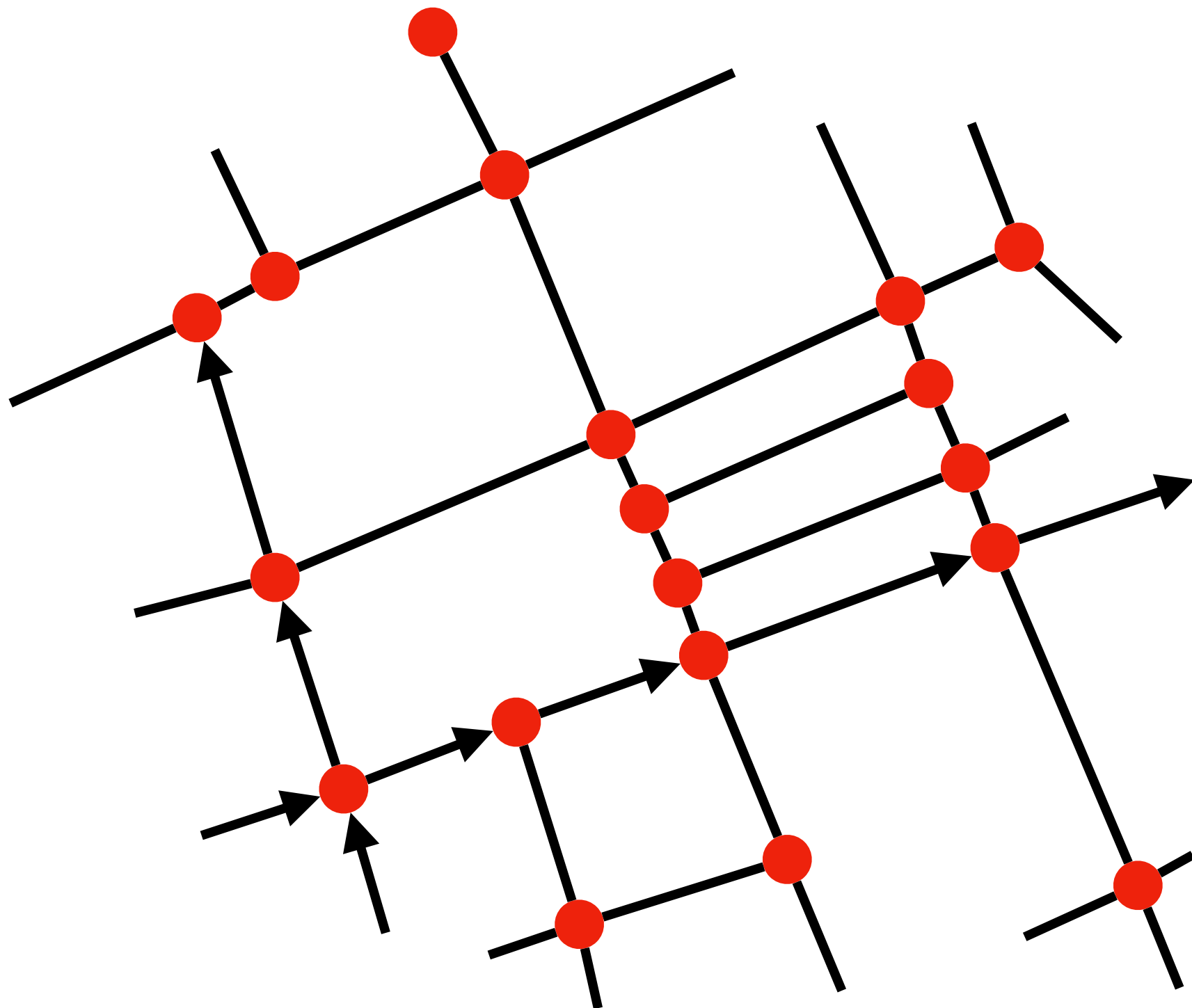
Graphes (orientés)



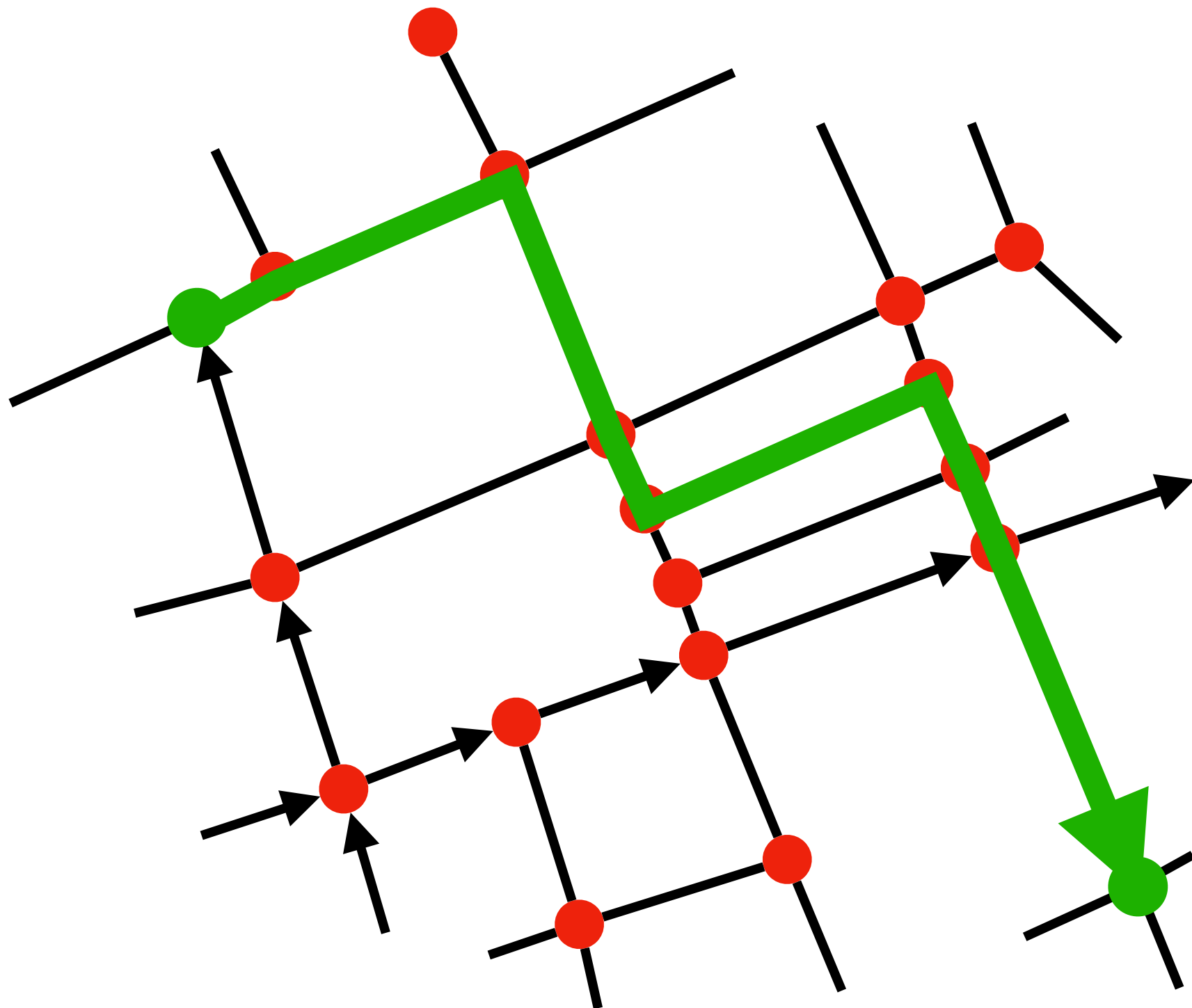
Plan d'une ville



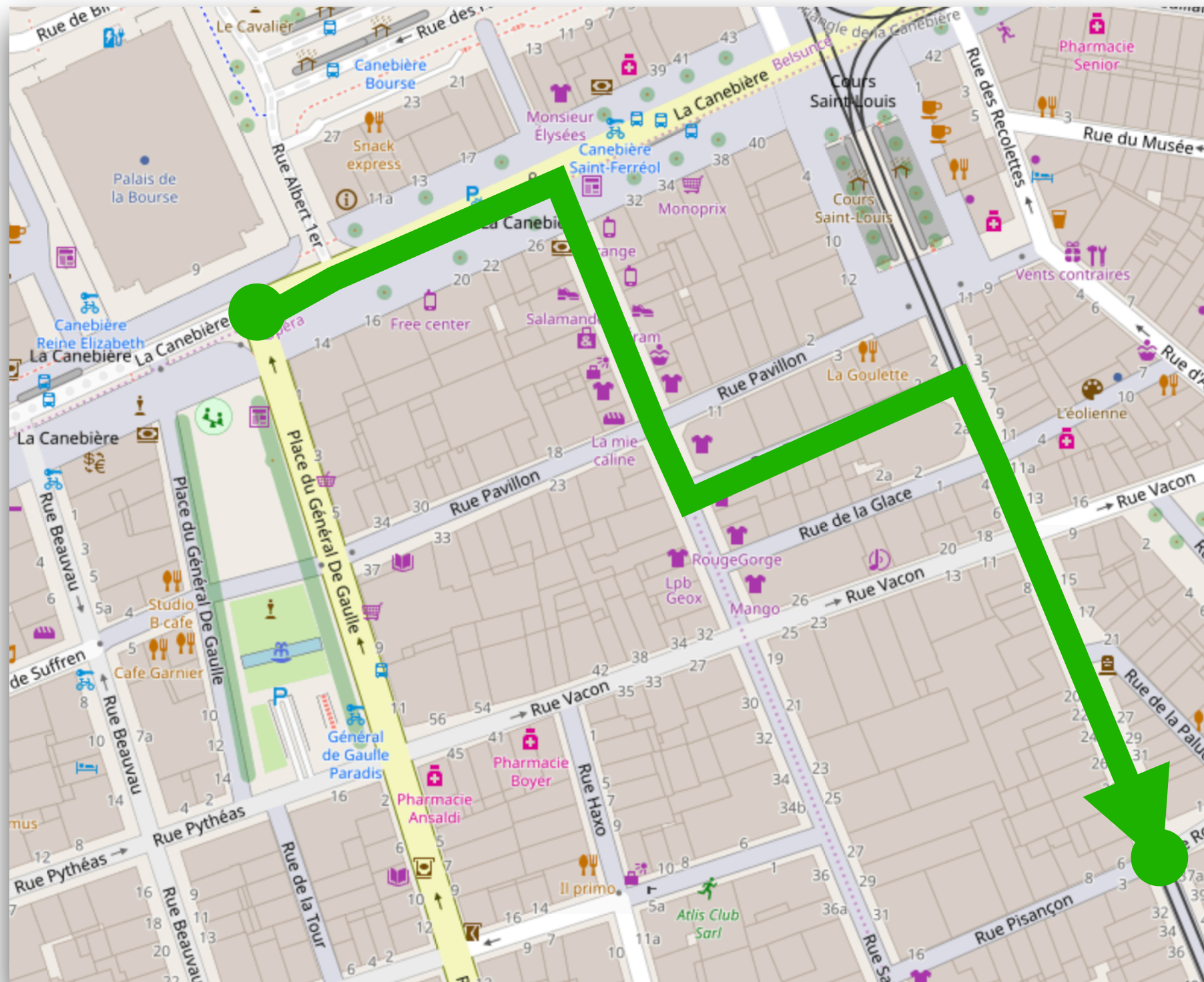
Plan d'une ville



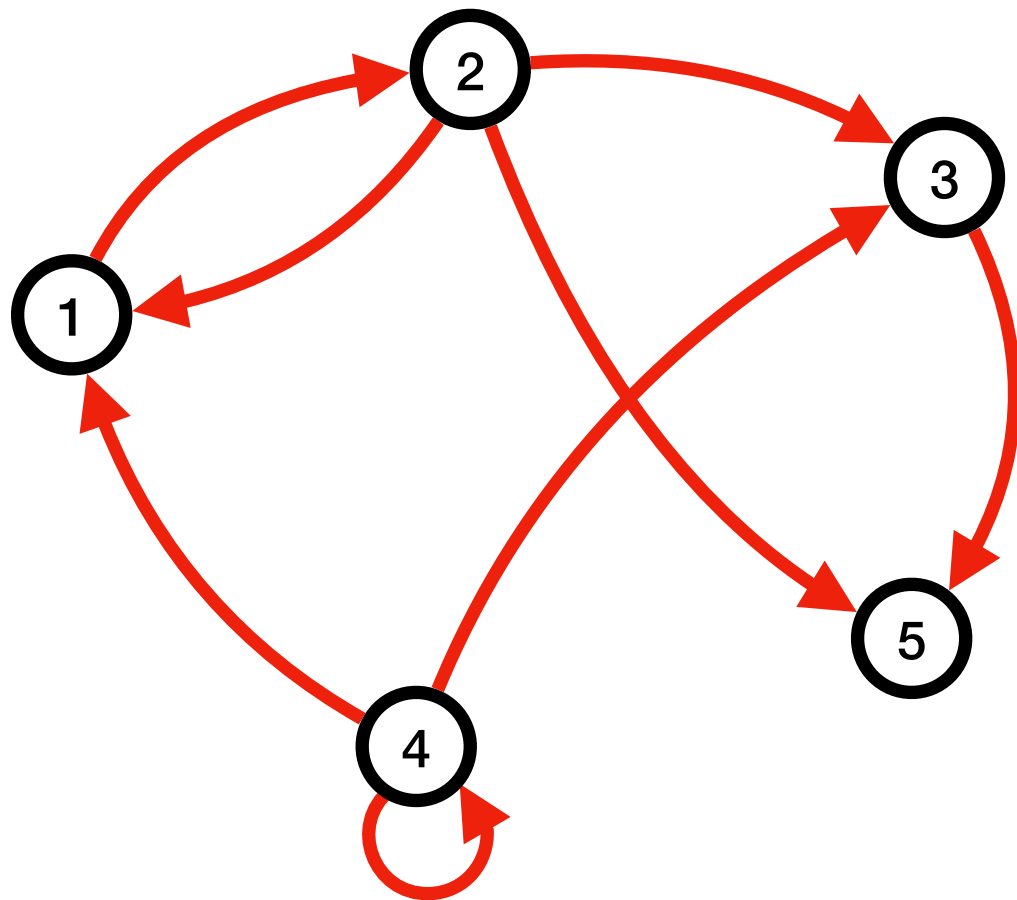
Plan d'une ville



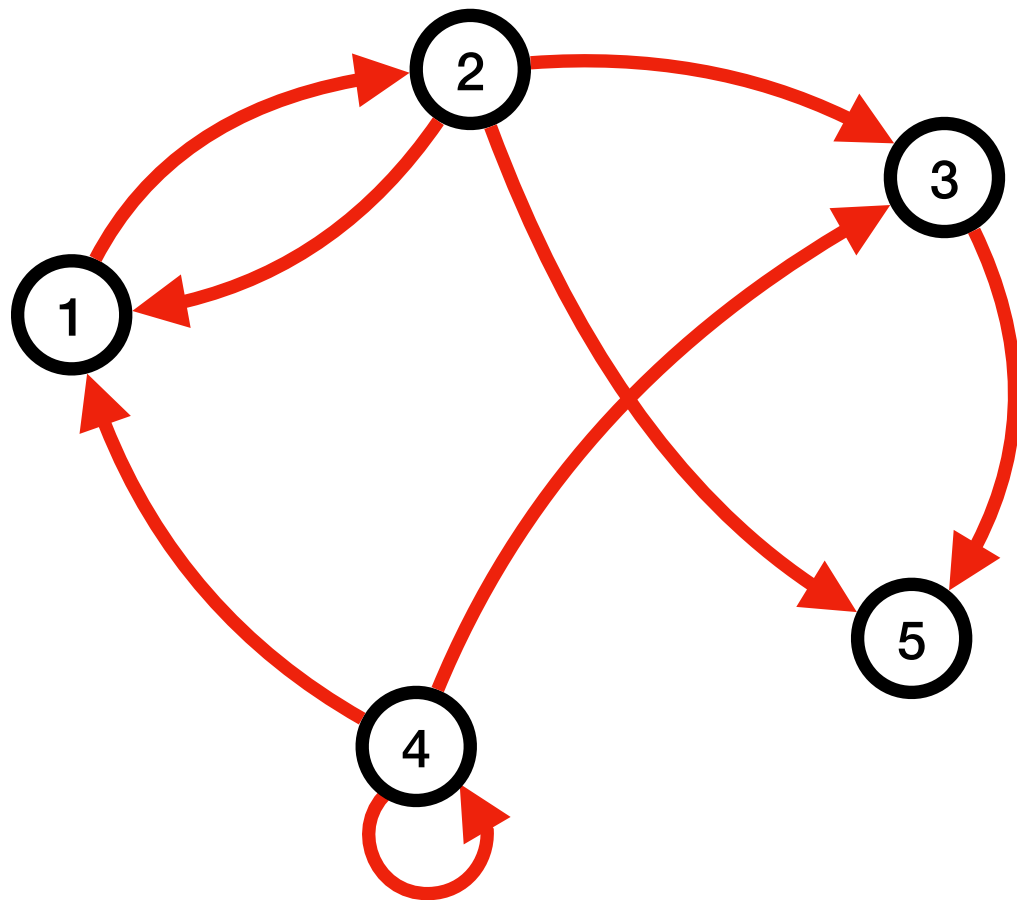
Plan d'une ville



Matrices d'adjacence

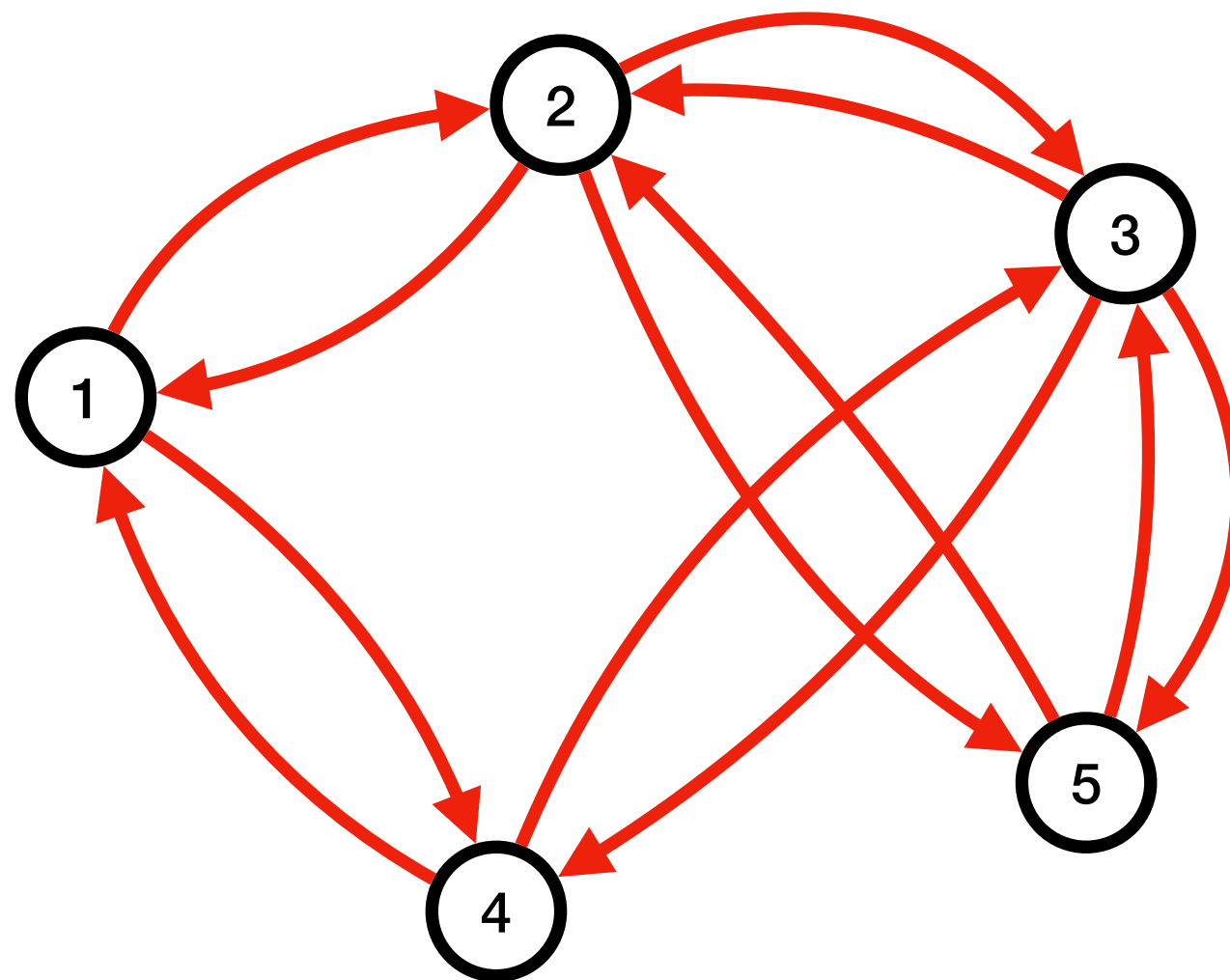


Matrices d'adjacence

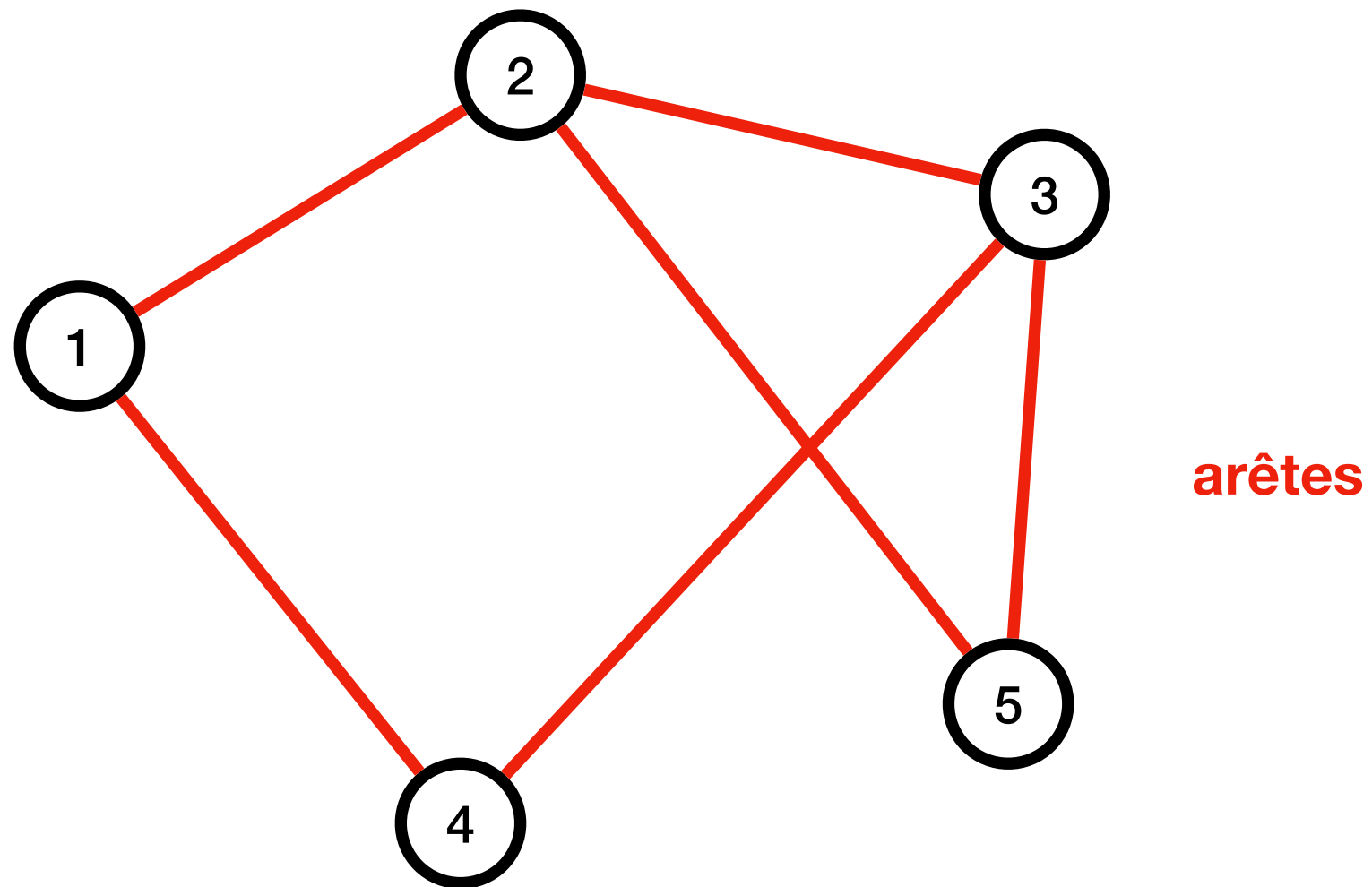


	1	2	3	4	5
1	0	1	0	0	0
2	1	0	1	0	1
3	0	0	0	0	1
4	1	0	1	1	0
5	0	0	0	0	0

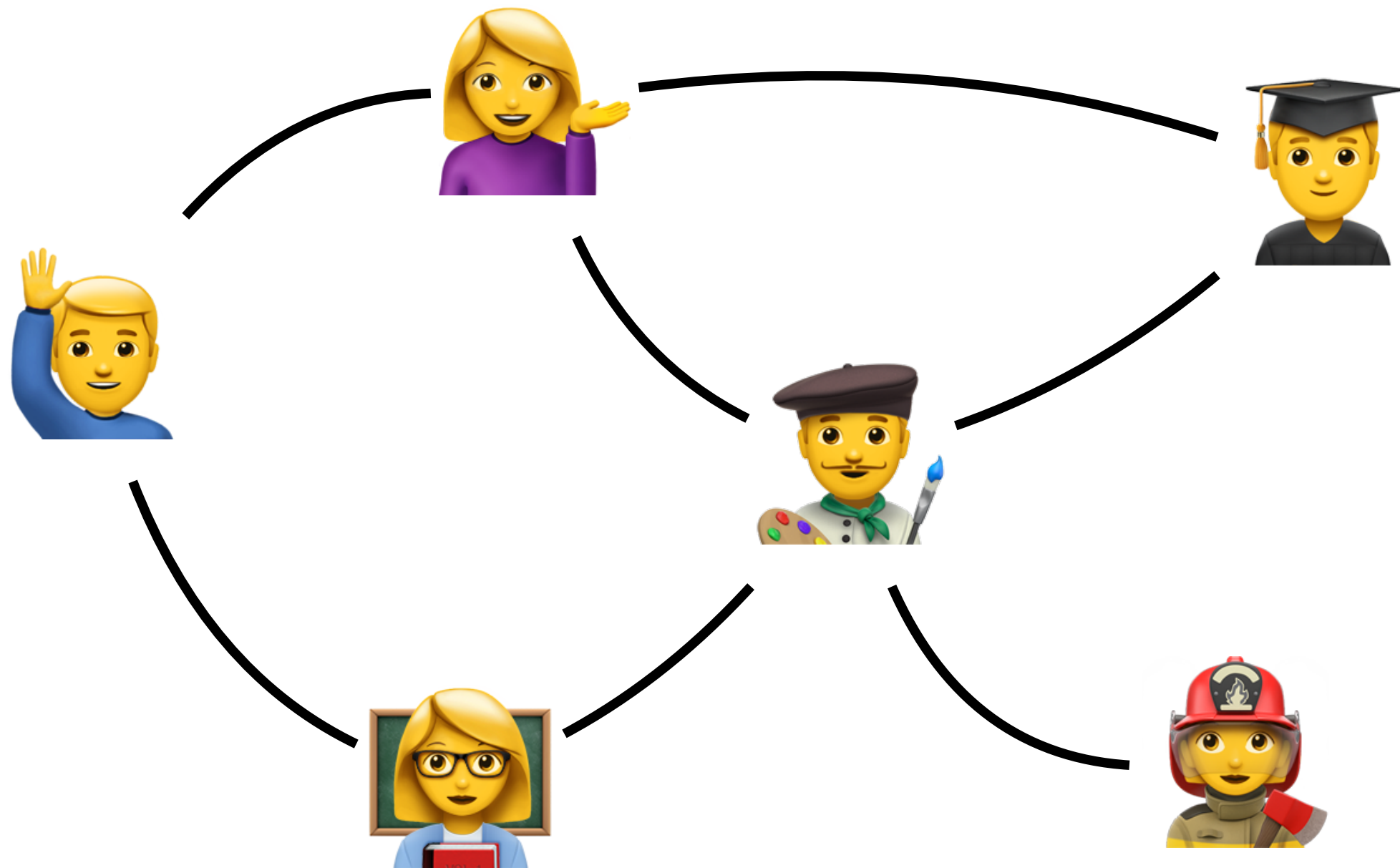
Graphes non orientés



Graphes non orientés



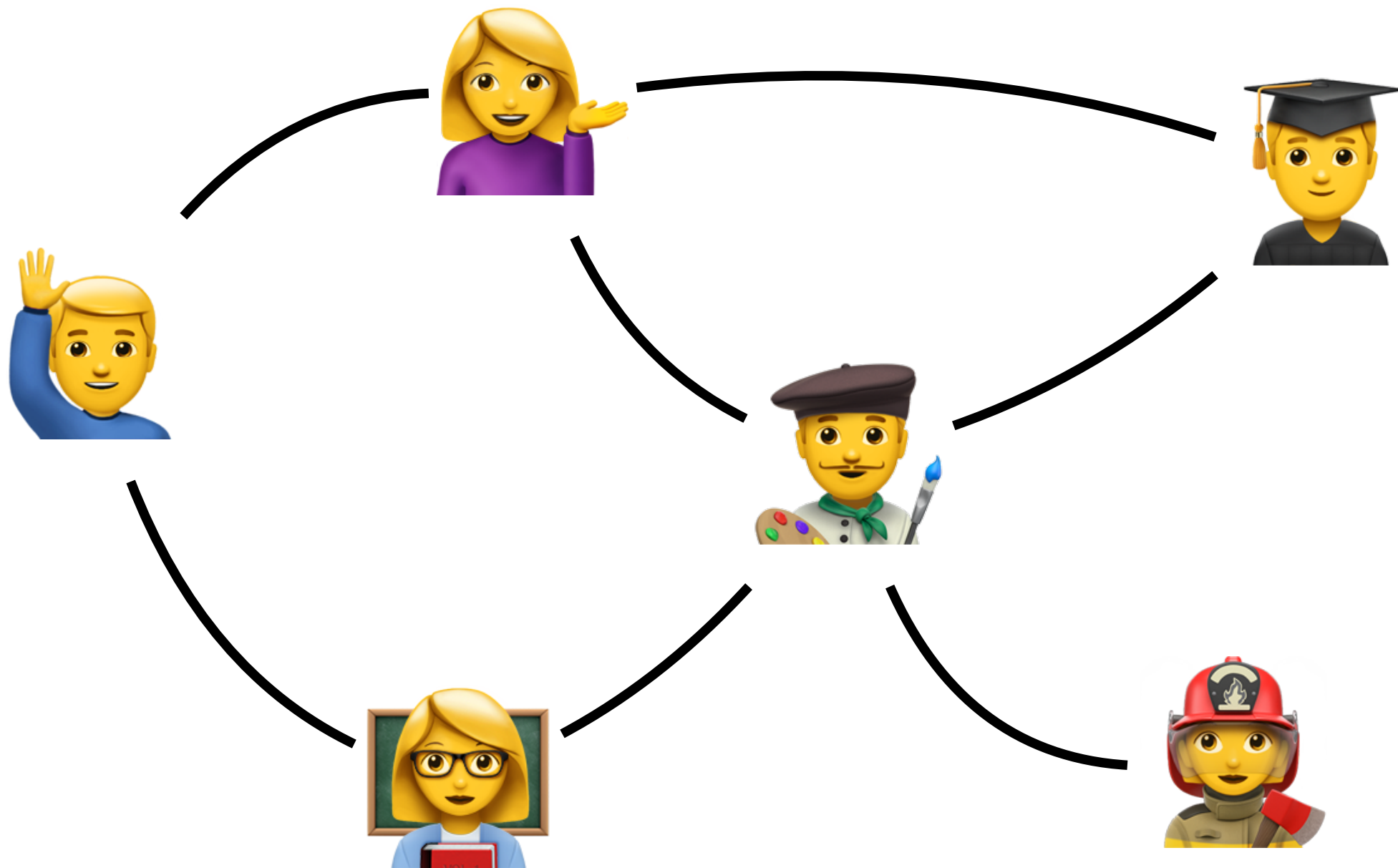
Réseau social (symétrique)



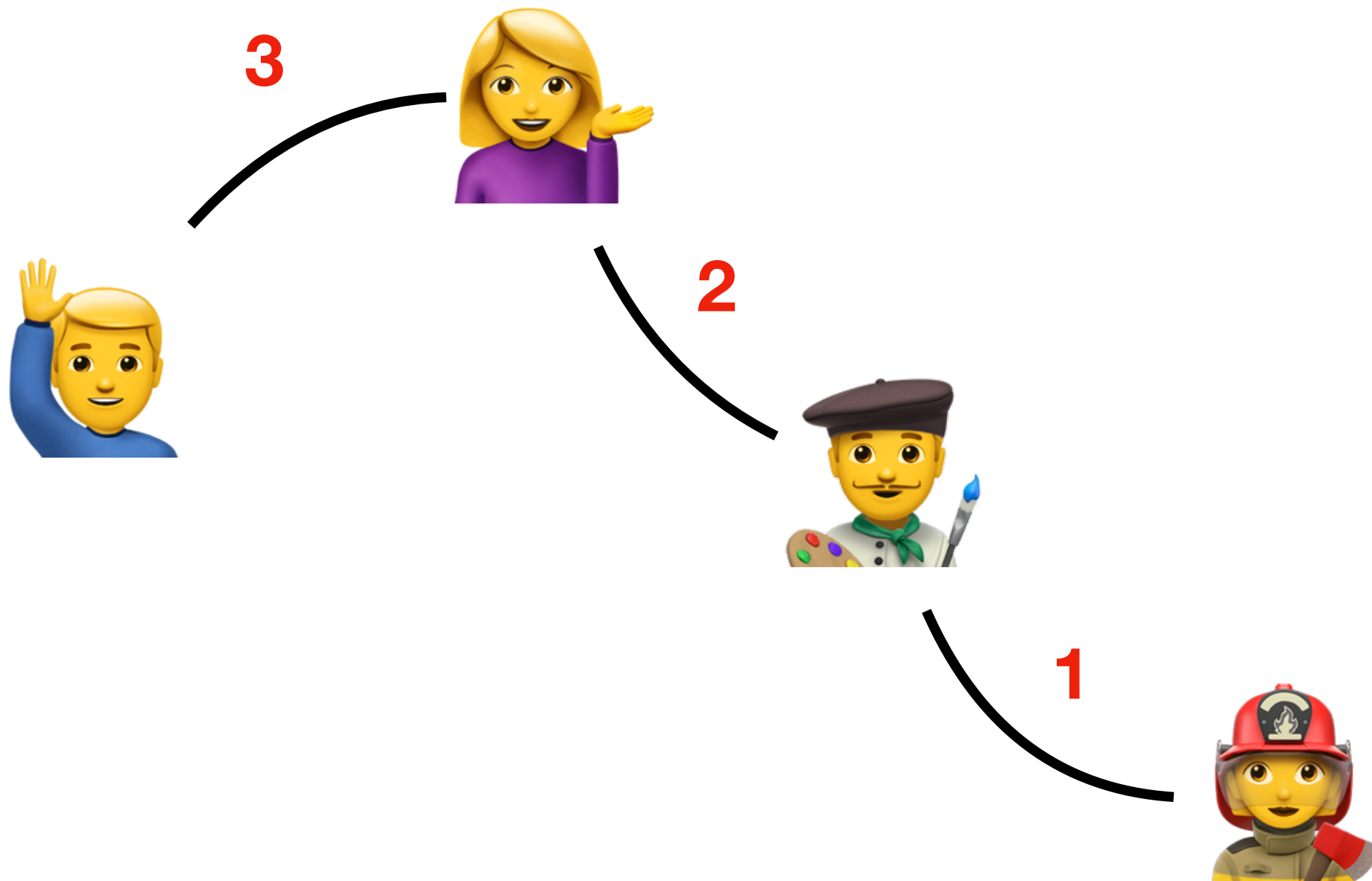
**Each person in the world (at least among the
1.59 billion people active on Facebook)
is connected to every other person
by an average of three and a half other people.**

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

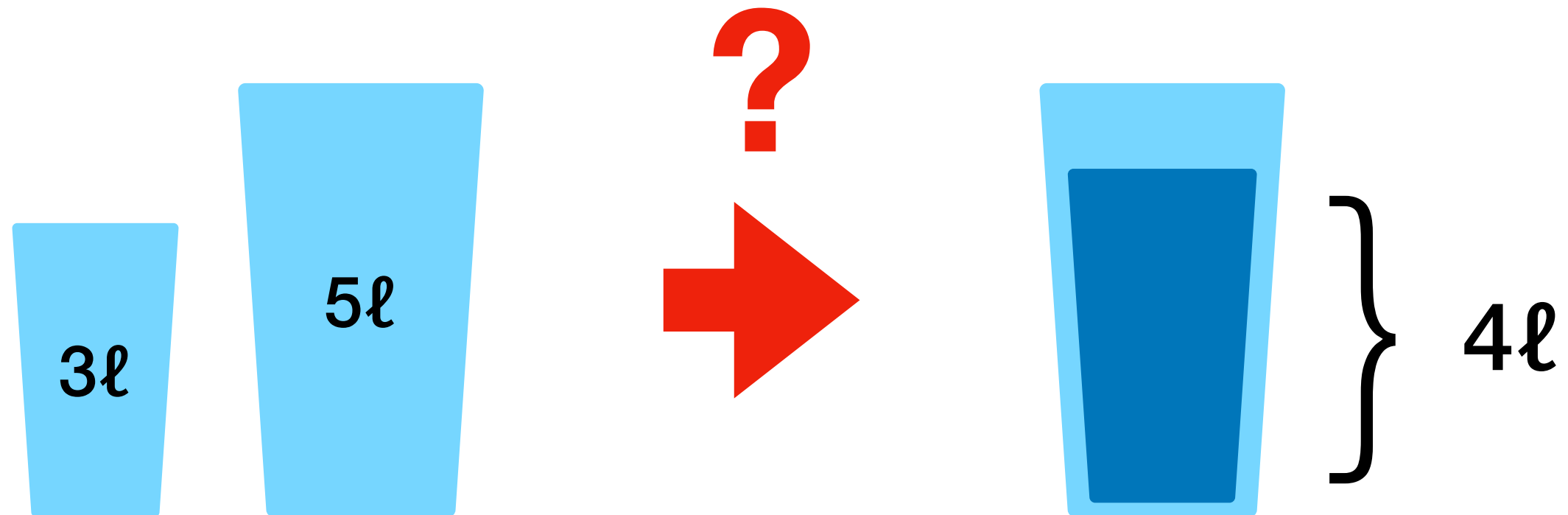
Diamètre d'un graphe



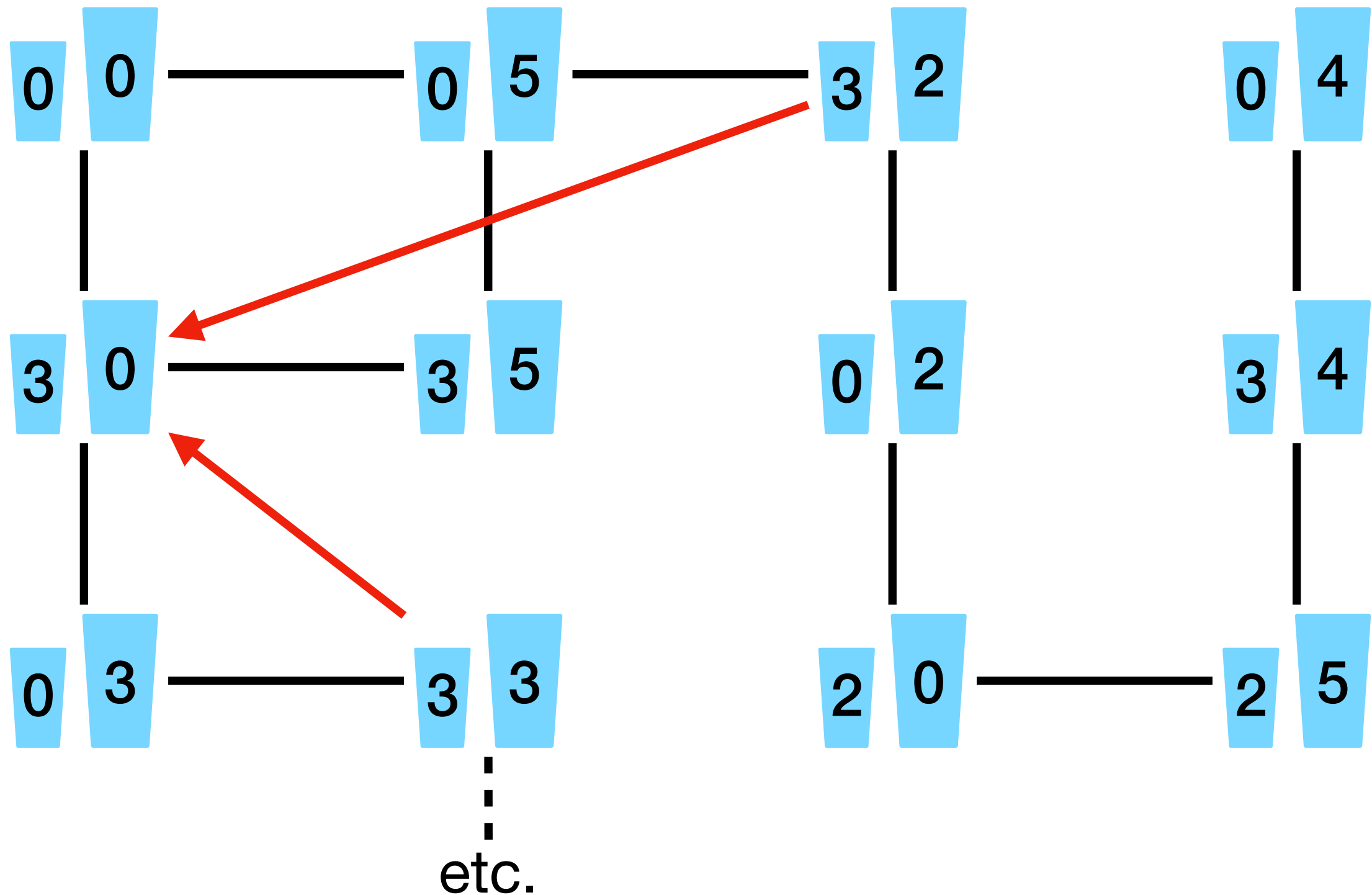
Diamètre d'un graphe



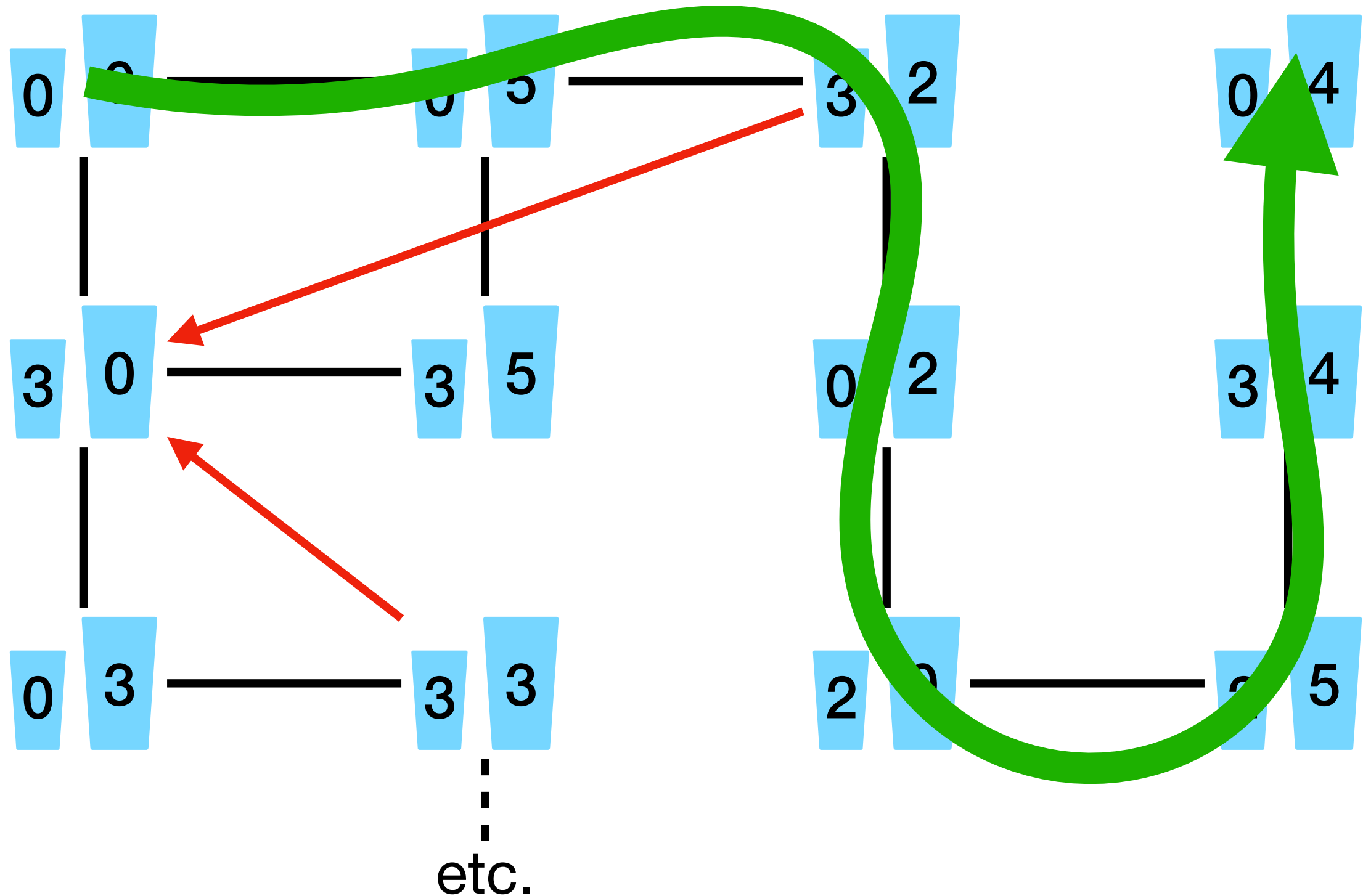
Enigme des récipients



Graphe des configurations

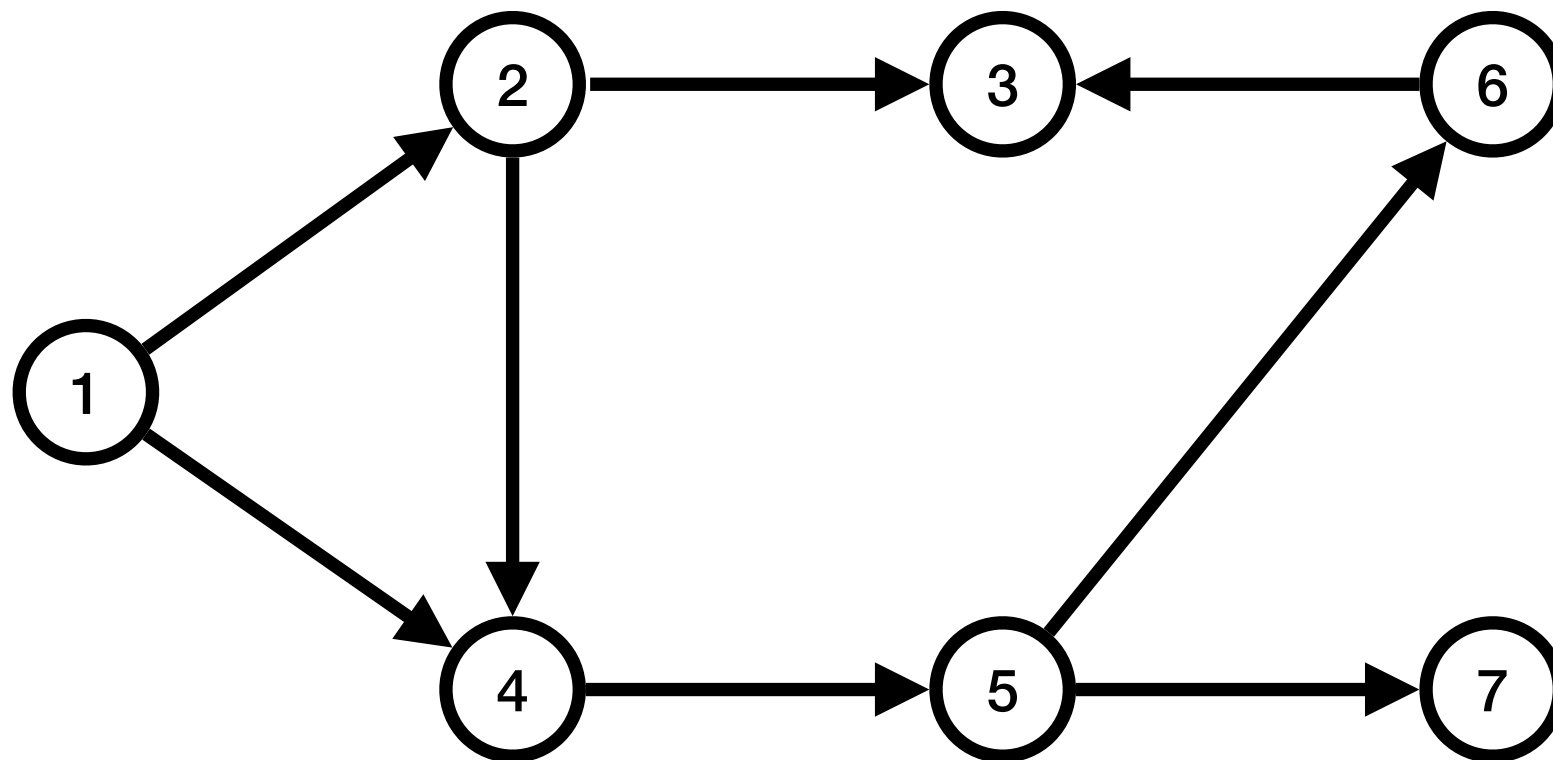


Graphe des configurations

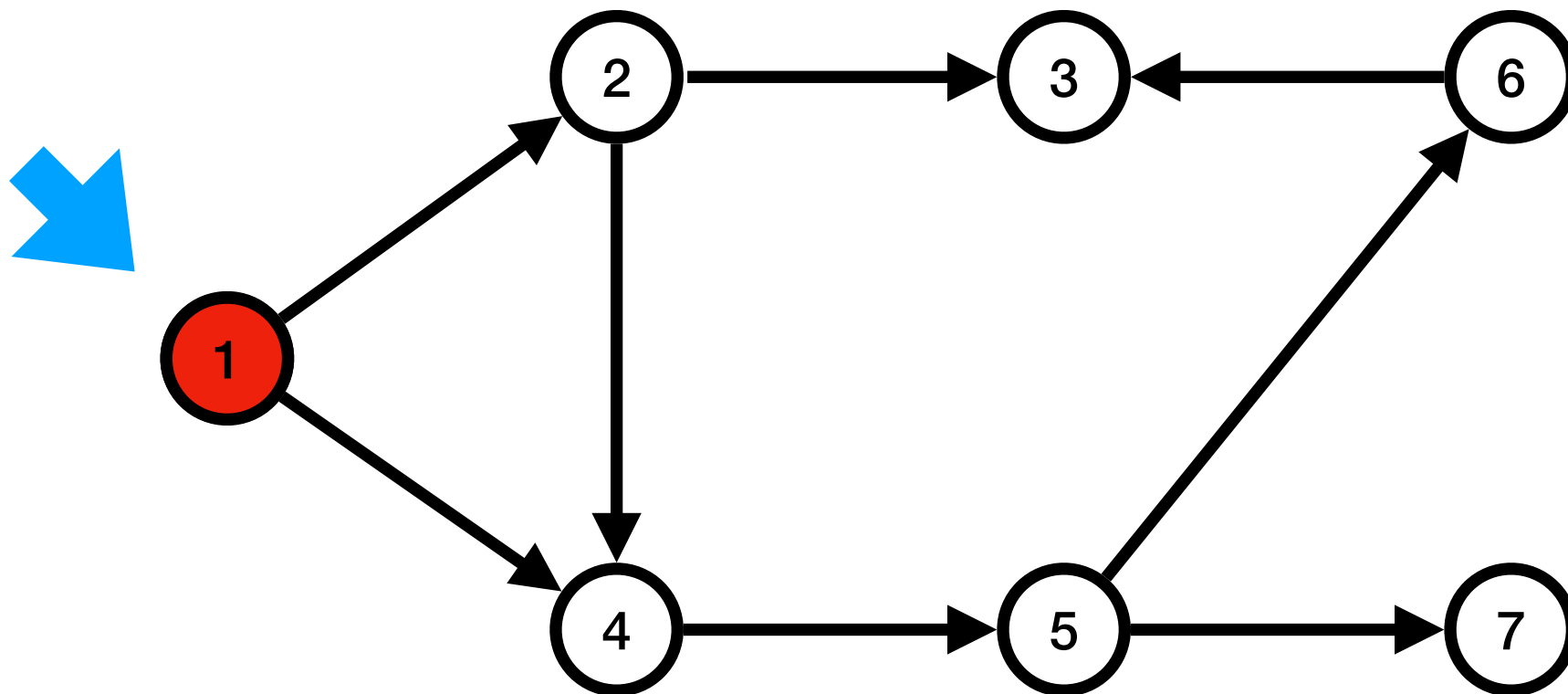


Parcours de graphes

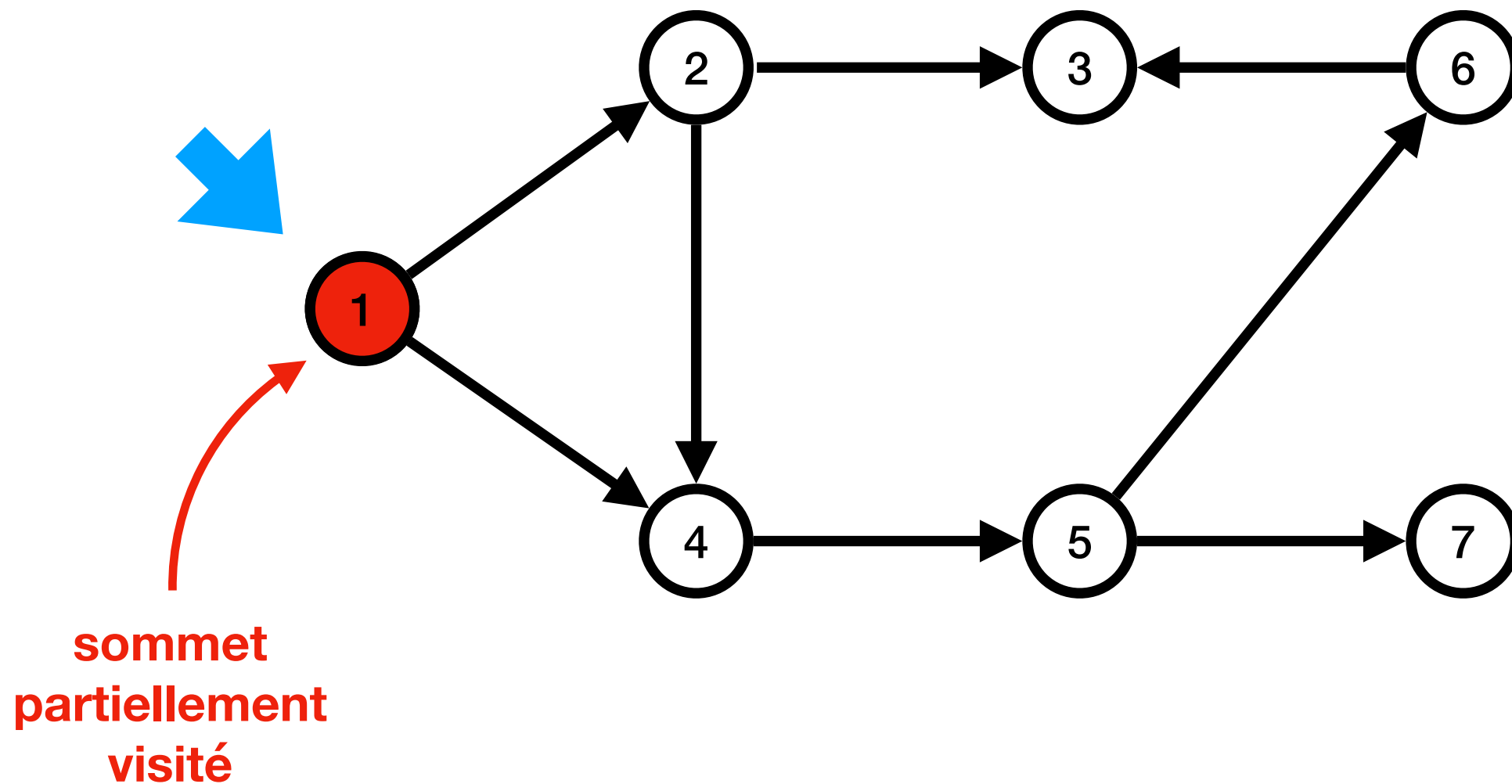
Parcours en largeur



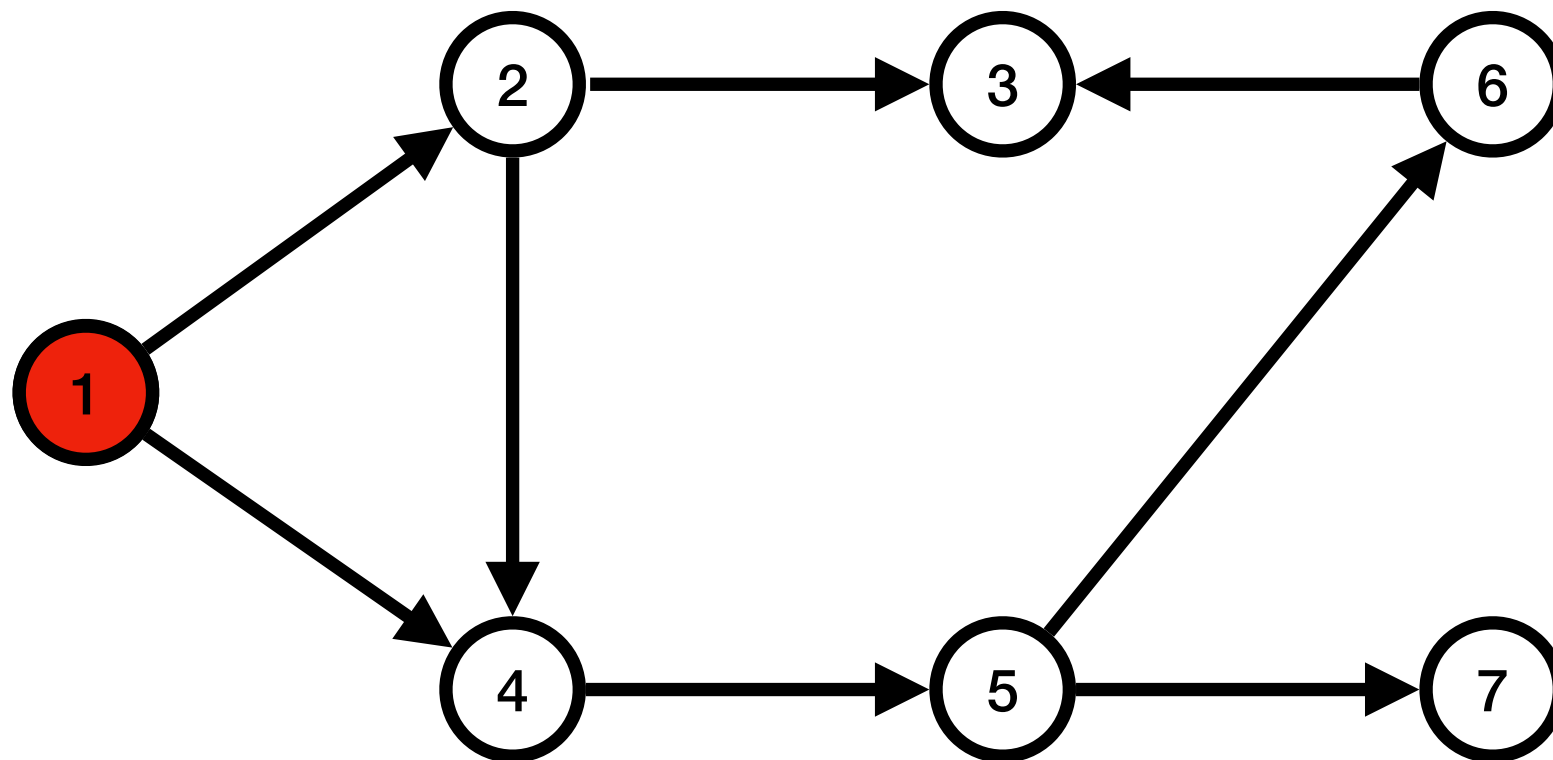
Parcours en largeur



Parcours en largeur

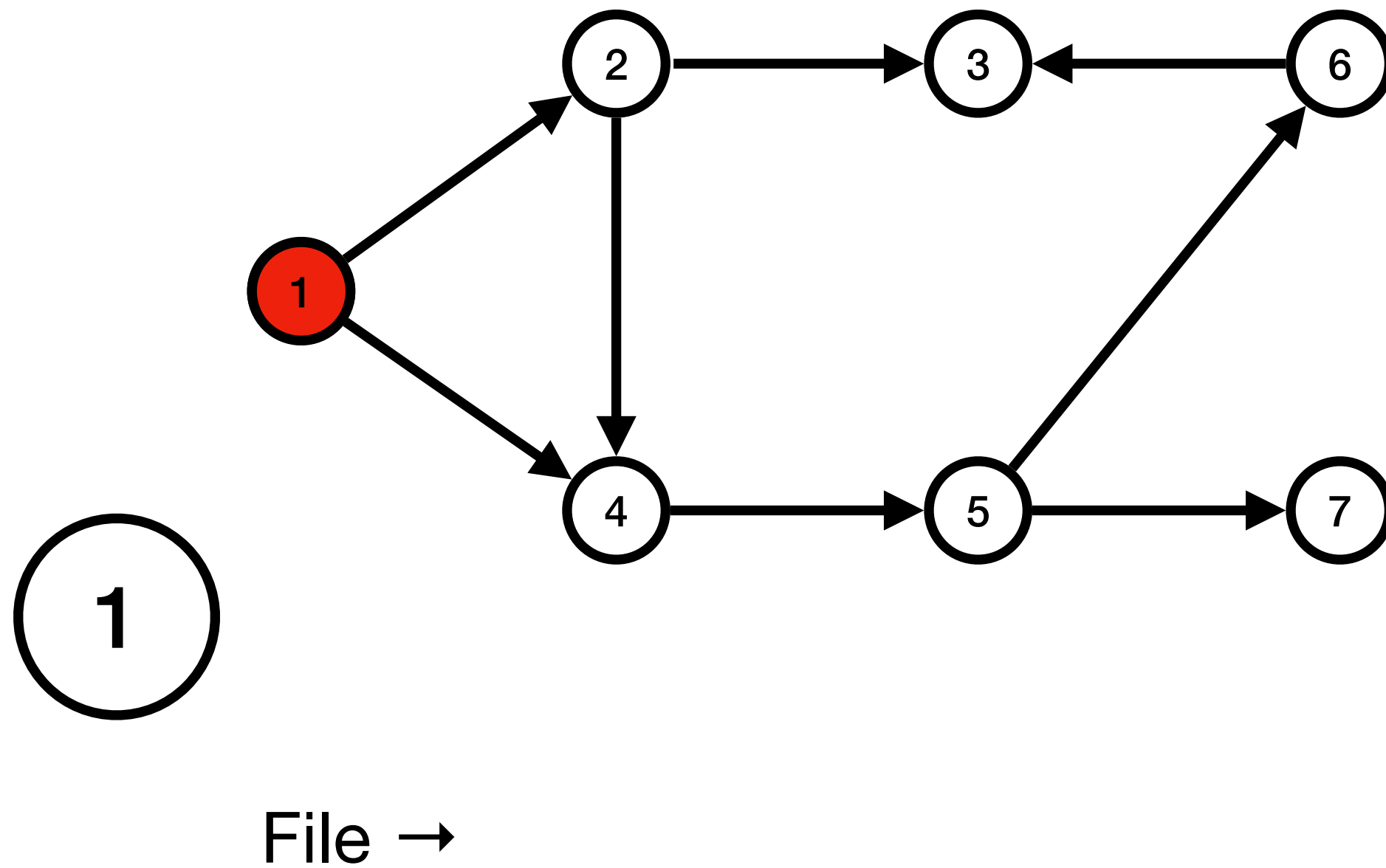


Parcours en largeur

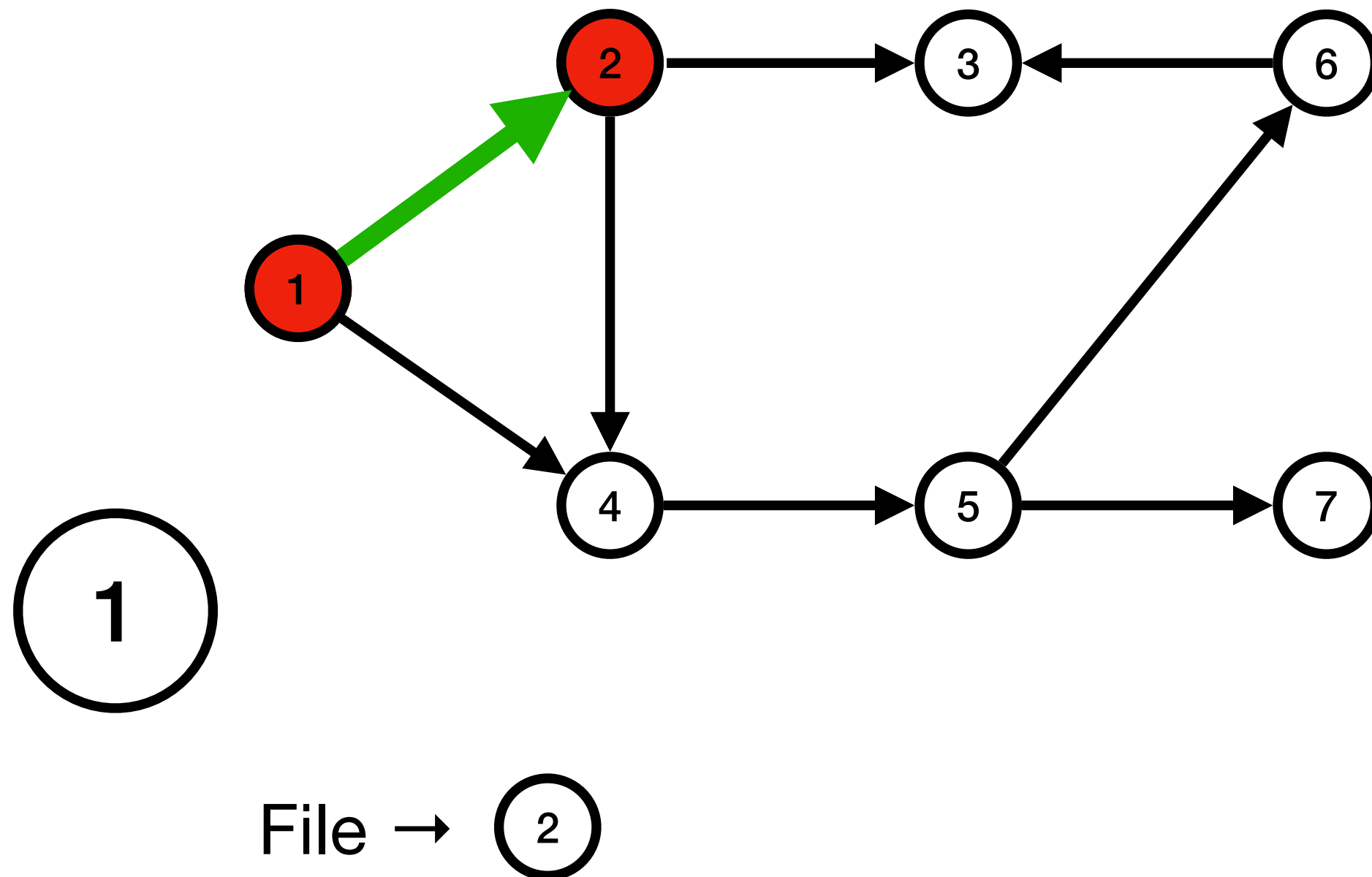


File → 1

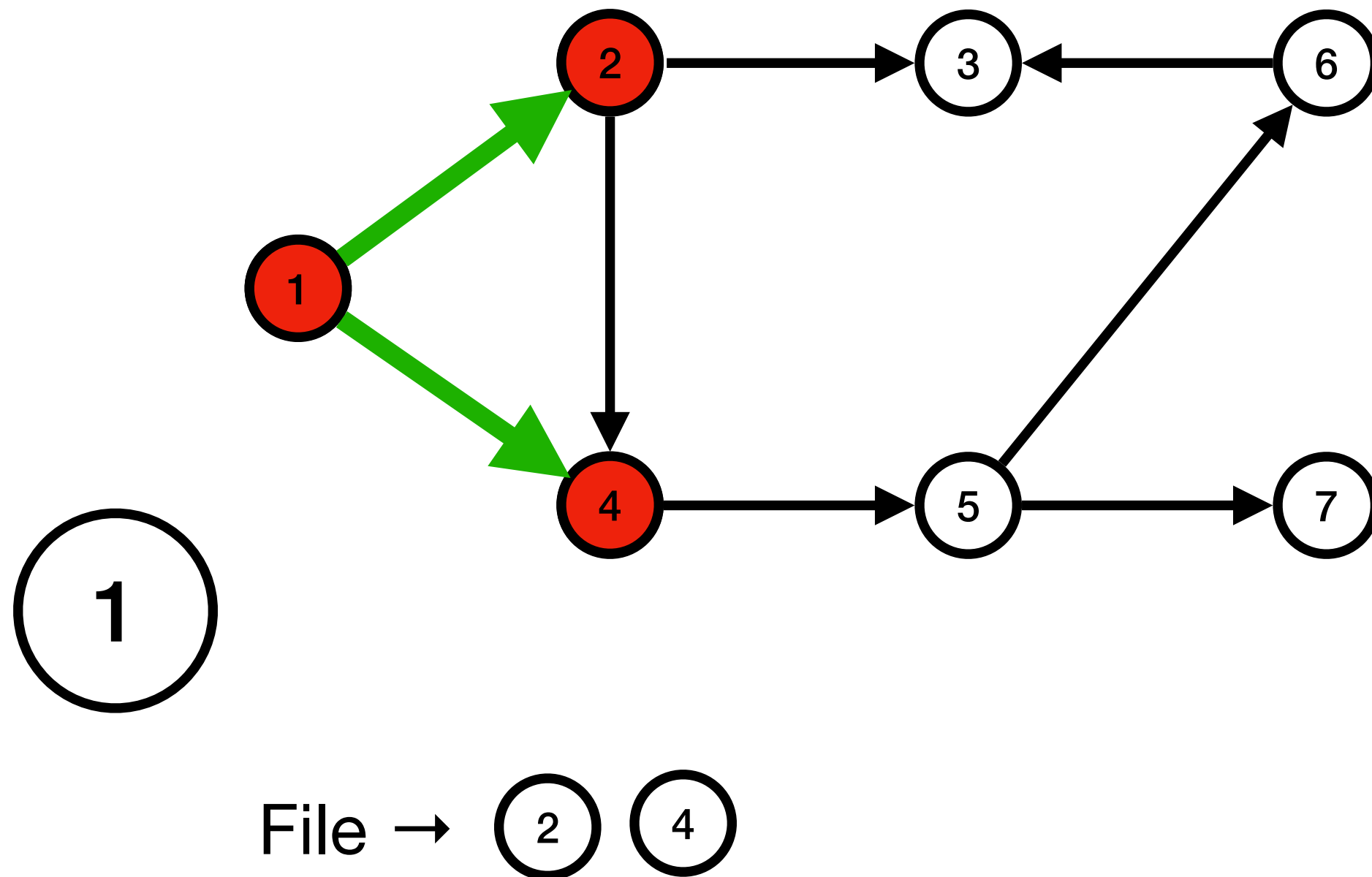
Parcours en largeur



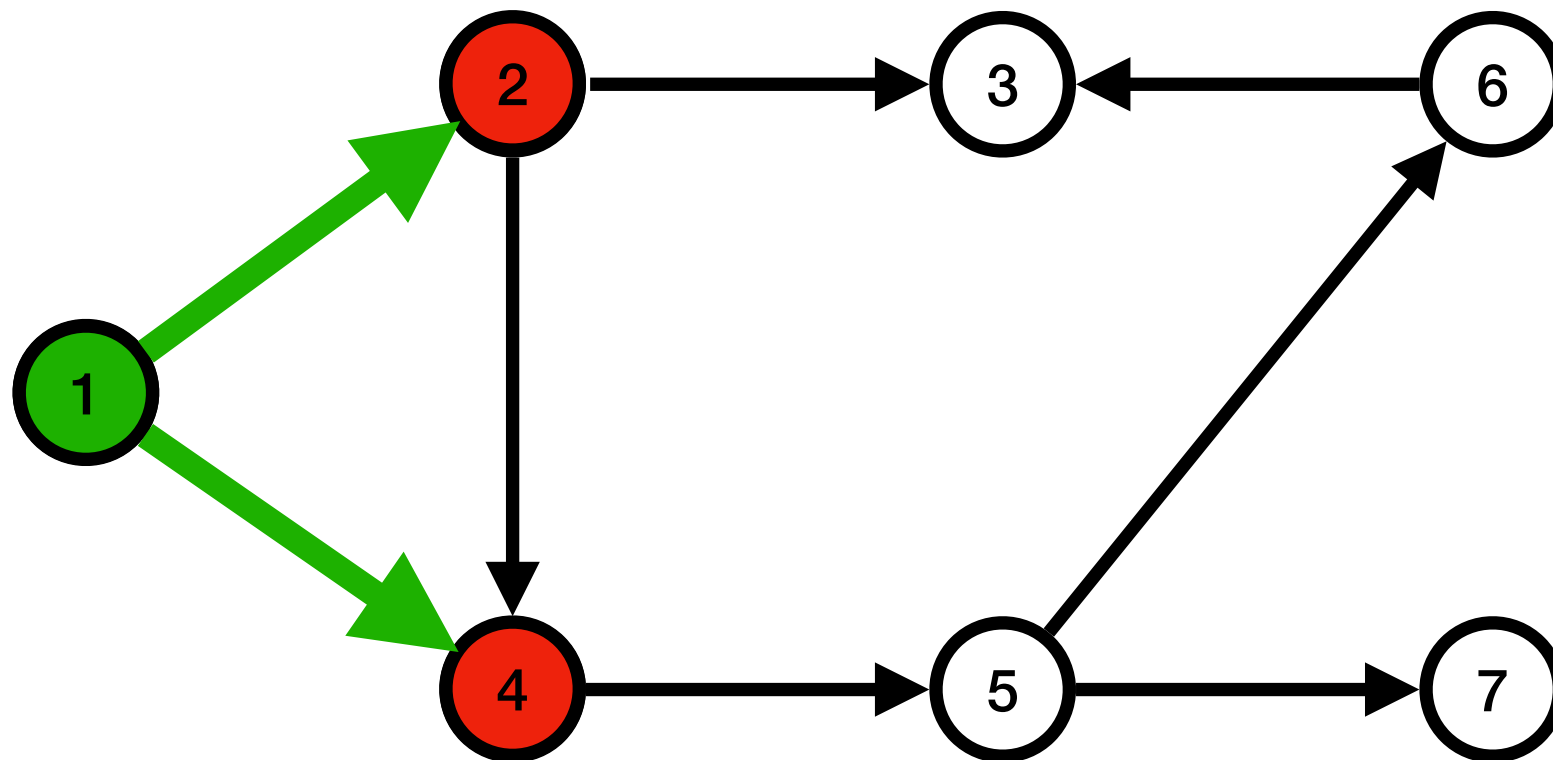
Parcours en largeur



Parcours en largeur

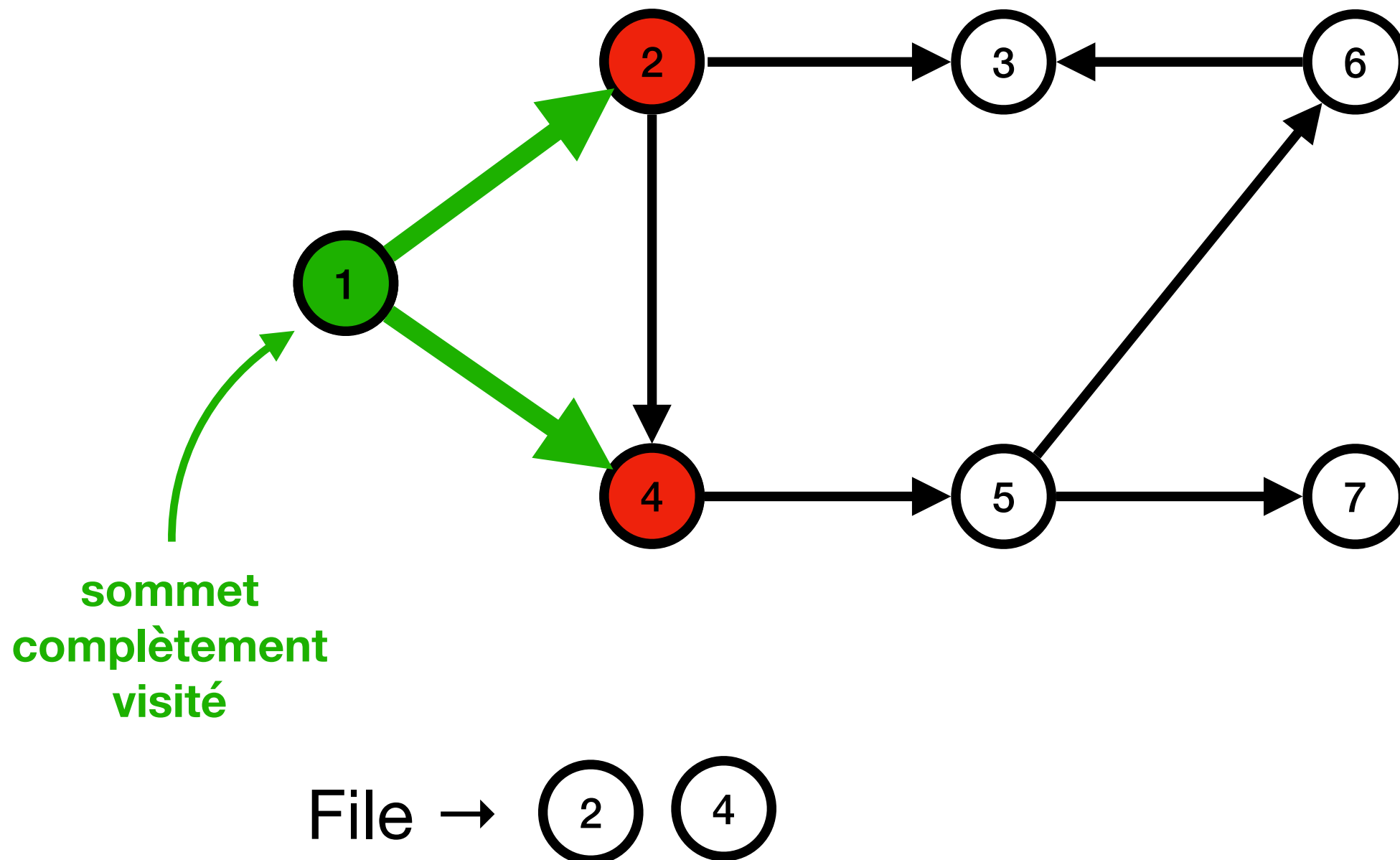


Parcours en largeur

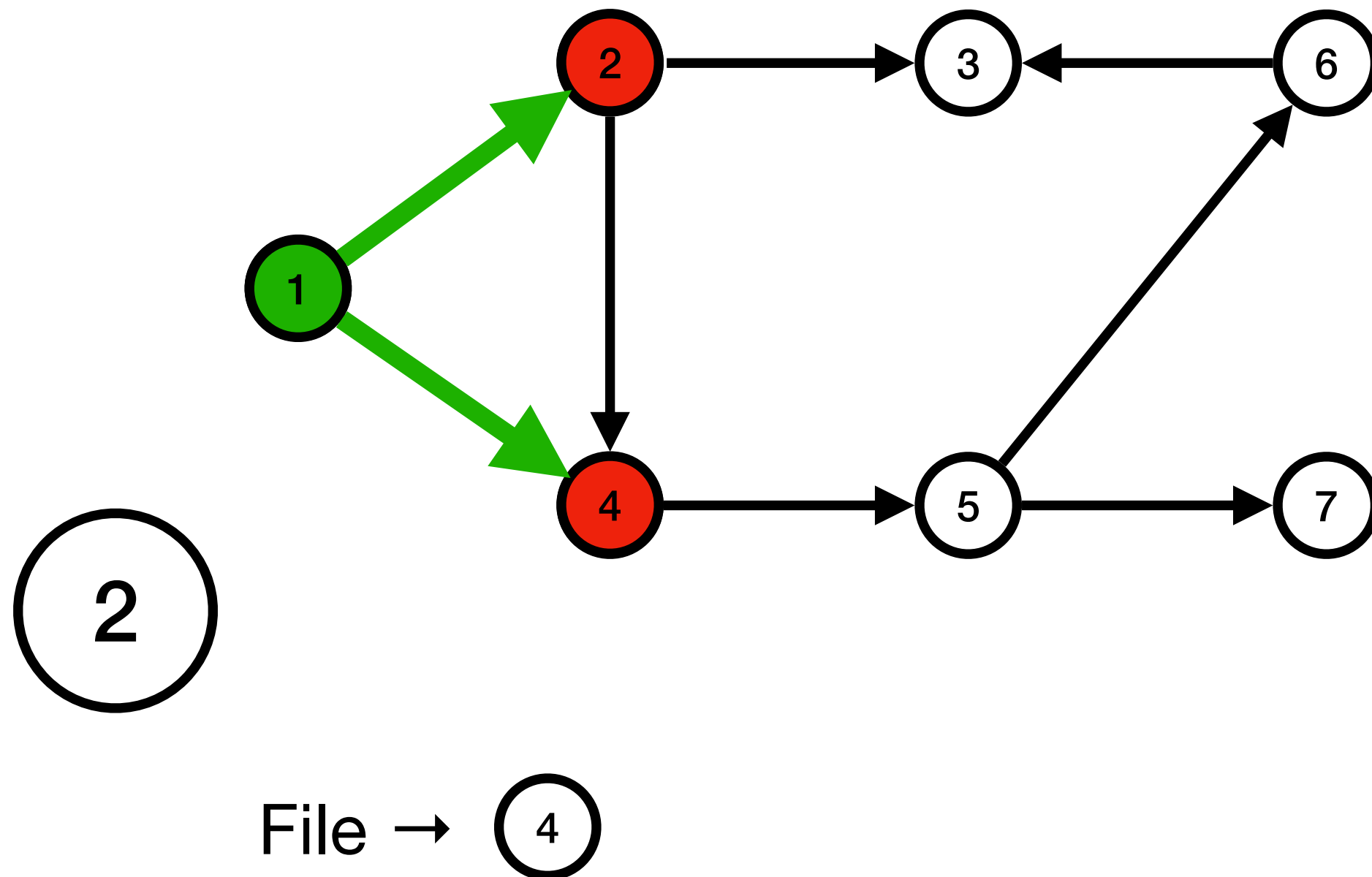


File → (2) (4)

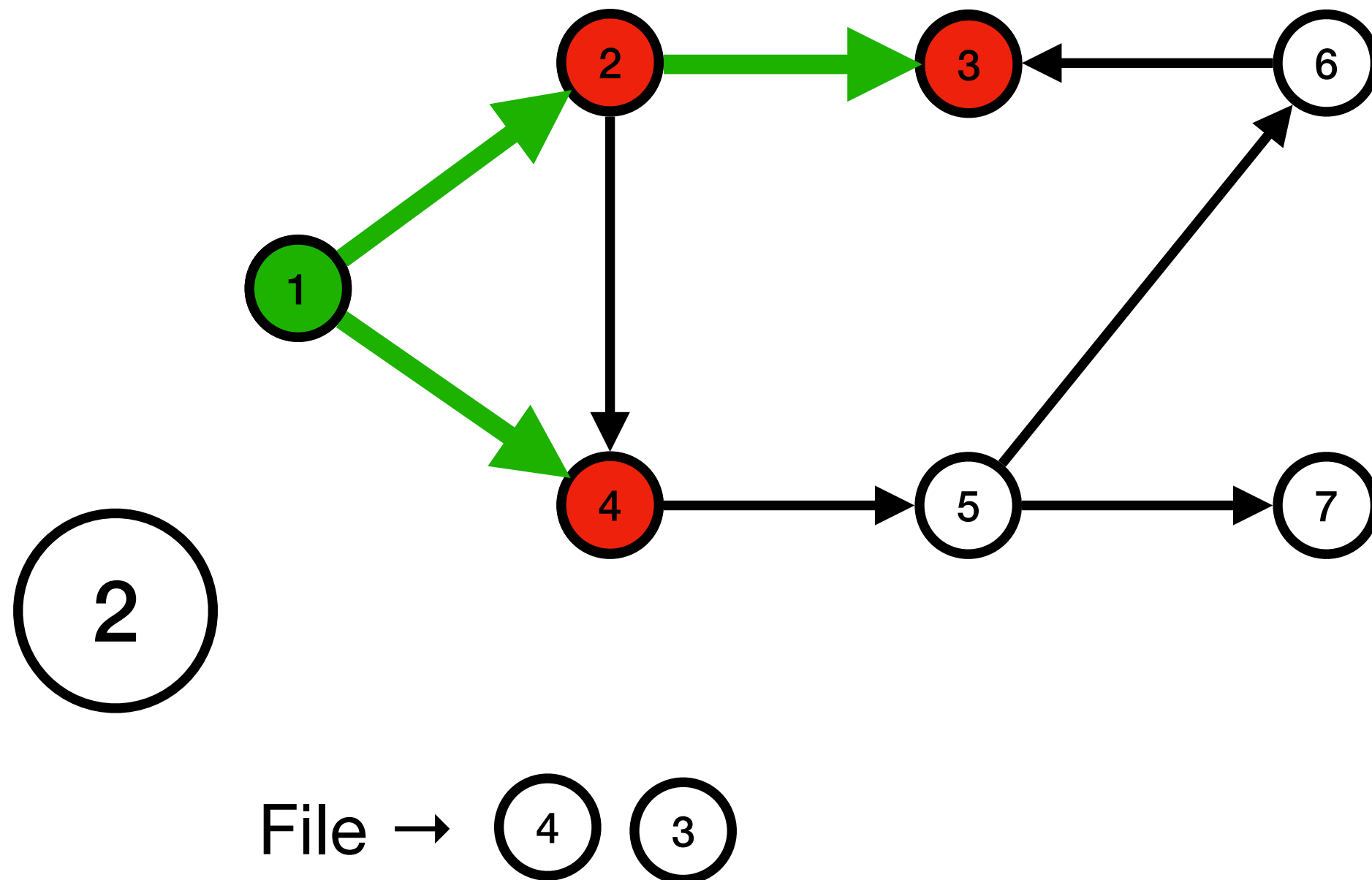
Parcours en largeur



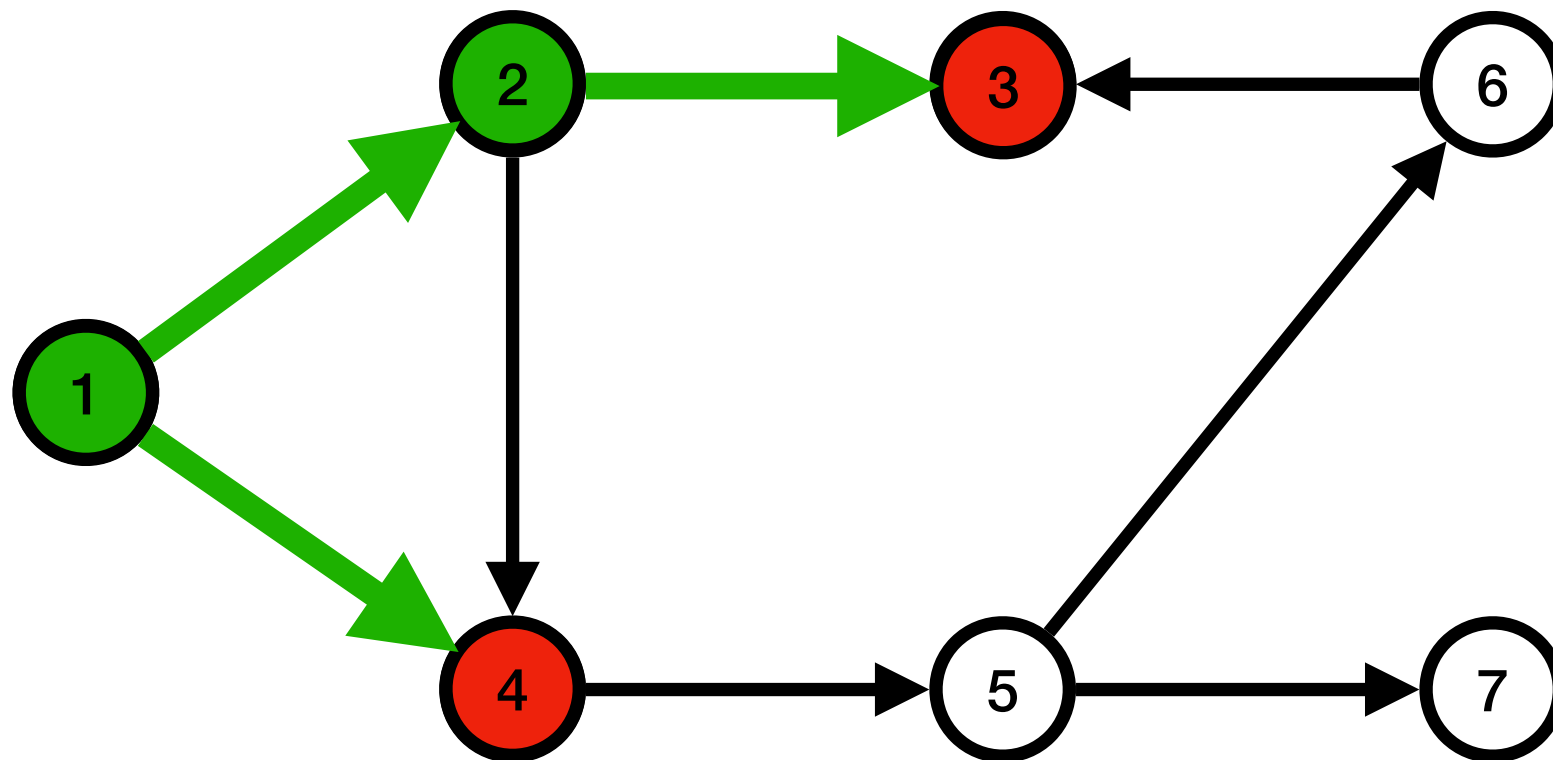
Parcours en largeur



Parcours en largeur

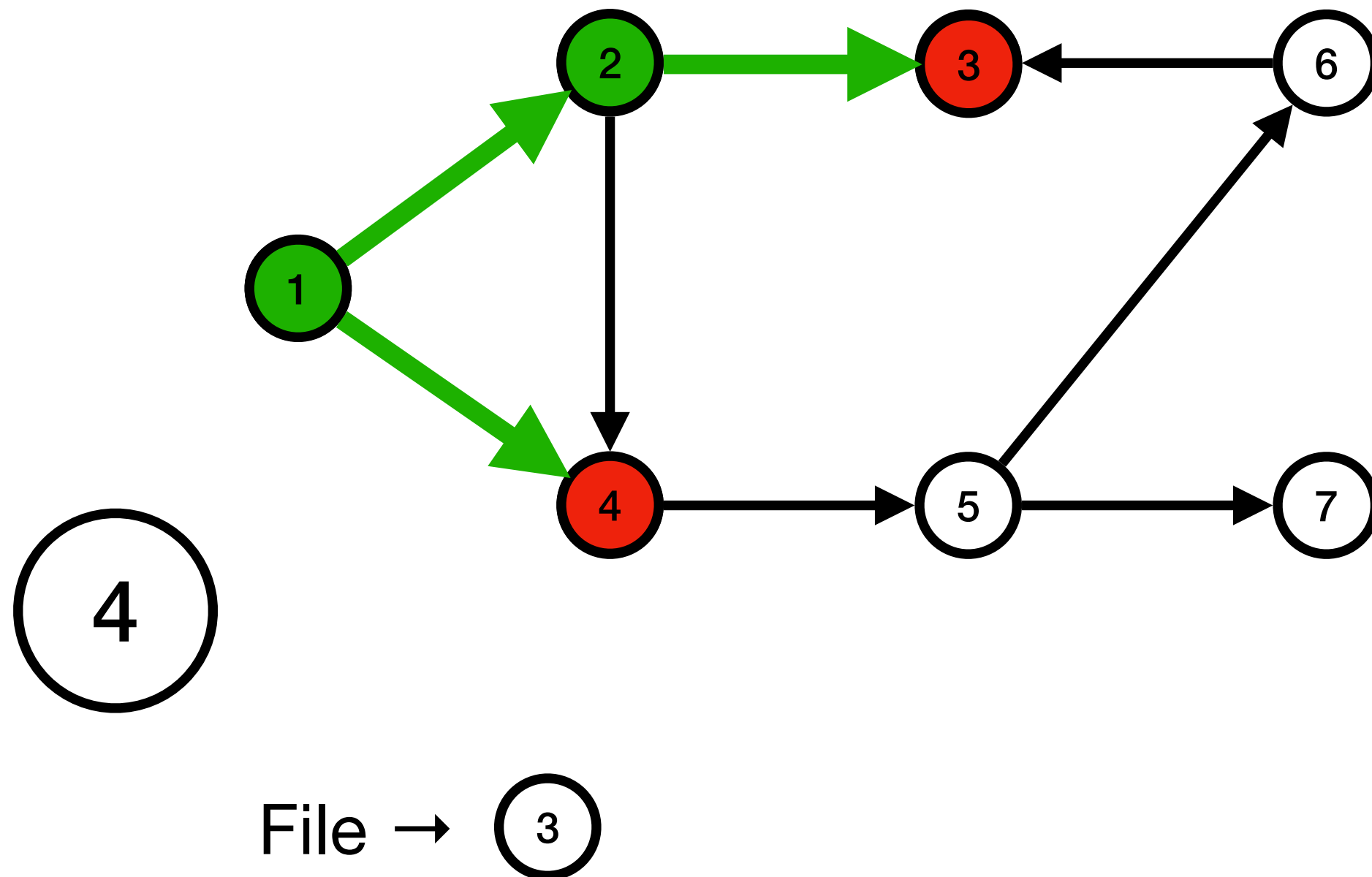


Parcours en largeur

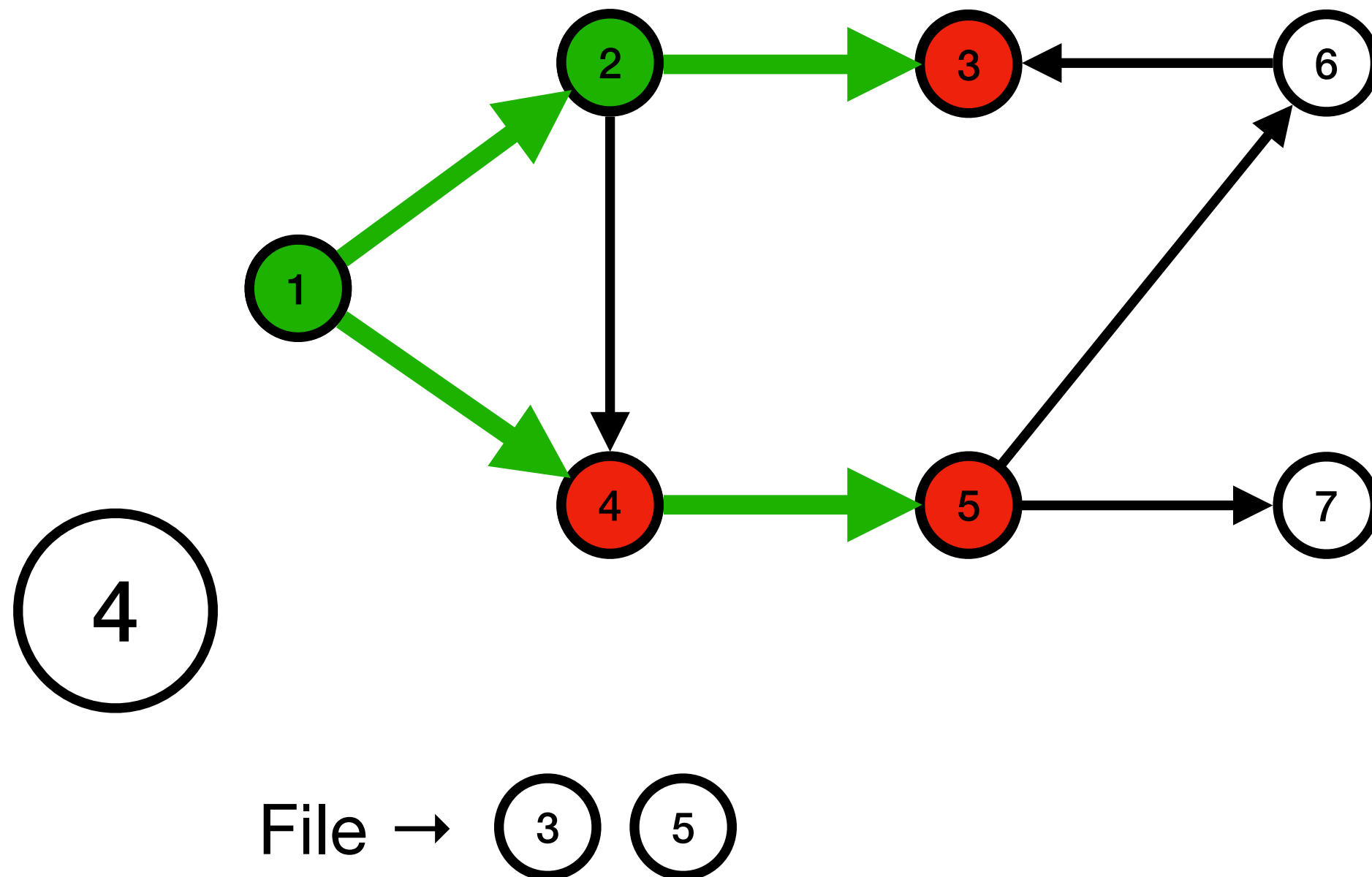


File → 4 3

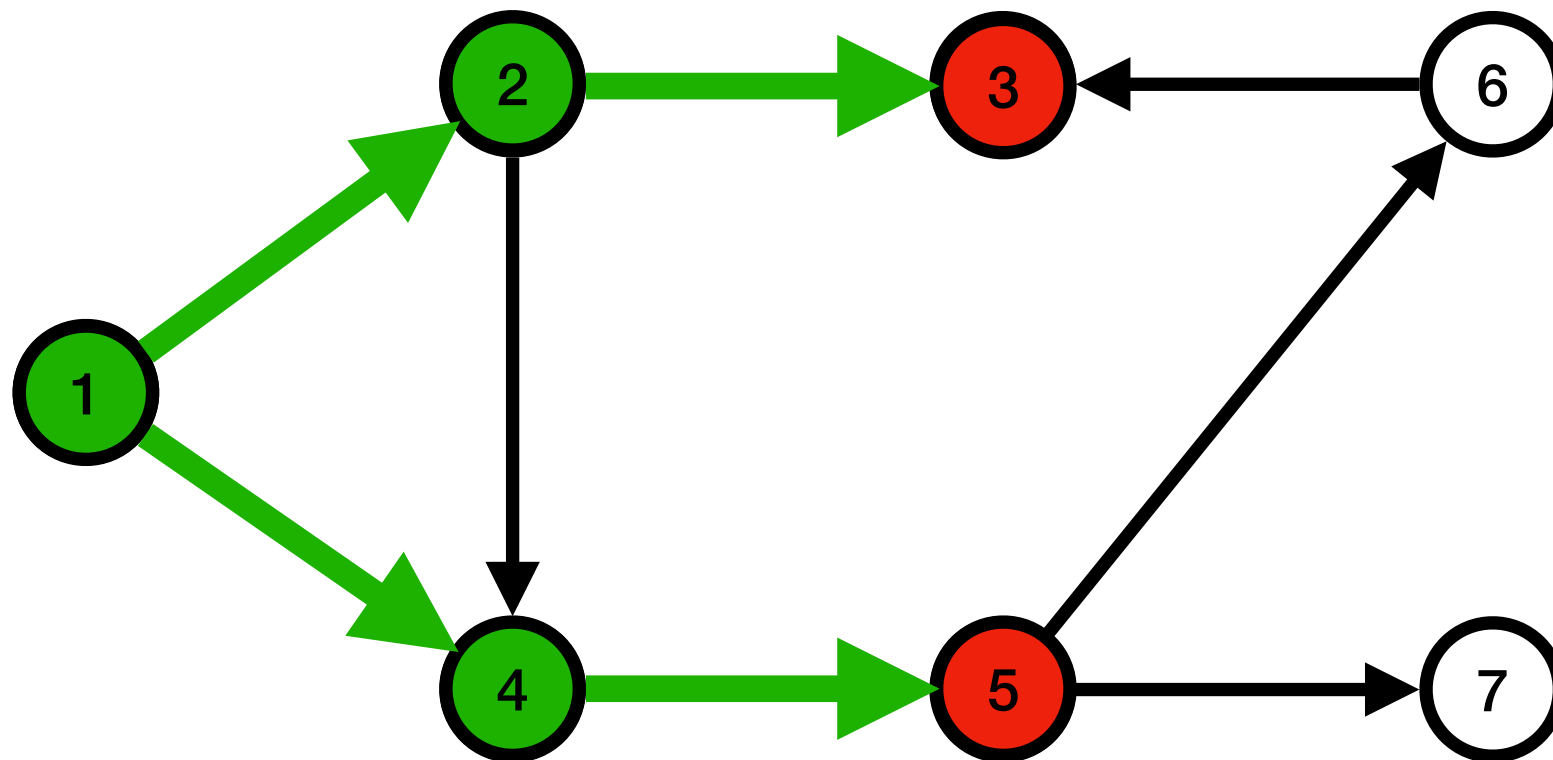
Parcours en largeur



Parcours en largeur

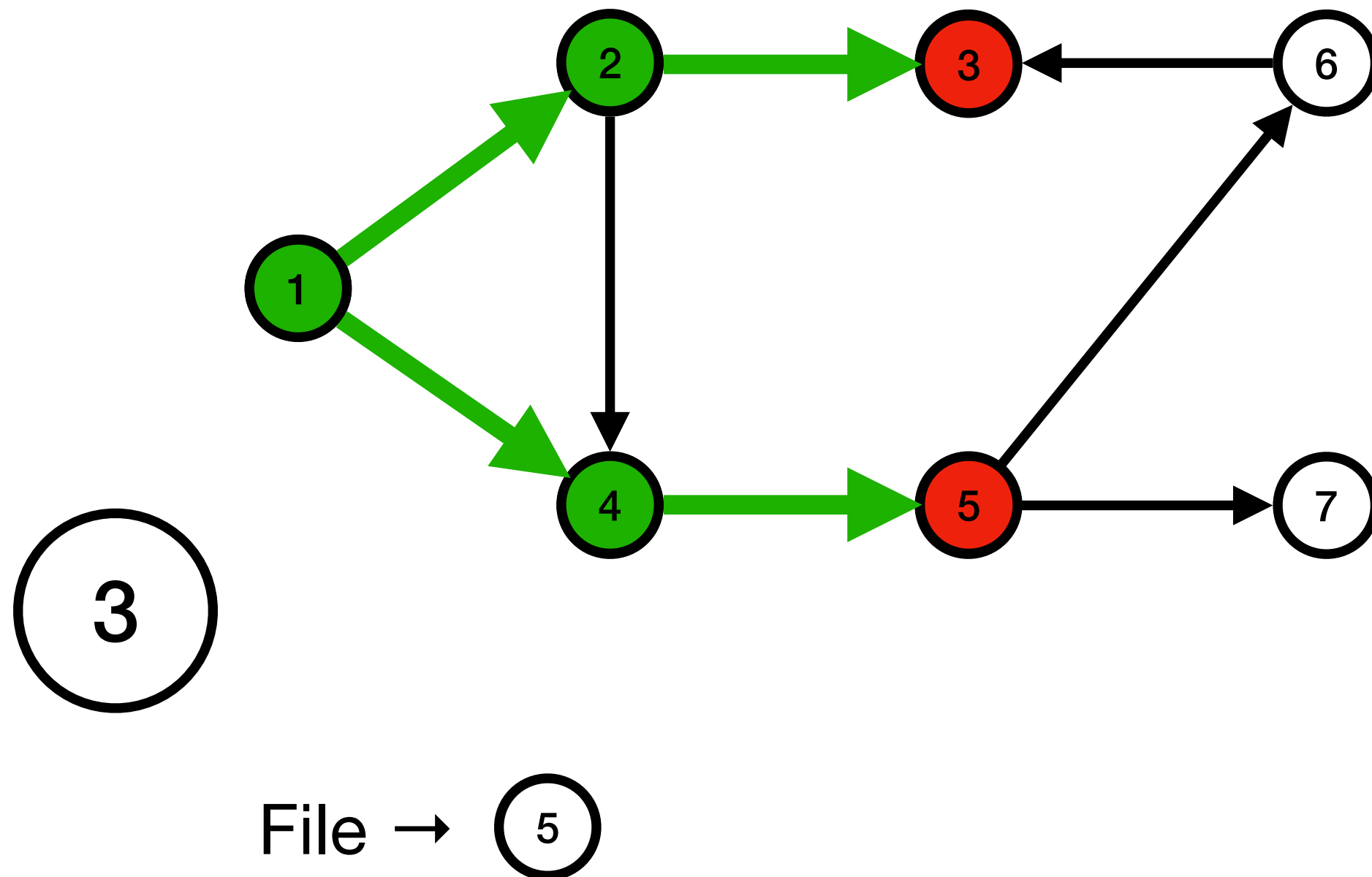


Parcours en largeur

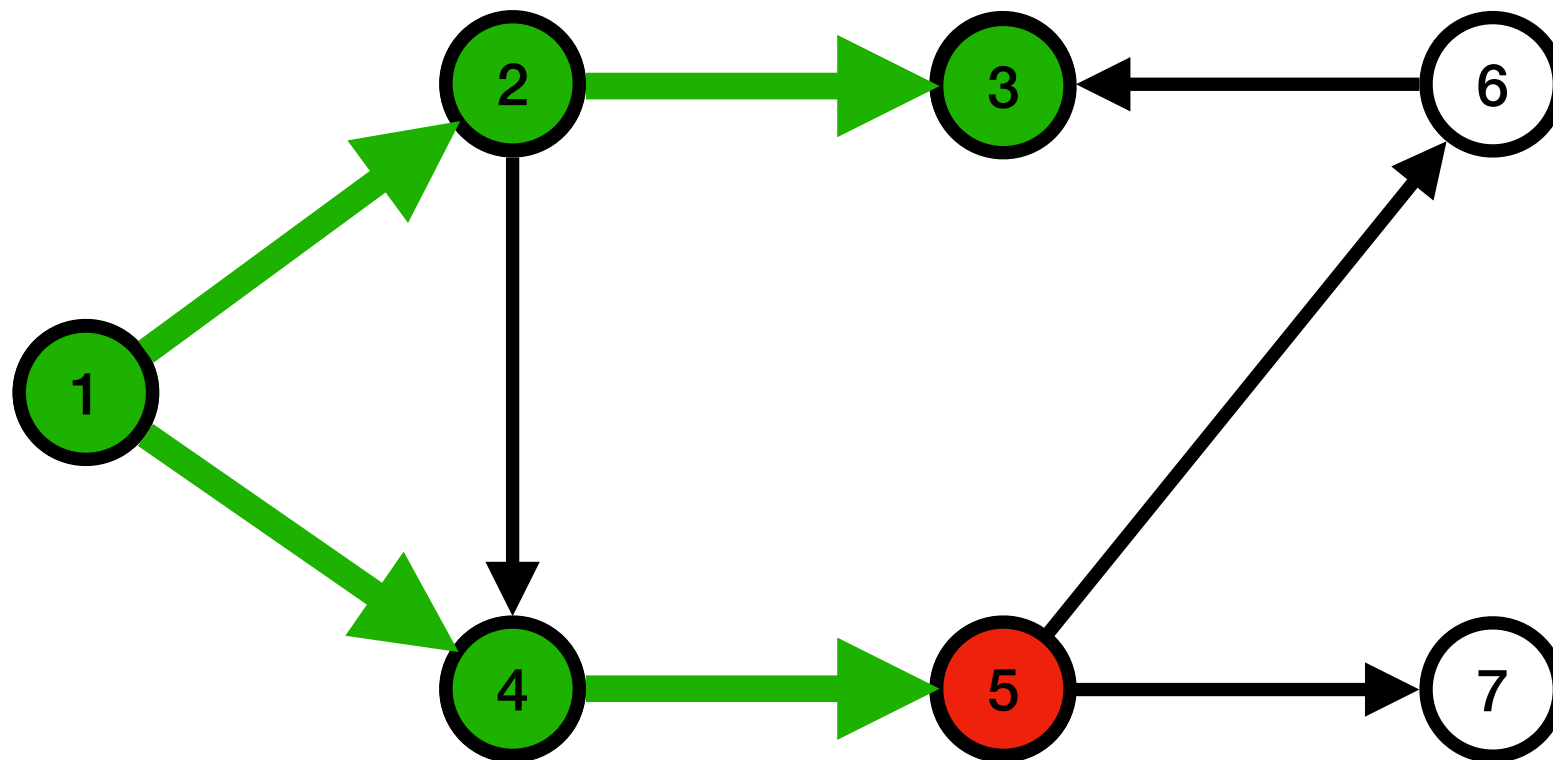


File → (3) (5)

Parcours en largeur

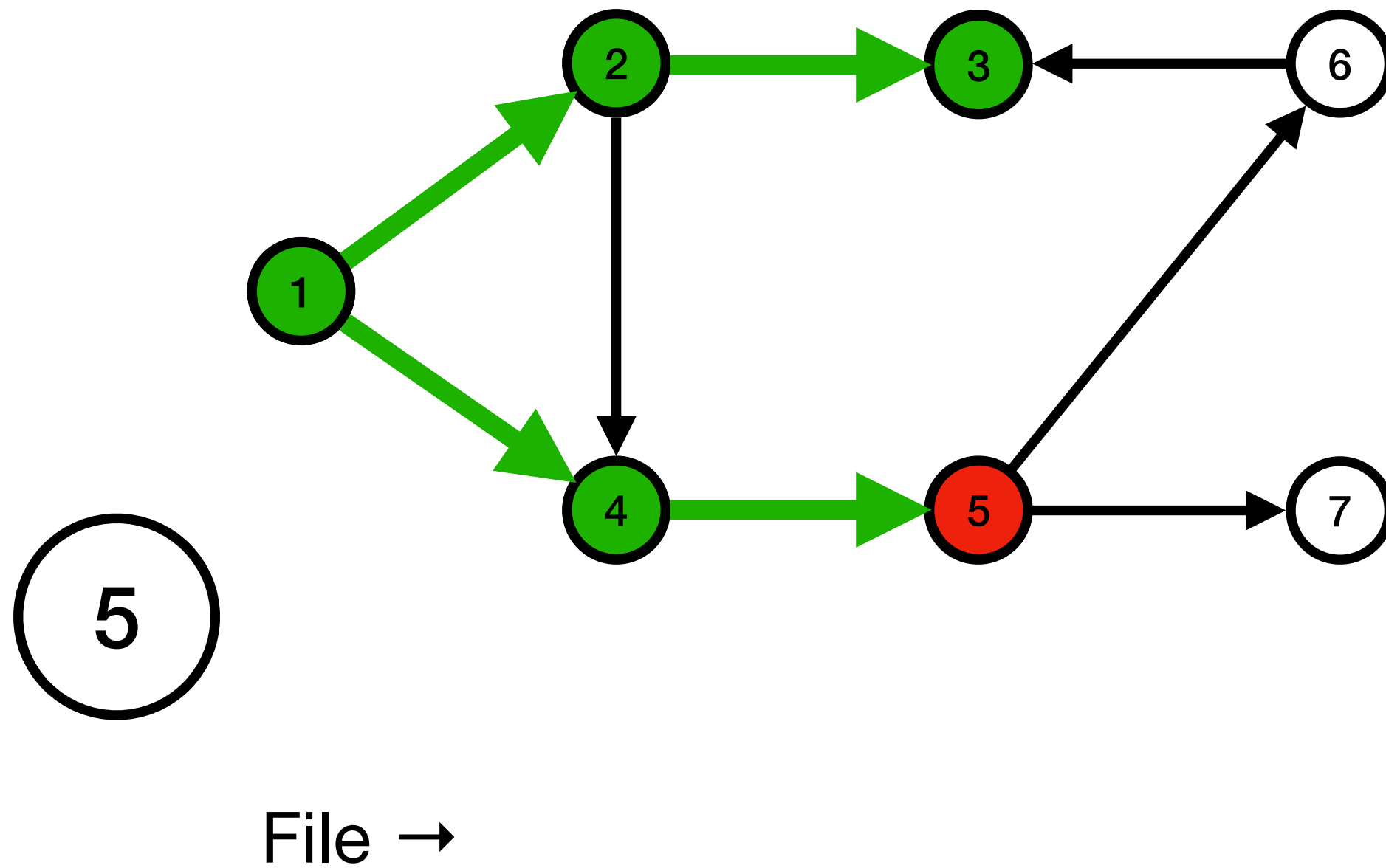


Parcours en largeur

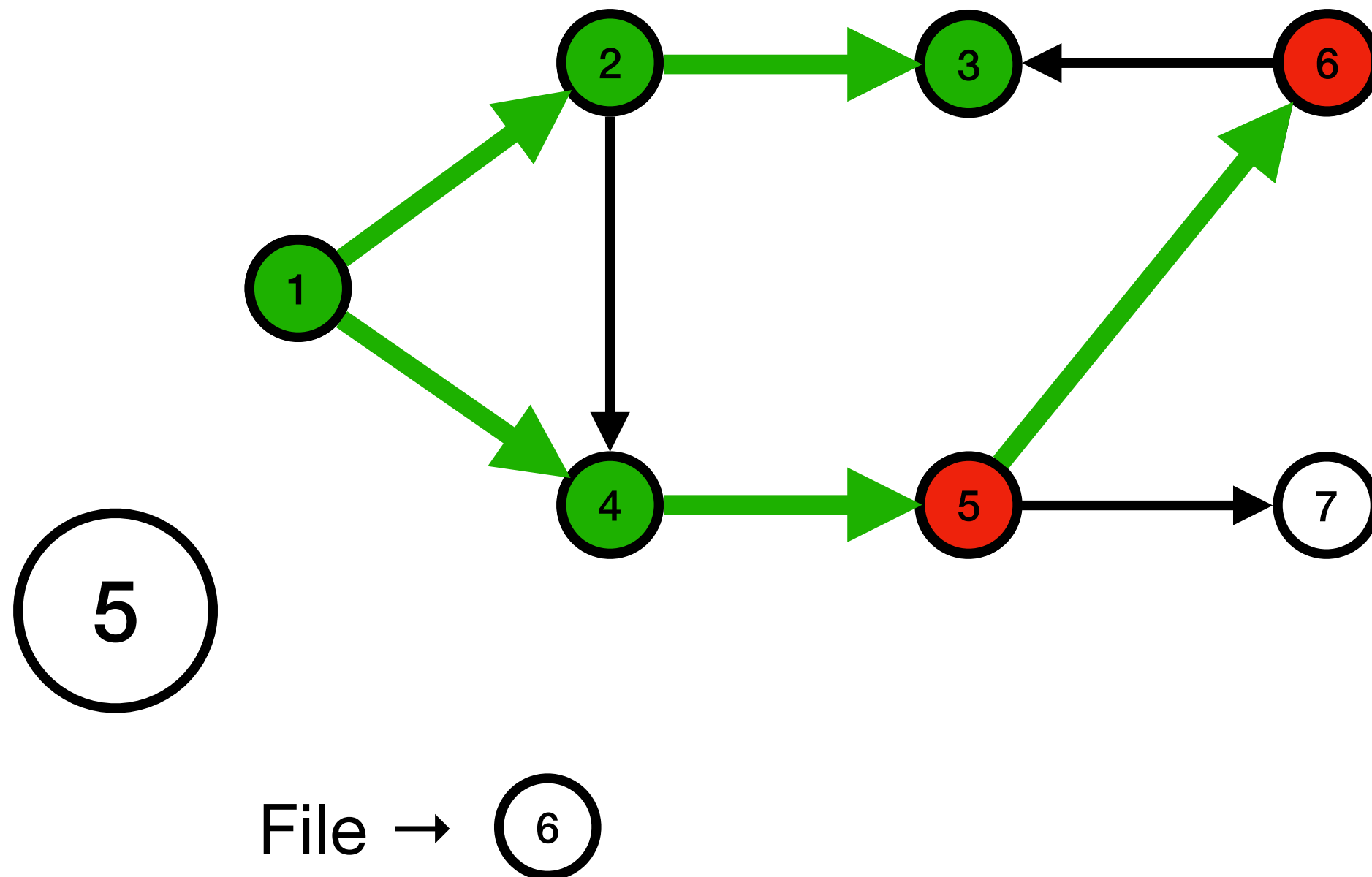


File → 5

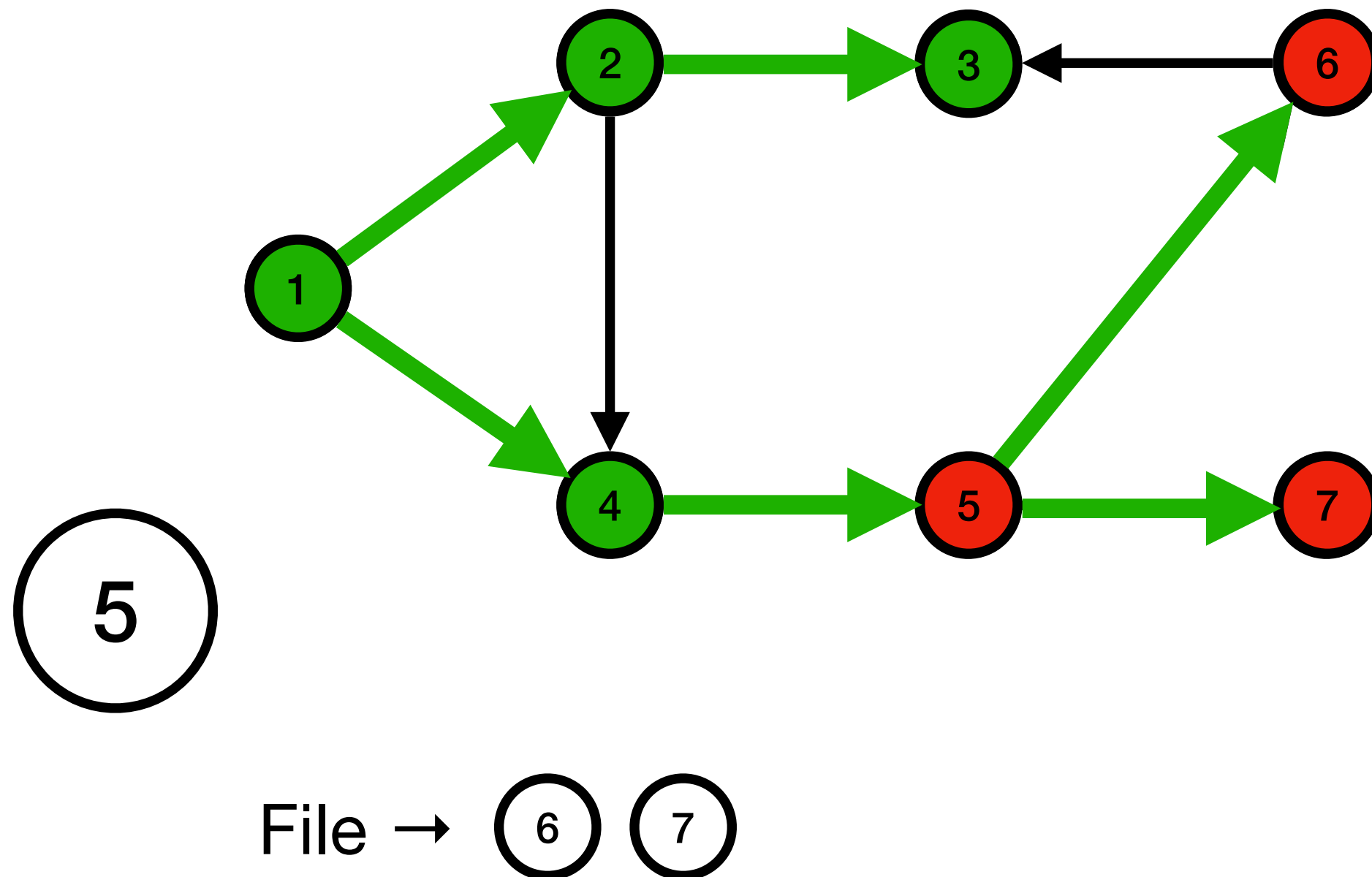
Parcours en largeur



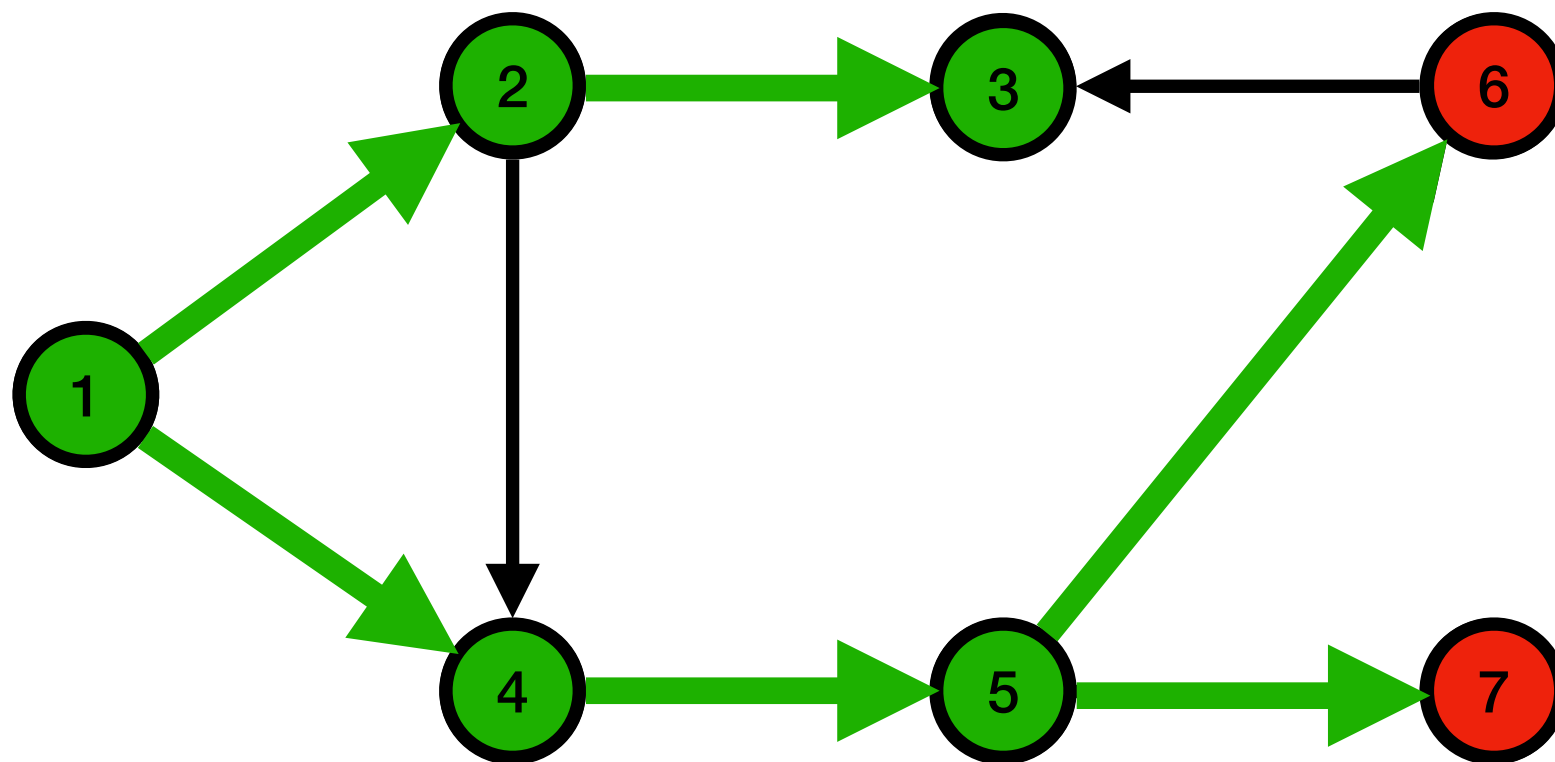
Parcours en largeur



Parcours en largeur

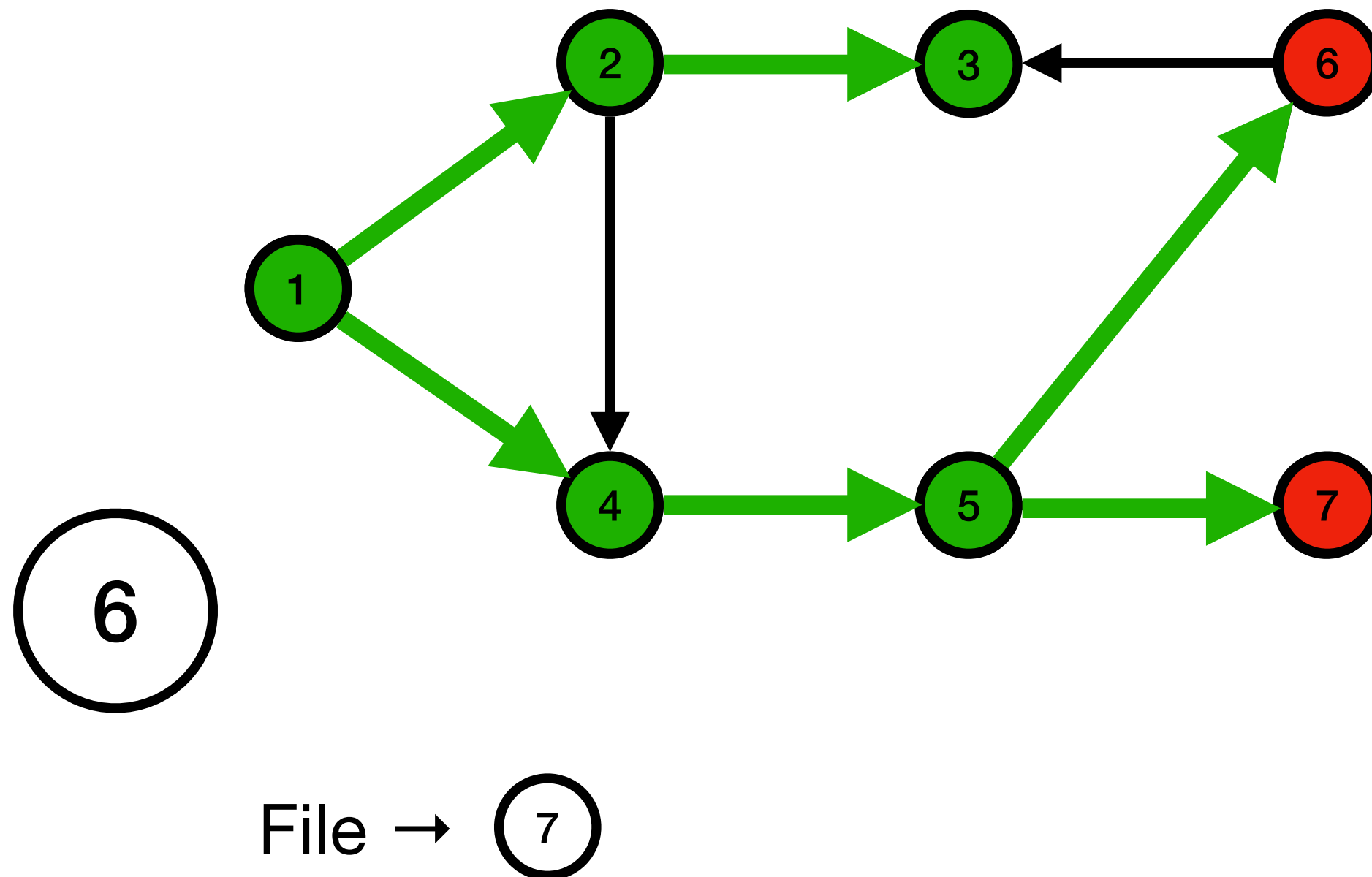


Parcours en largeur

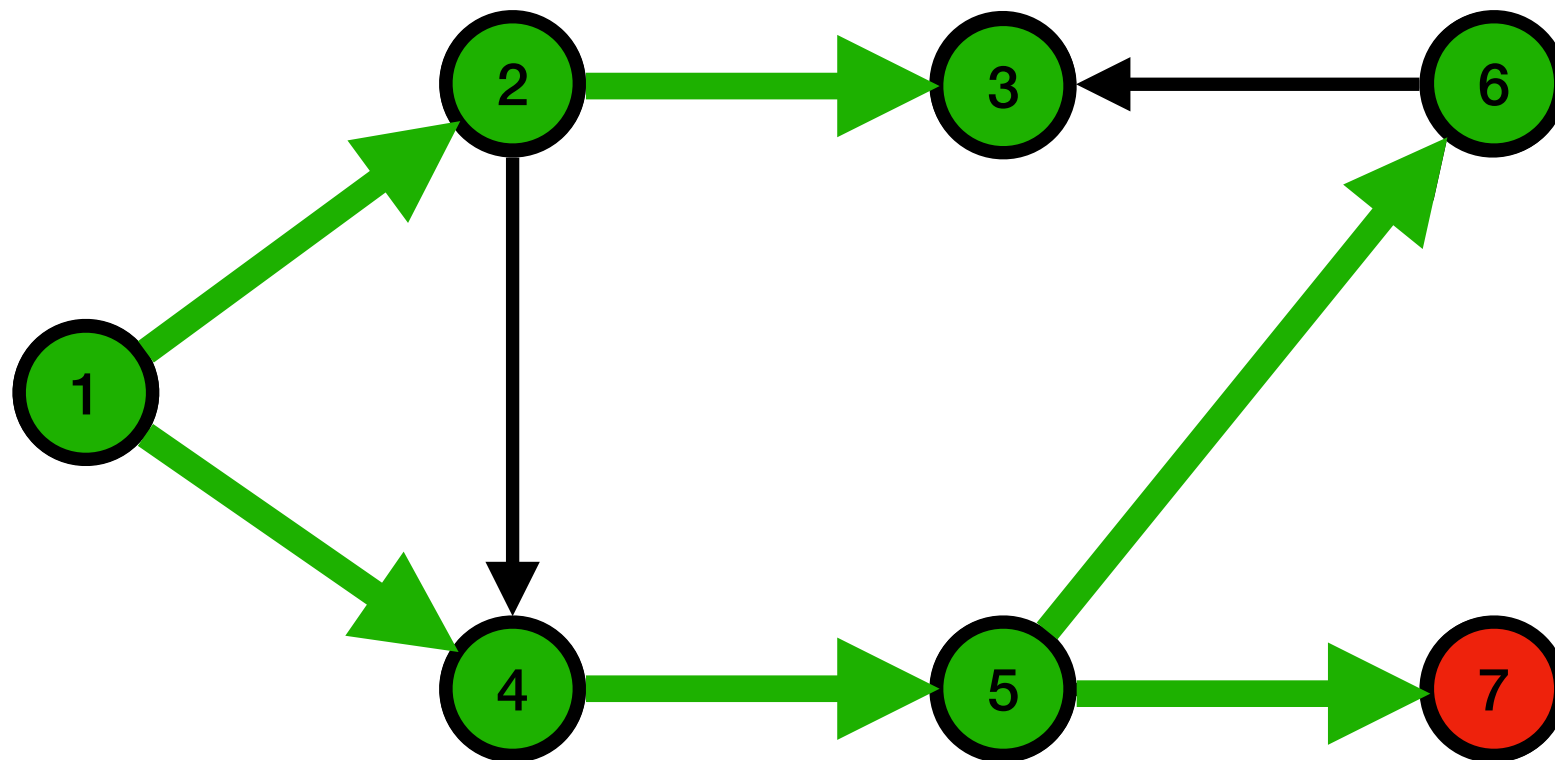


File → (6) (7)

Parcours en largeur

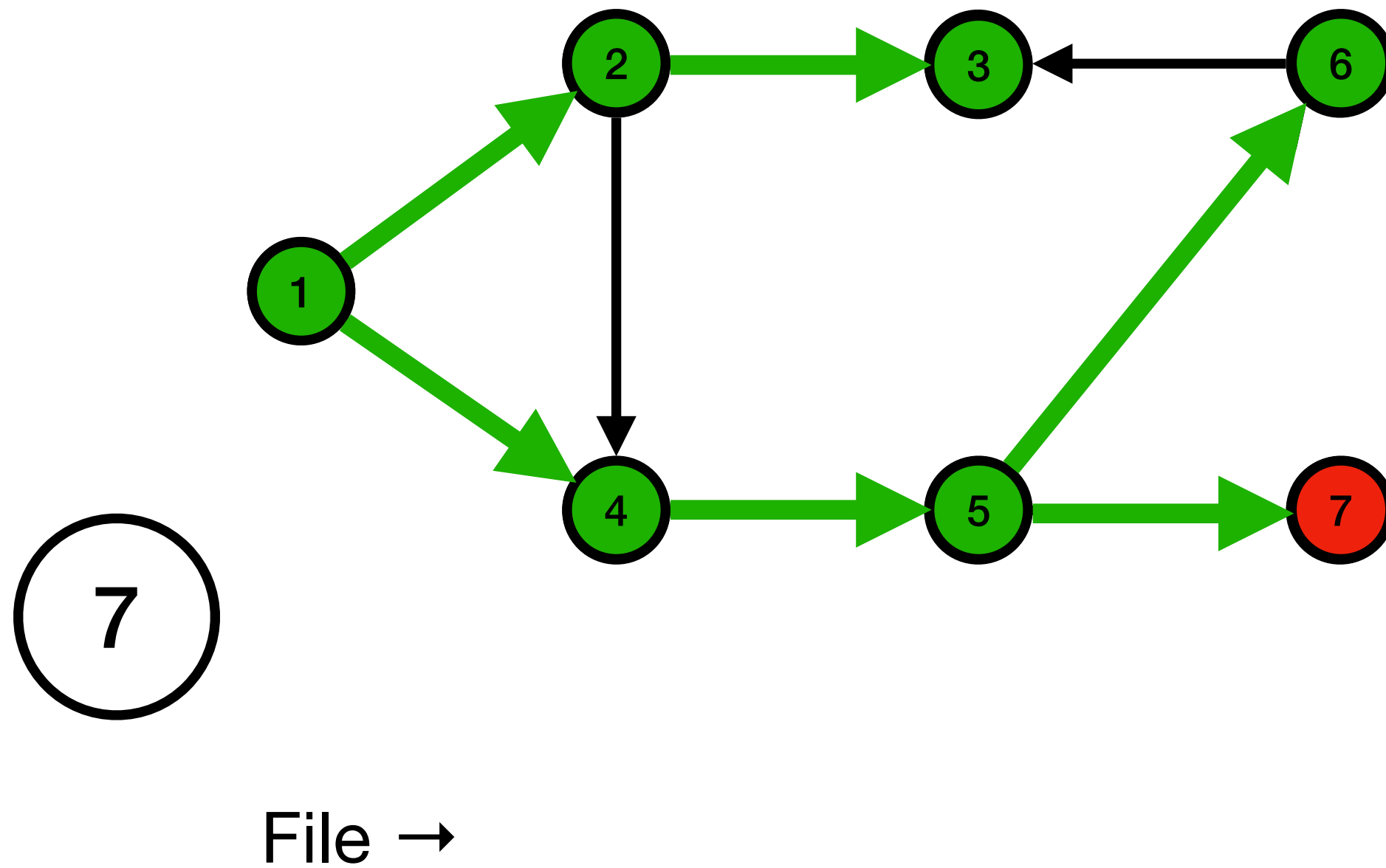


Parcours en largeur

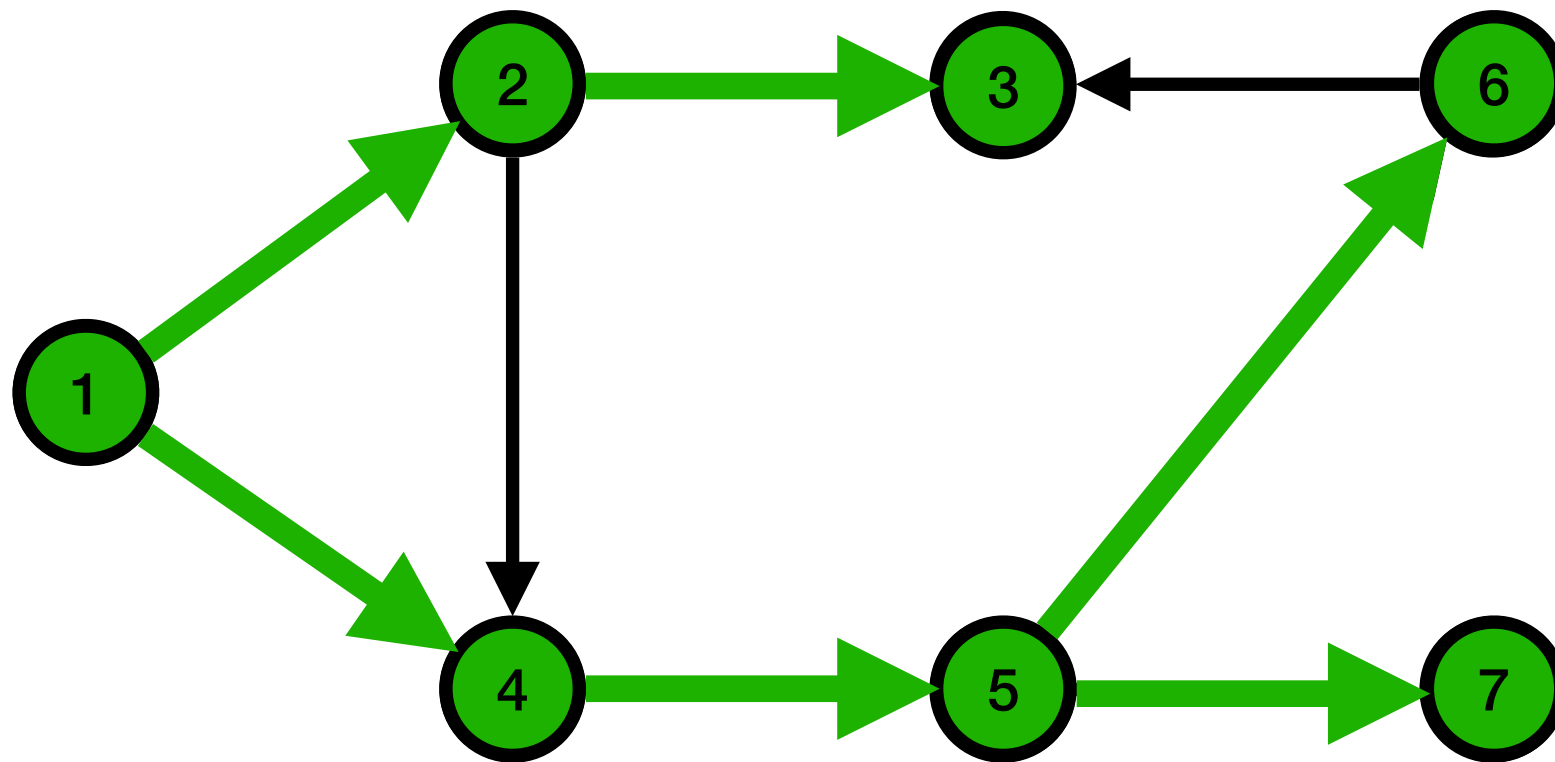


File → (7)

Parcours en largeur

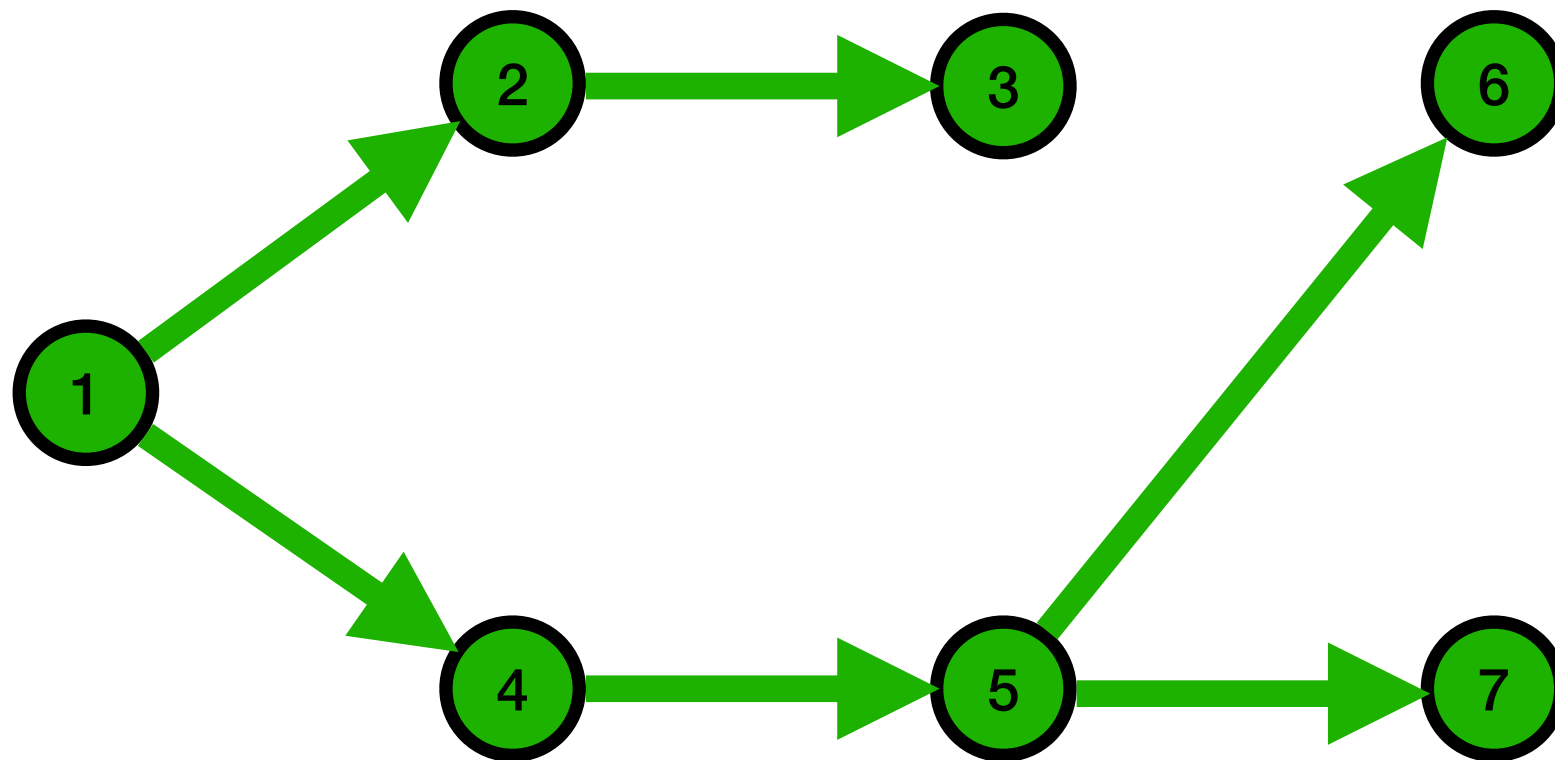


Parcours en largeur



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)

$F \rightarrow 1$

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 »

$x := \text{défiler}(F)$

$F \rightarrow 4 \quad 5$

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 »

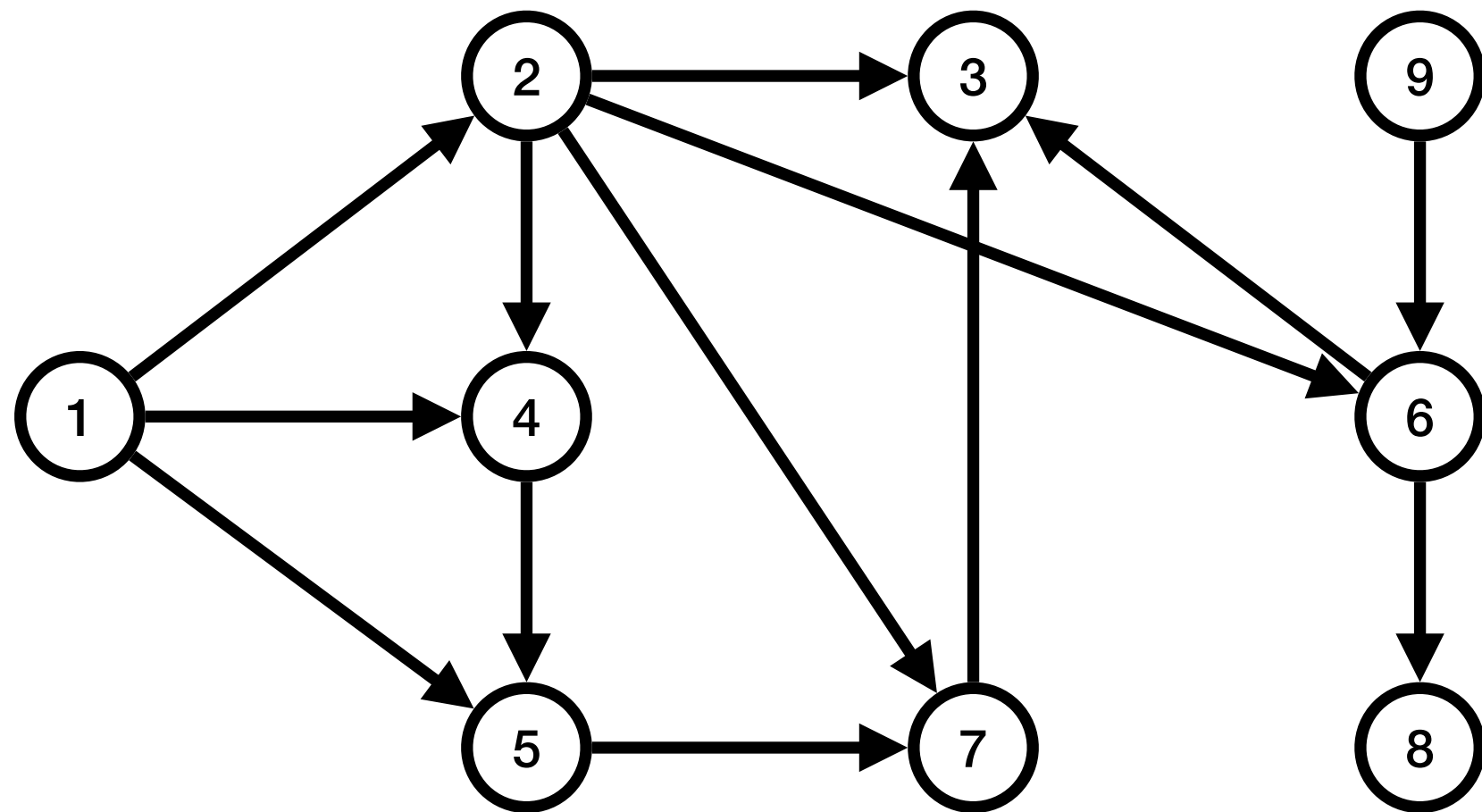
$F \rightarrow$

```

fonction parcours-en-largeur(G, s)
  n := |G| (nombre de sommets)
  pour v := 0 jusqu'à n – 1 faire
    couleur[v] := blanc
  fin pour
  H := graphe(n) (graphe vide)
  F := ∅ (file vide)
  couleur[s] := rouge
  enfiler(F, s)
  tant que F ≠ ∅ faire
    u := défiler(F)
    pour v := 0 jusqu'à n – 1 faire
      si G[u, v] = 1 et couleur[v] = blanc alors
        couleur[v] := rouge
        H[u, v] := 1
        enfiler(F, v)
      fin si
    fin pour
    couleur[u] := vert
  fin tant que
  retourner H (graphe des chemins minimaux)
fin fonction

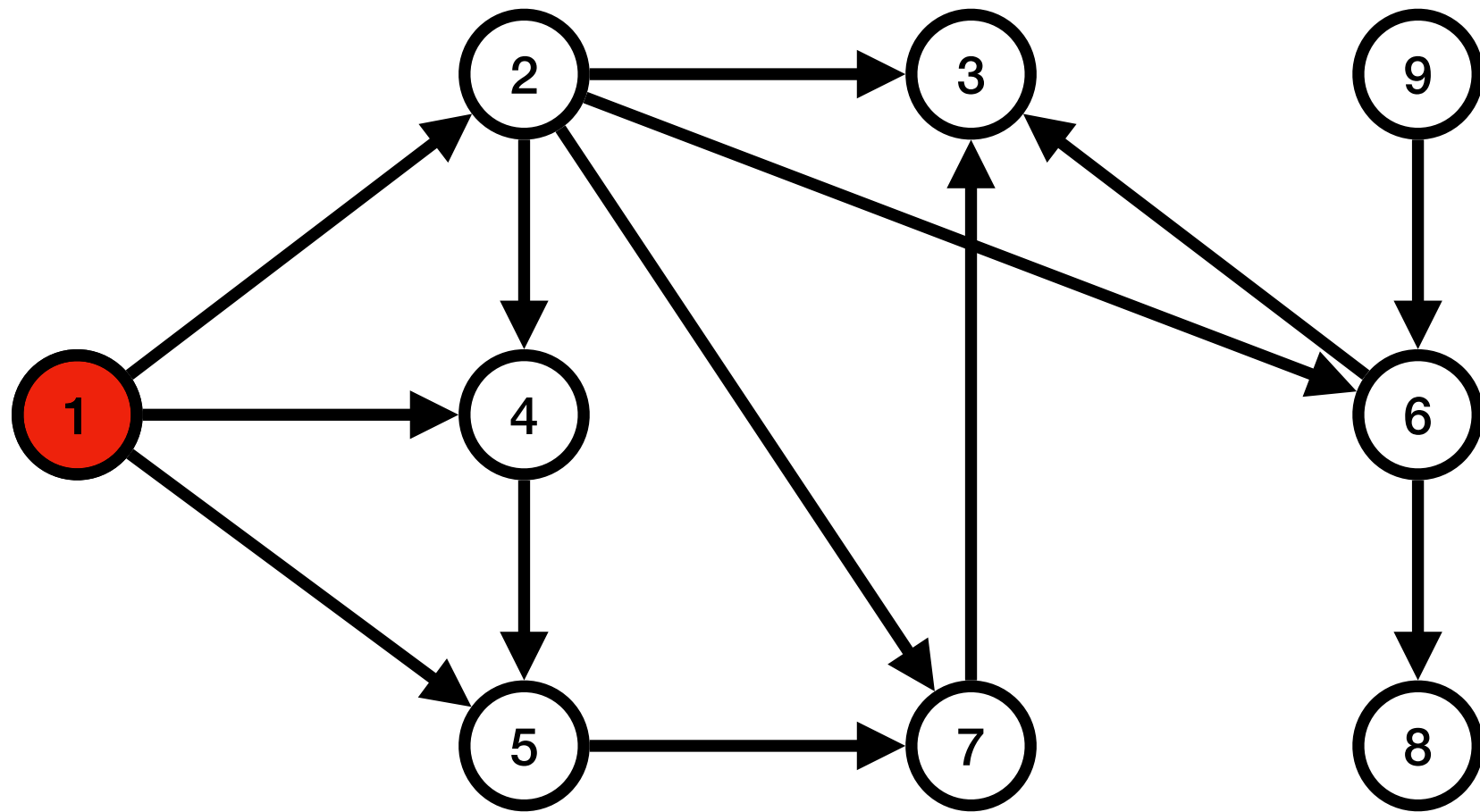
```

Un autre exemple



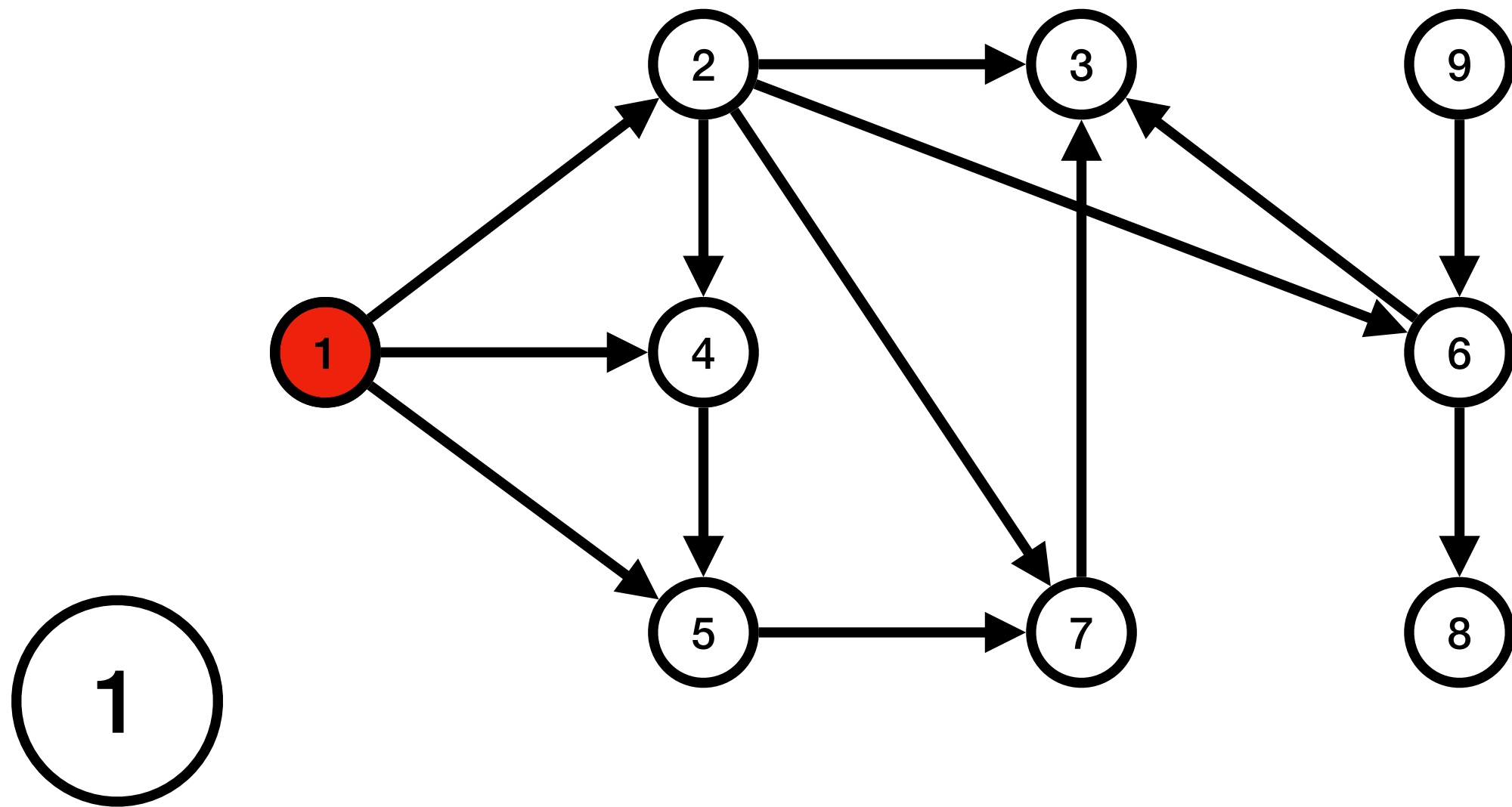
File →

Un autre exemple



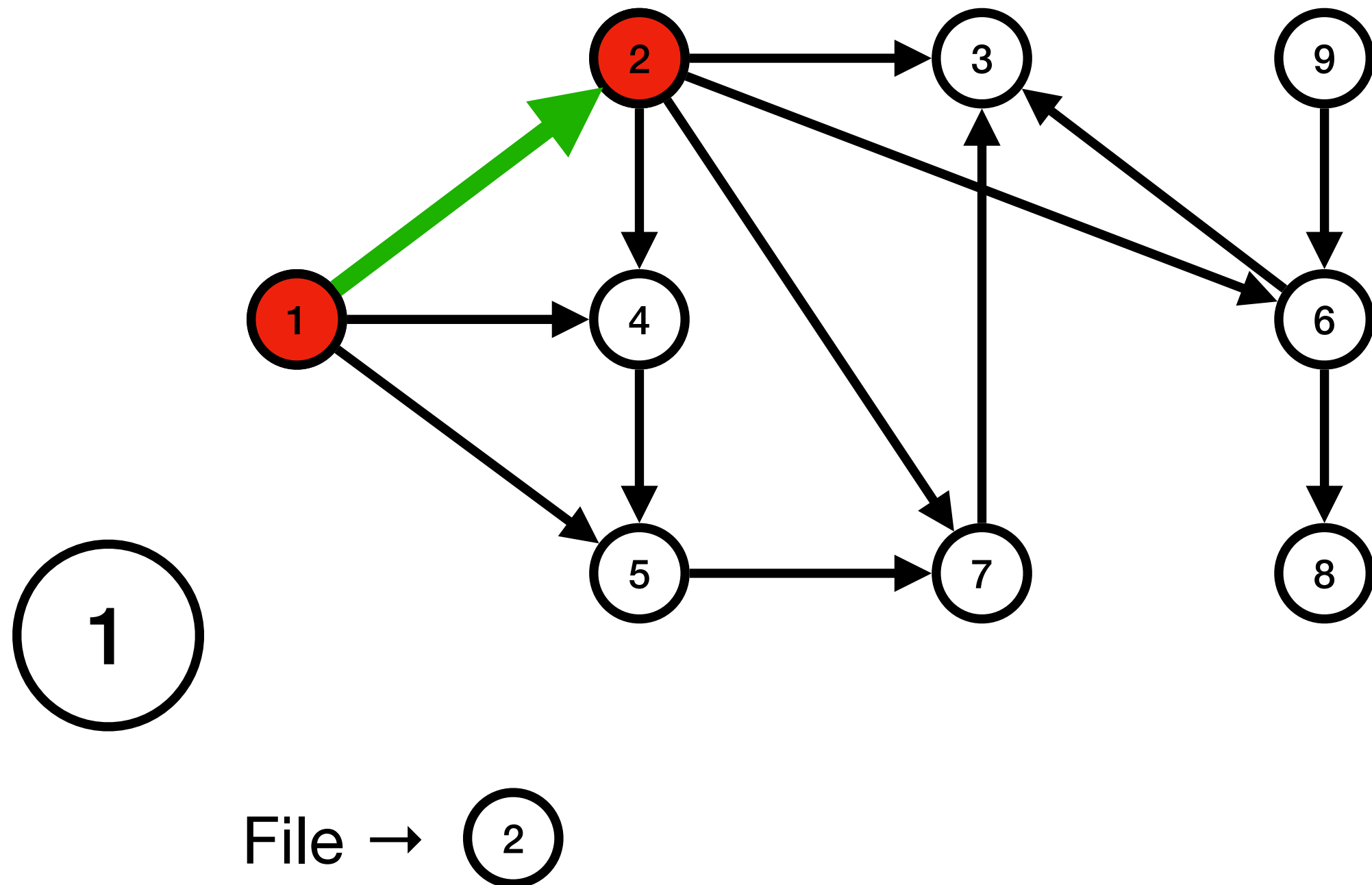
File → **1**

Un autre exemple

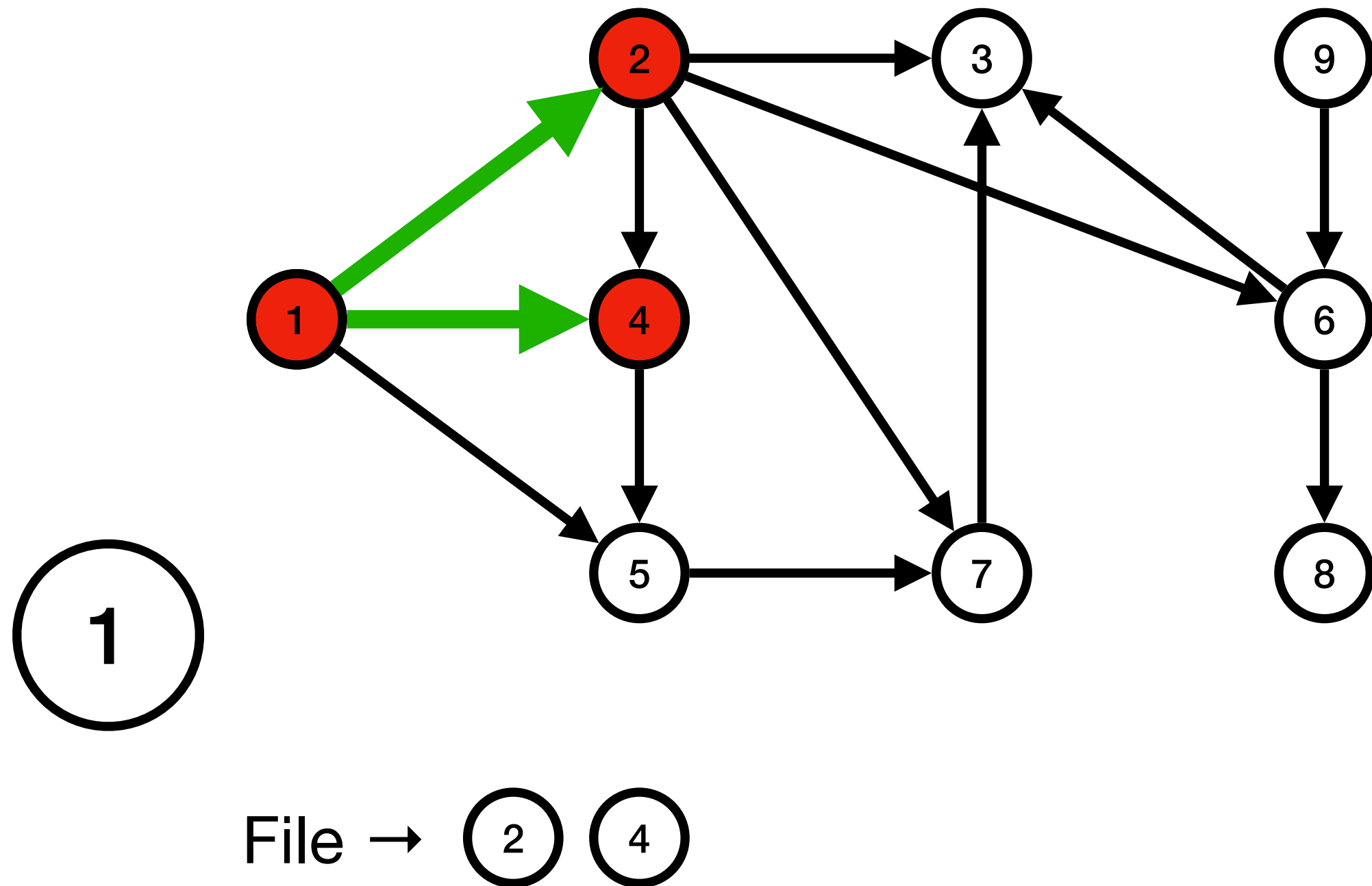


File →

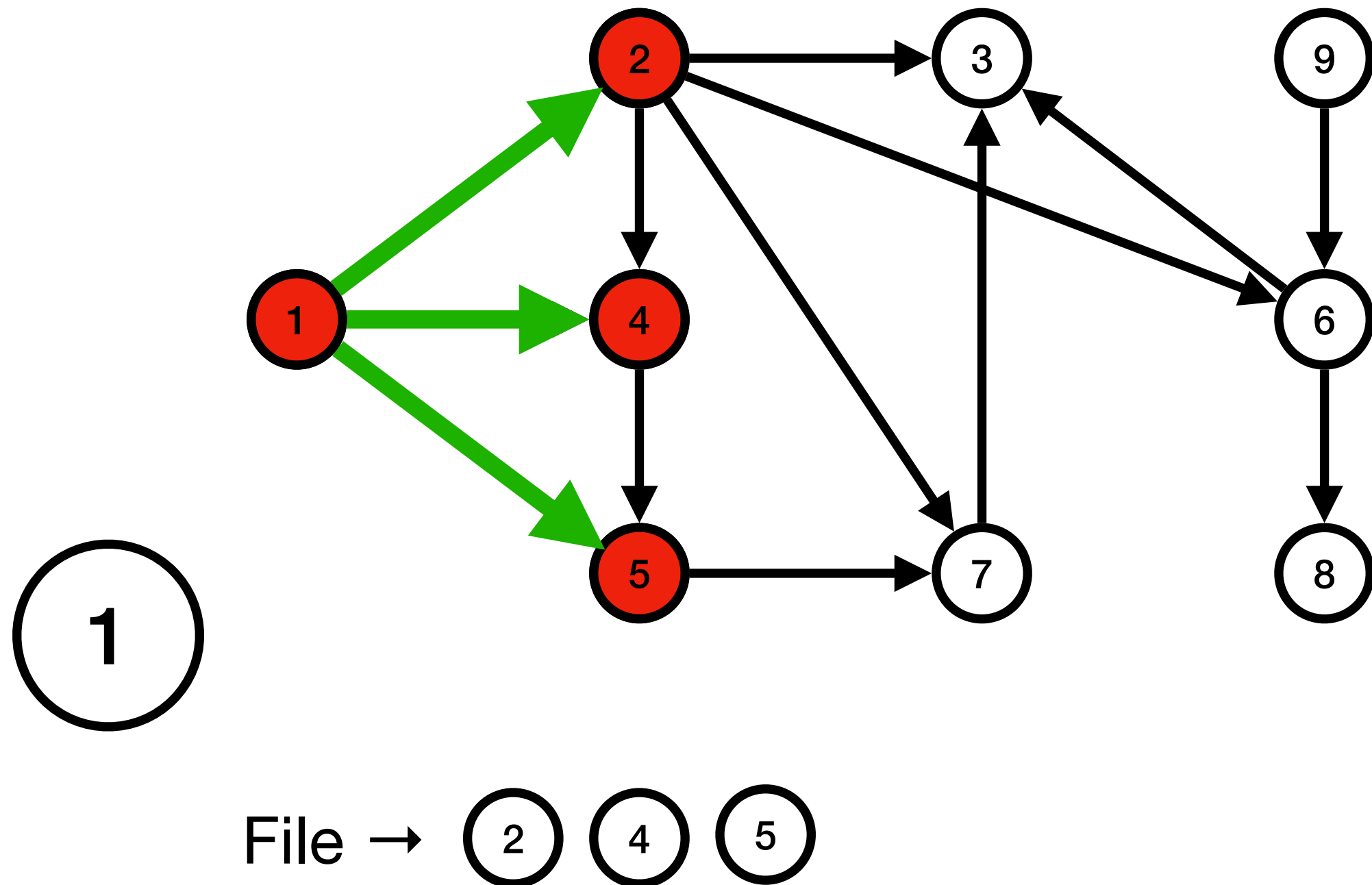
Un autre exemple



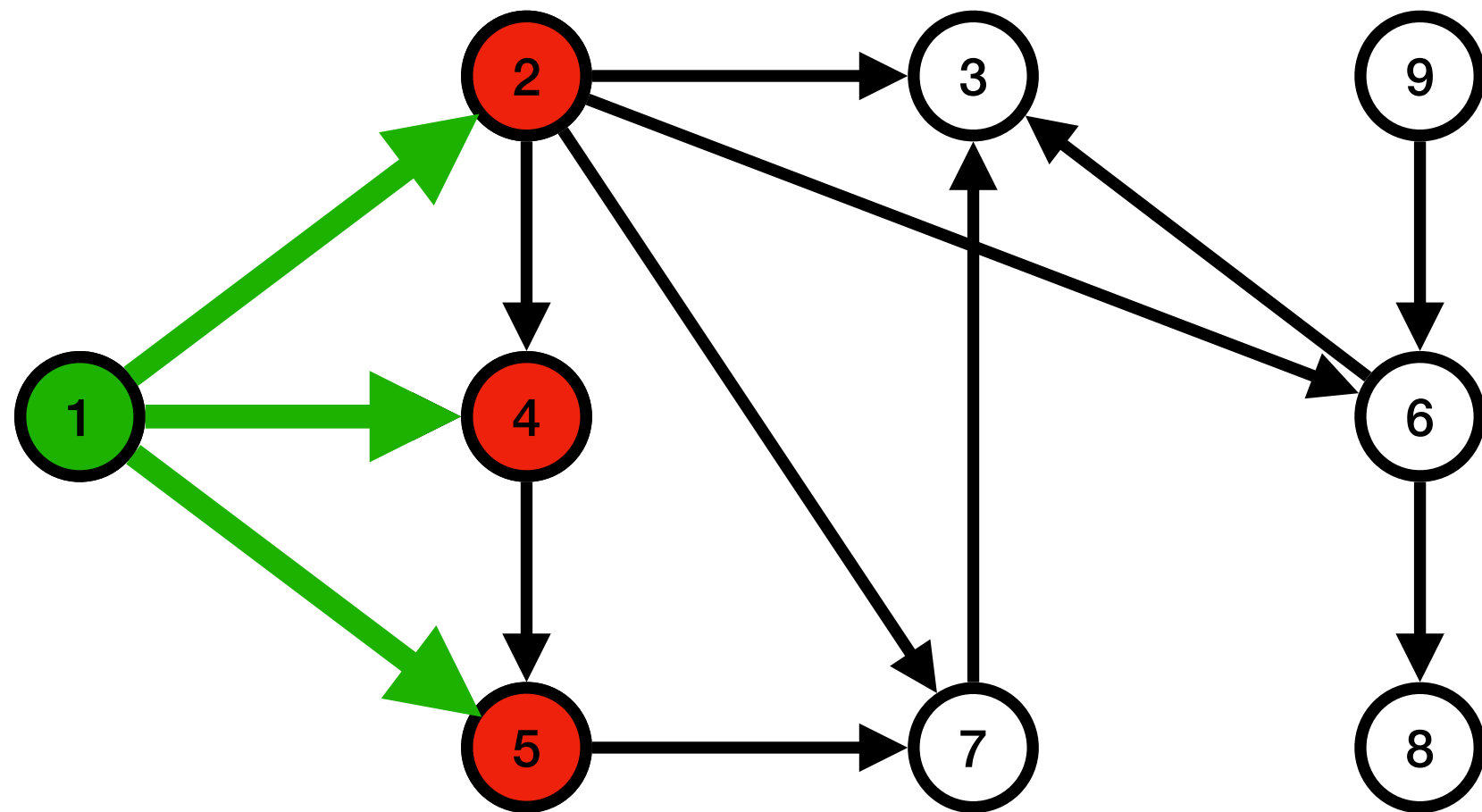
Un autre exemple



Un autre exemple

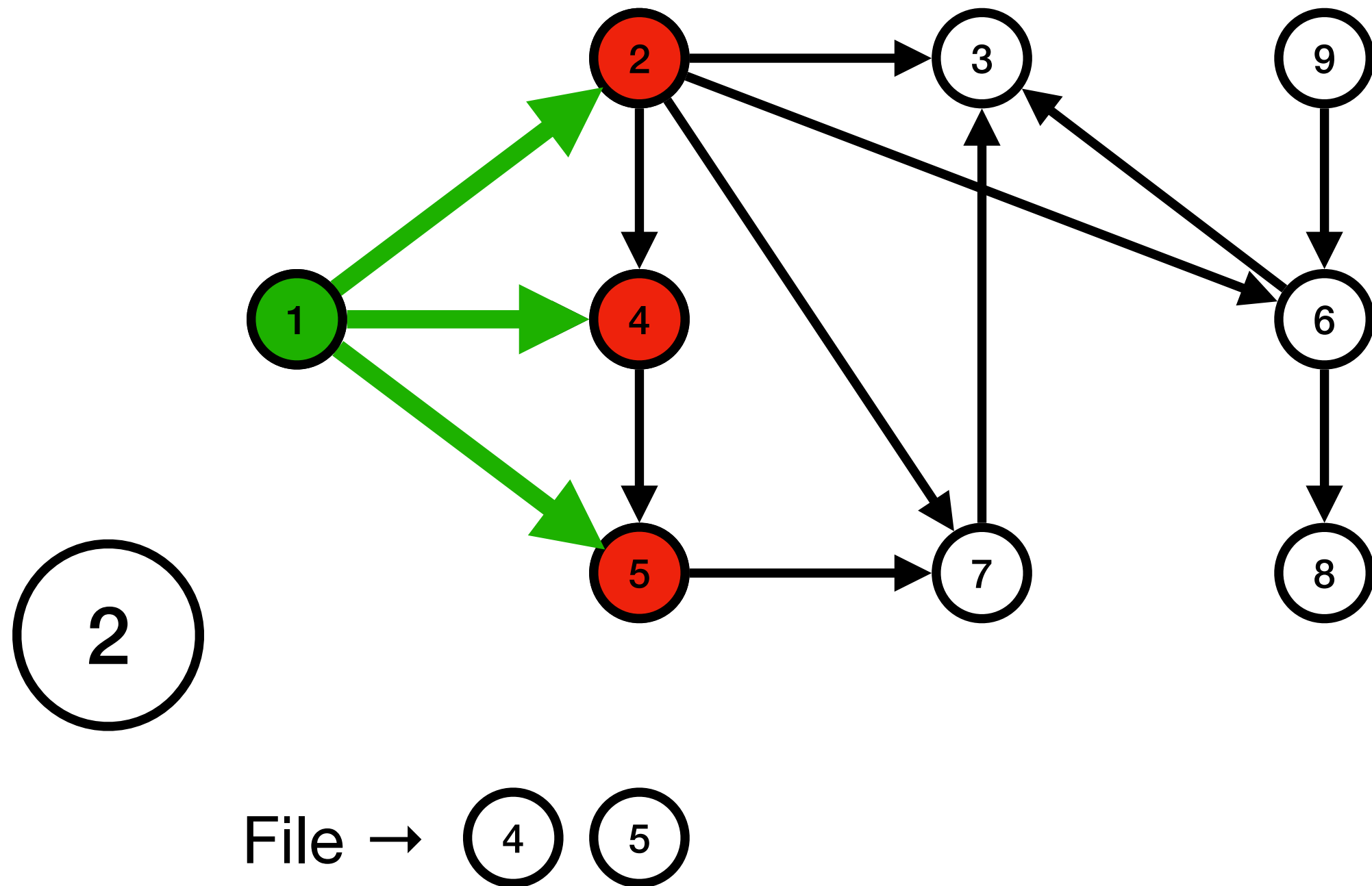


Un autre exemple

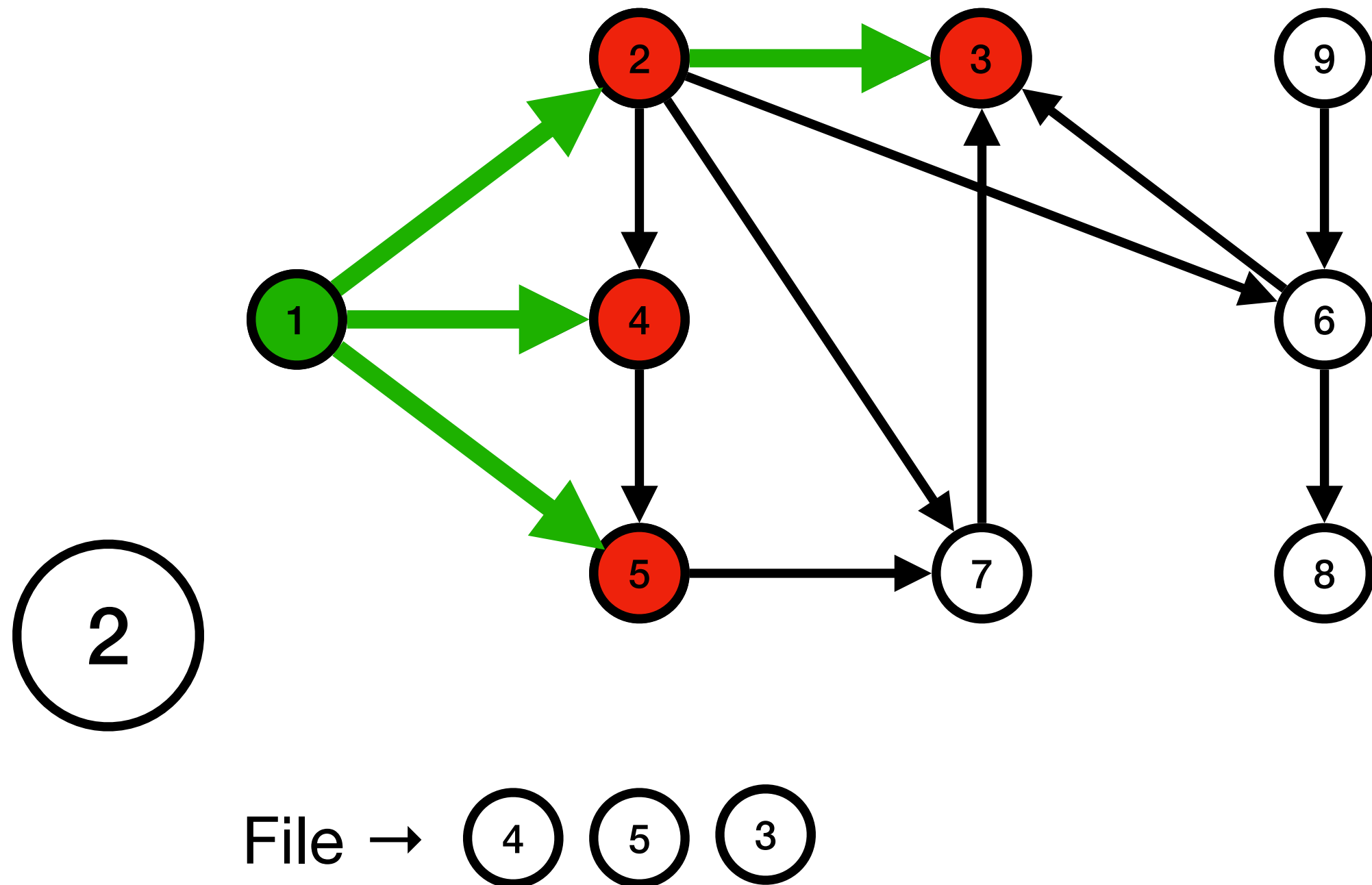


File → 2 4 5

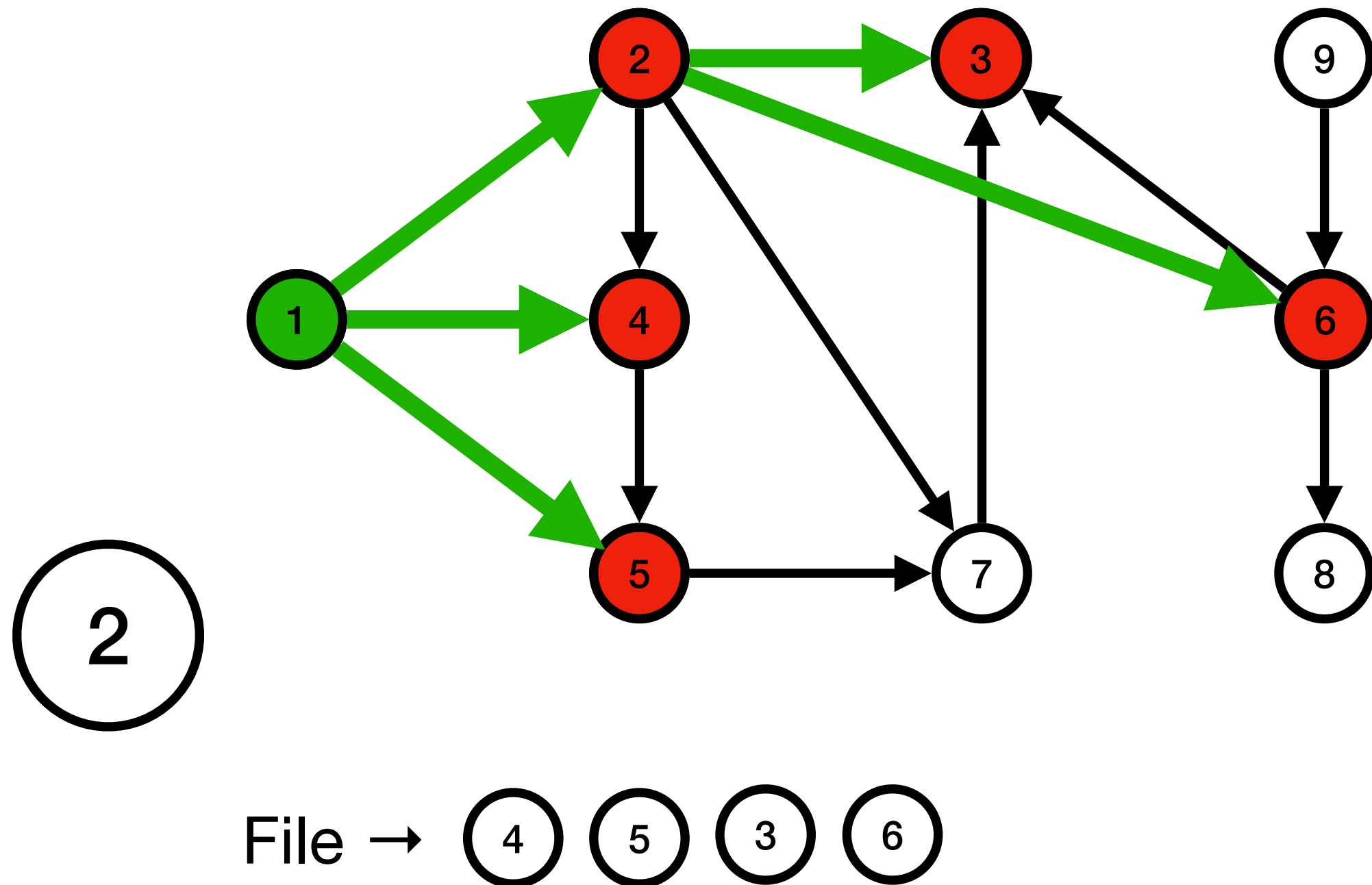
Un autre exemple



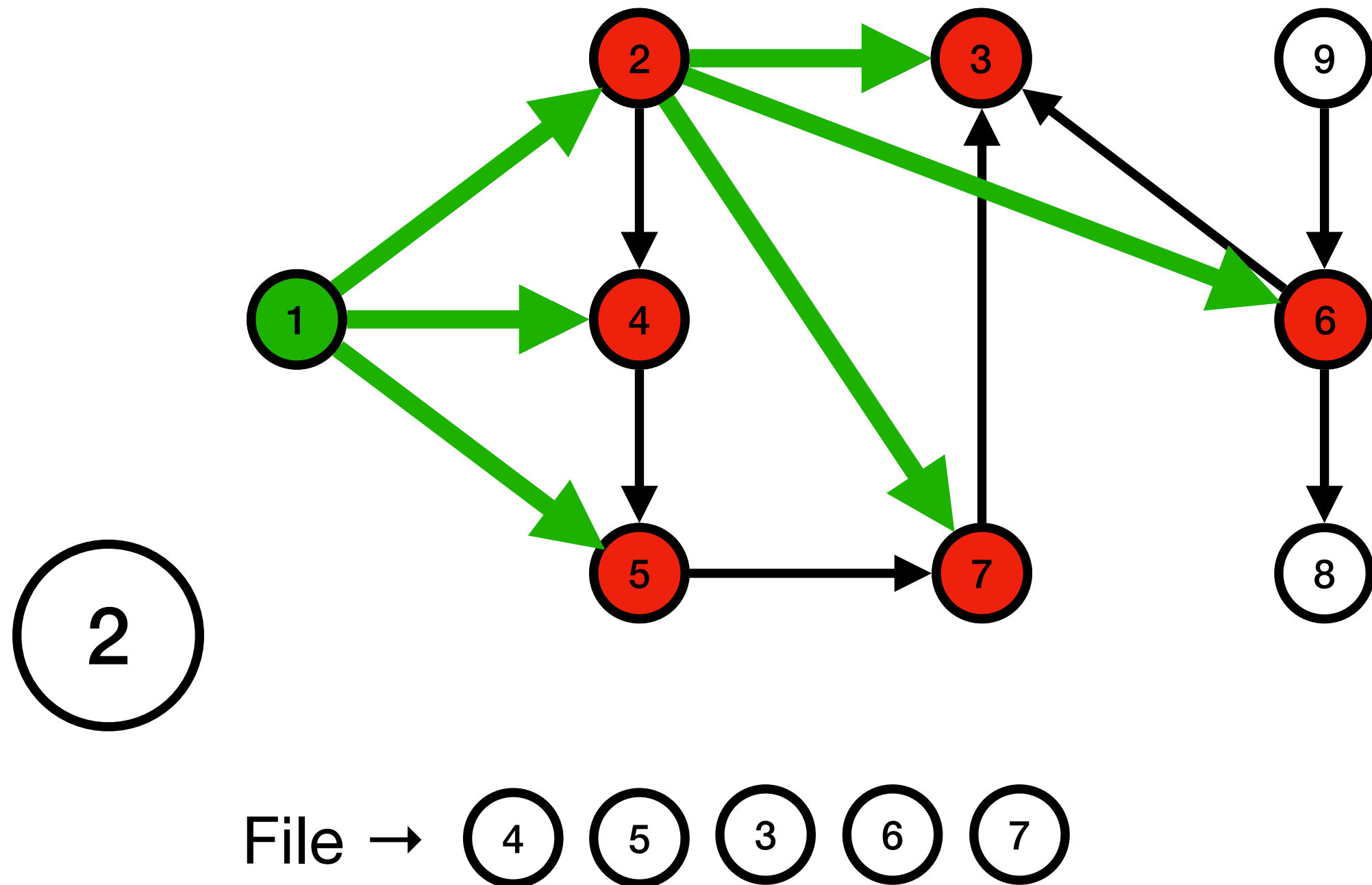
Un autre exemple



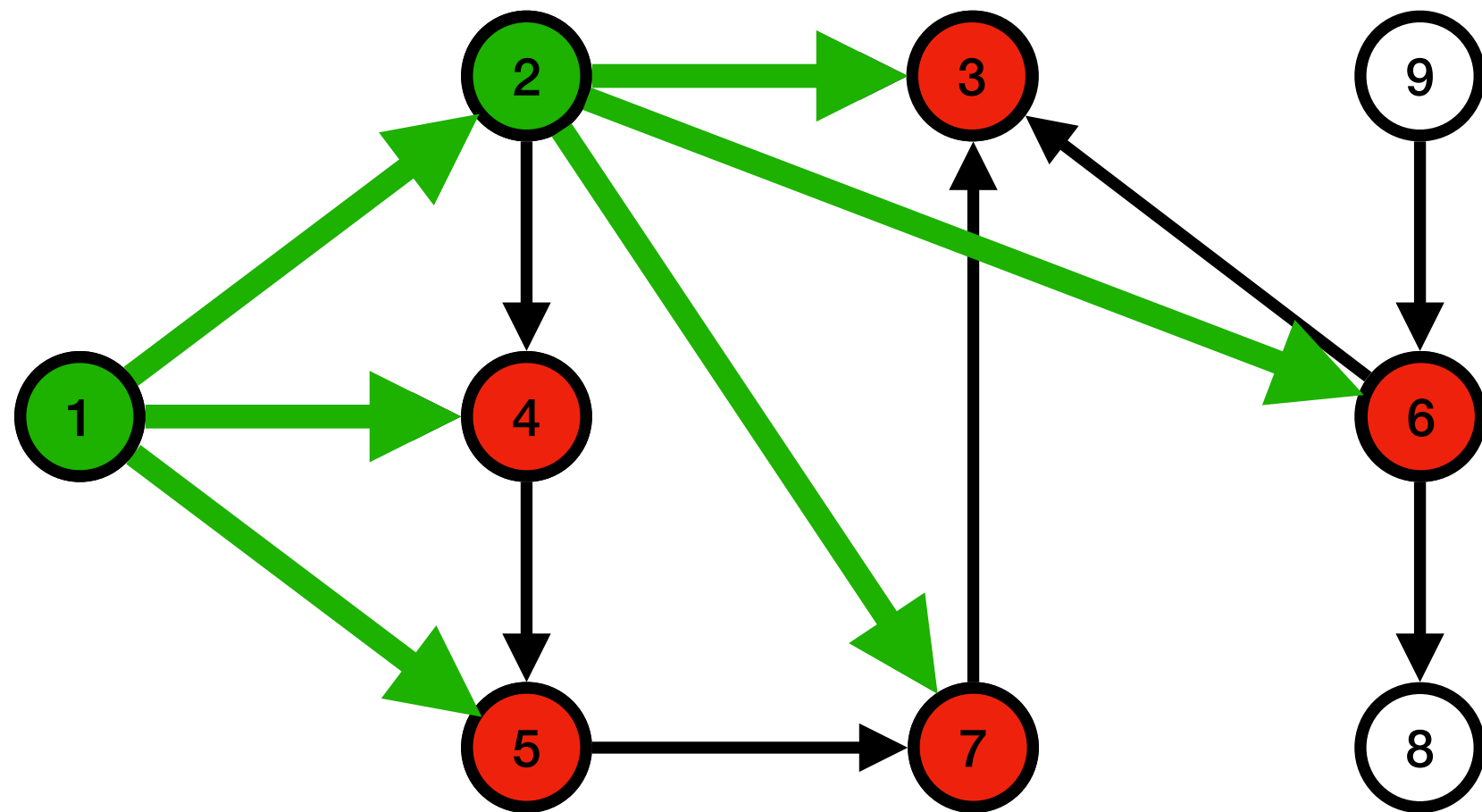
Un autre exemple



Un autre exemple

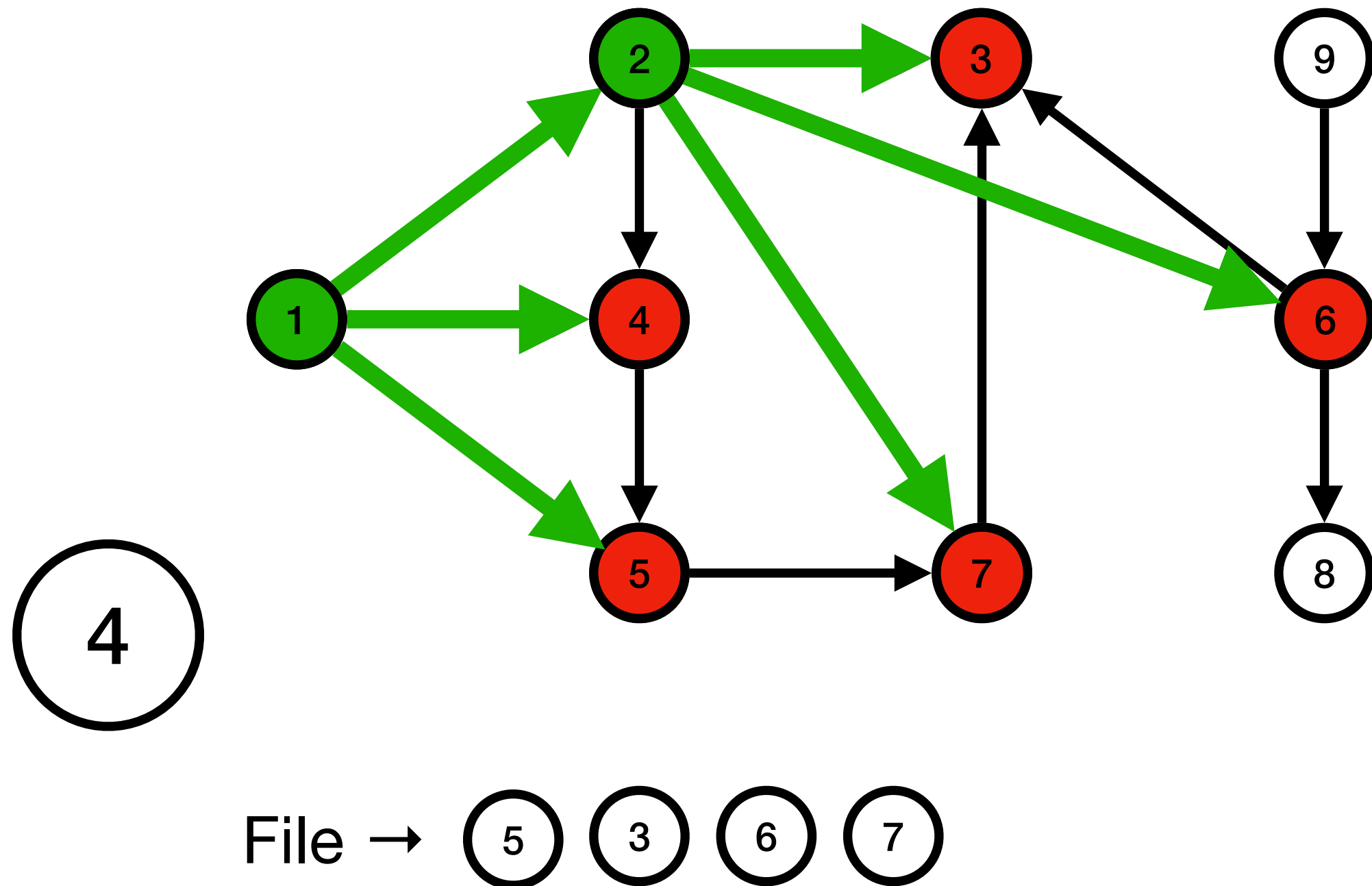


Un autre exemple

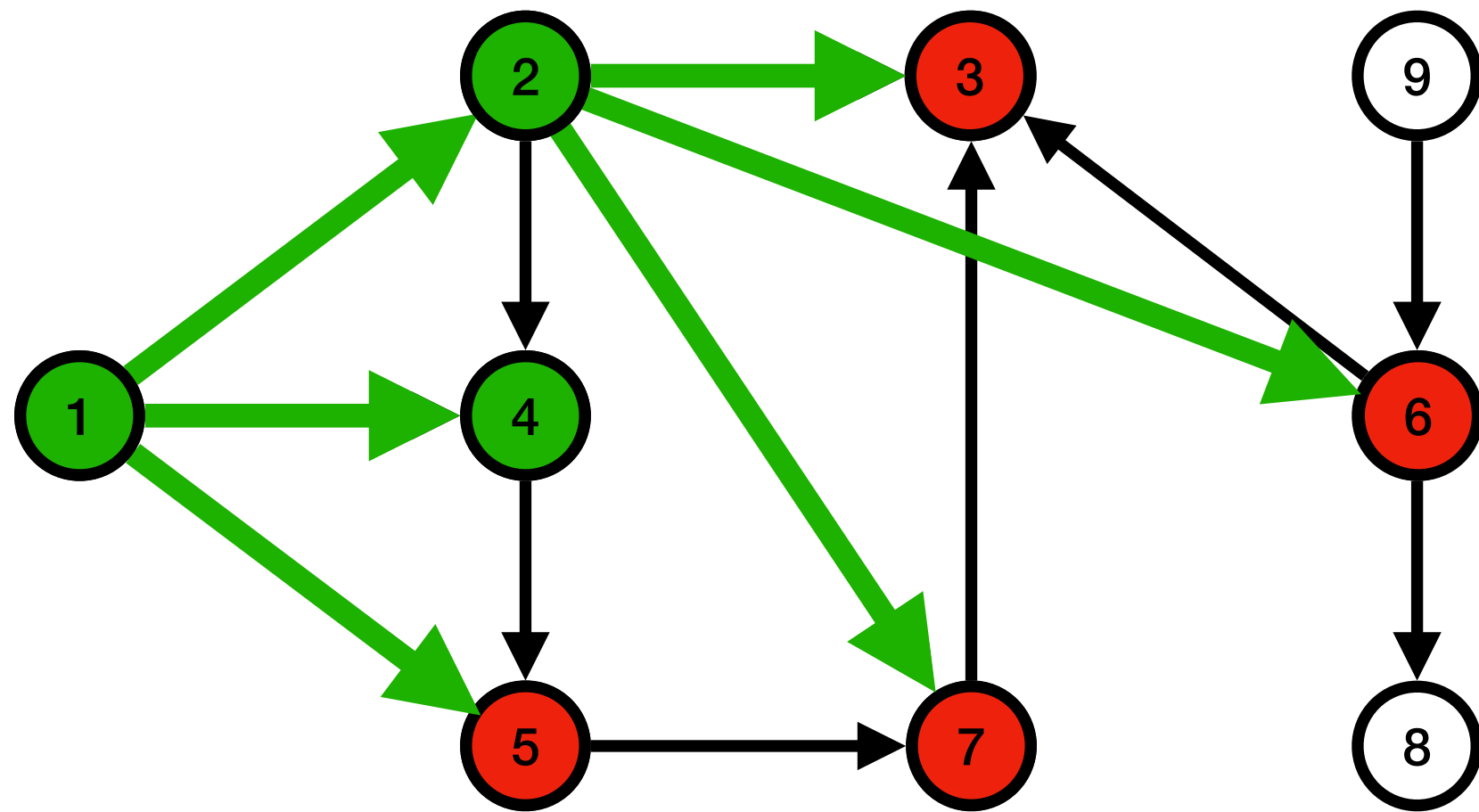


File → 4 5 3 6 7

Un autre exemple

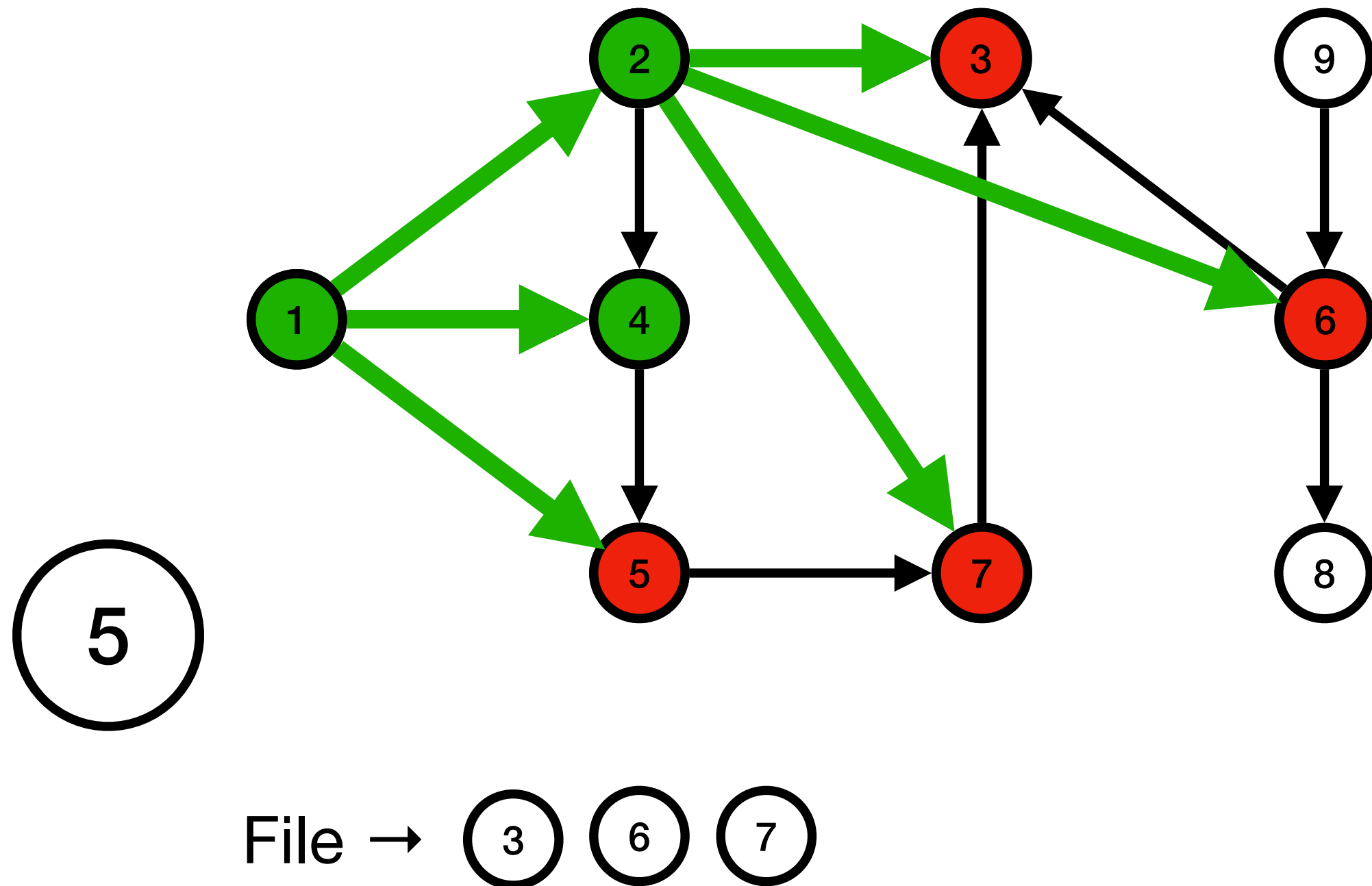


Un autre exemple

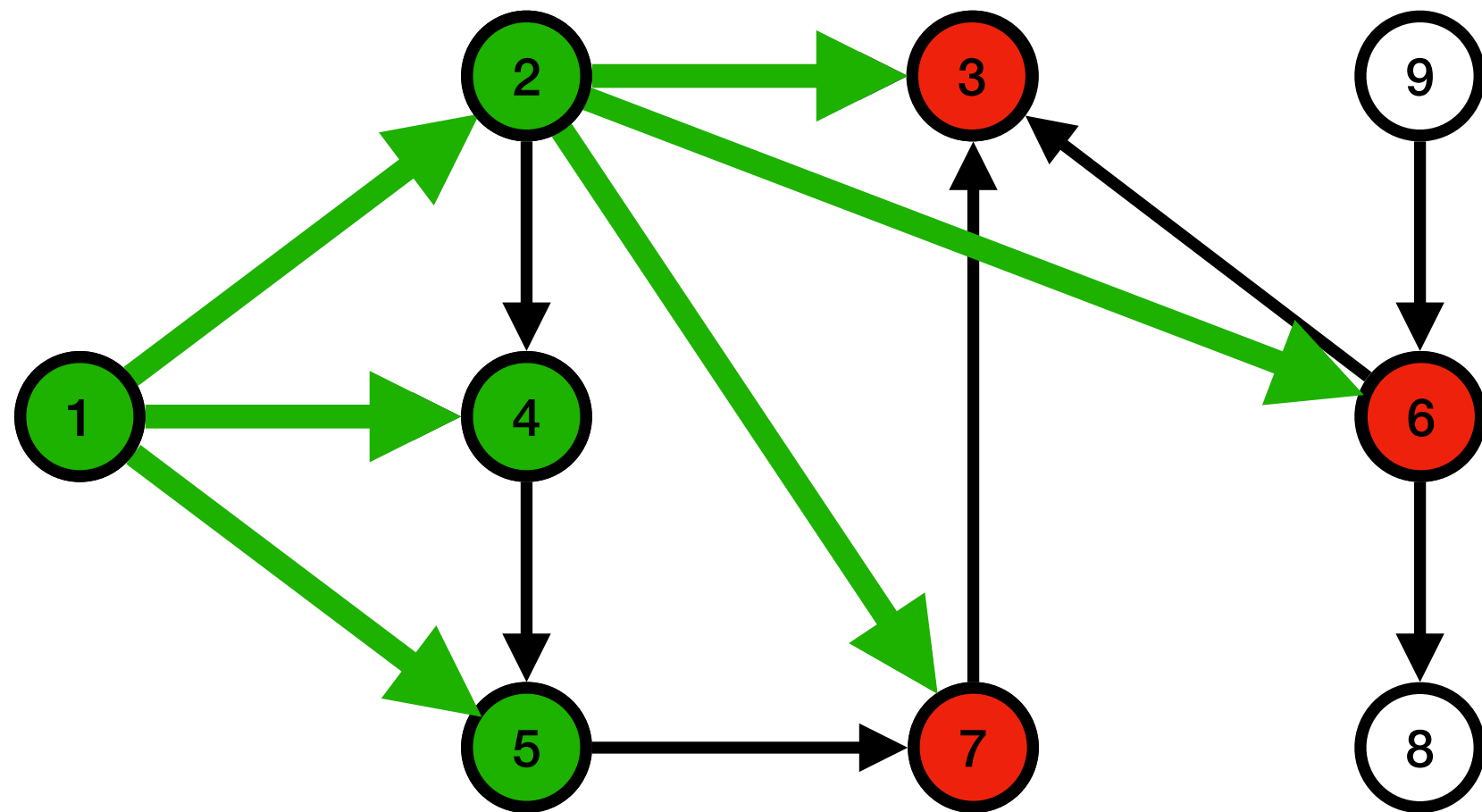


File → 5 3 6 7

Un autre exemple

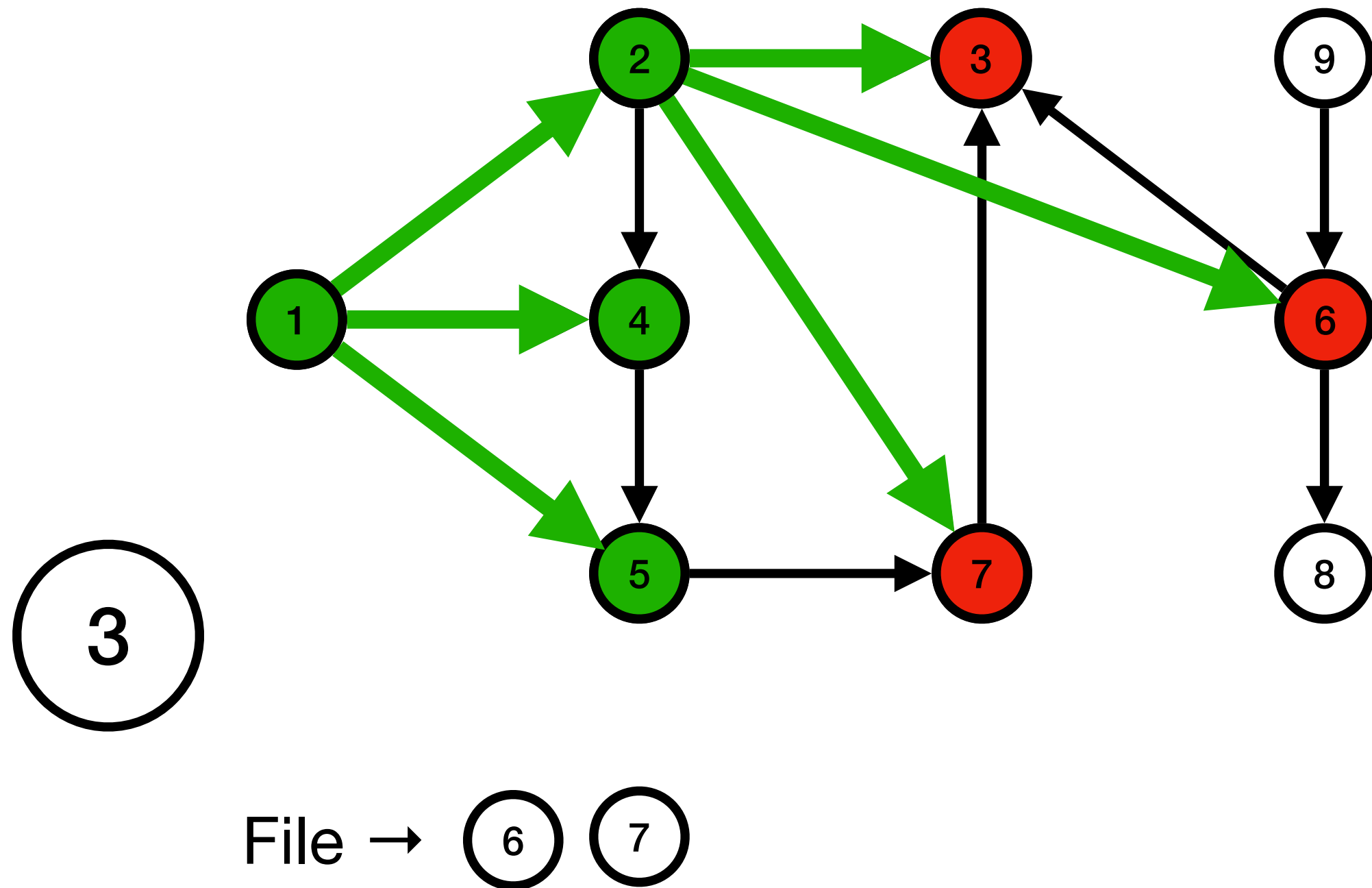


Un autre exemple

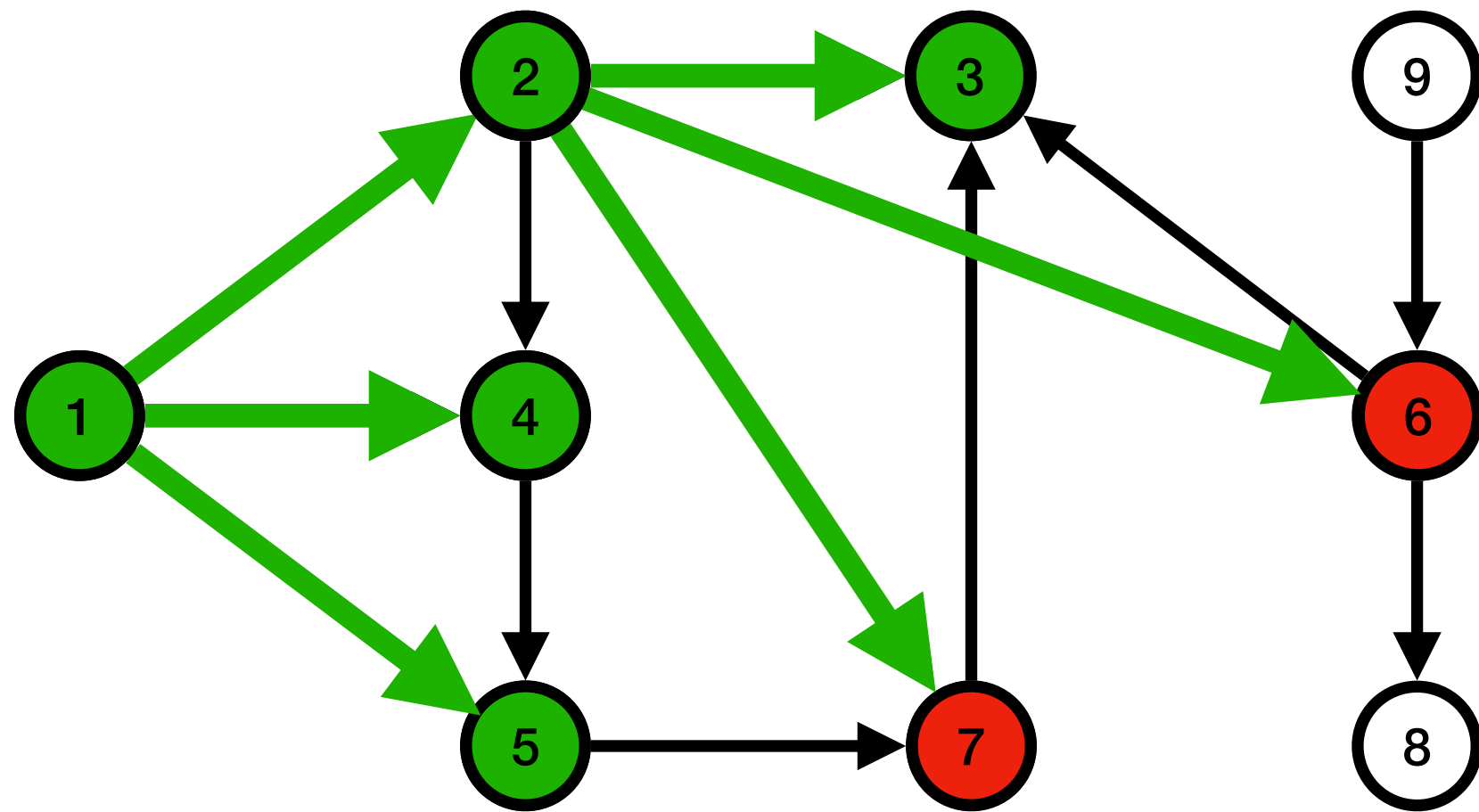




File → 3 6 7

Un autre exemple

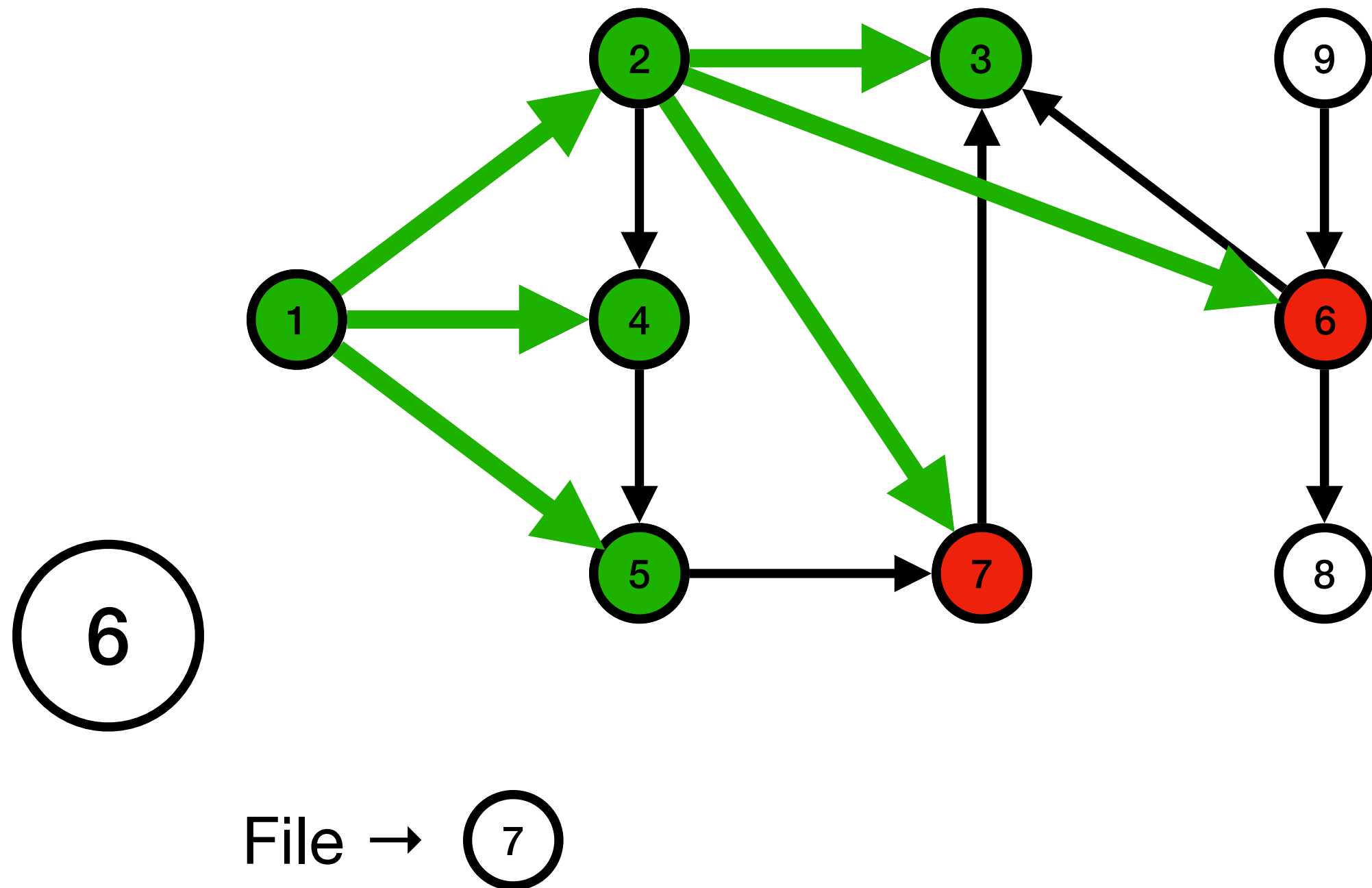


Un autre exemple

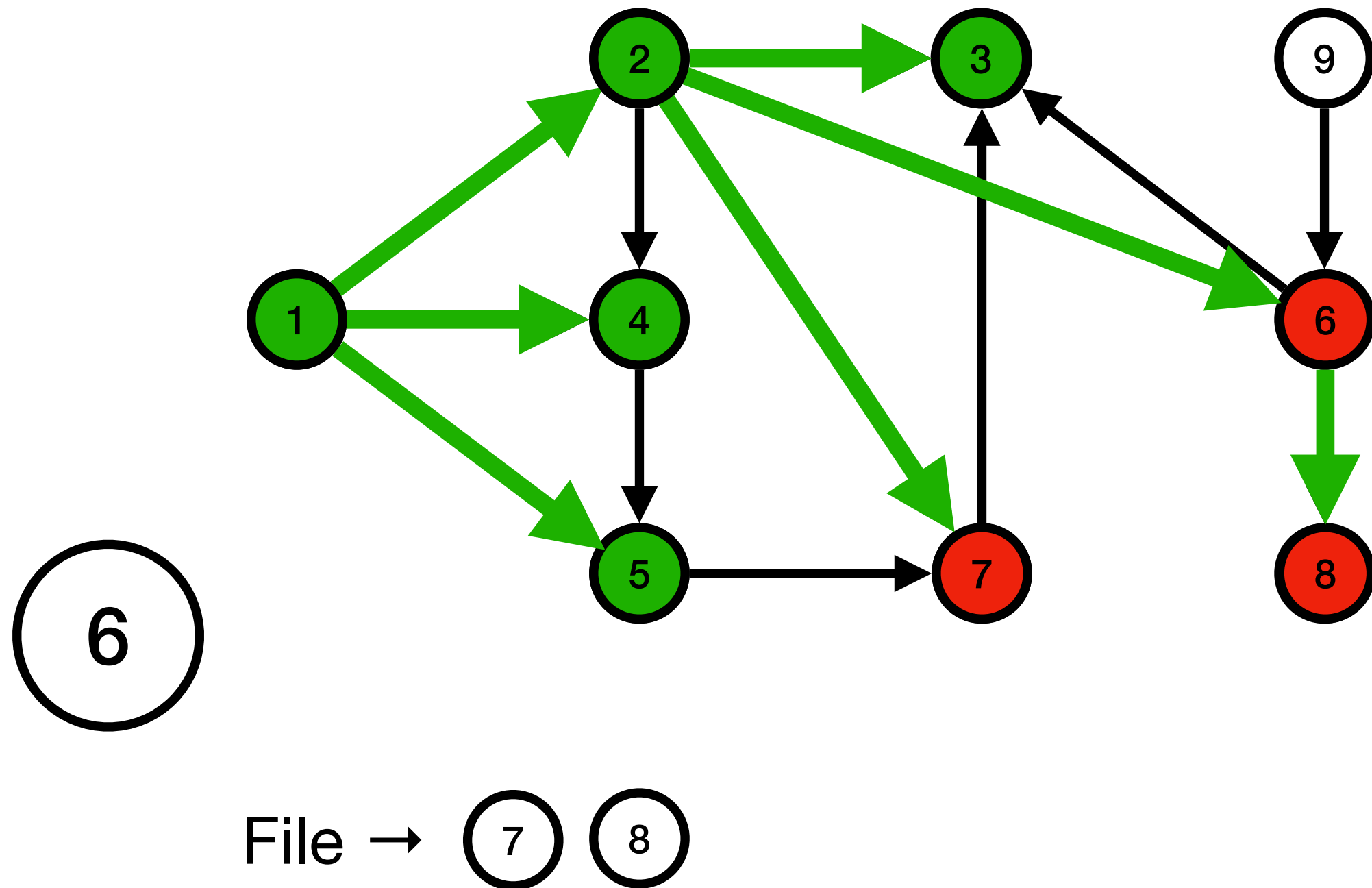


File →  

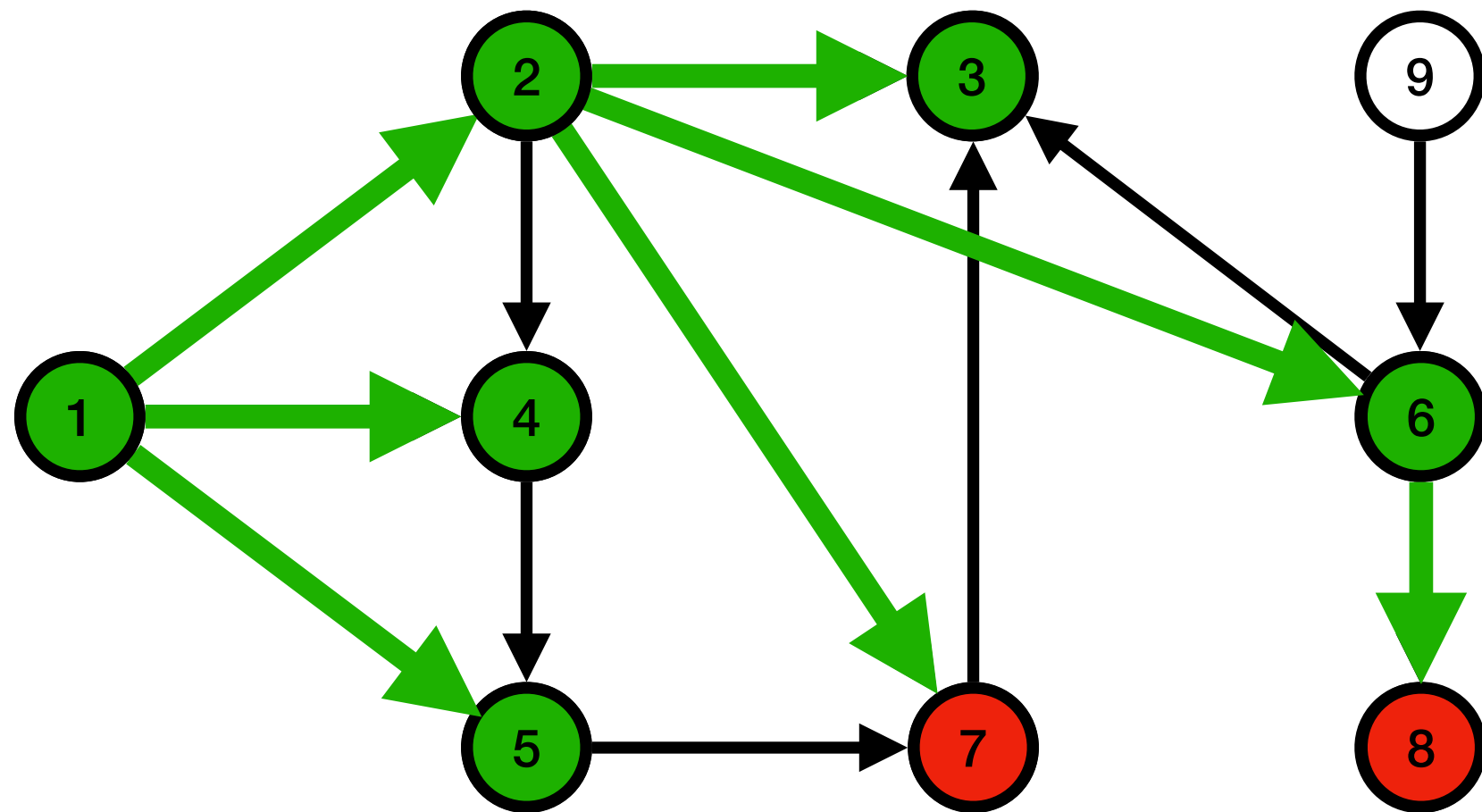
Un autre exemple



Un autre exemple

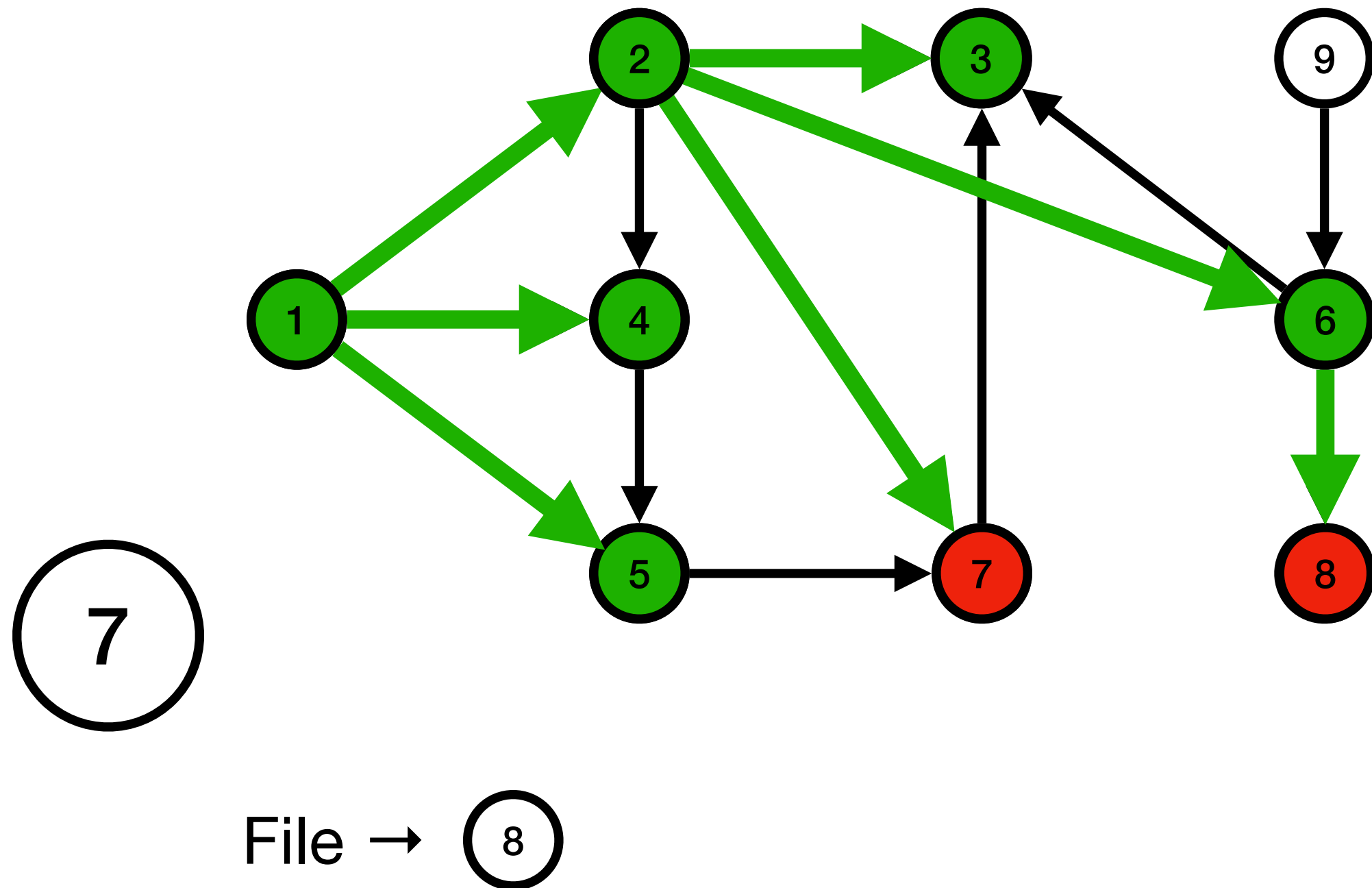


Un autre exemple

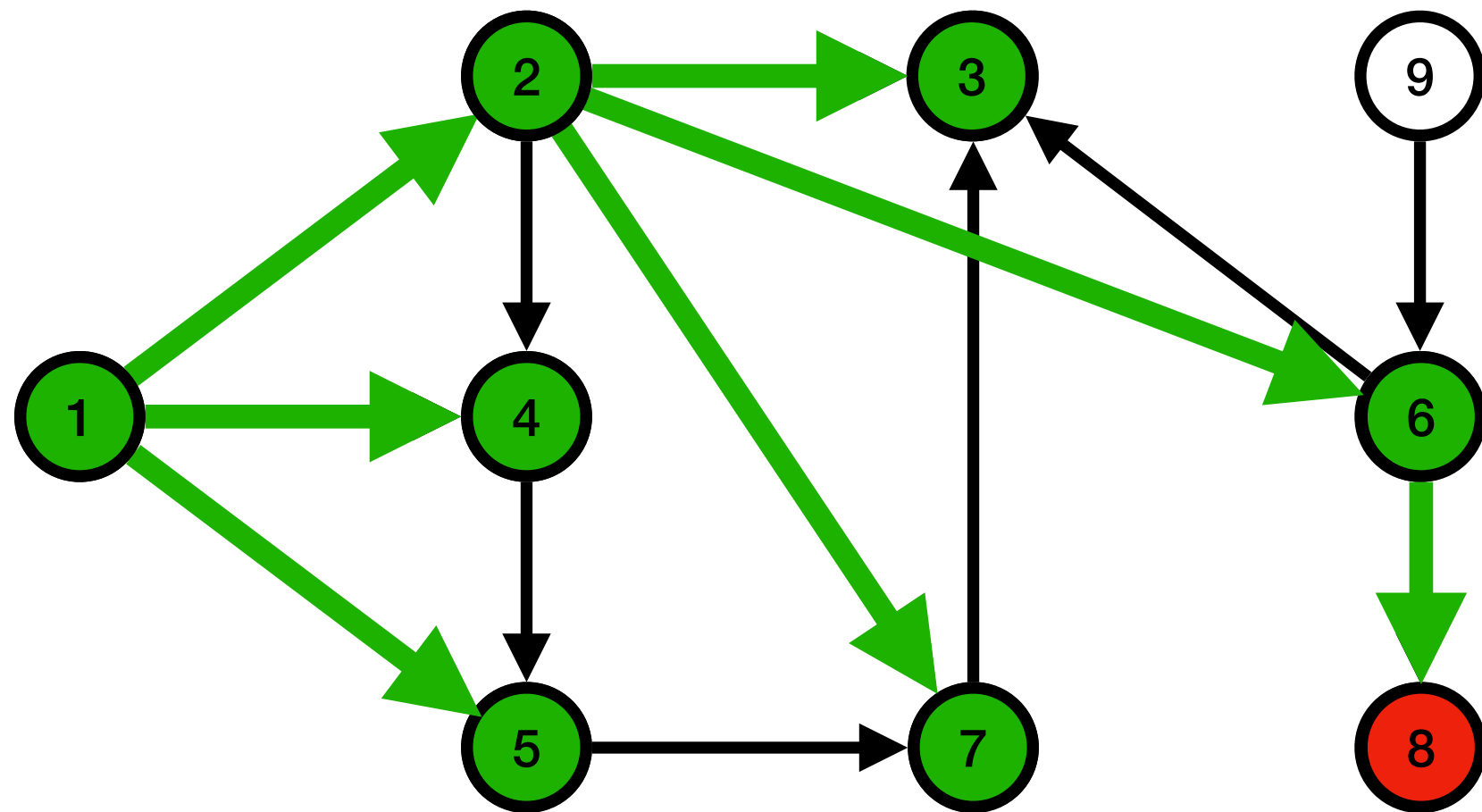


File → (7) (8)

Un autre exemple

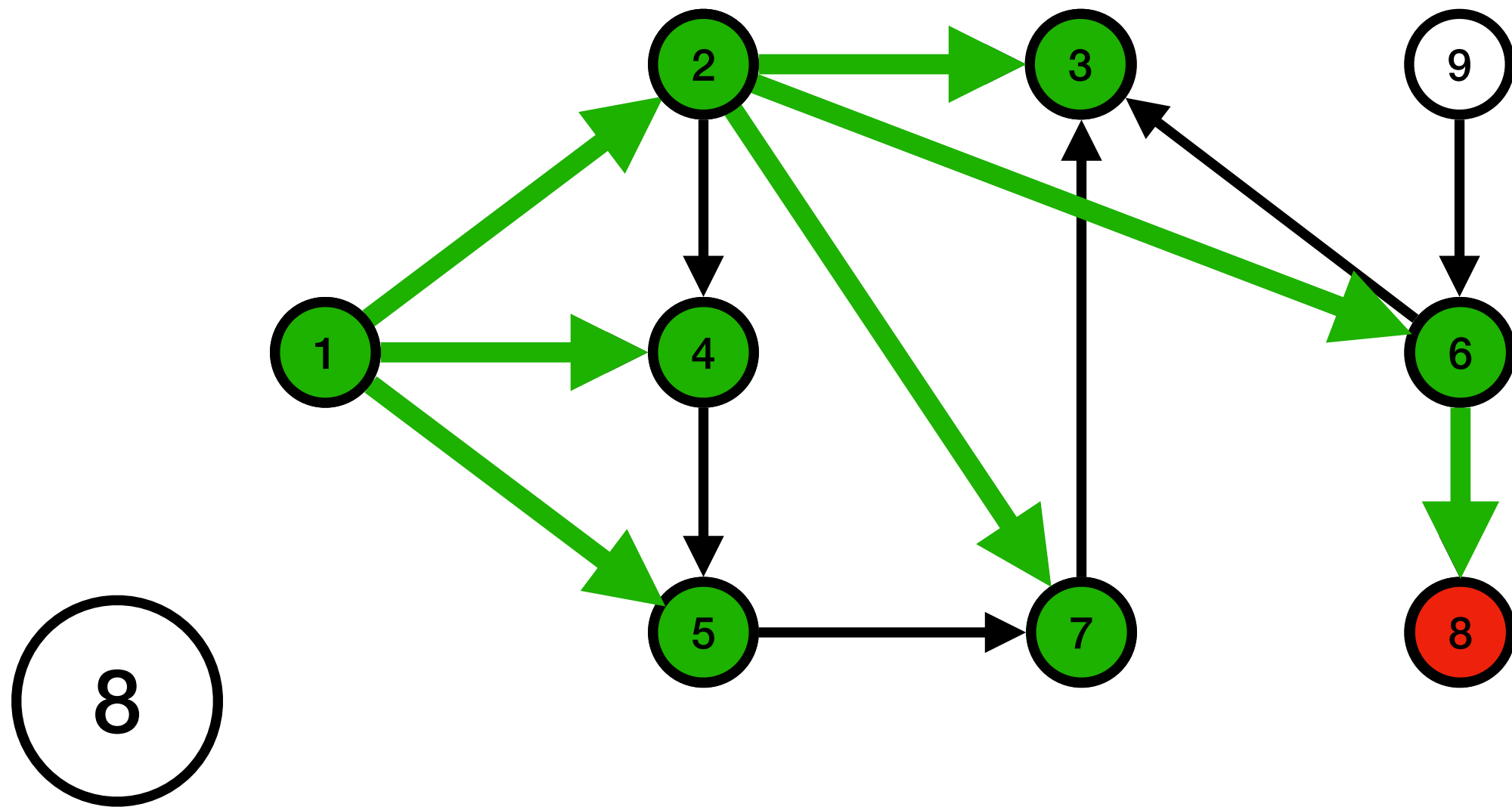


Un autre exemple



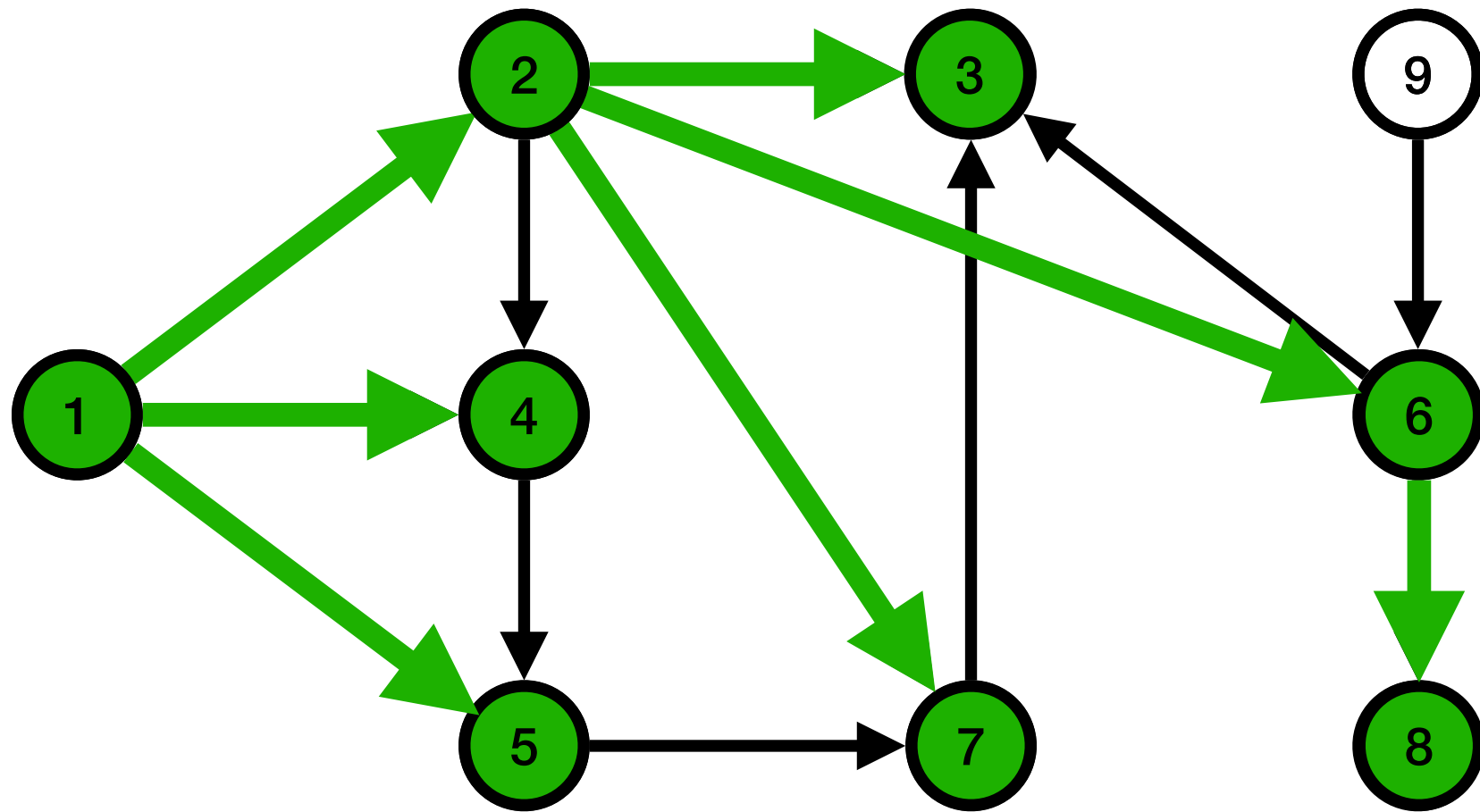
File → 8

Un autre exemple



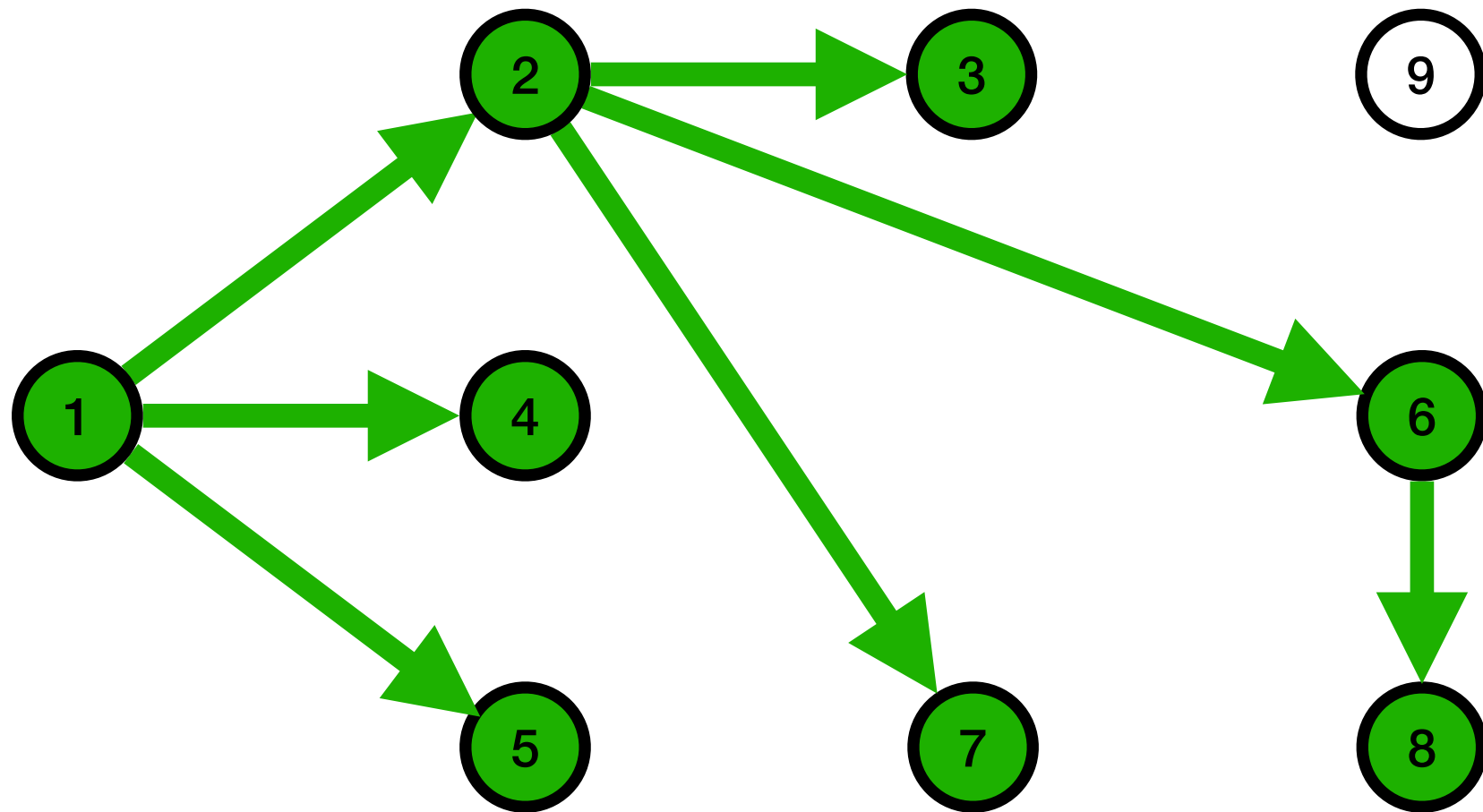
File →

Un autre exemple

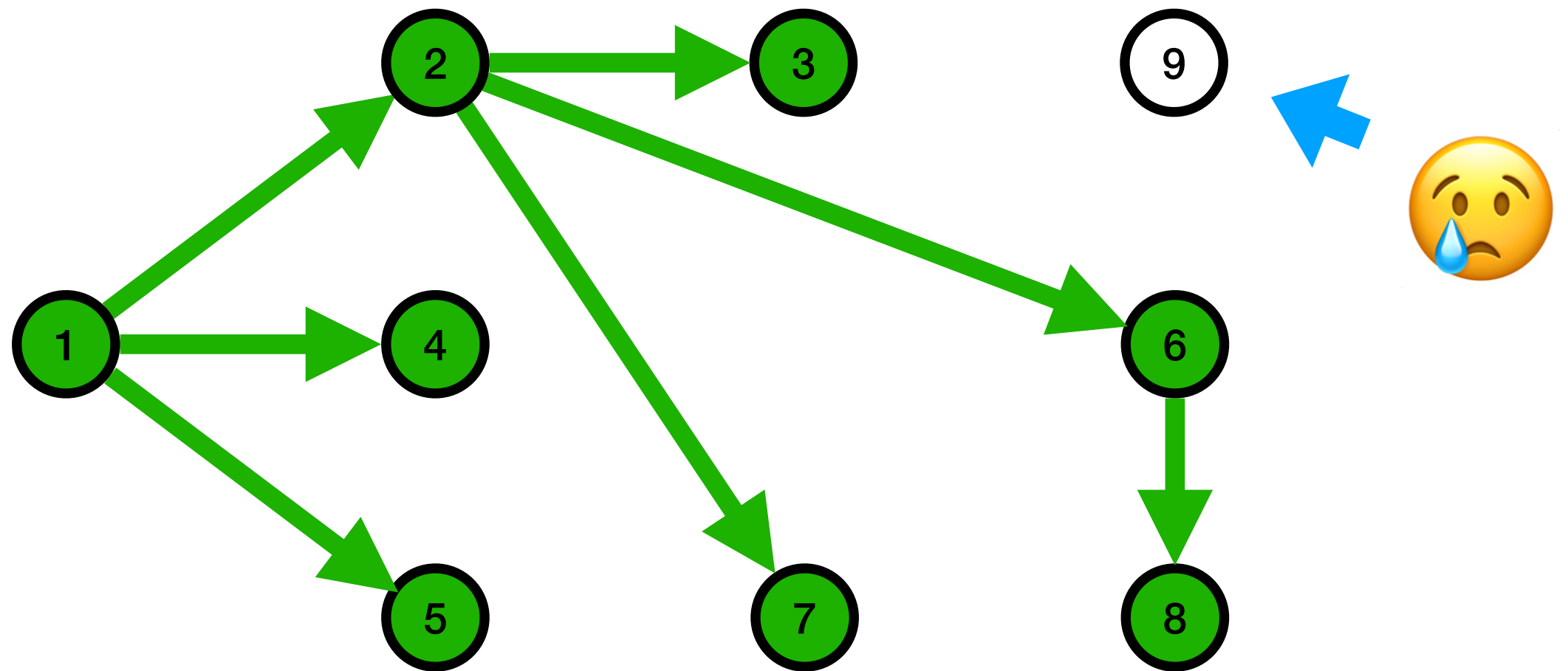


File →

Un autre exemple

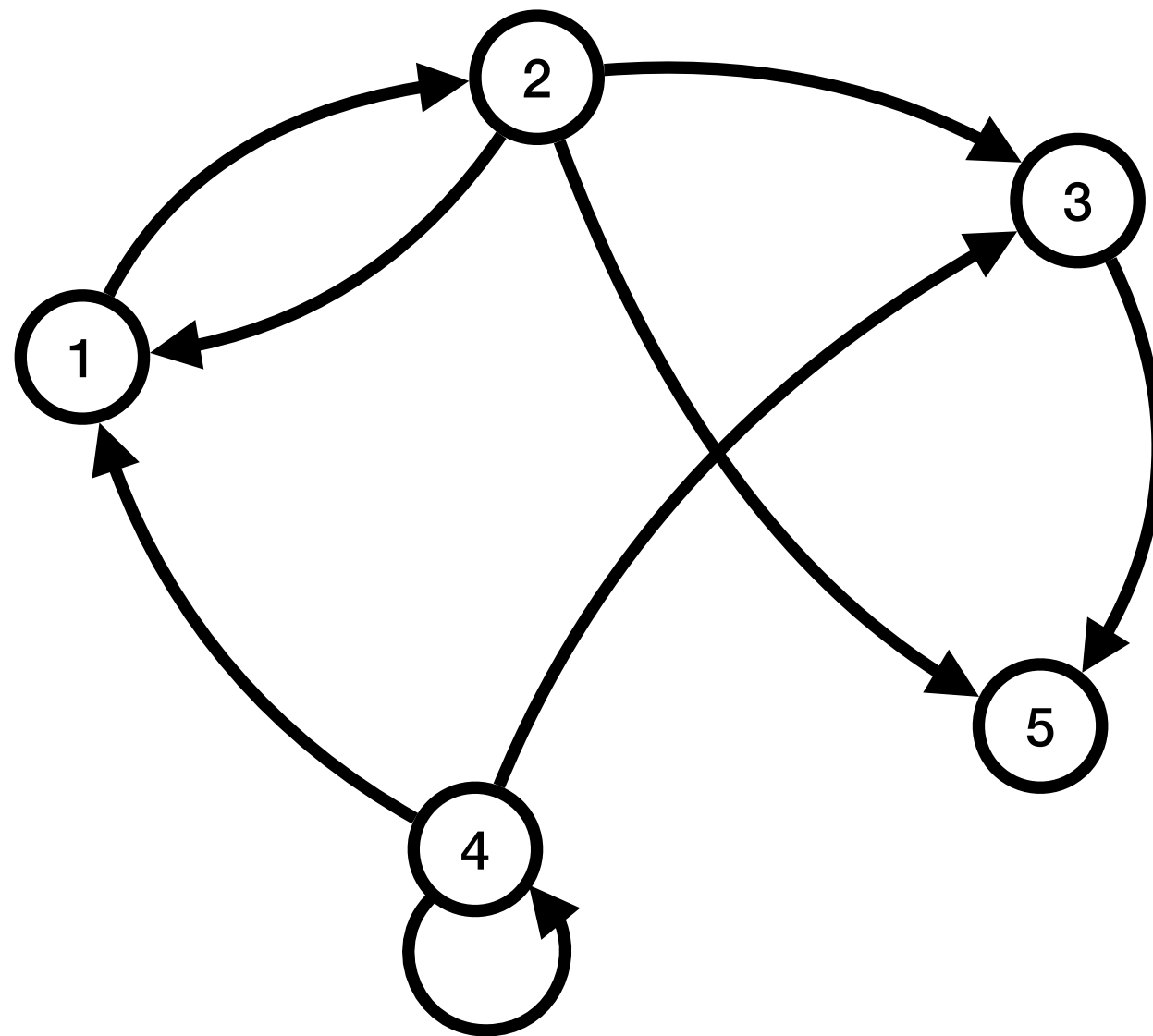


Un autre exemple

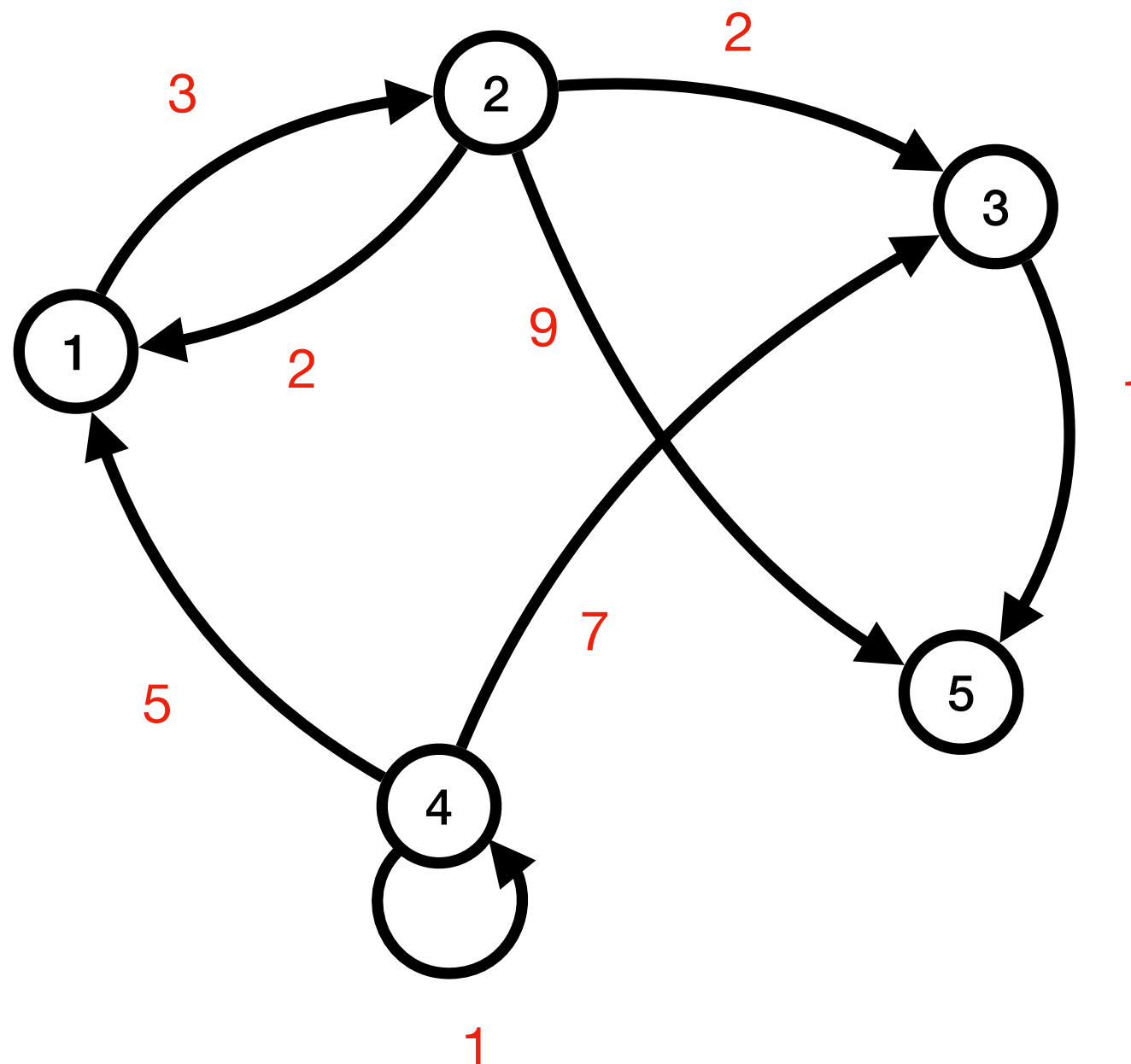


Le plus court chemin

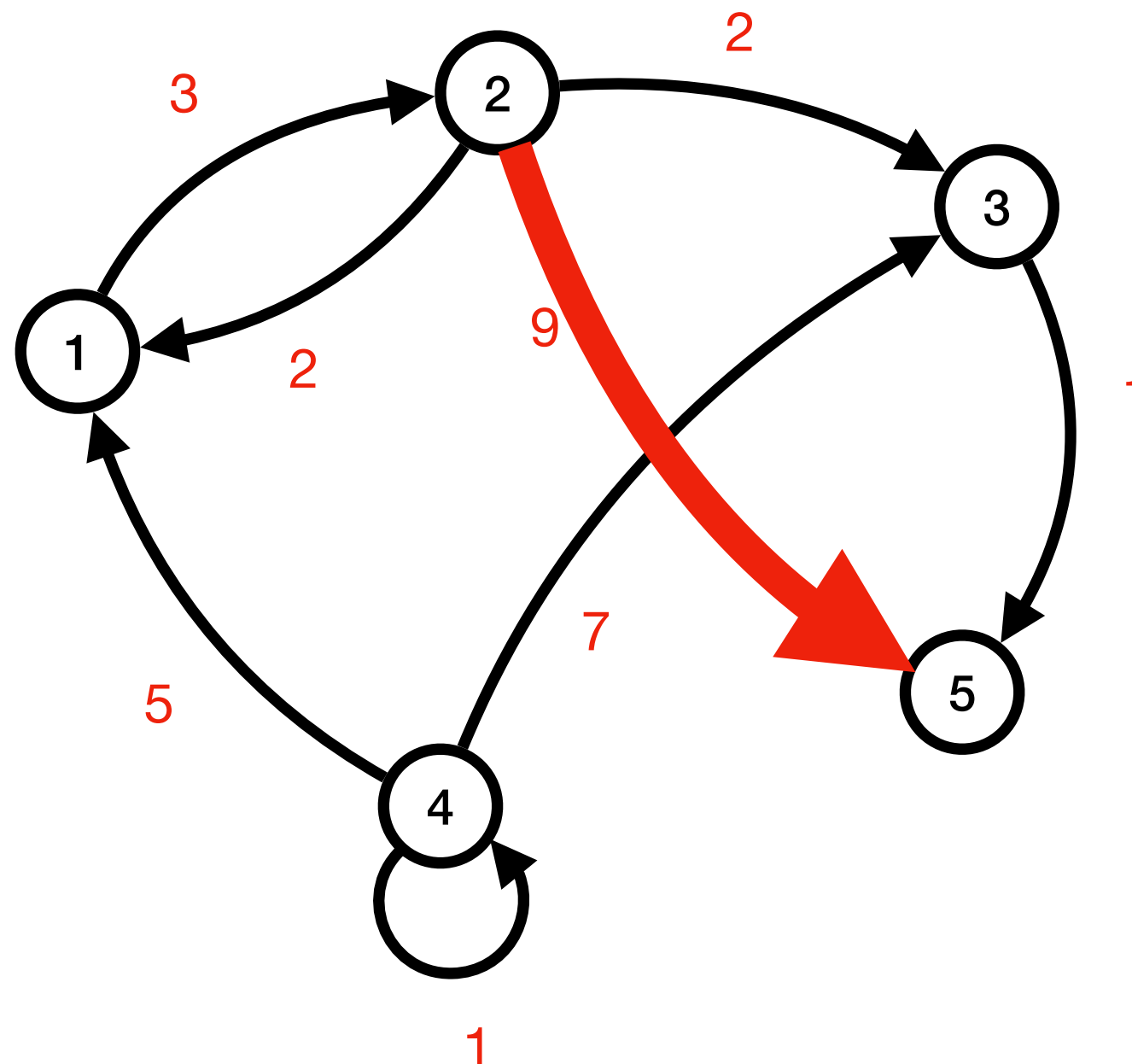
Graphes pondérés



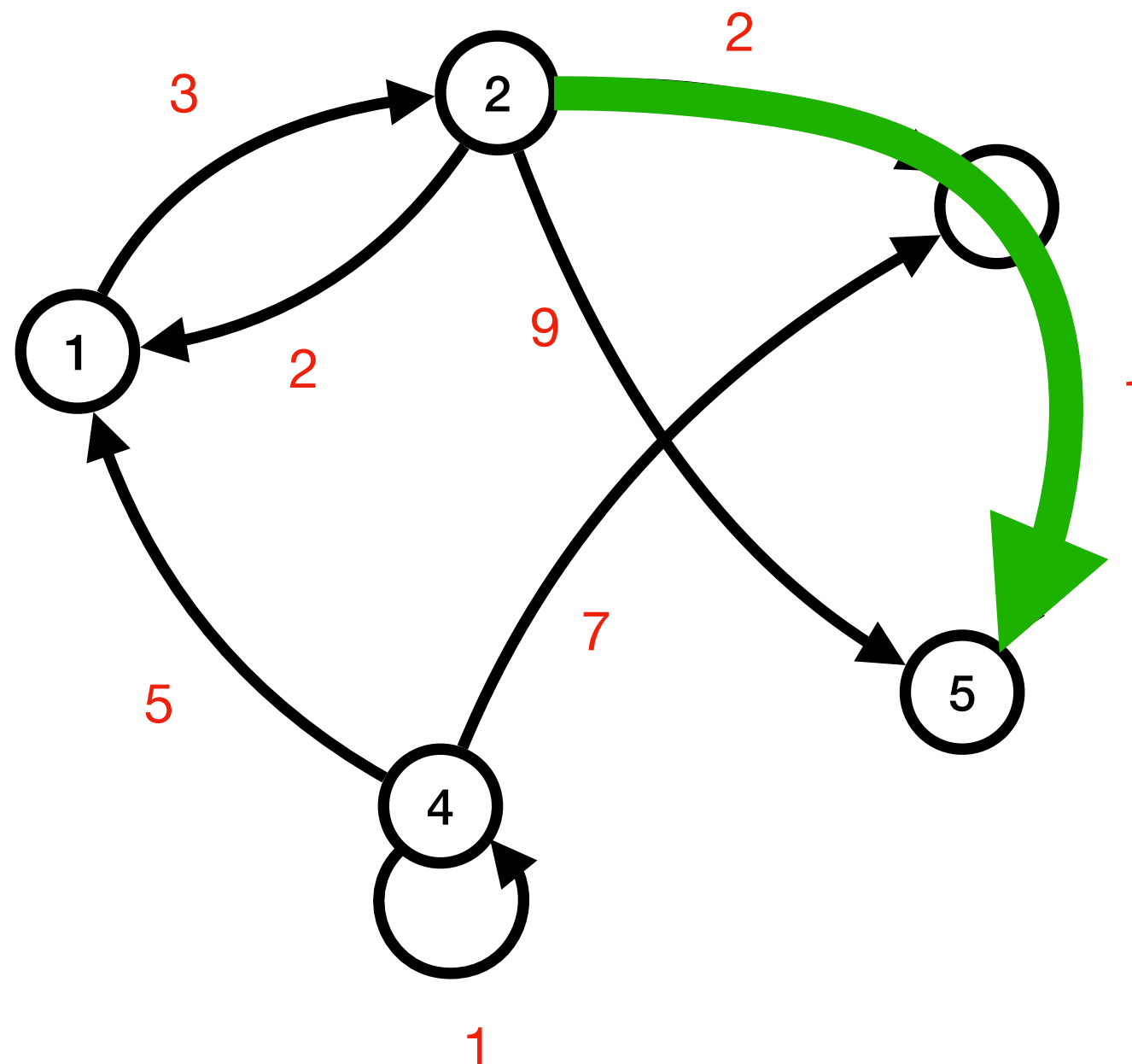
Graphes pondérés



Graphes pondérés

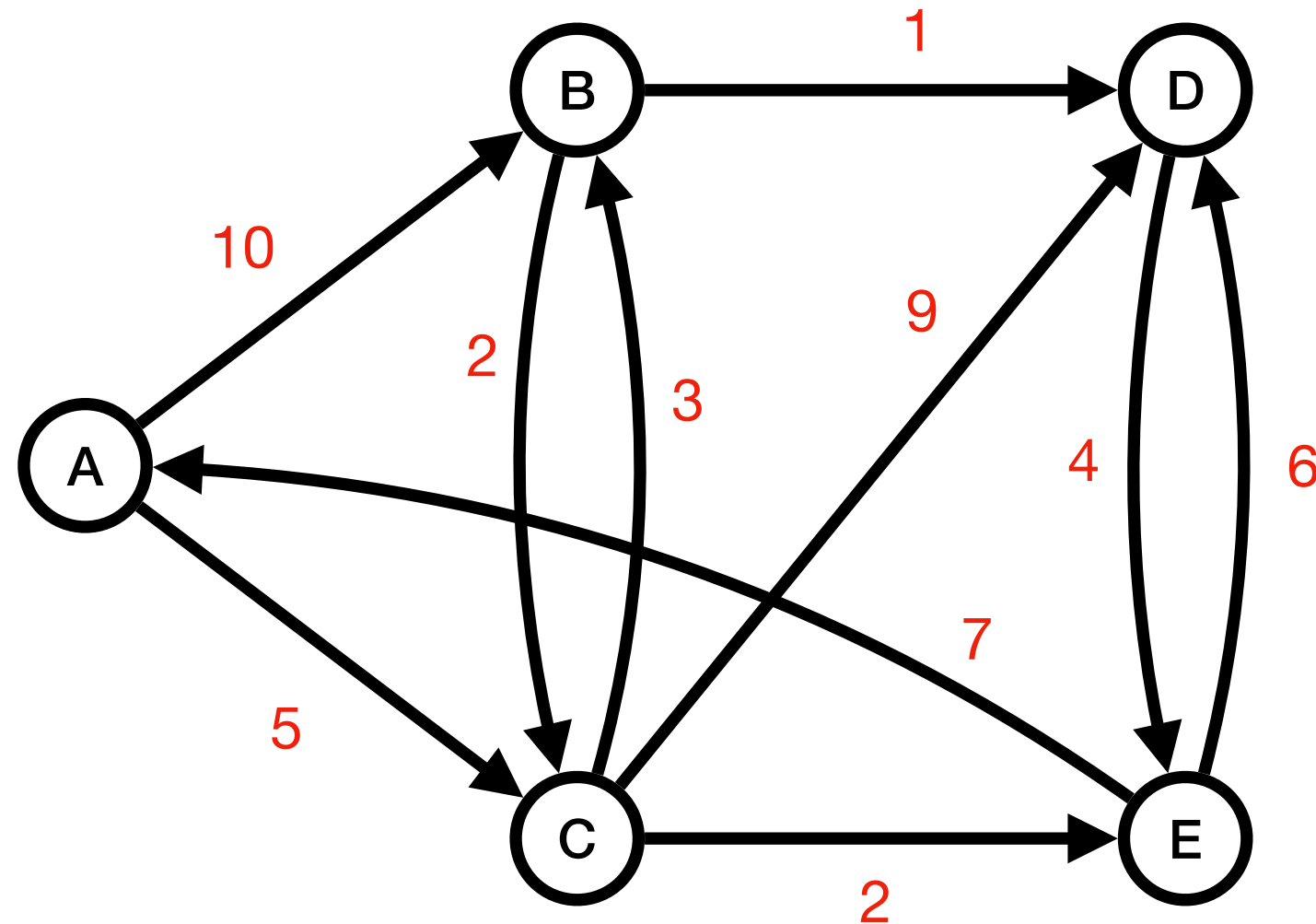


Graphes pondérés

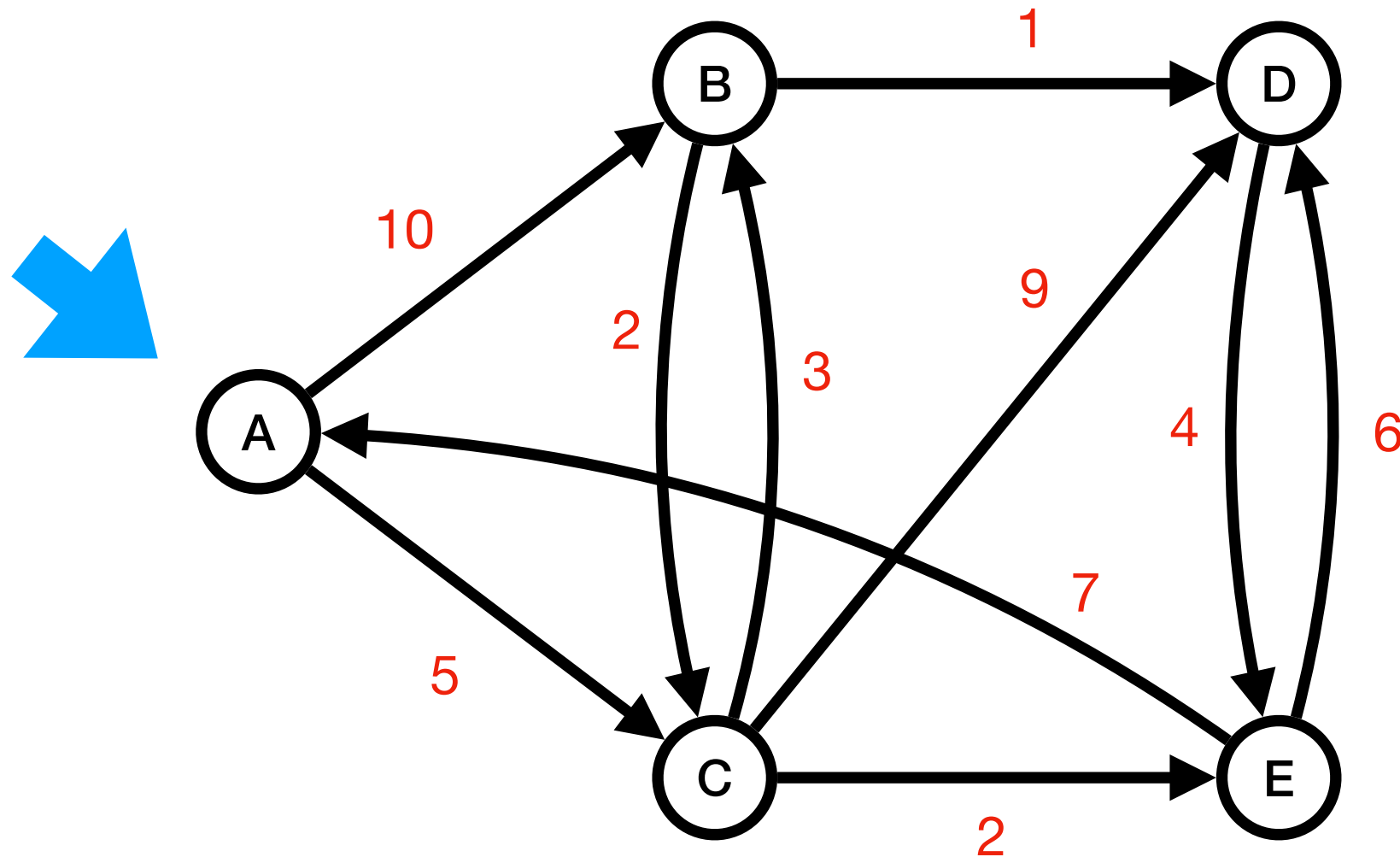


**L'algorithme de parcours
en largeur ne garantit pas
d'obtenir un chemin minimal
sur un graphe pondéré**

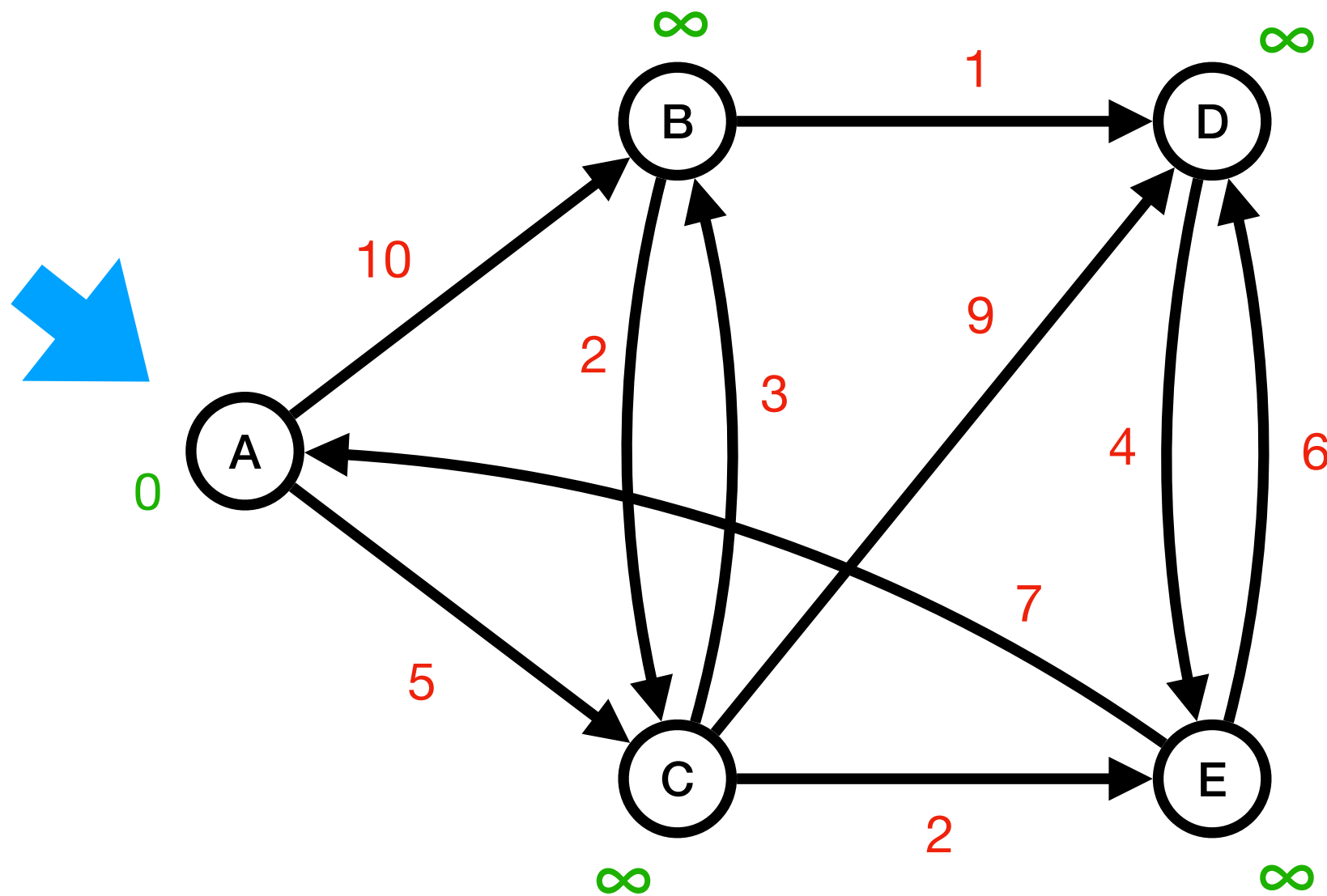
Algorithme de Dijkstra



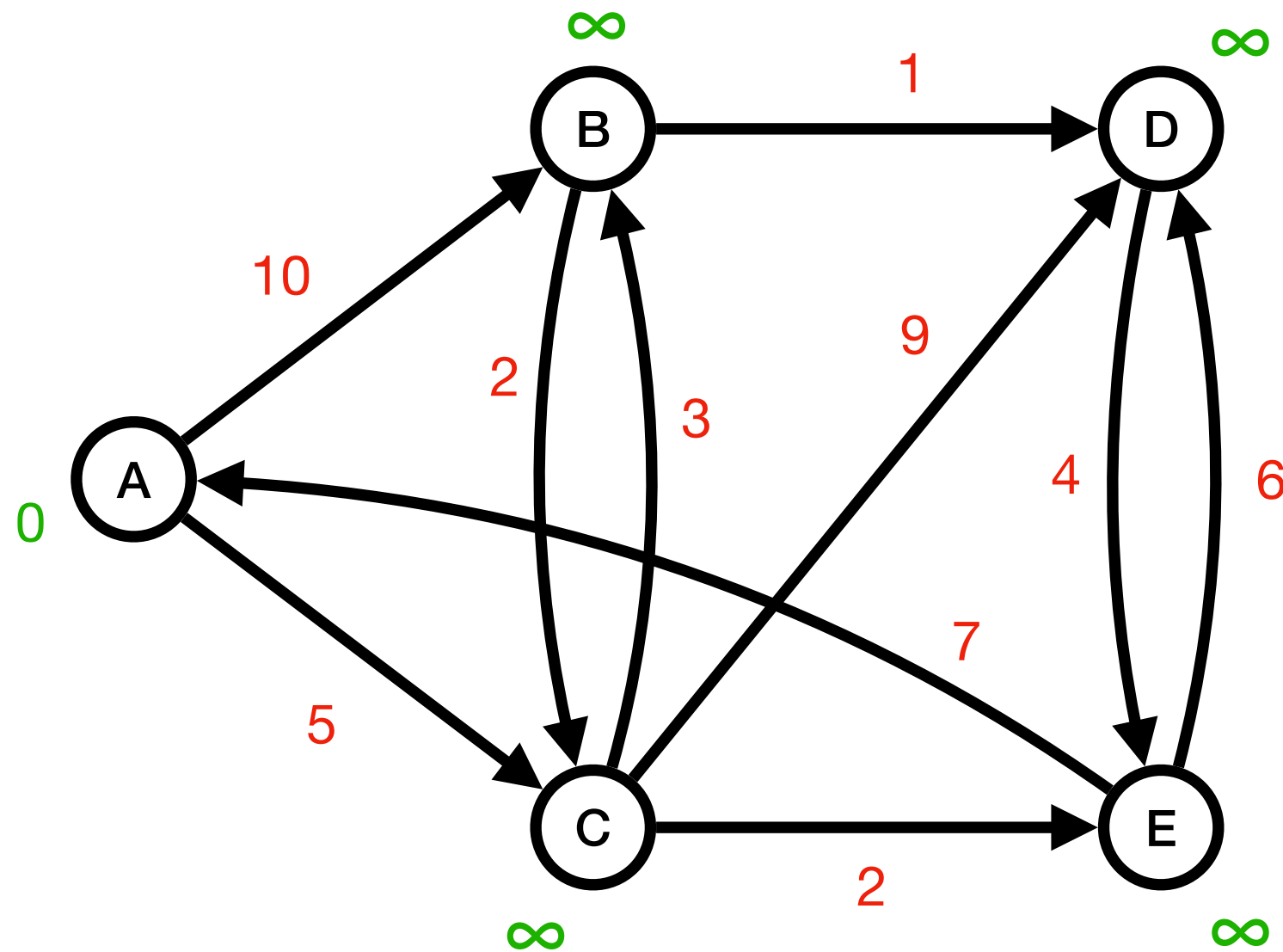
Algorithme de Dijkstra



Algorithme de Dijkstra



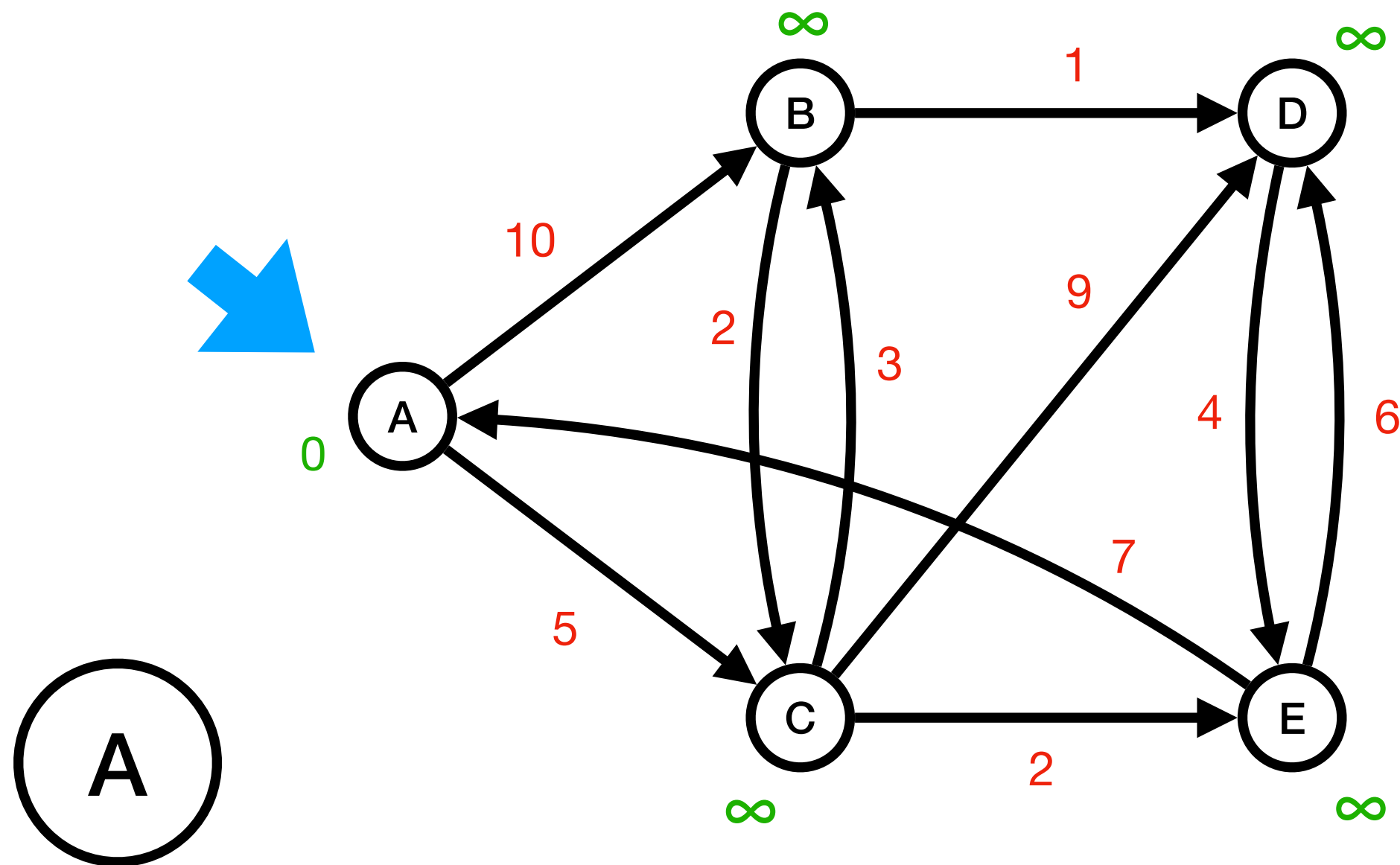
Algorithme de Dijkstra



File de
priorité

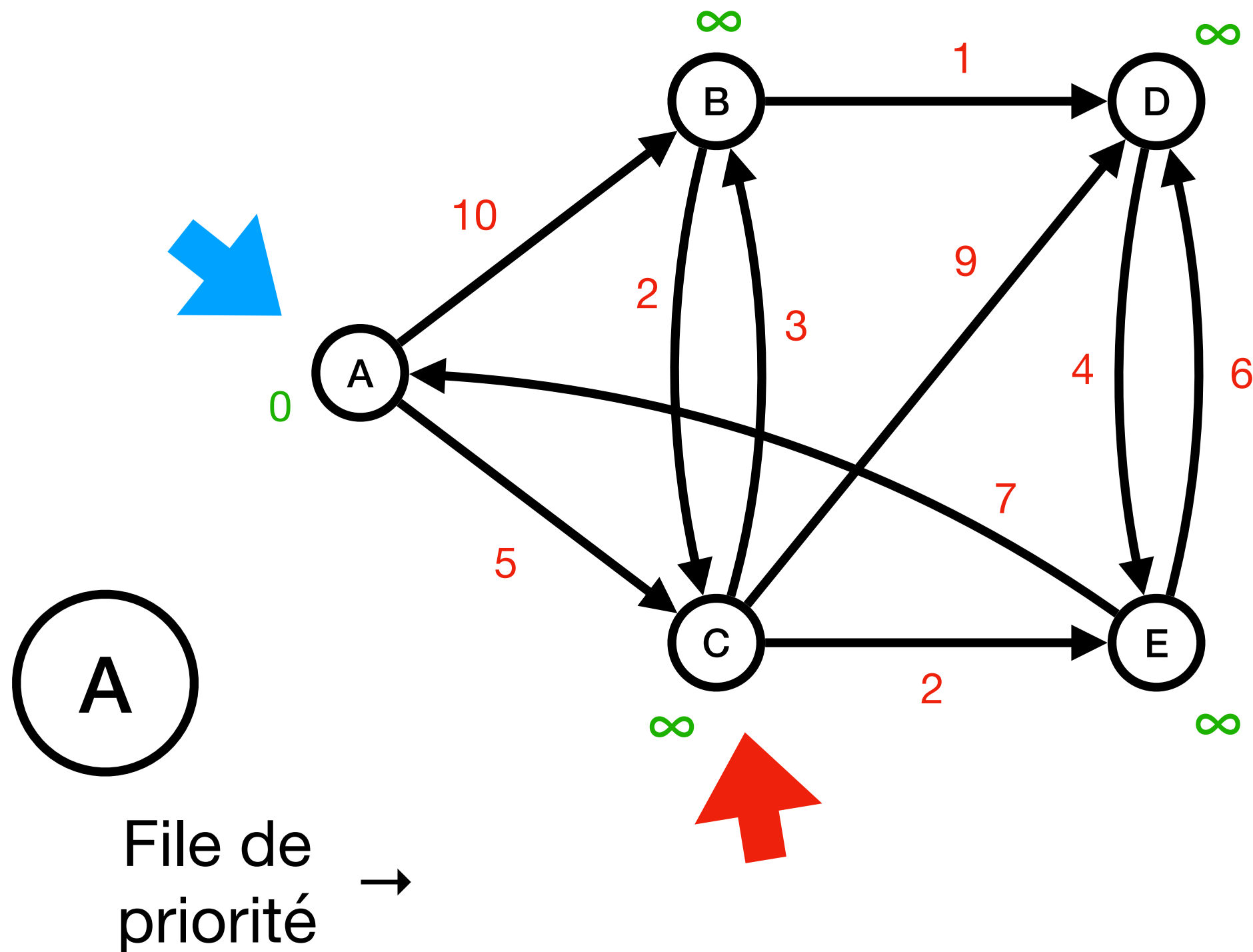


Algorithme de Dijkstra

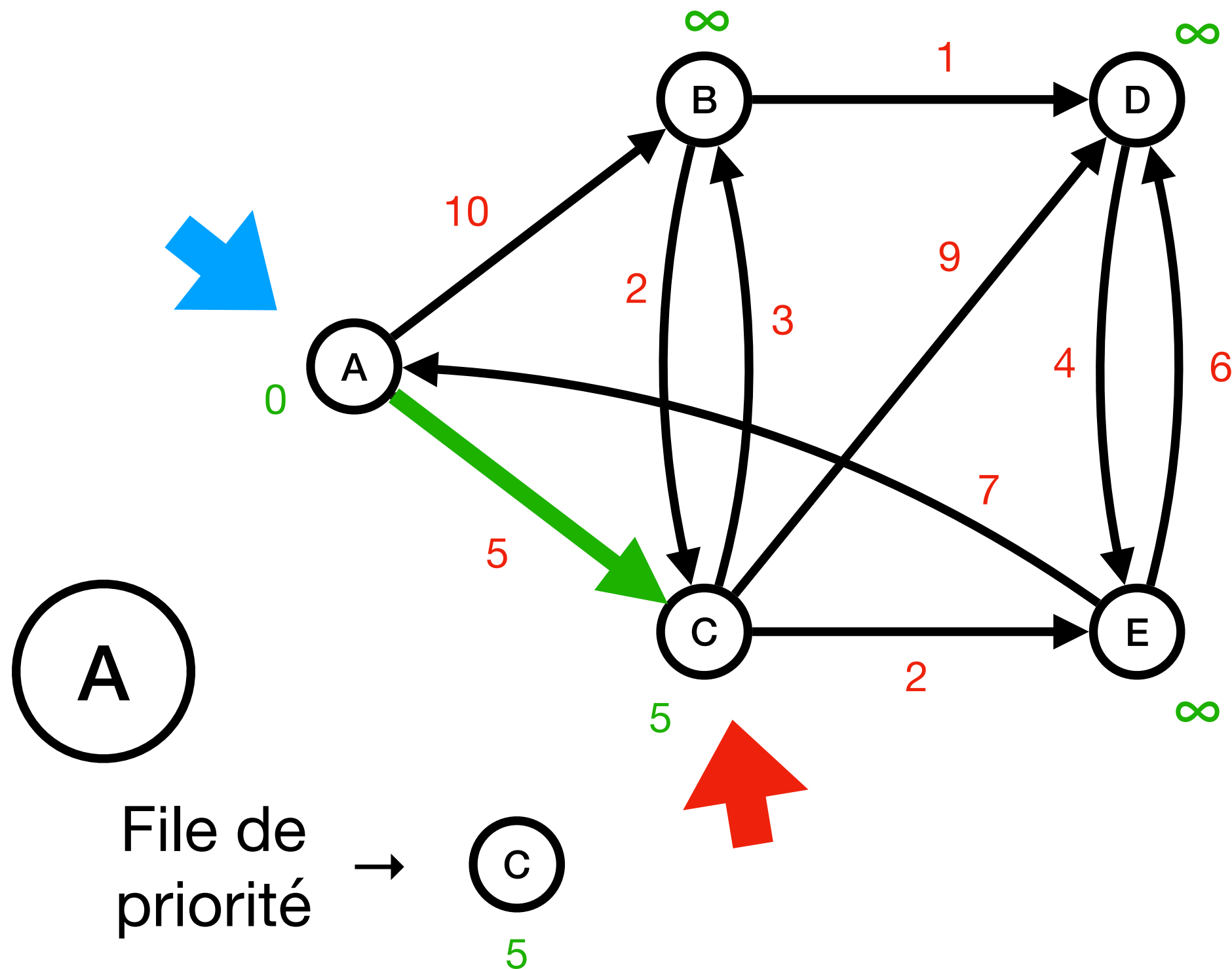


File de
priorité →

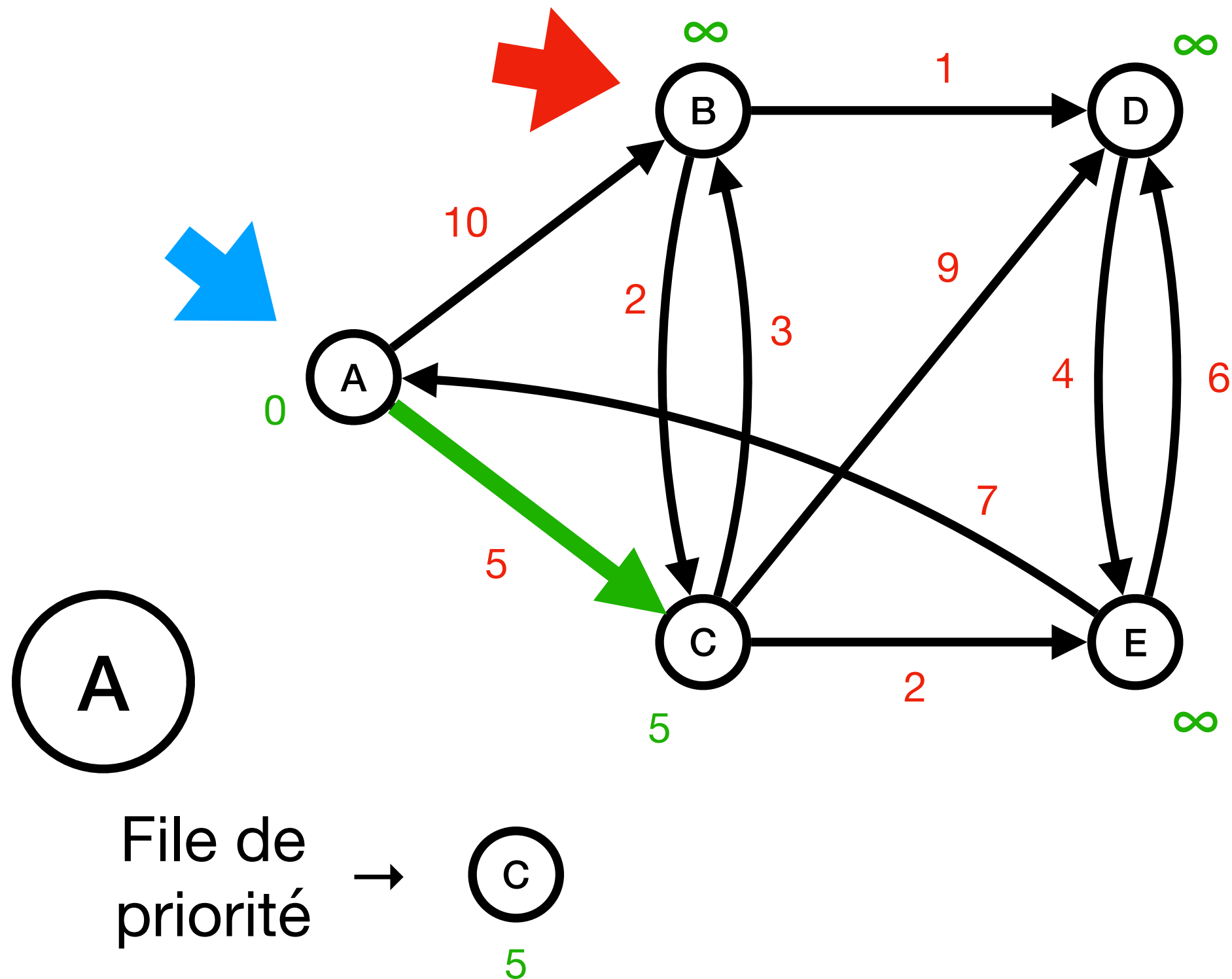
Algorithme de Dijkstra



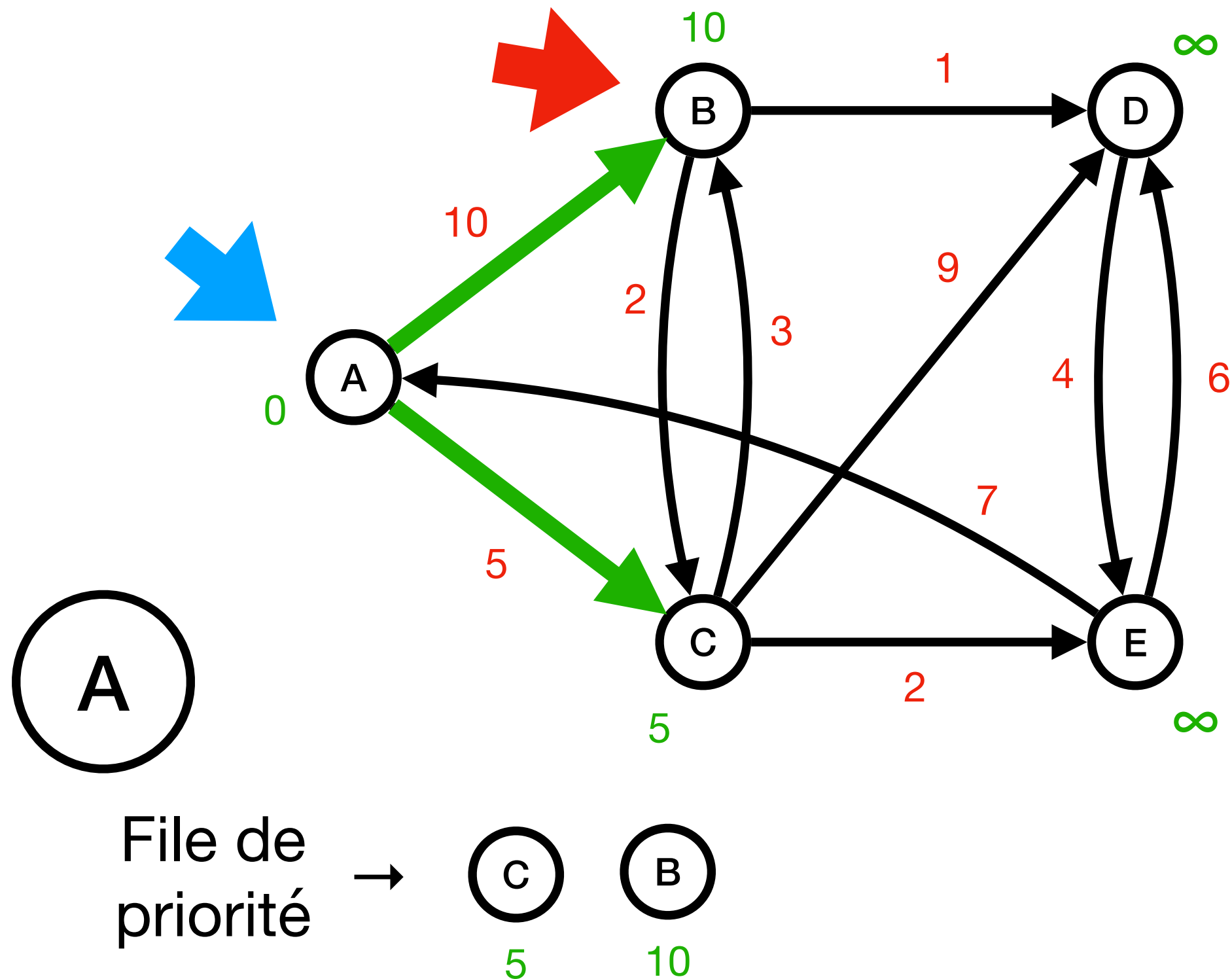
Algorithme de Dijkstra



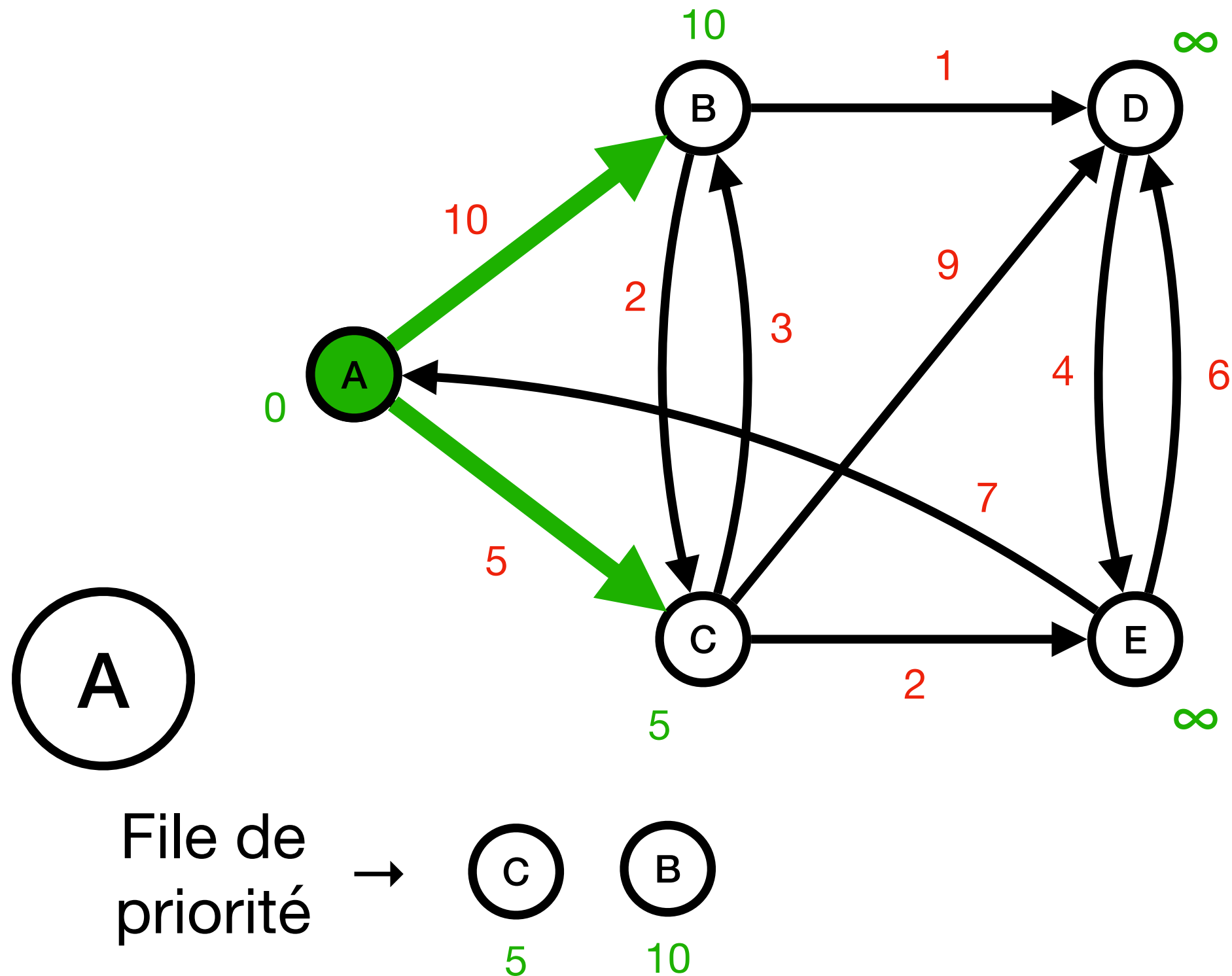
Algorithme de Dijkstra



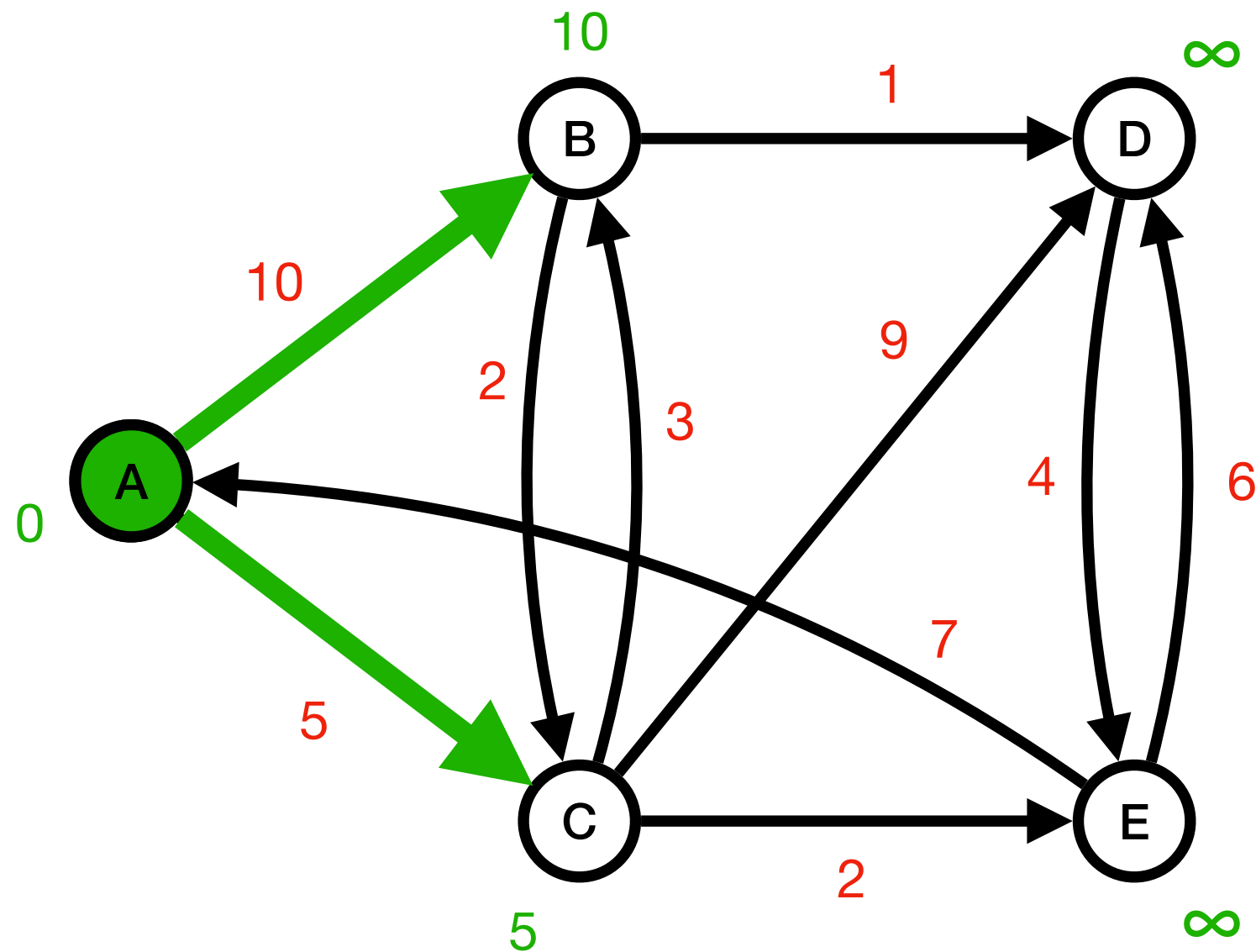
Algorithme de Dijkstra



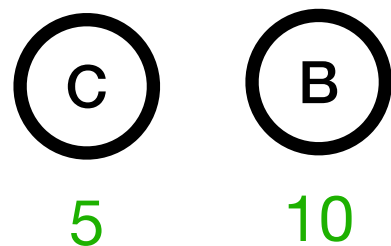
Algorithme de Dijkstra



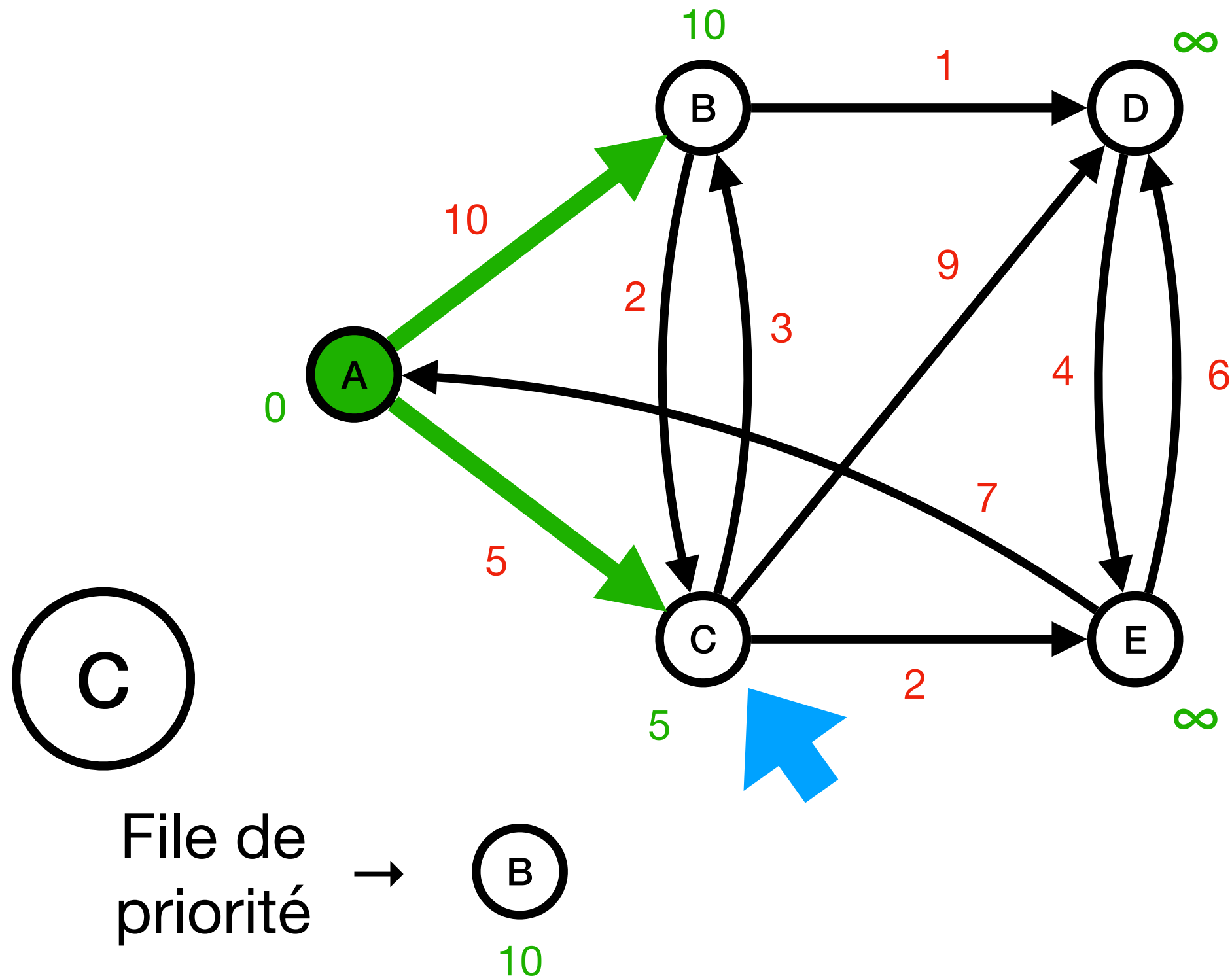
Algorithme de Dijkstra



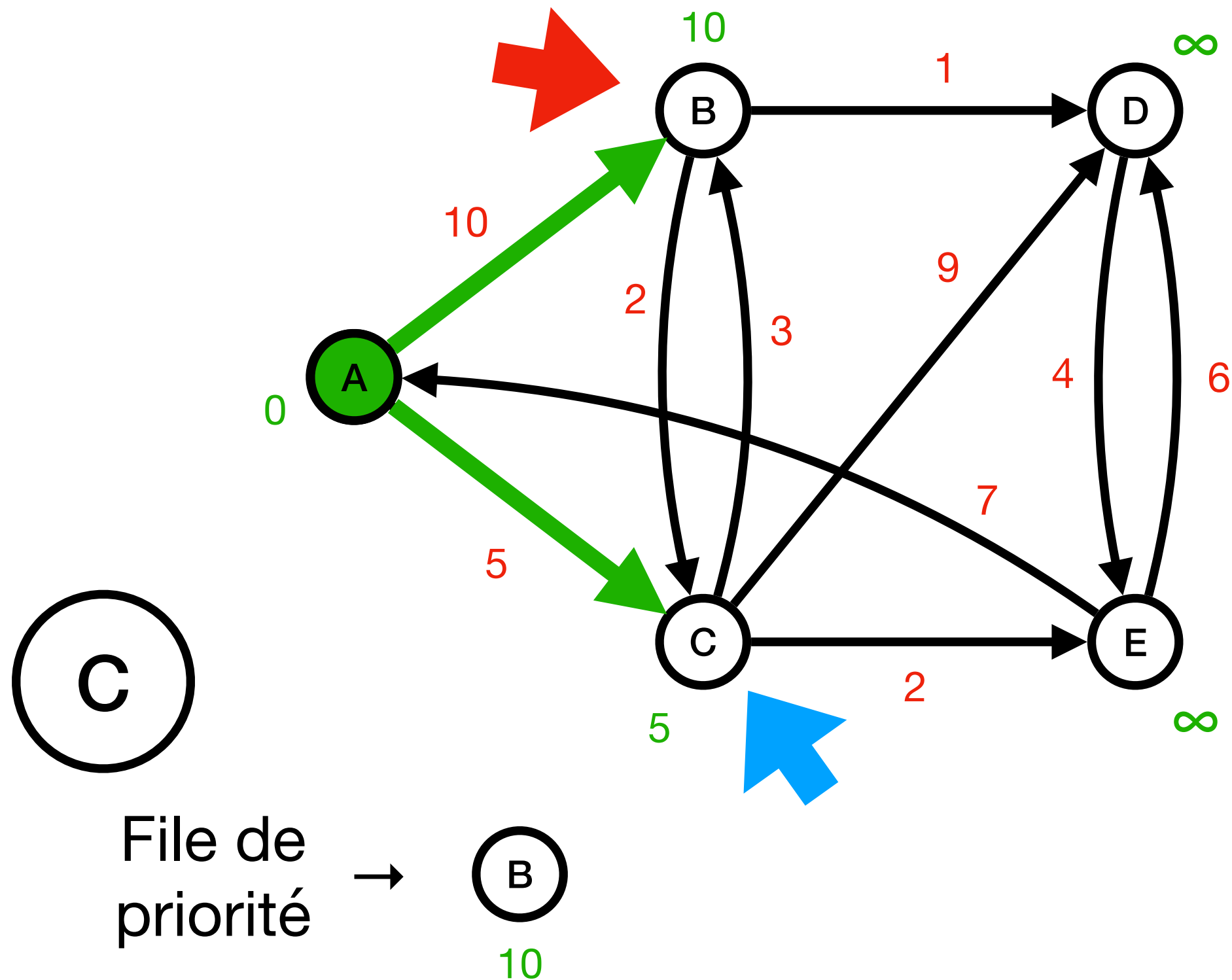
File de
priorité →



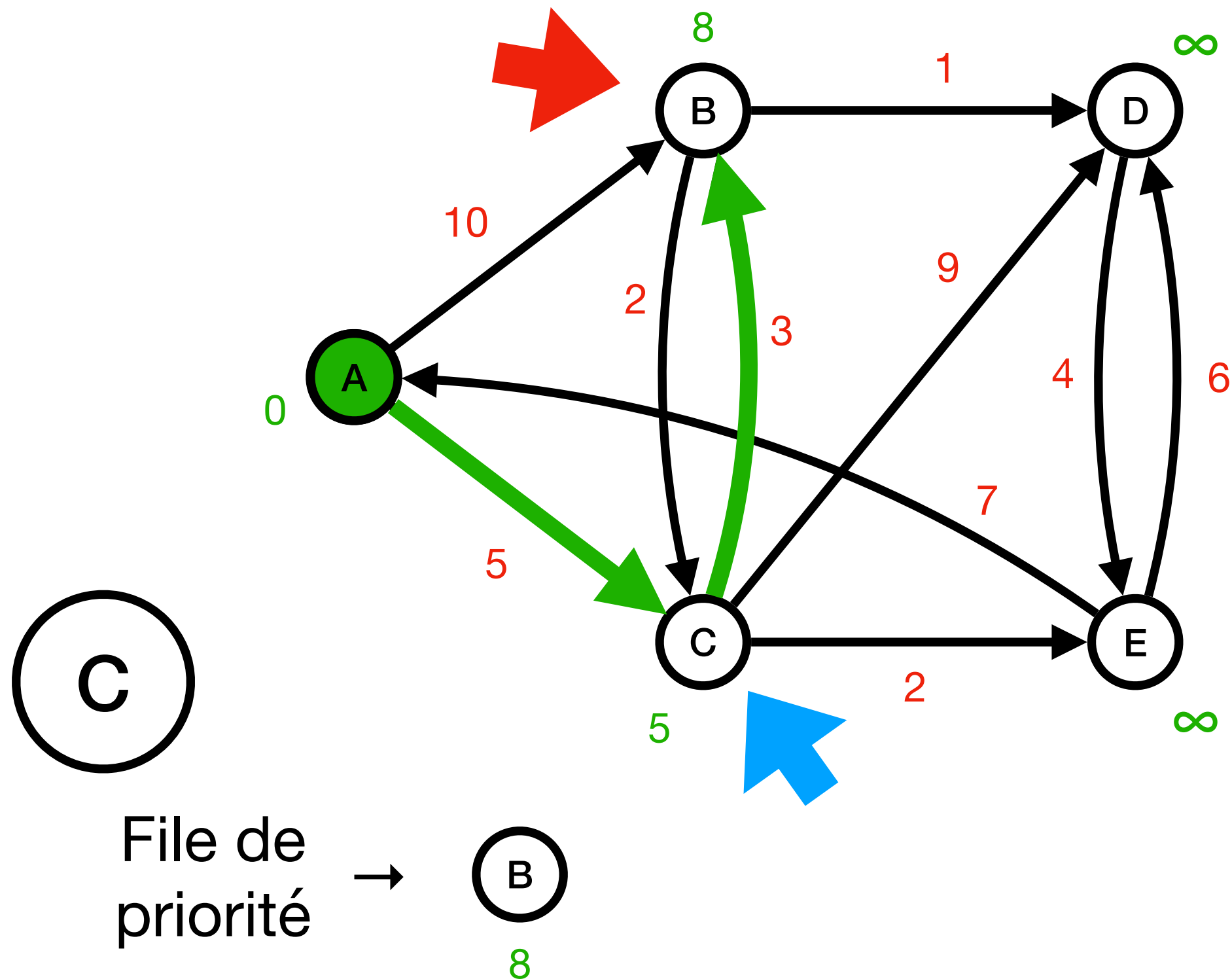
Algorithme de Dijkstra



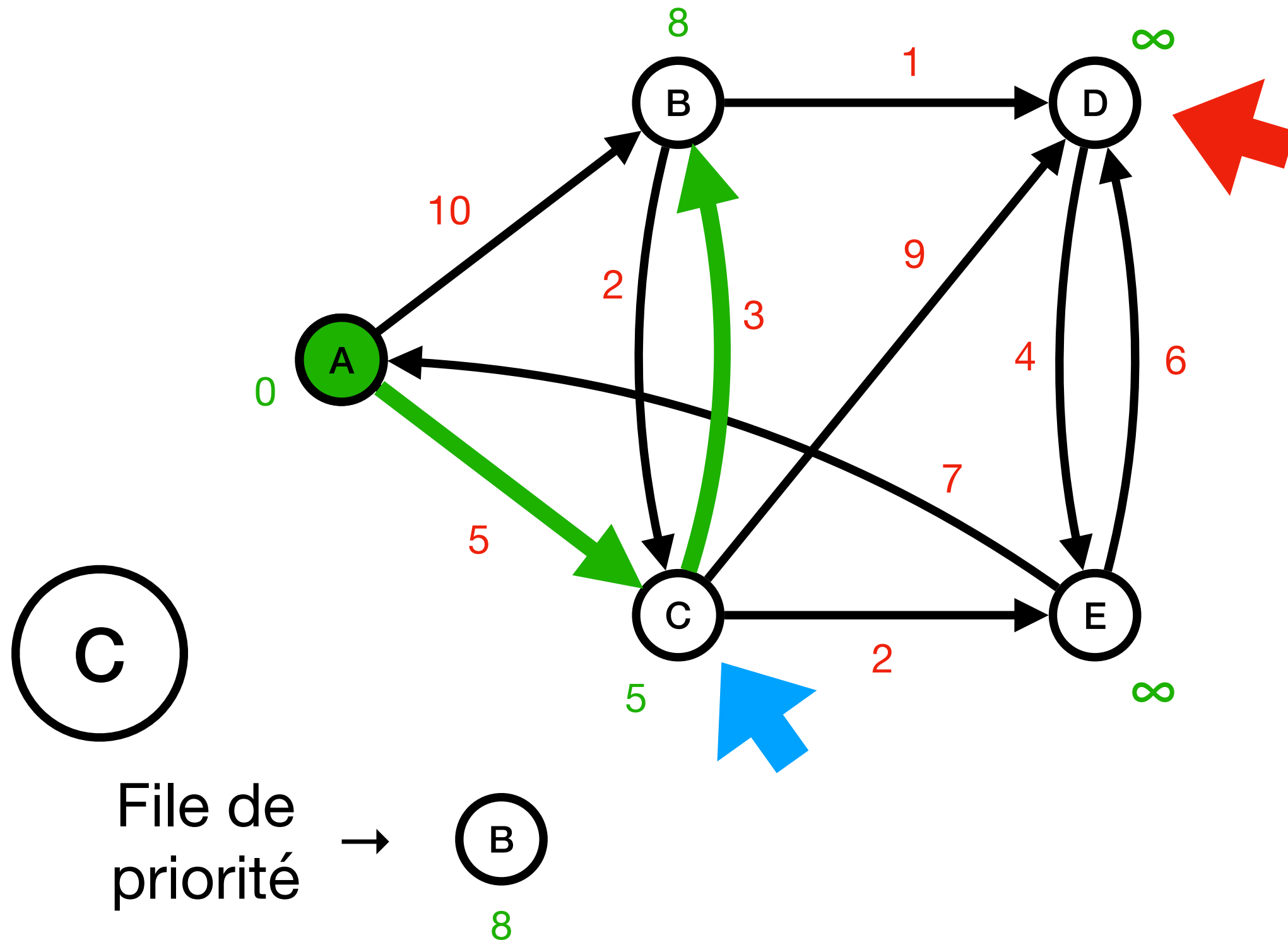
Algorithme de Dijkstra



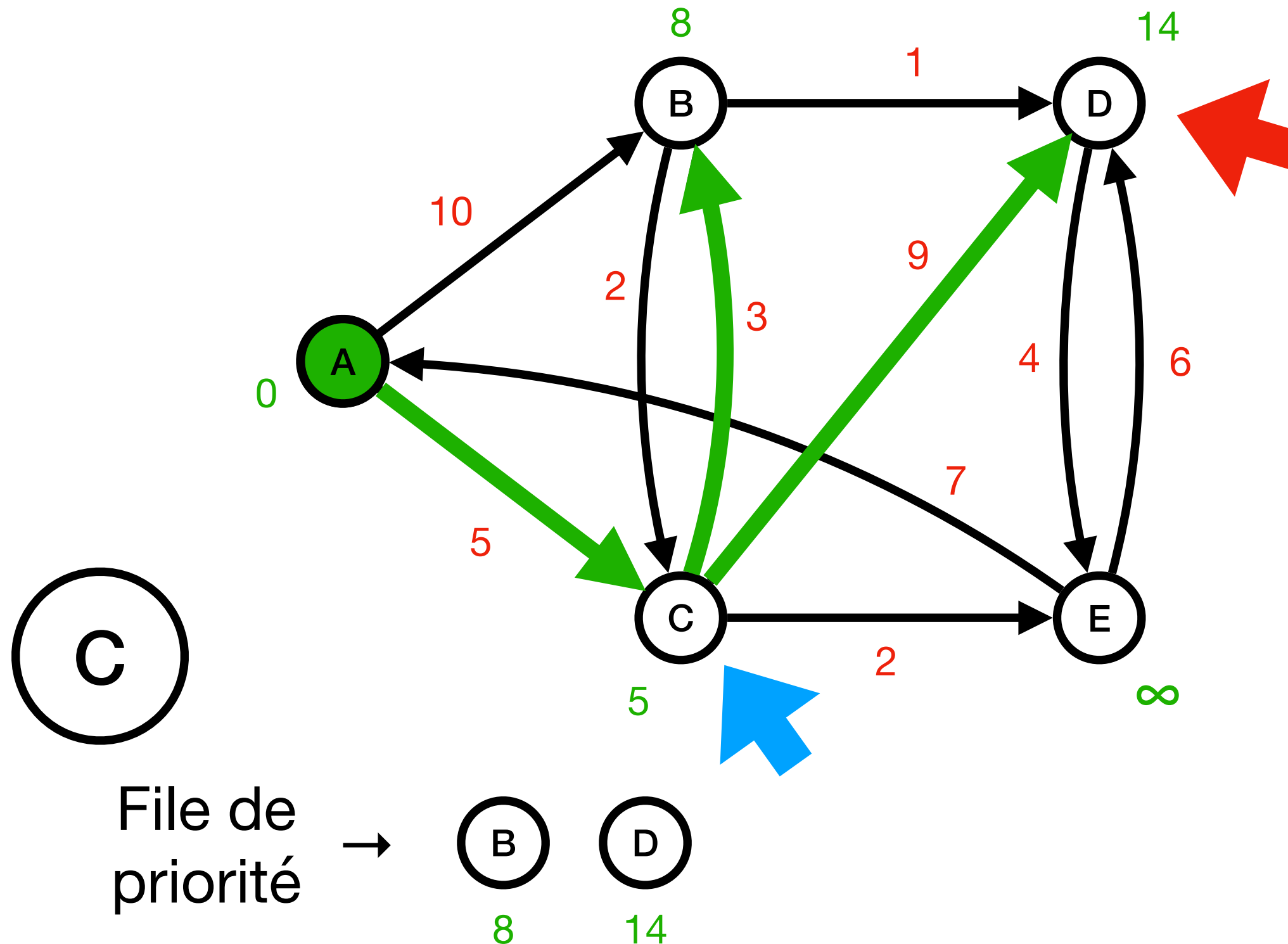
Algorithme de Dijkstra



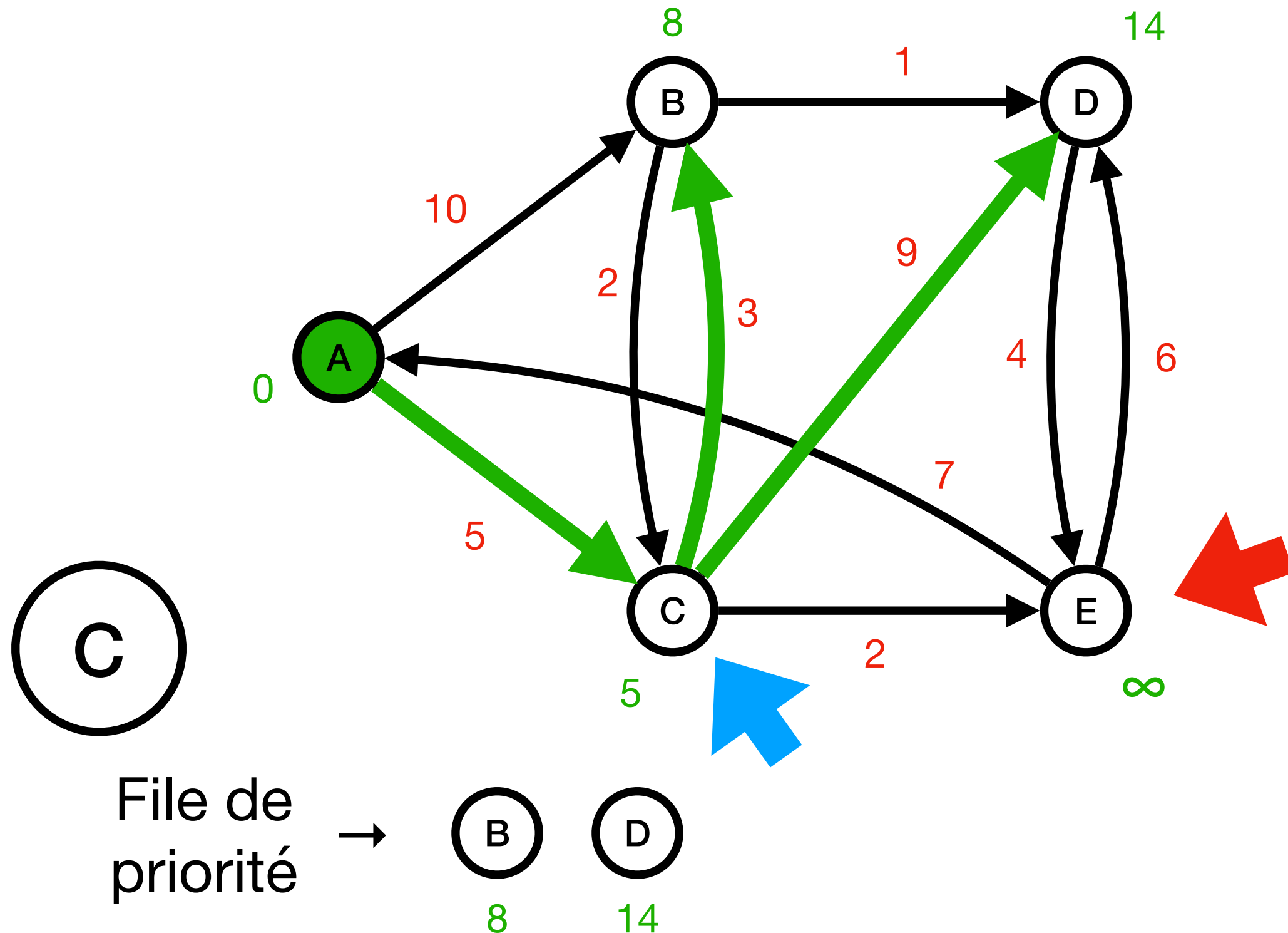
Algorithme de Dijkstra



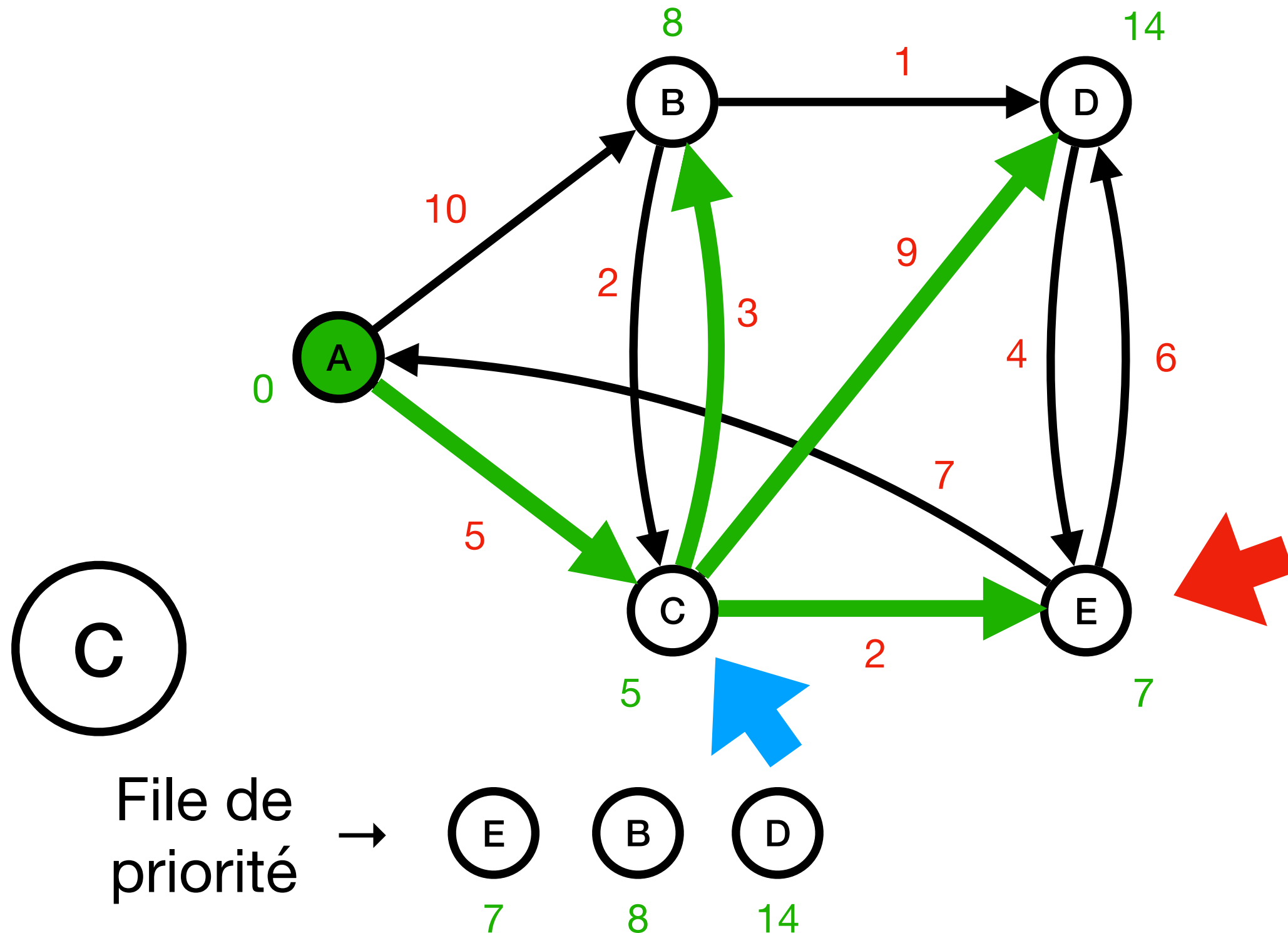
Algorithme de Dijkstra



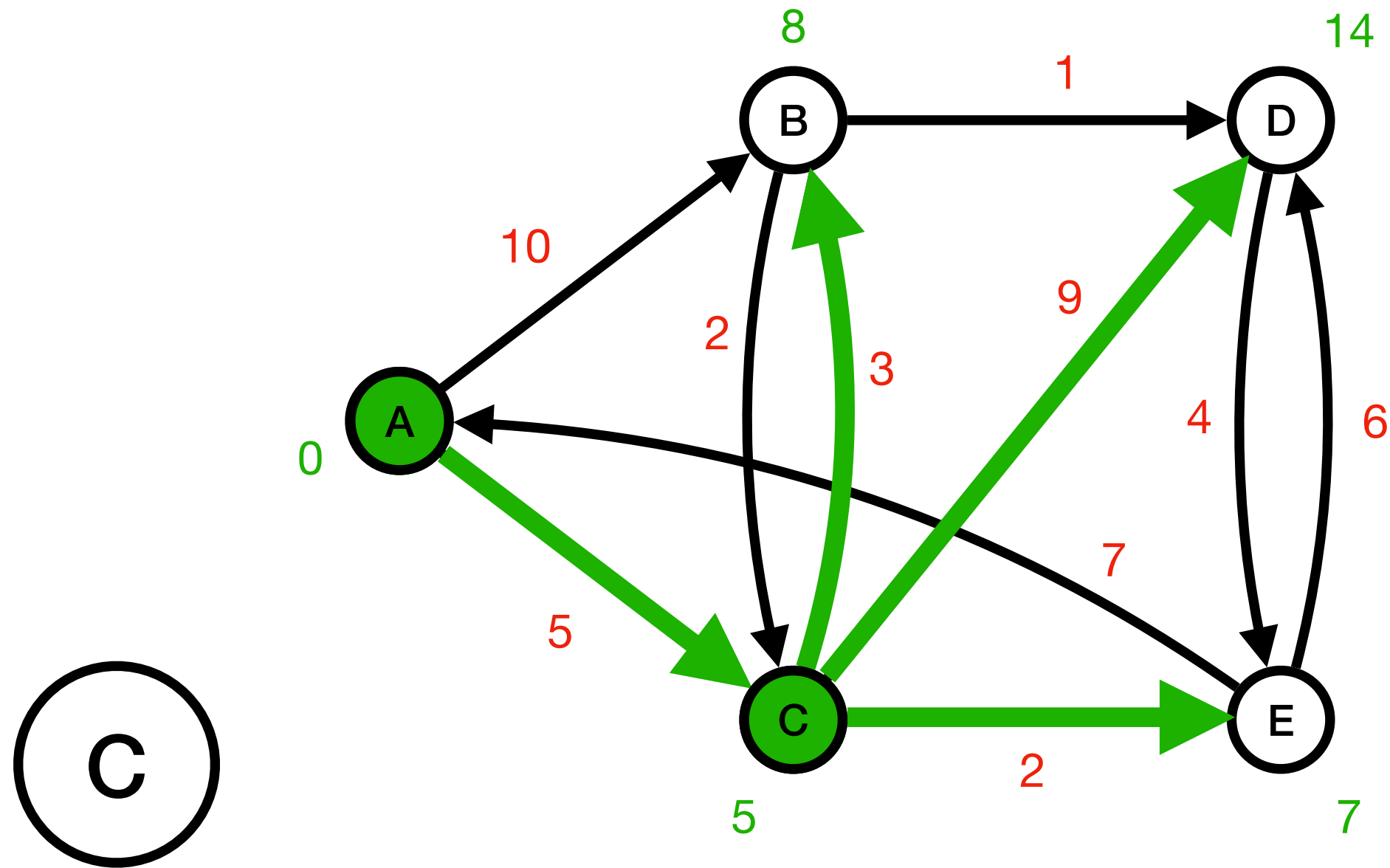
Algorithme de Dijkstra



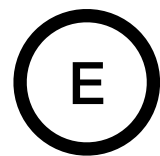
Algorithme de Dijkstra



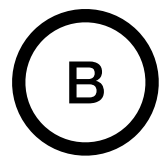
Algorithme de Dijkstra



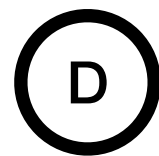
File de
priorité



7

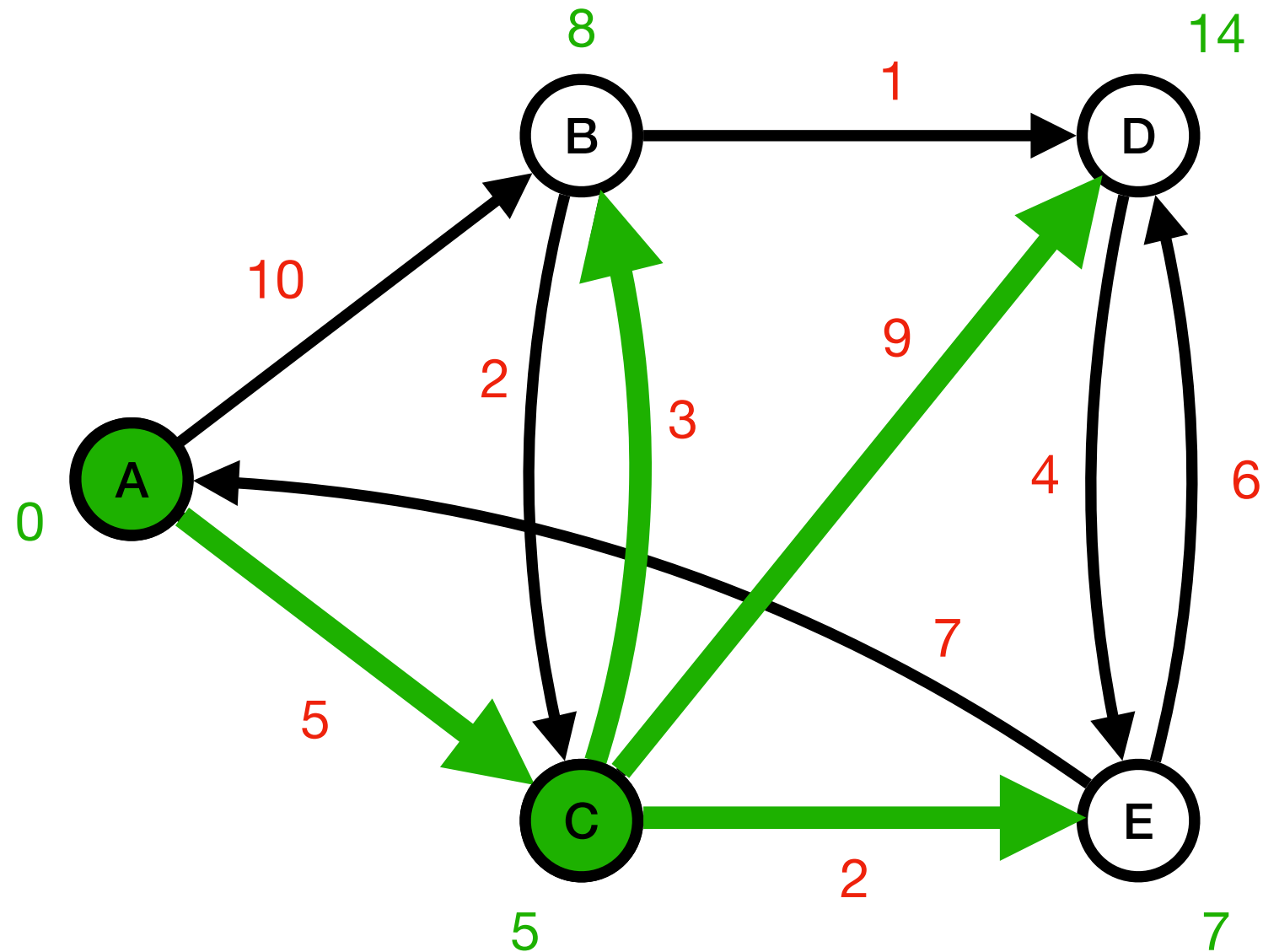


8

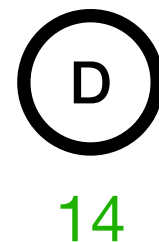
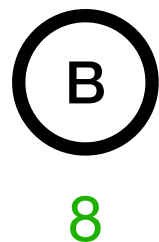
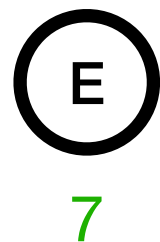


14

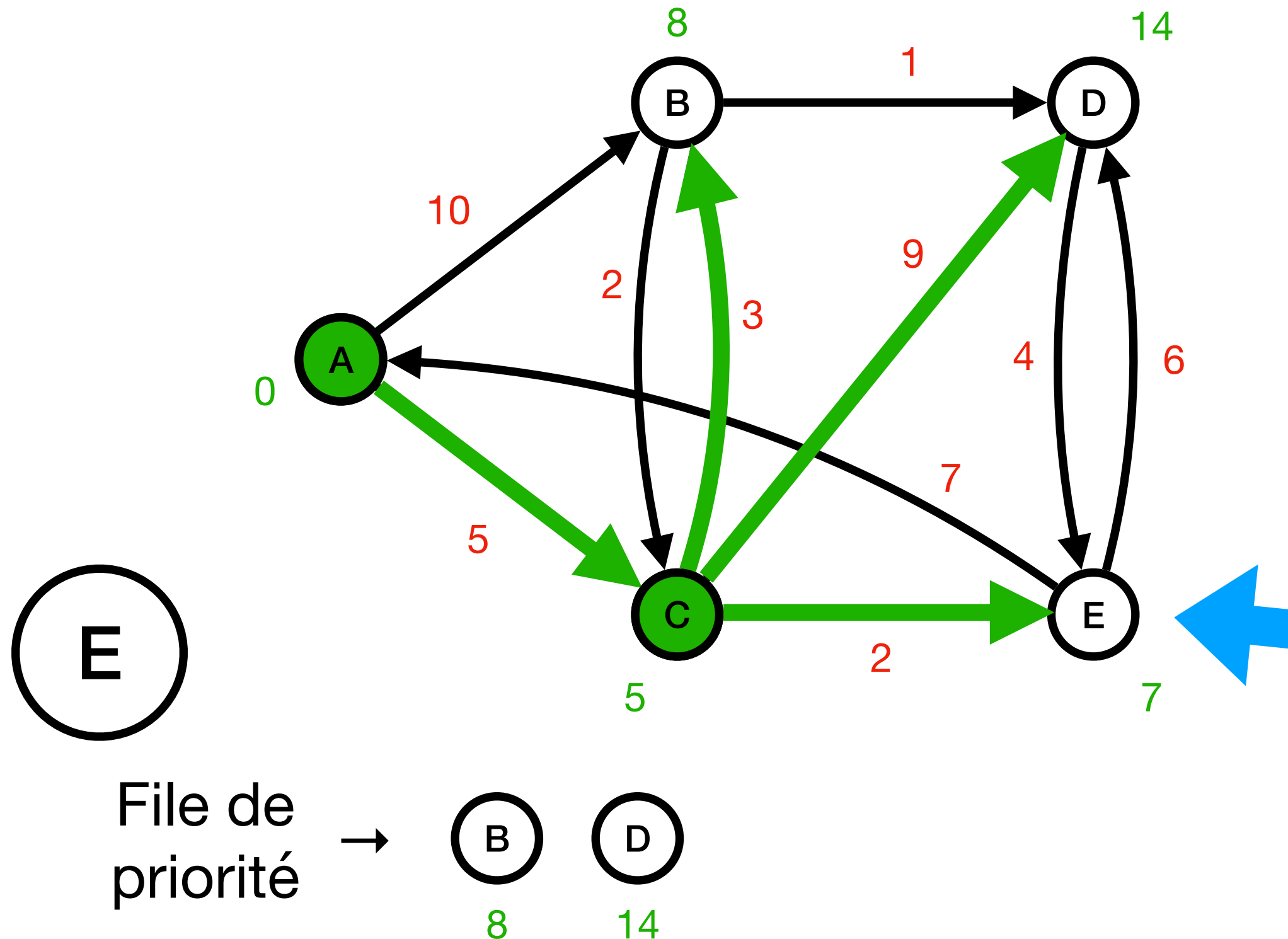
Algorithme de Dijkstra



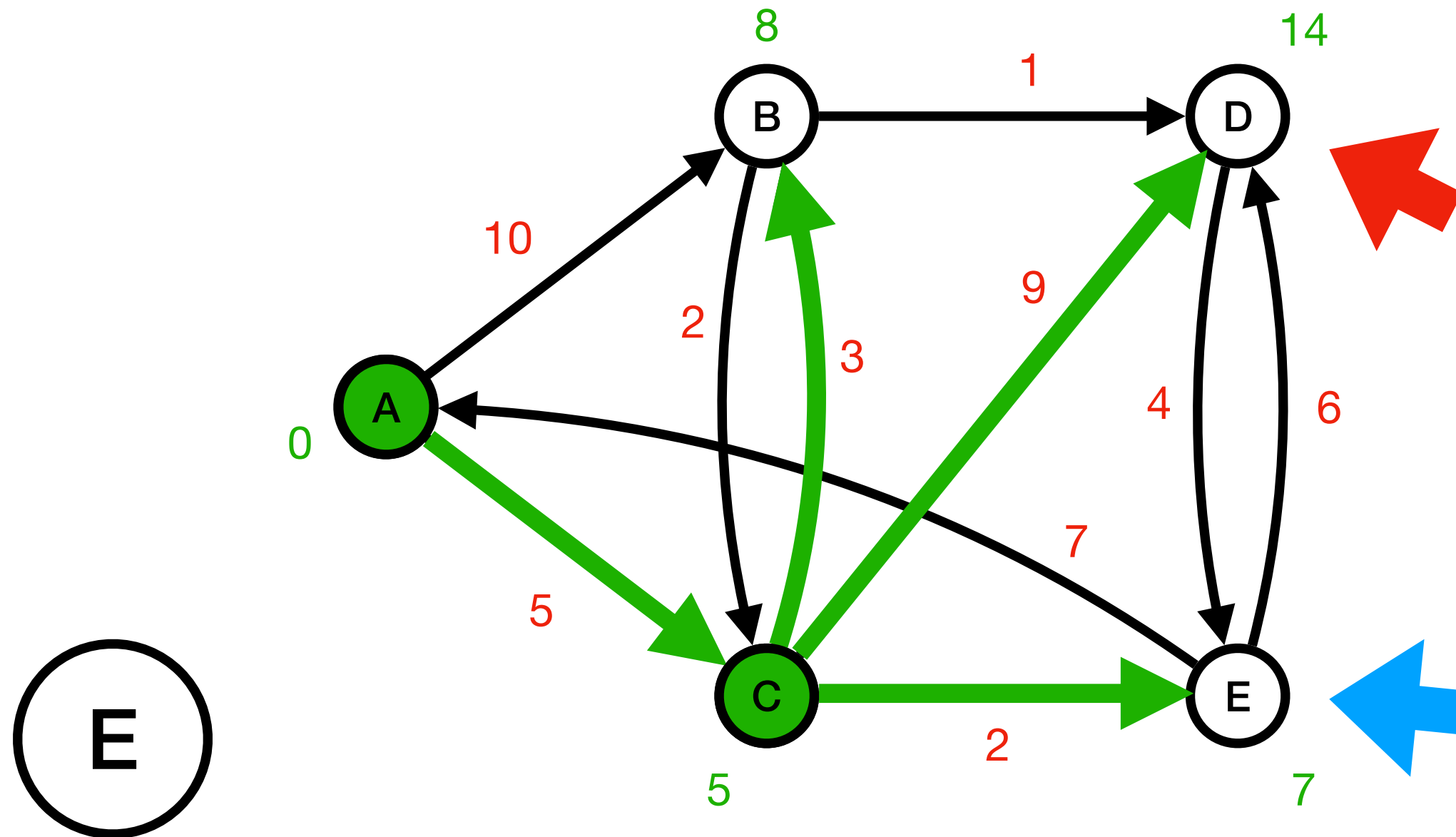
File de
priorité →



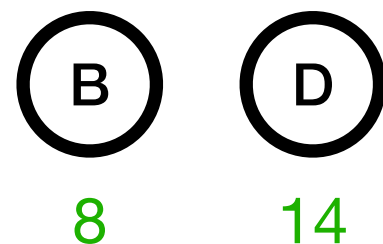
Algorithme de Dijkstra



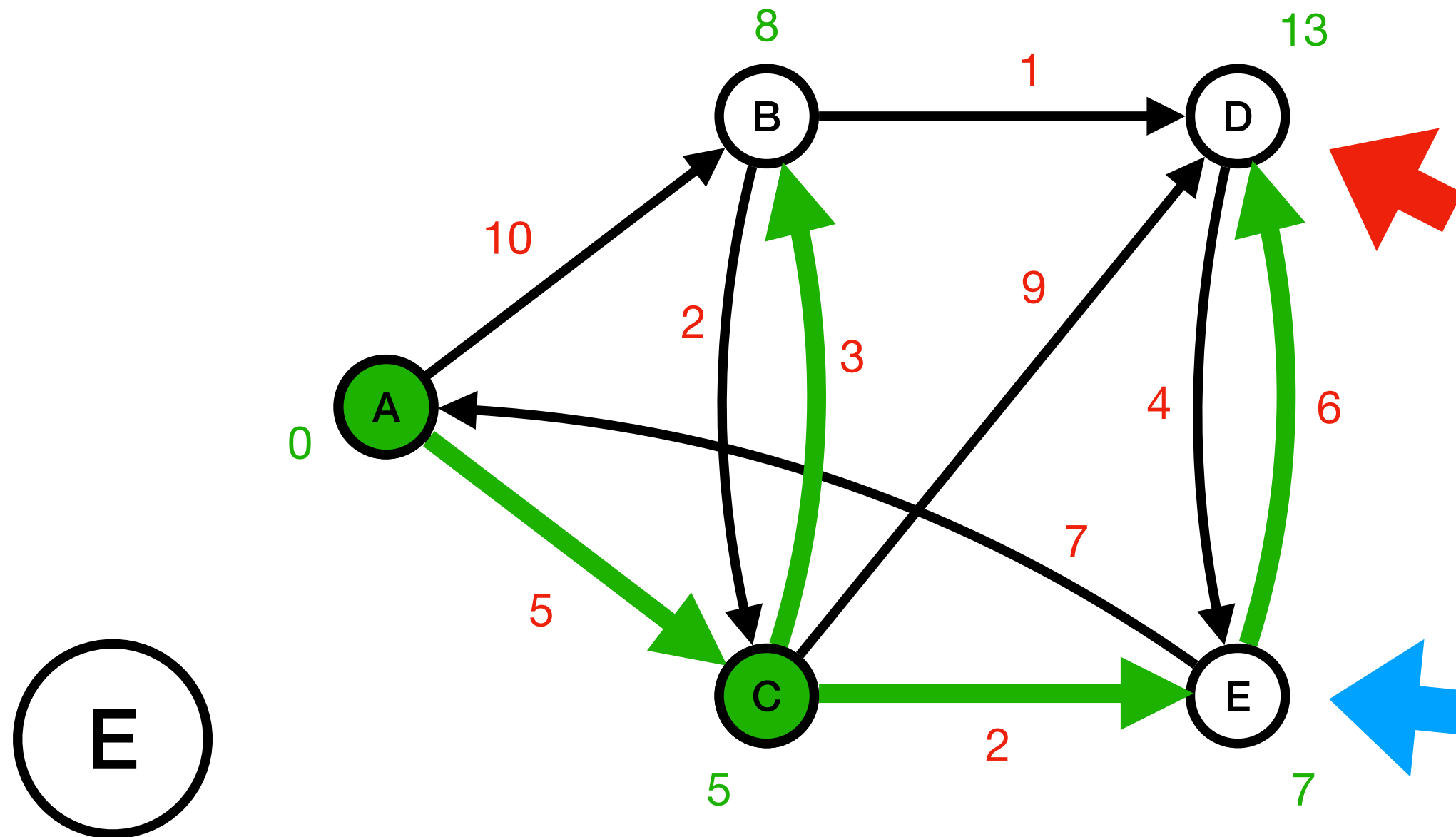
Algorithme de Dijkstra



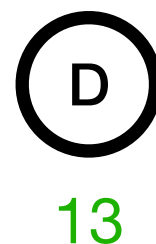
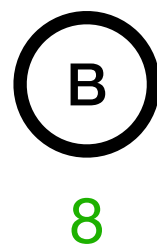
File de
priorité →



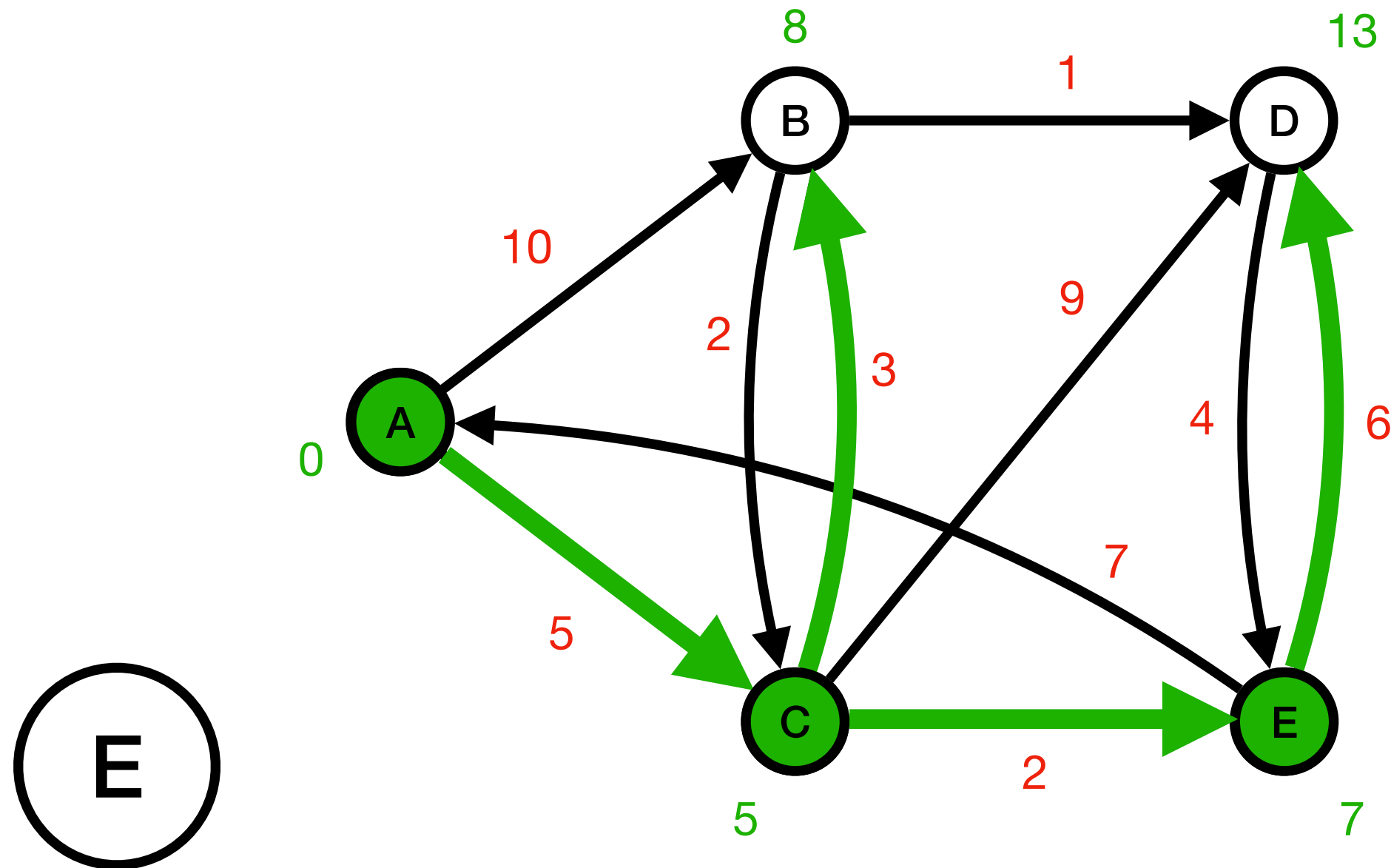
Algorithme de Dijkstra



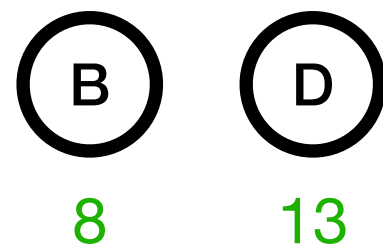
File de
priorité →



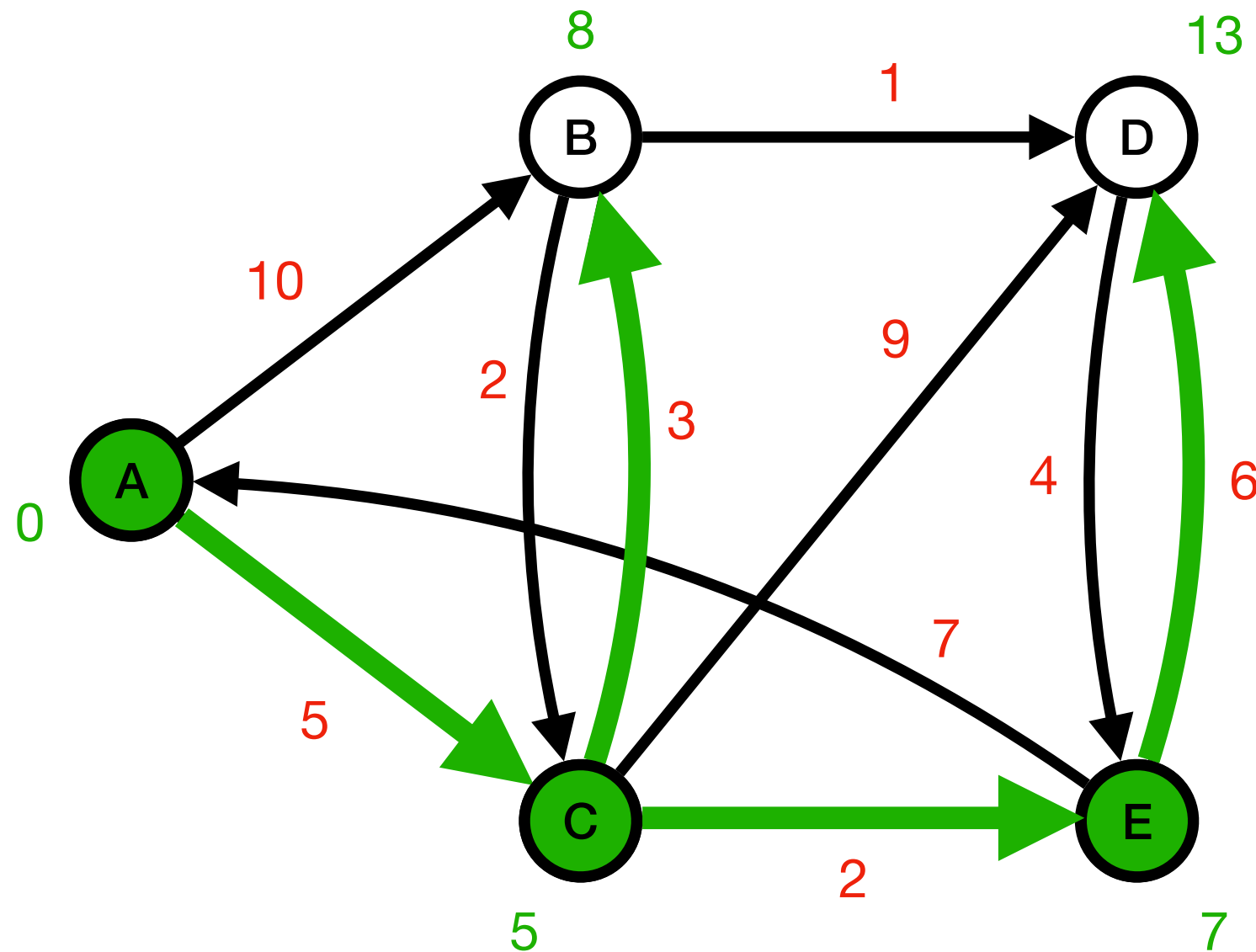
Algorithme de Dijkstra



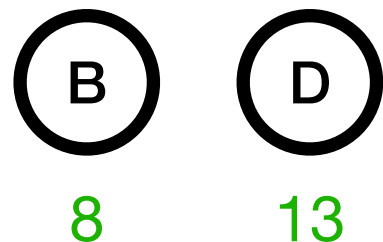
File de
priorité →



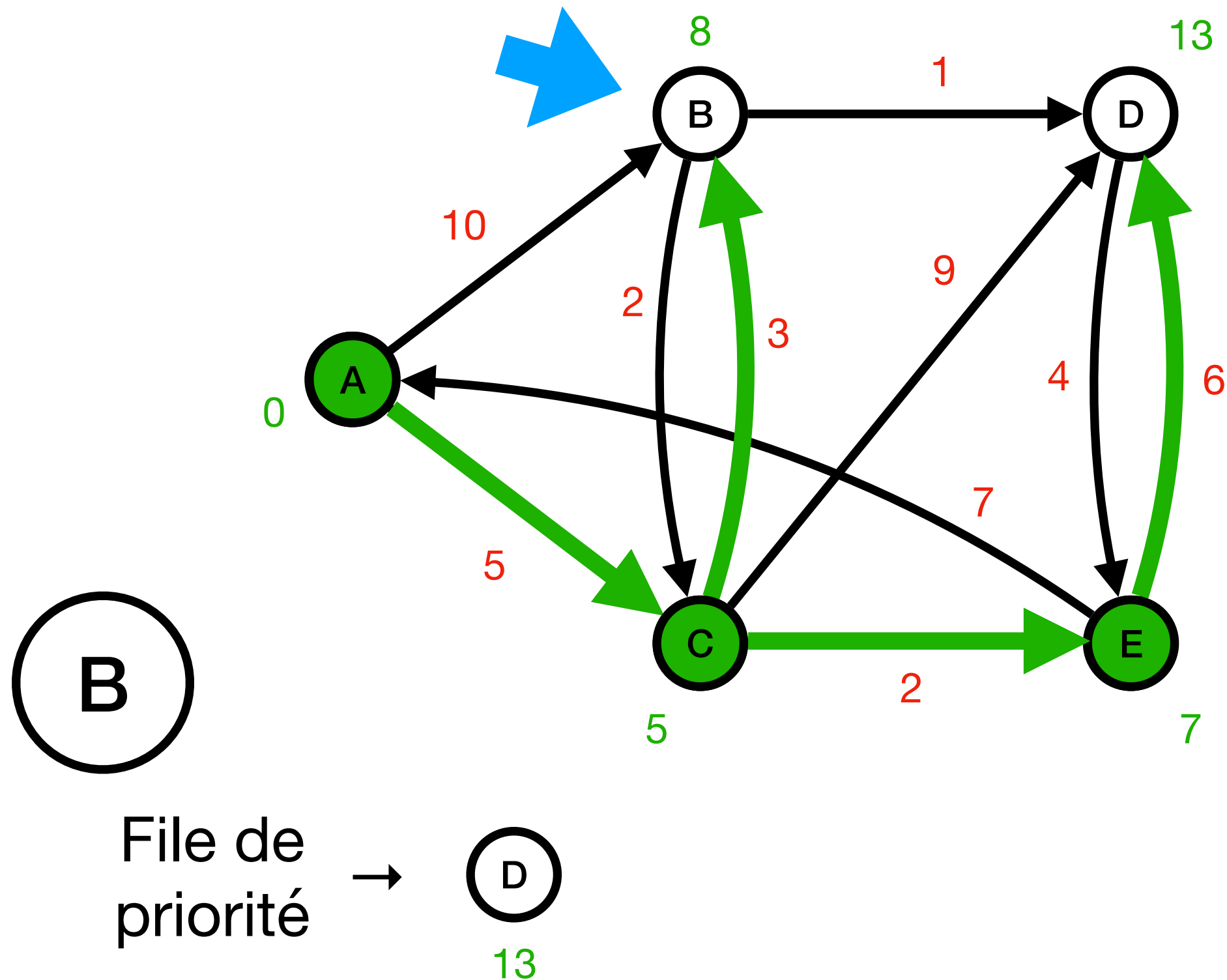
Algorithme de Dijkstra



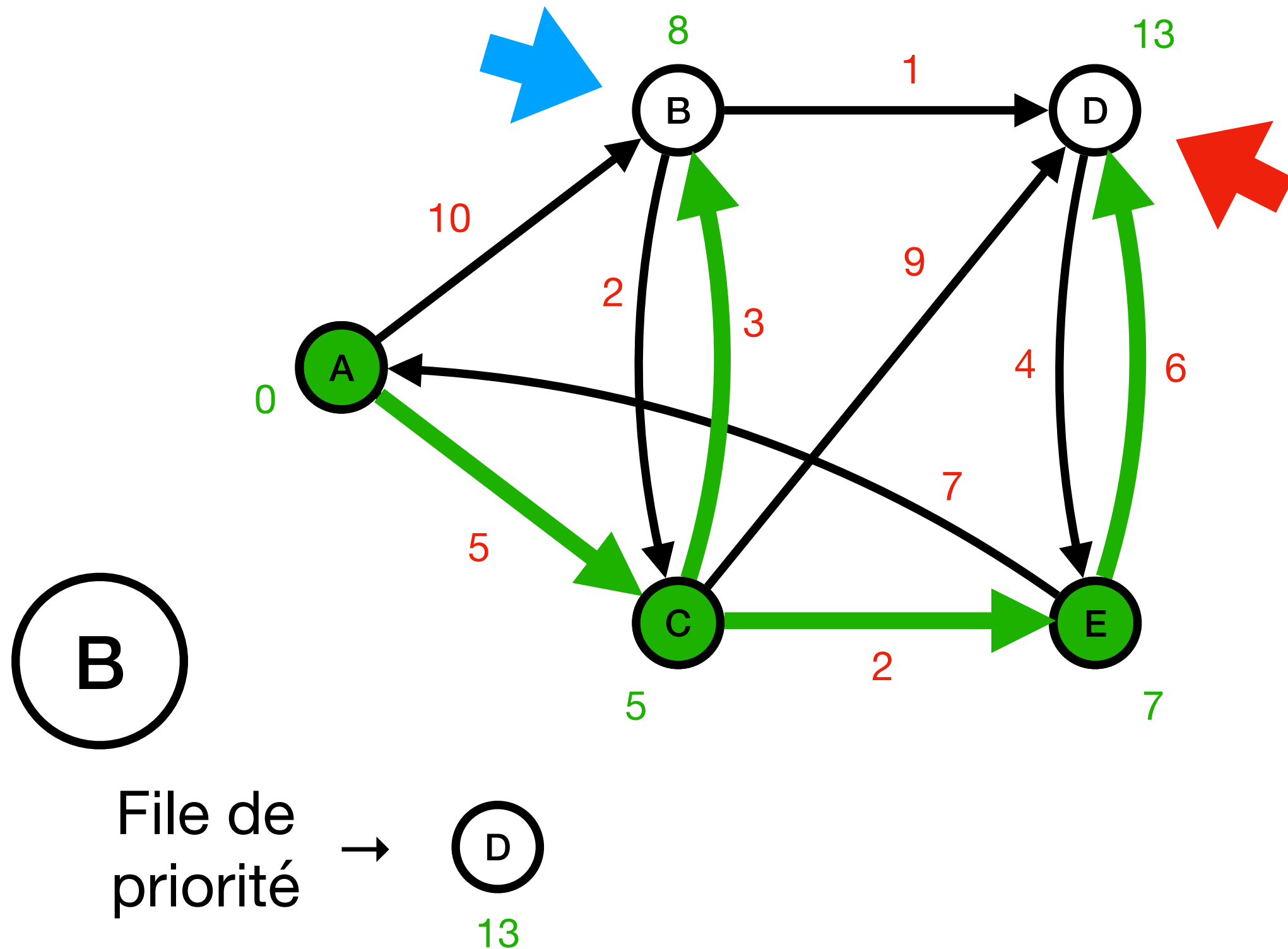
File de
priorité →



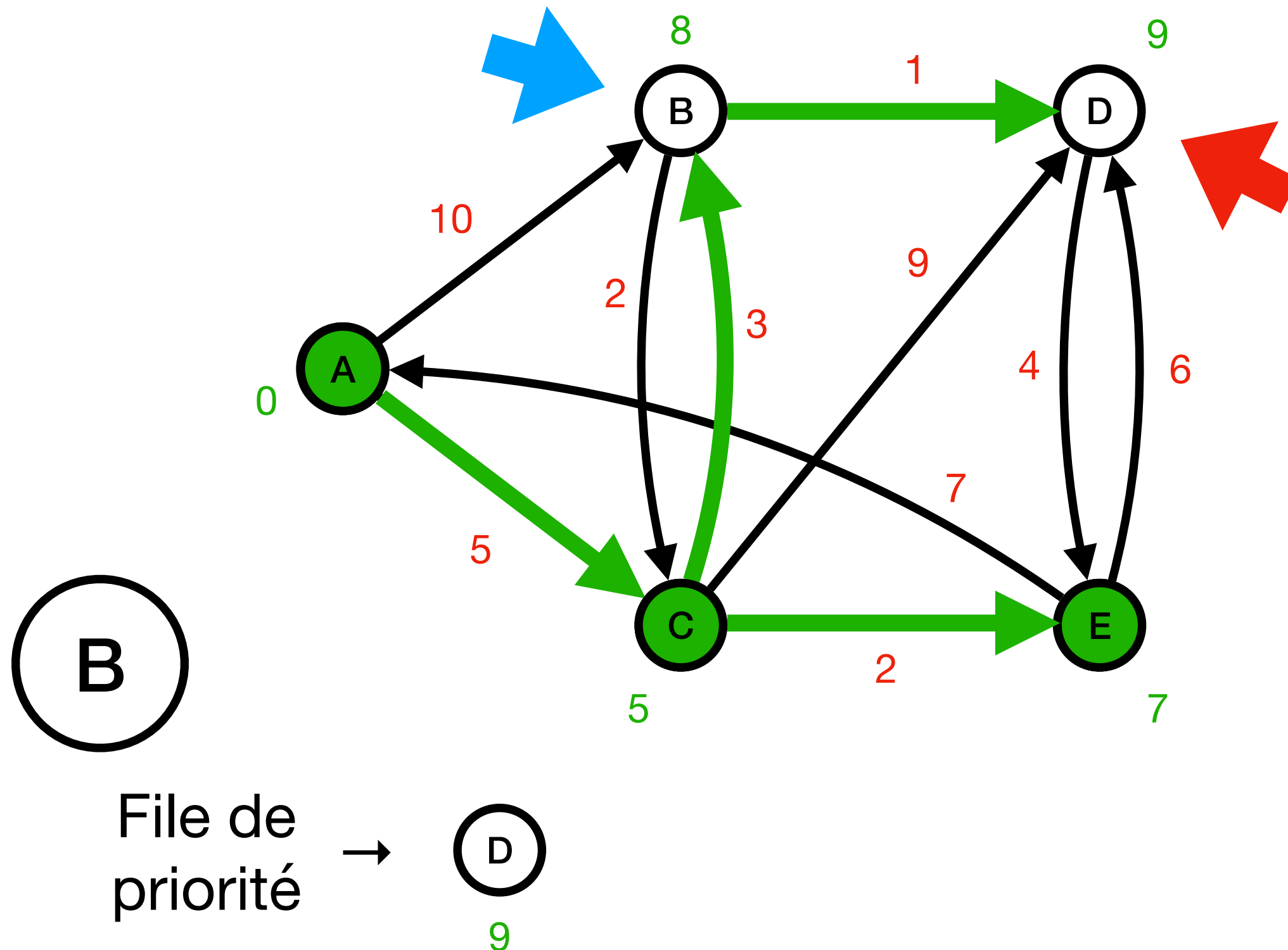
Algorithme de Dijkstra



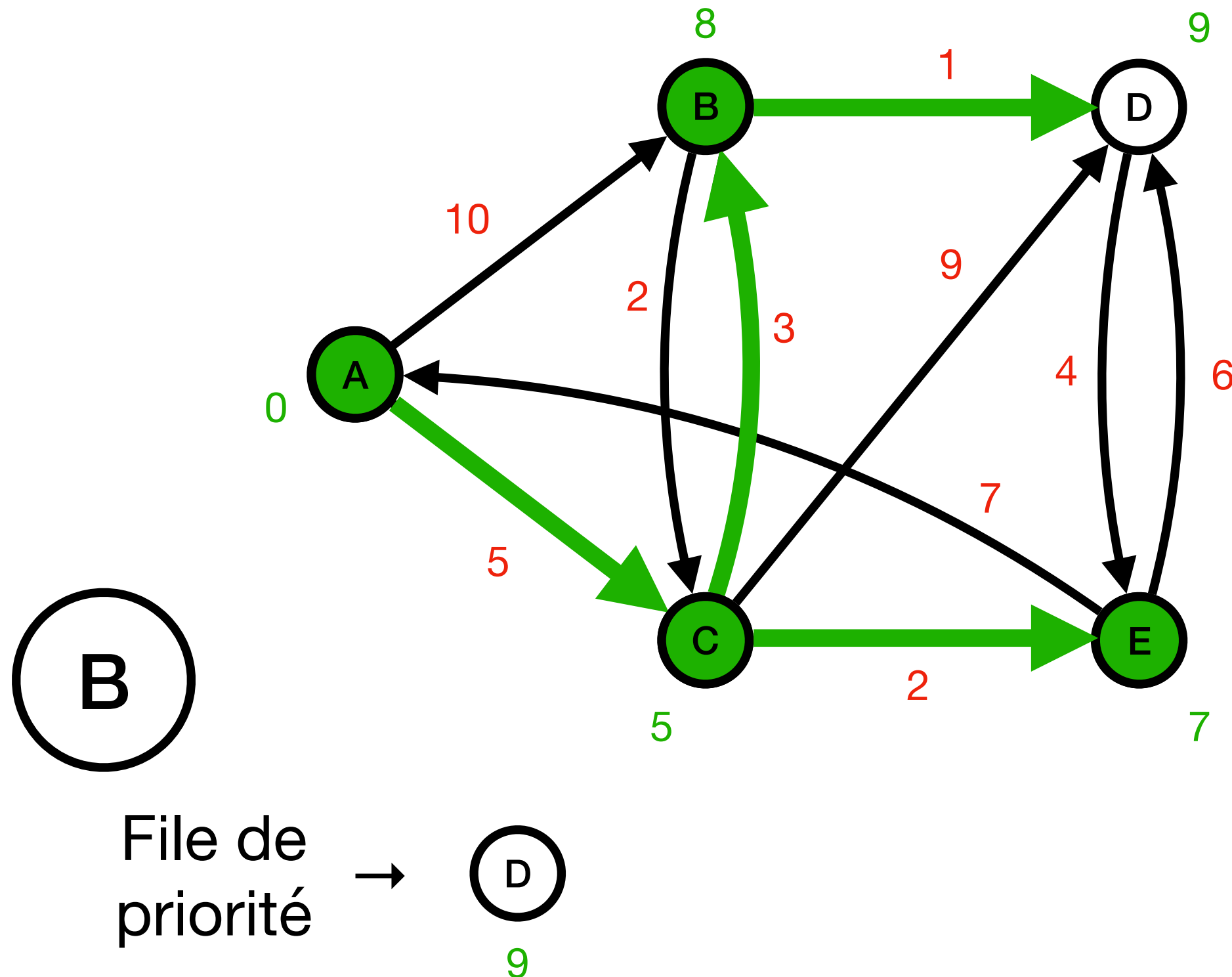
Algorithme de Dijkstra



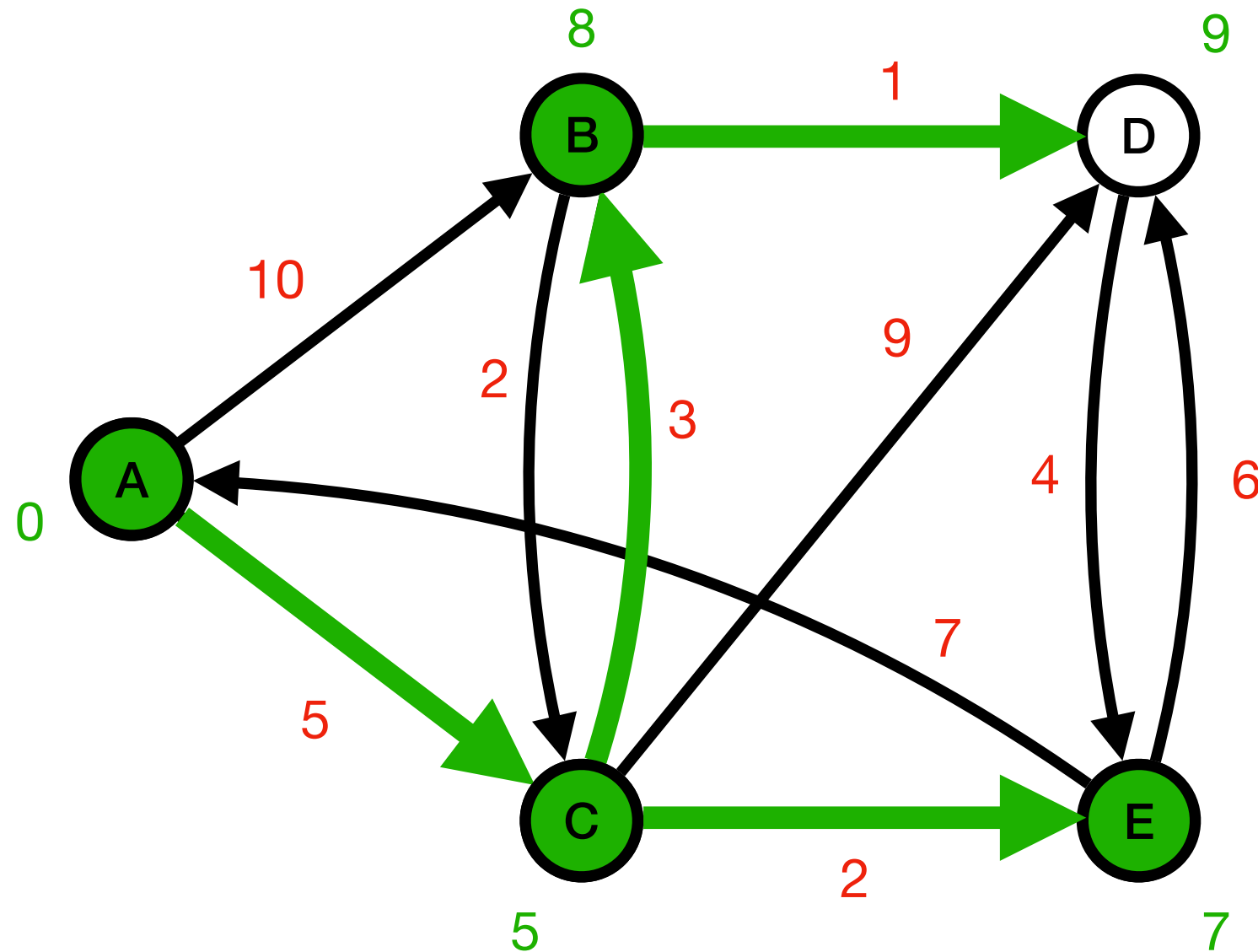
Algorithme de Dijkstra



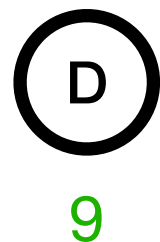
Algorithme de Dijkstra



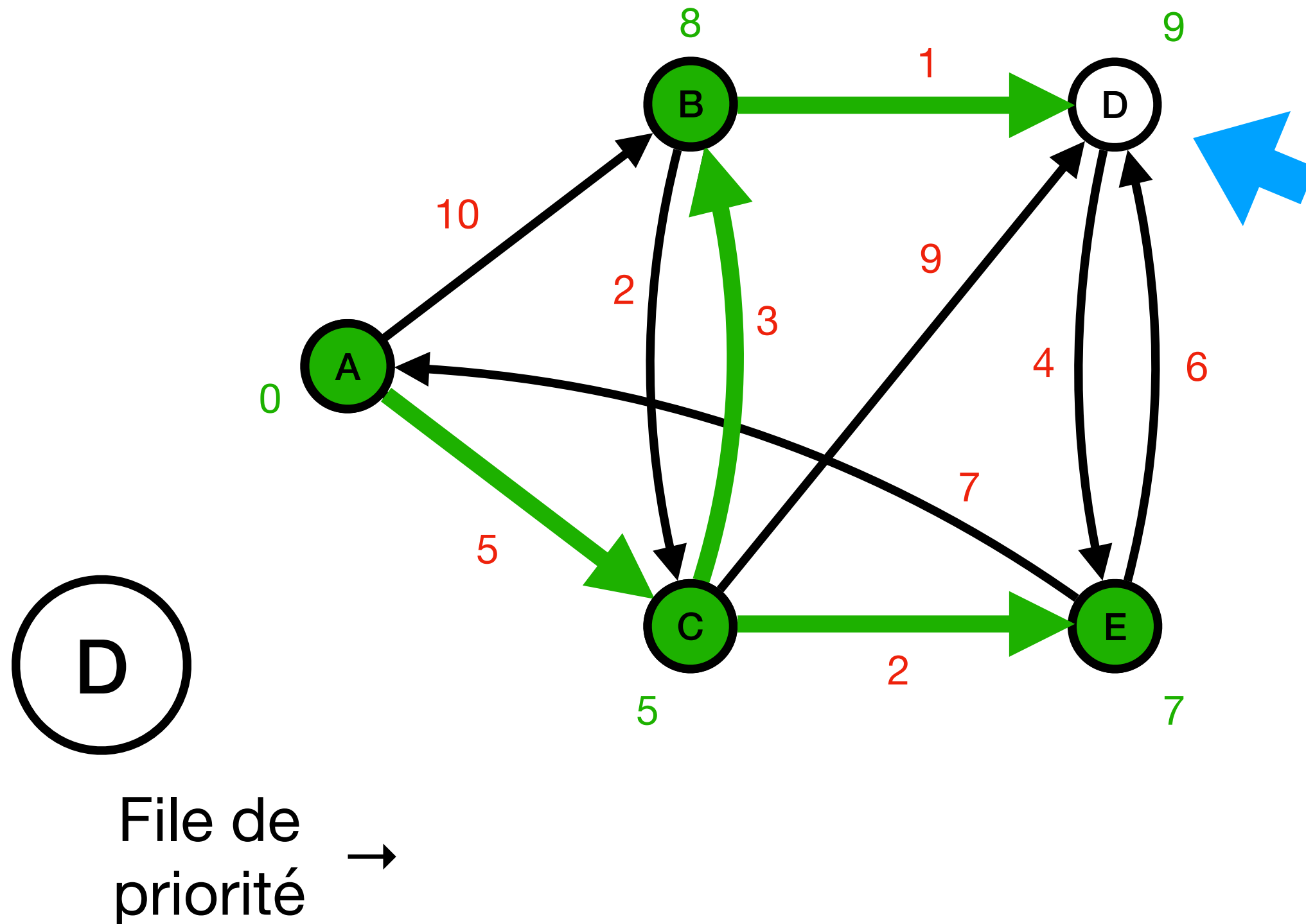
Algorithme de Dijkstra



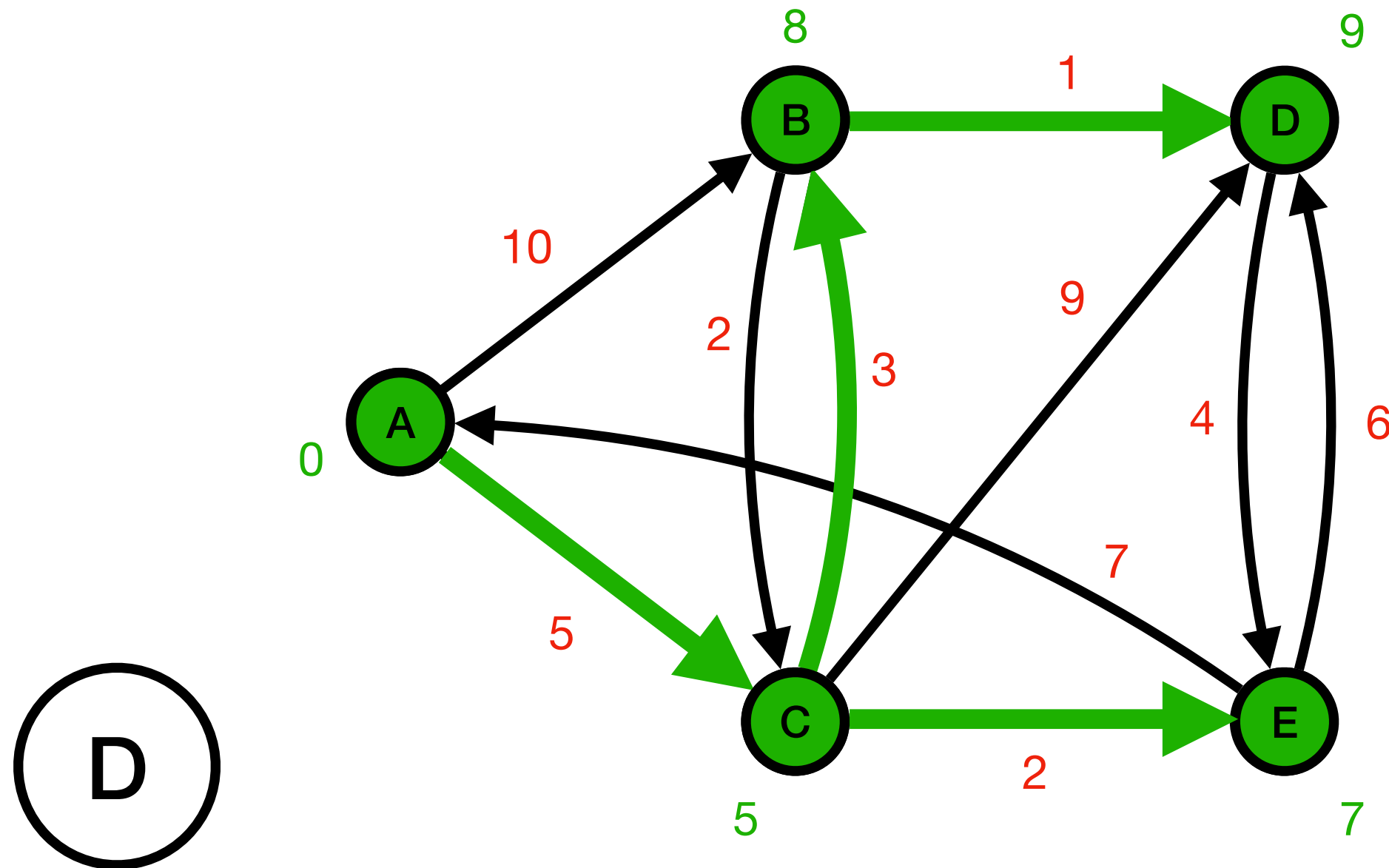
File de
priorité



Algorithme de Dijkstra

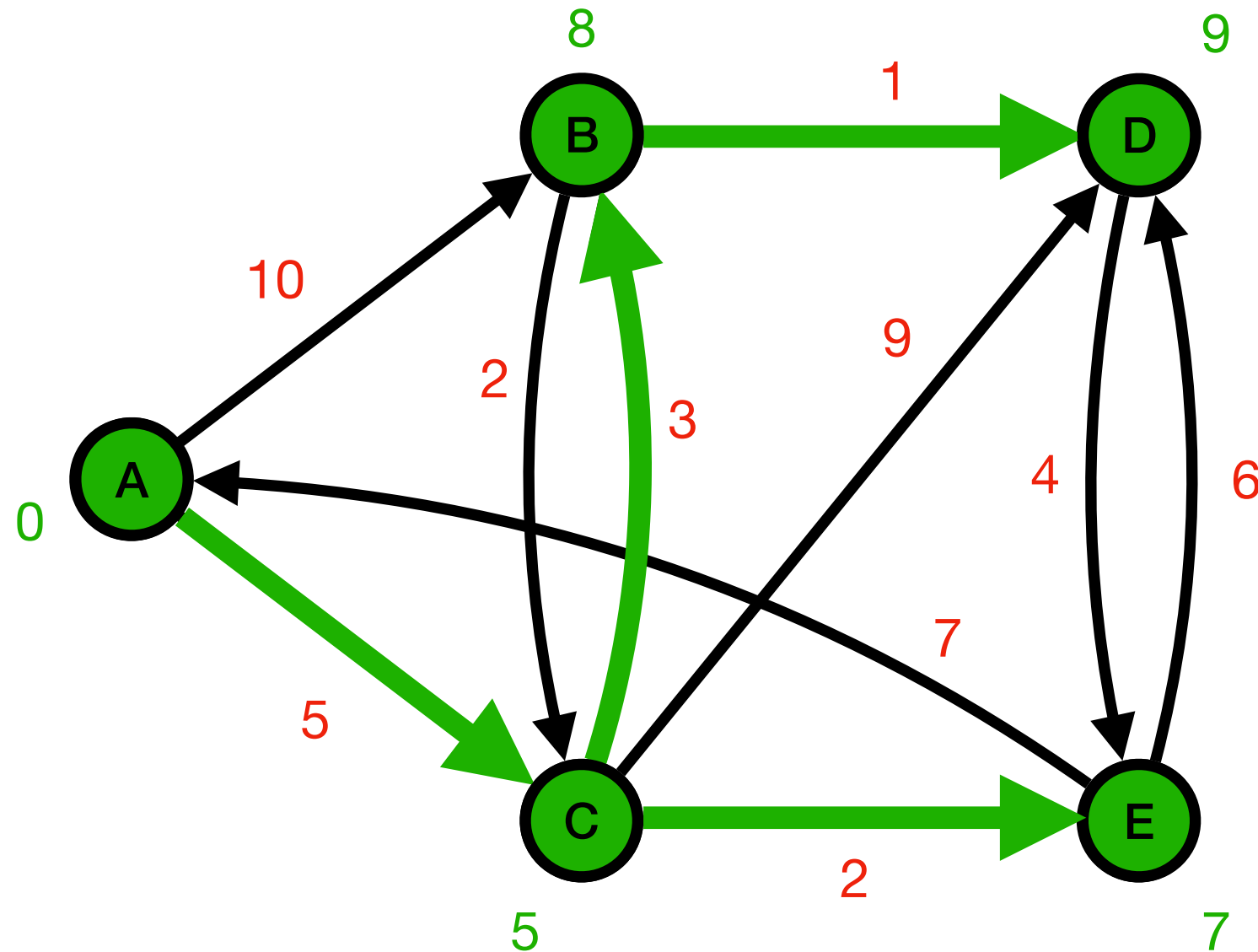


Algorithme de Dijkstra



File de
priorité →

Algorithme de Dijkstra



File de
priorité →

Algorithme de Dijkstra

