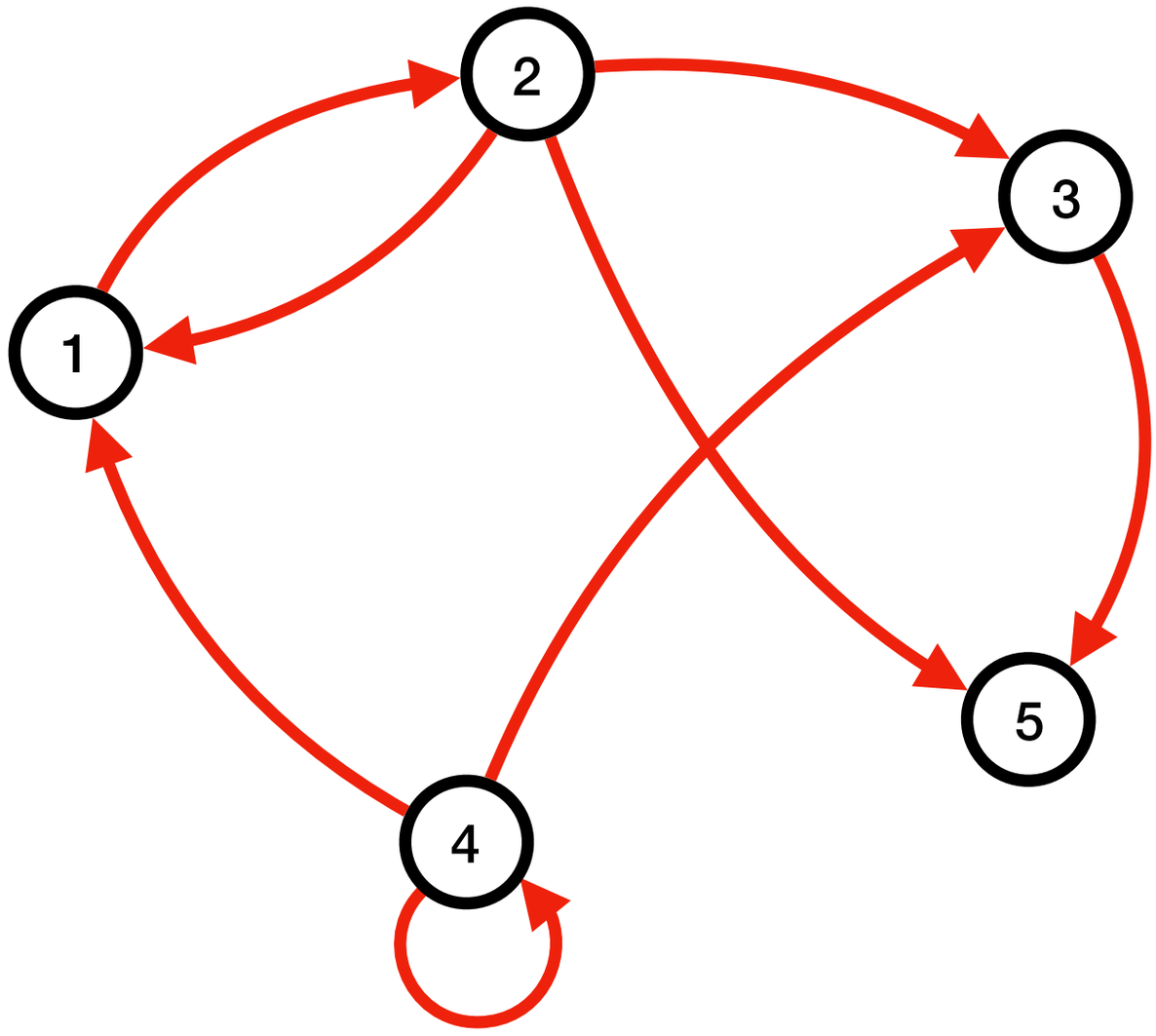
Introduction à l’informatique CM6

Antonio E. Porreca
https://aeporreca.org/teaching

Rappel :
la semaine prochaine
(23 ou 25 octobre)
on fera un TD noté
comme contrôle continu !

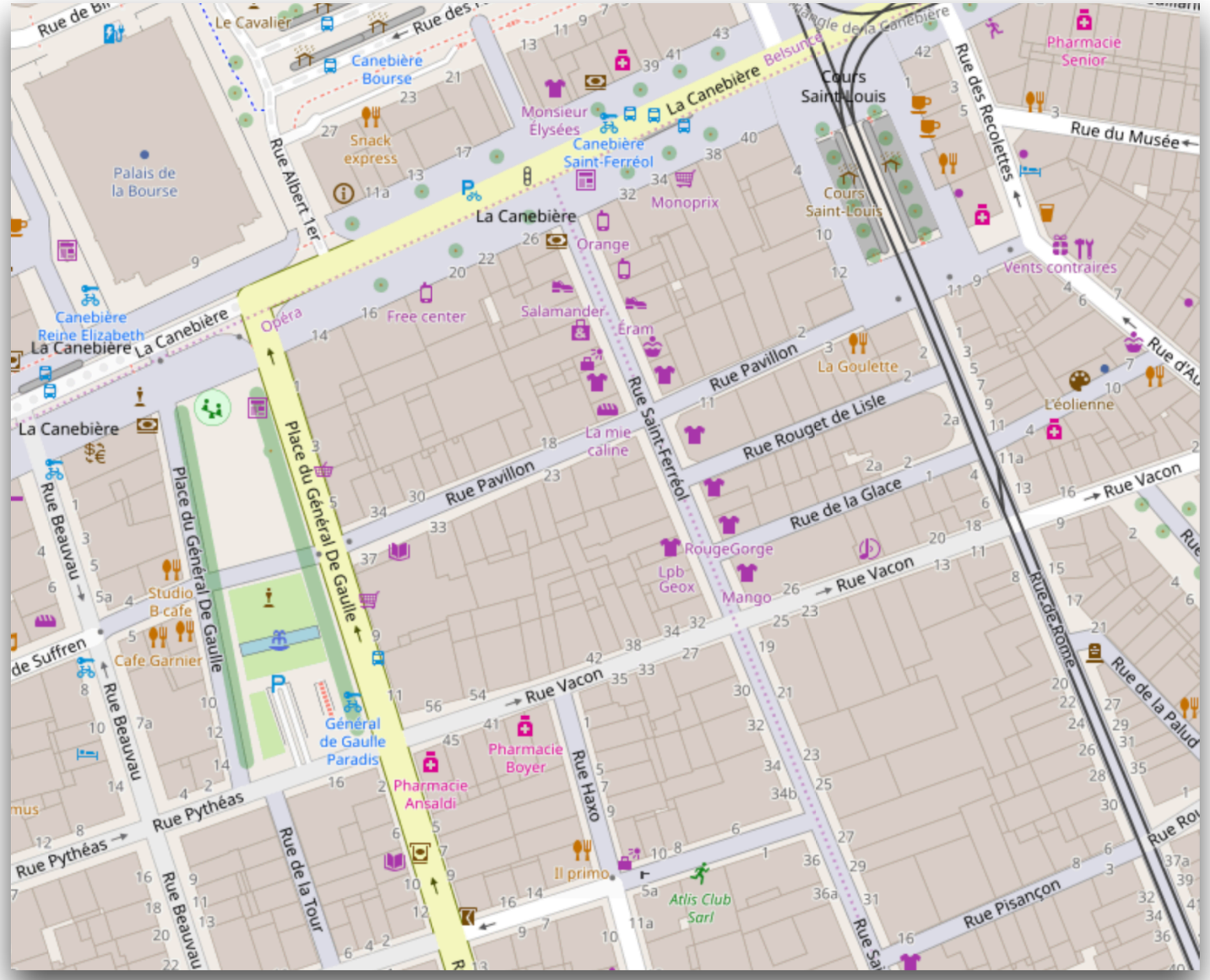
Graphes (orientés)

sommets (nœuds, points)



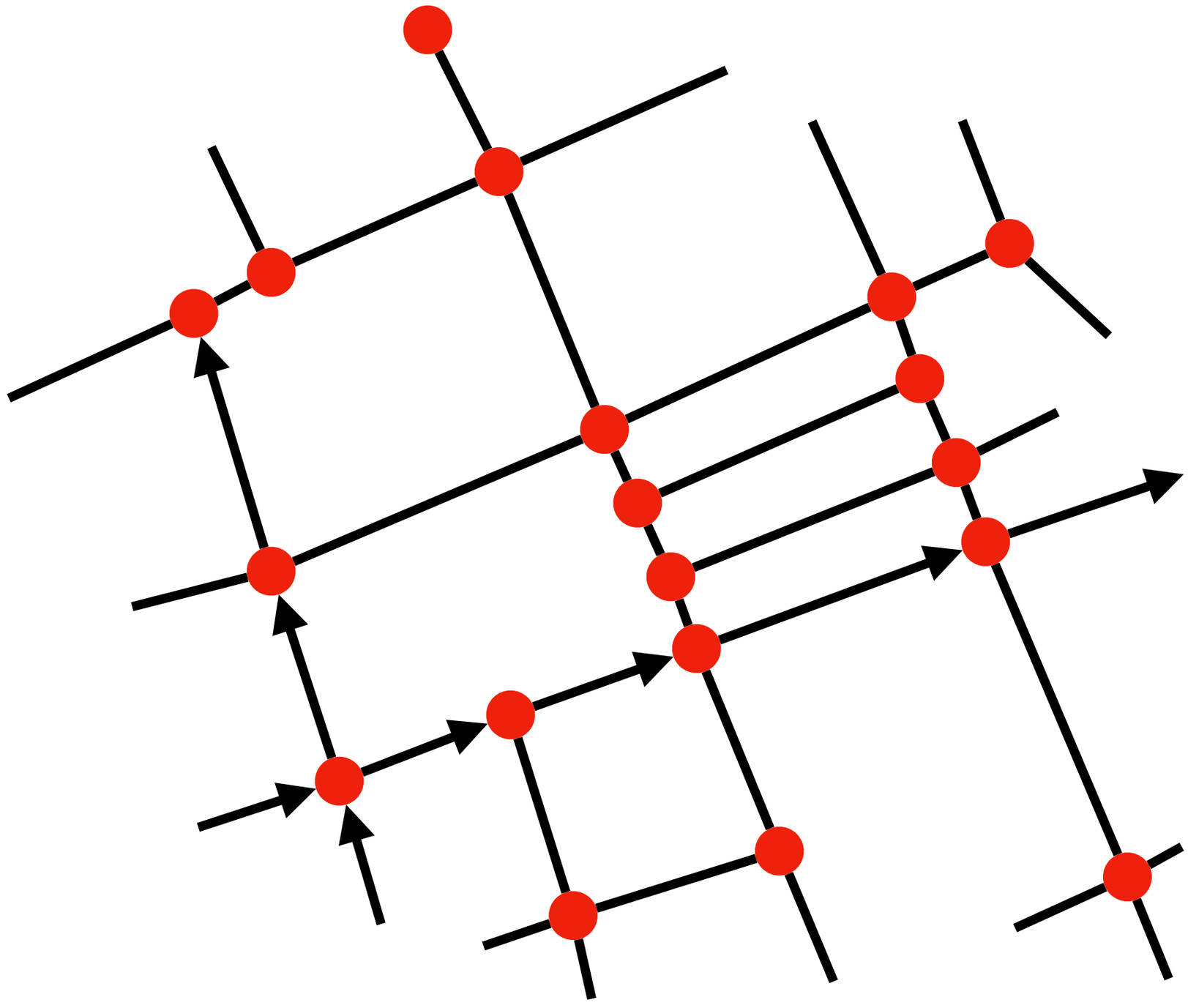
arcs

Plan d’une ville

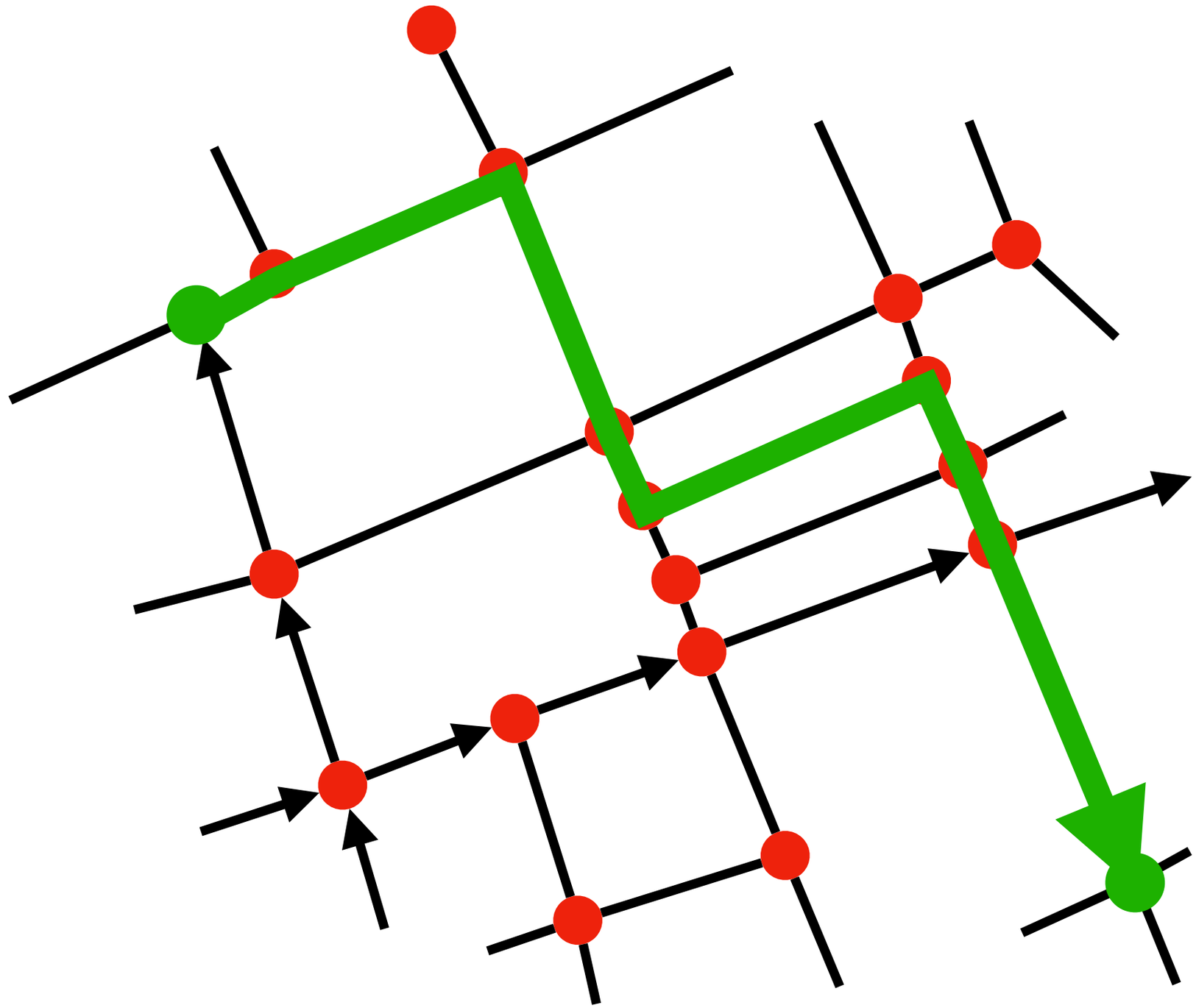


Plan d’une ville

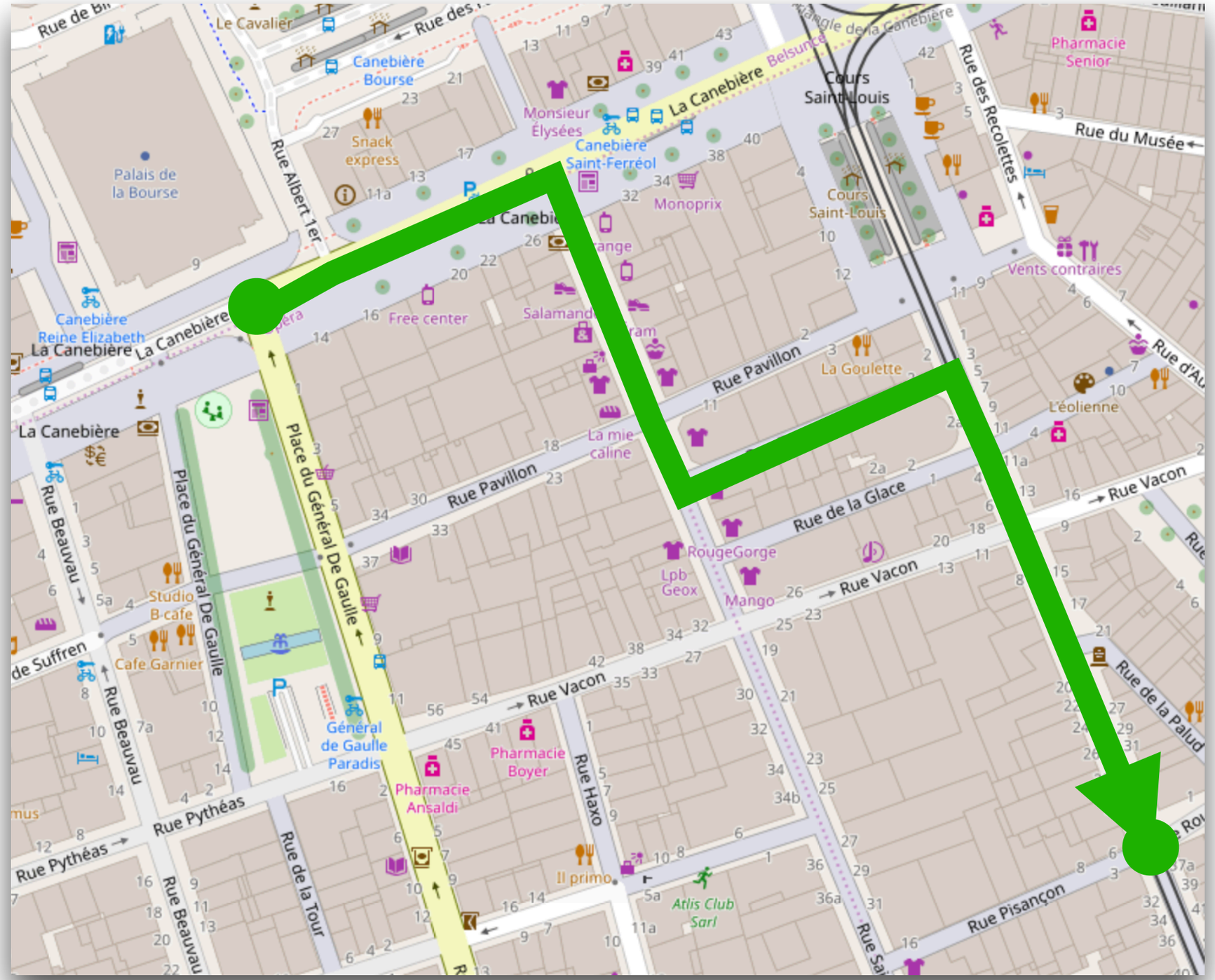
Plan d’une ville



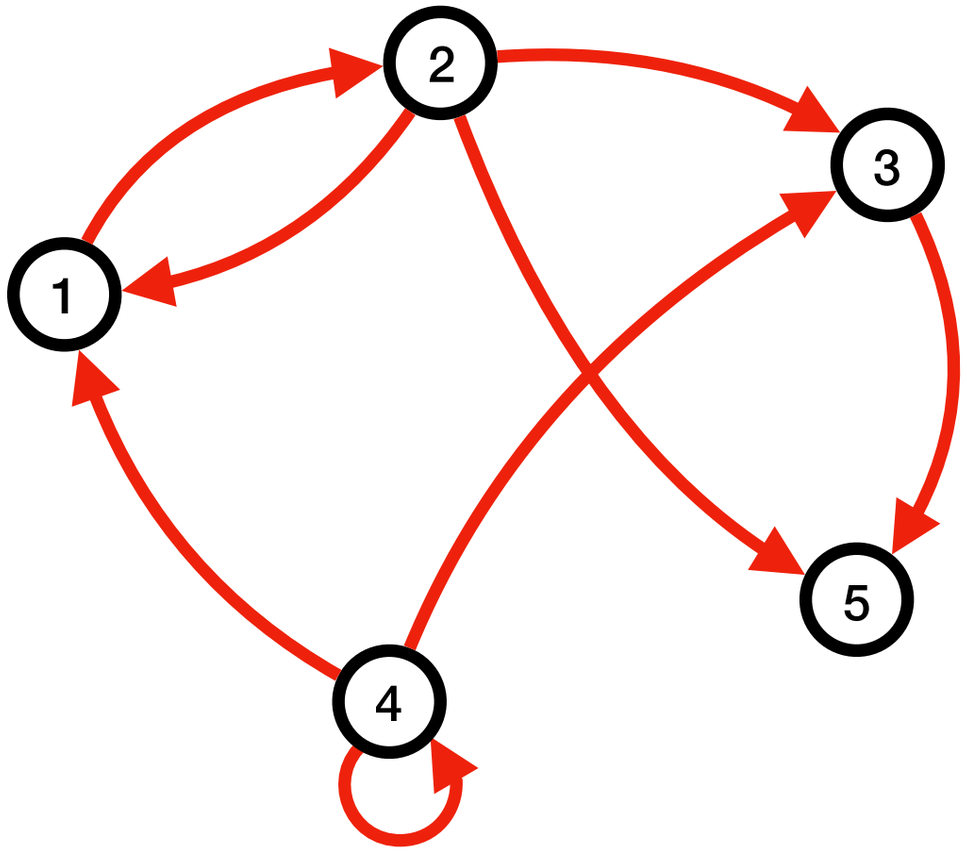
Plan d’une ville



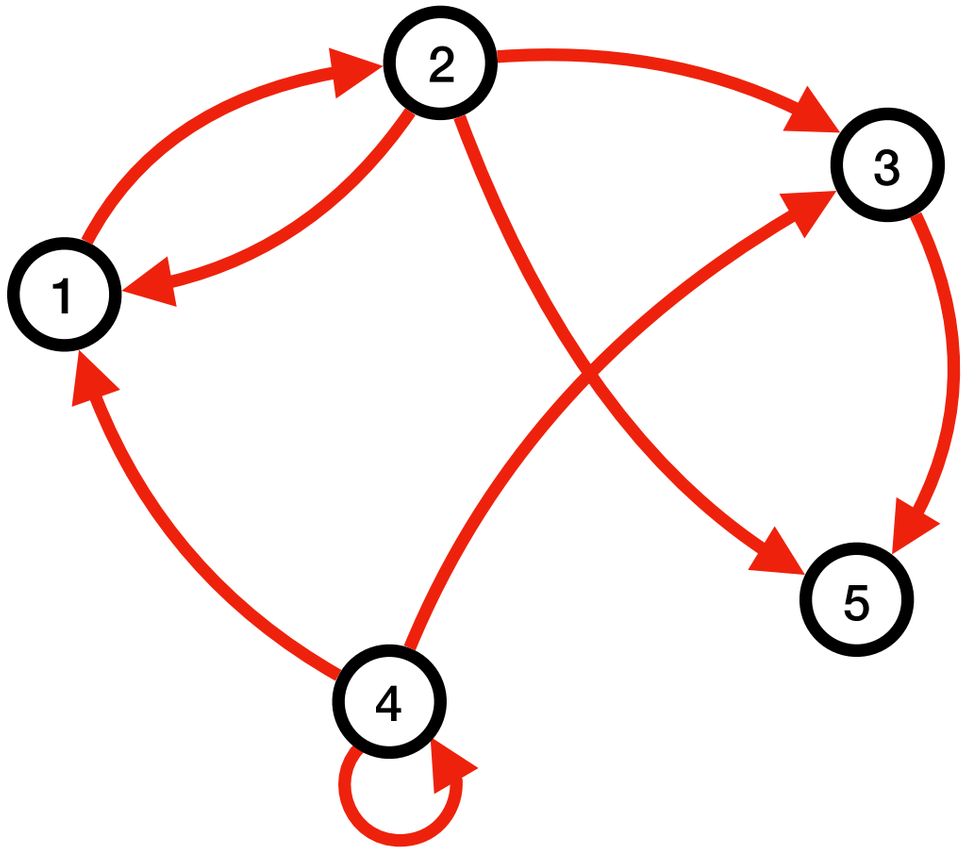
Plan d’une ville



Matrices d’adjacence

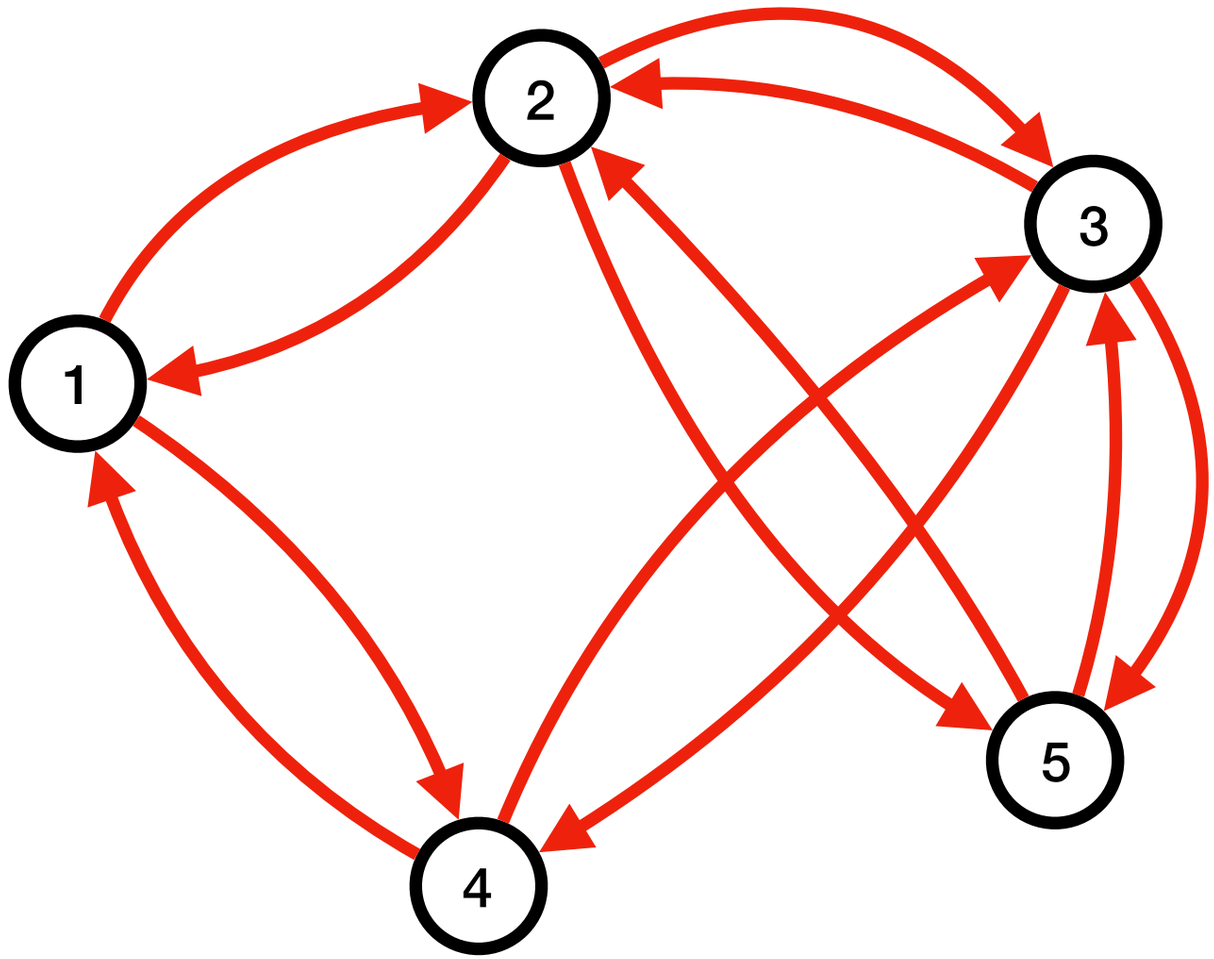


Matrices d’adjacence

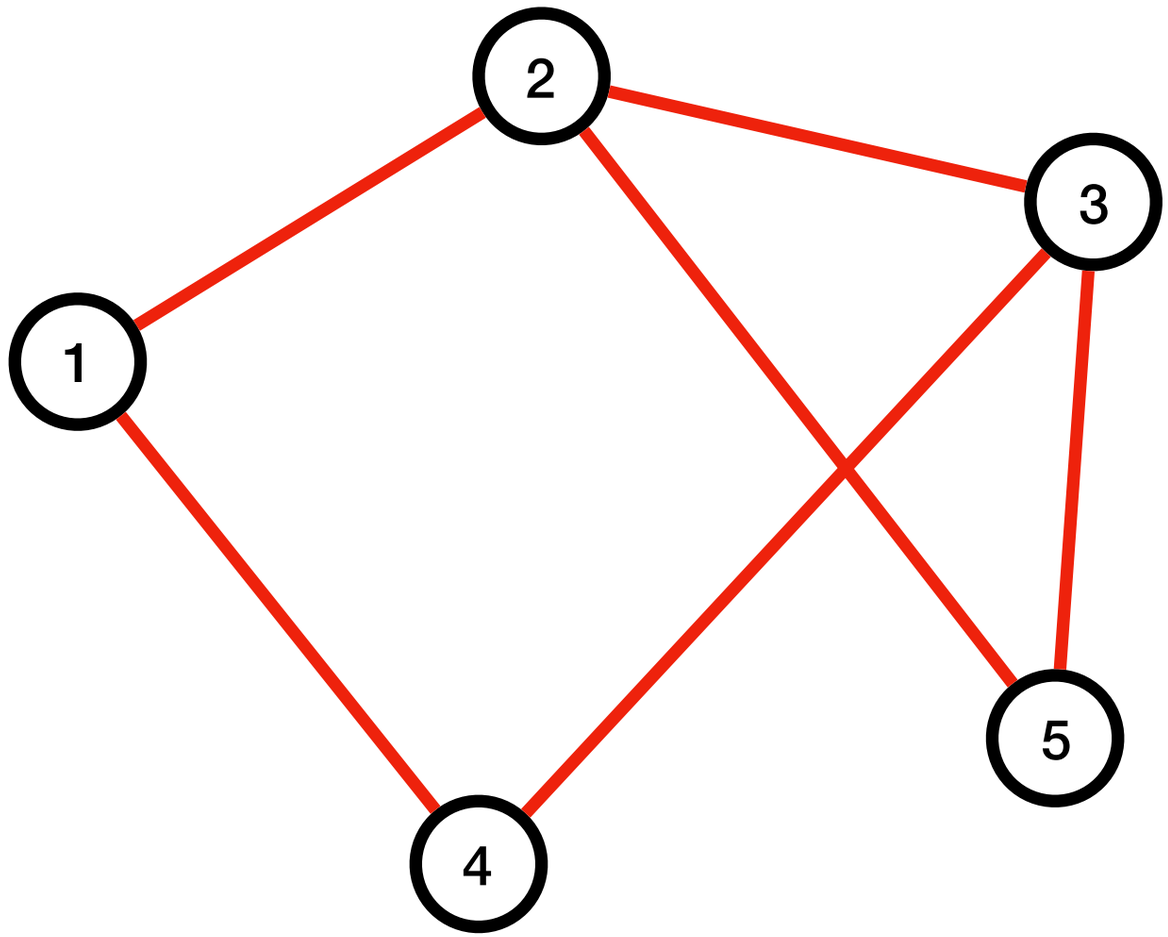


	1	2	3	4	5
1	0	**1**	0	0	0
2	**1**	0	**1**	0	**1**
3	0	0	0	0	**1**
4	**1**	0	**1**	**1**	0
5	0	0	0	0	0

Graphes non orientés

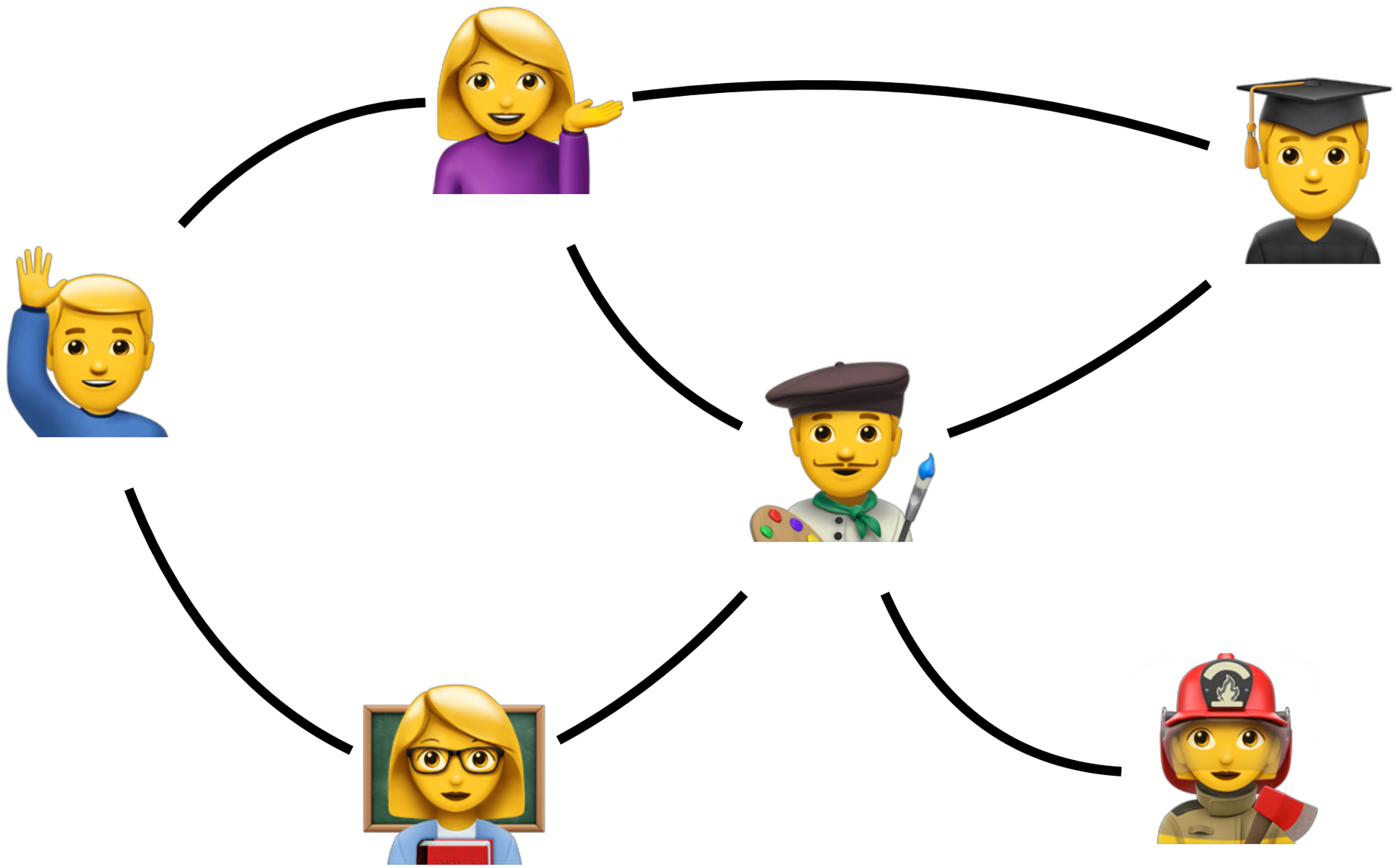


Graphes non orientés



arêtes

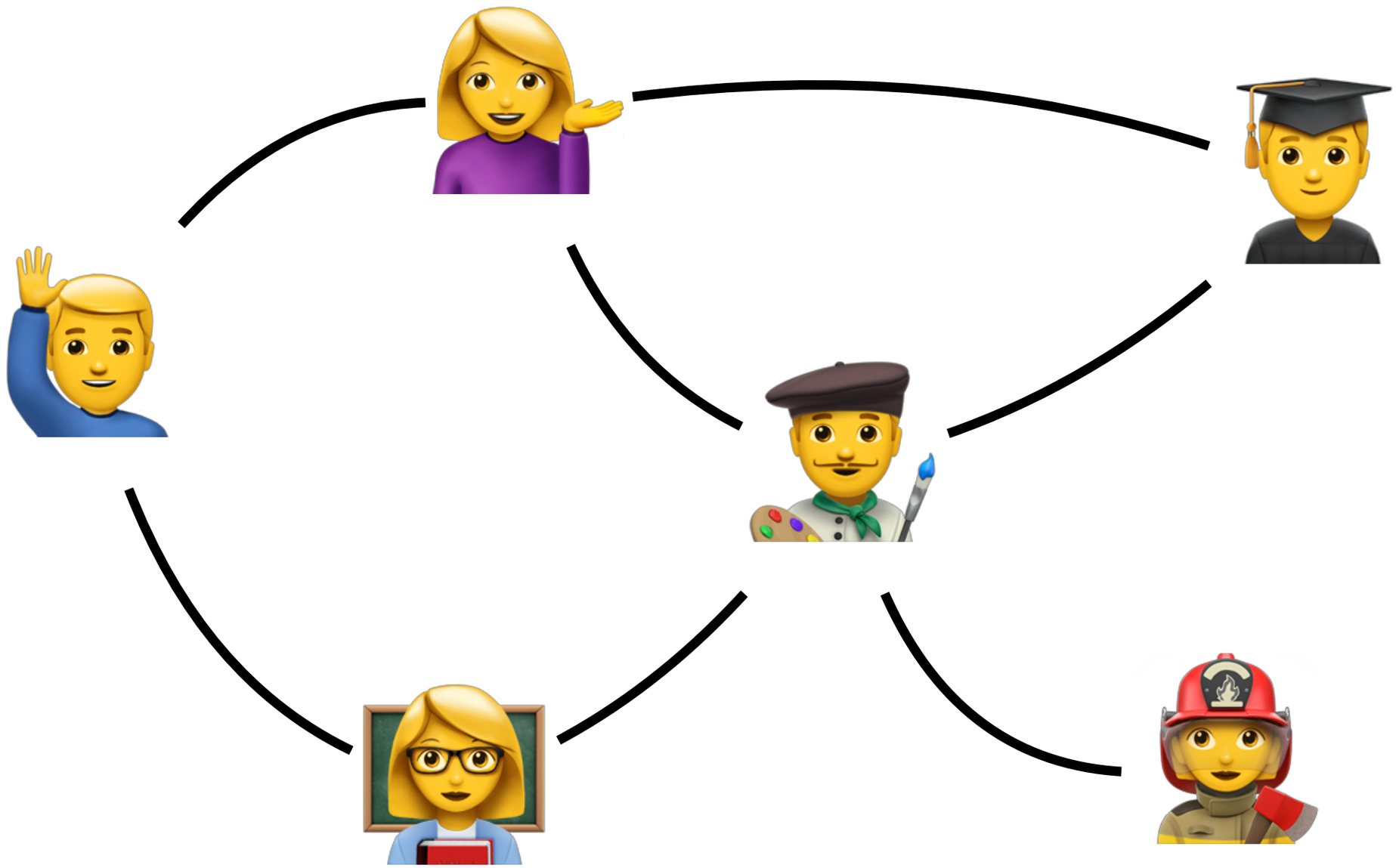
Réseau social (symétrique)



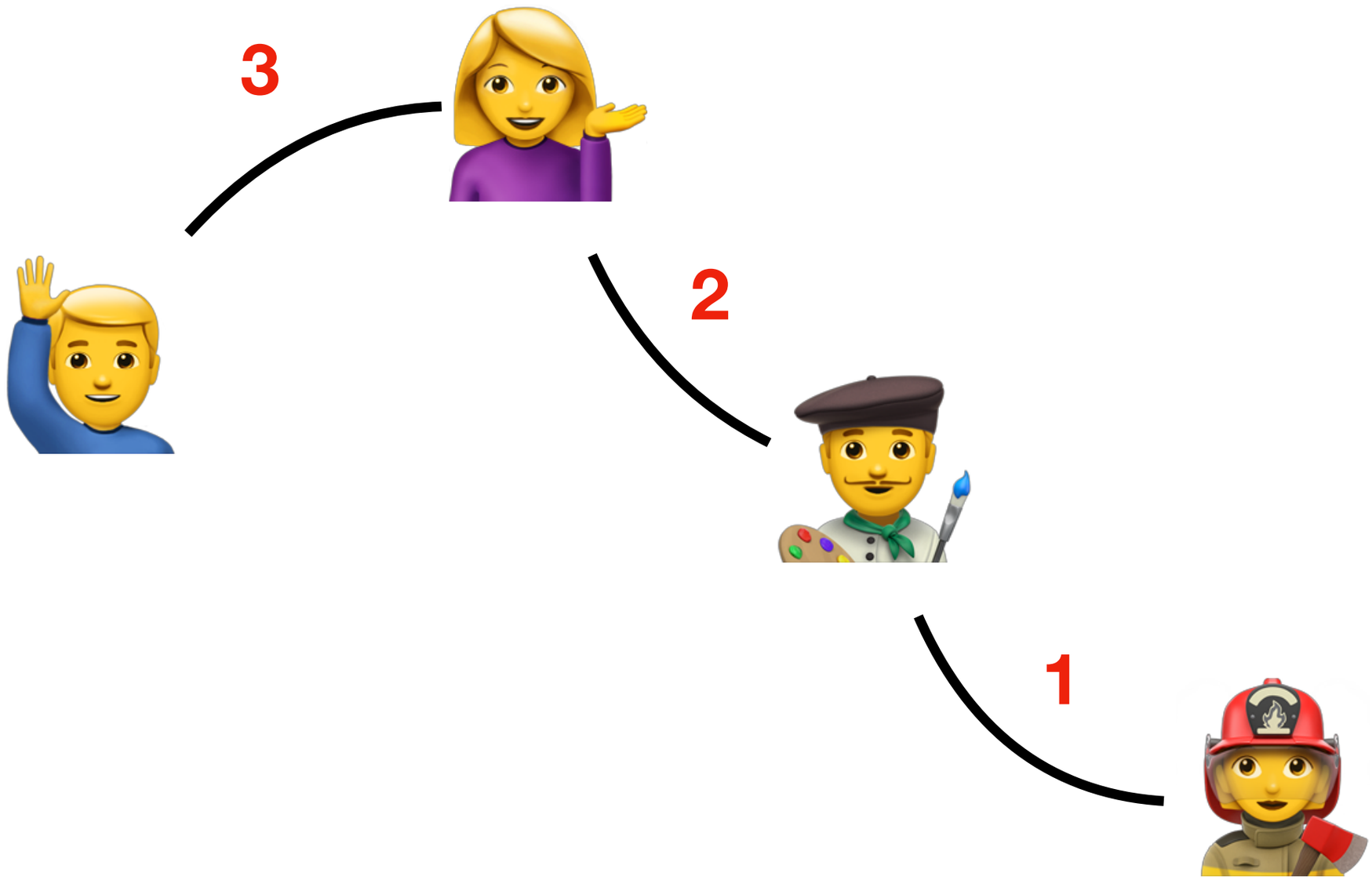
Each person in the world (at least among the 1.59 billion people active on Facebook) is connected to every other person by an average of three and a half other people.

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

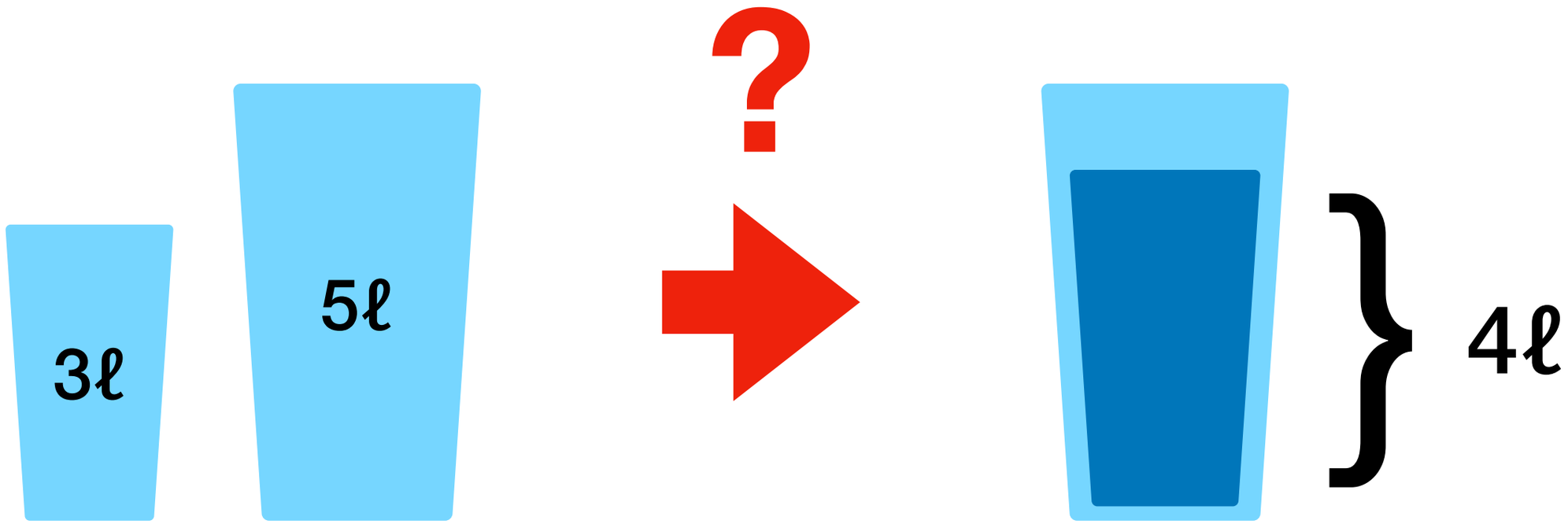
Diamètre d’un graphe



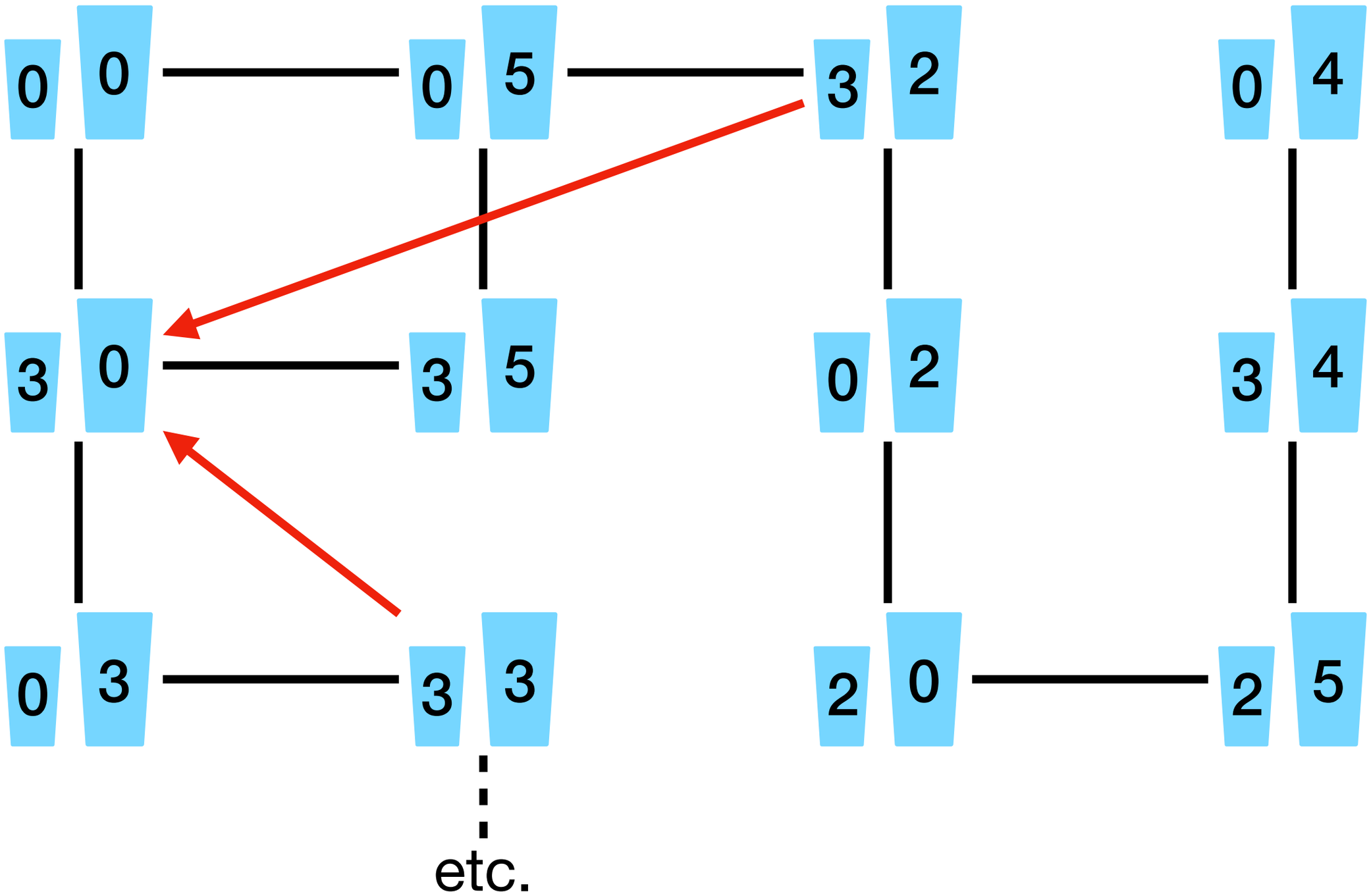
Diamètre d’un graphe



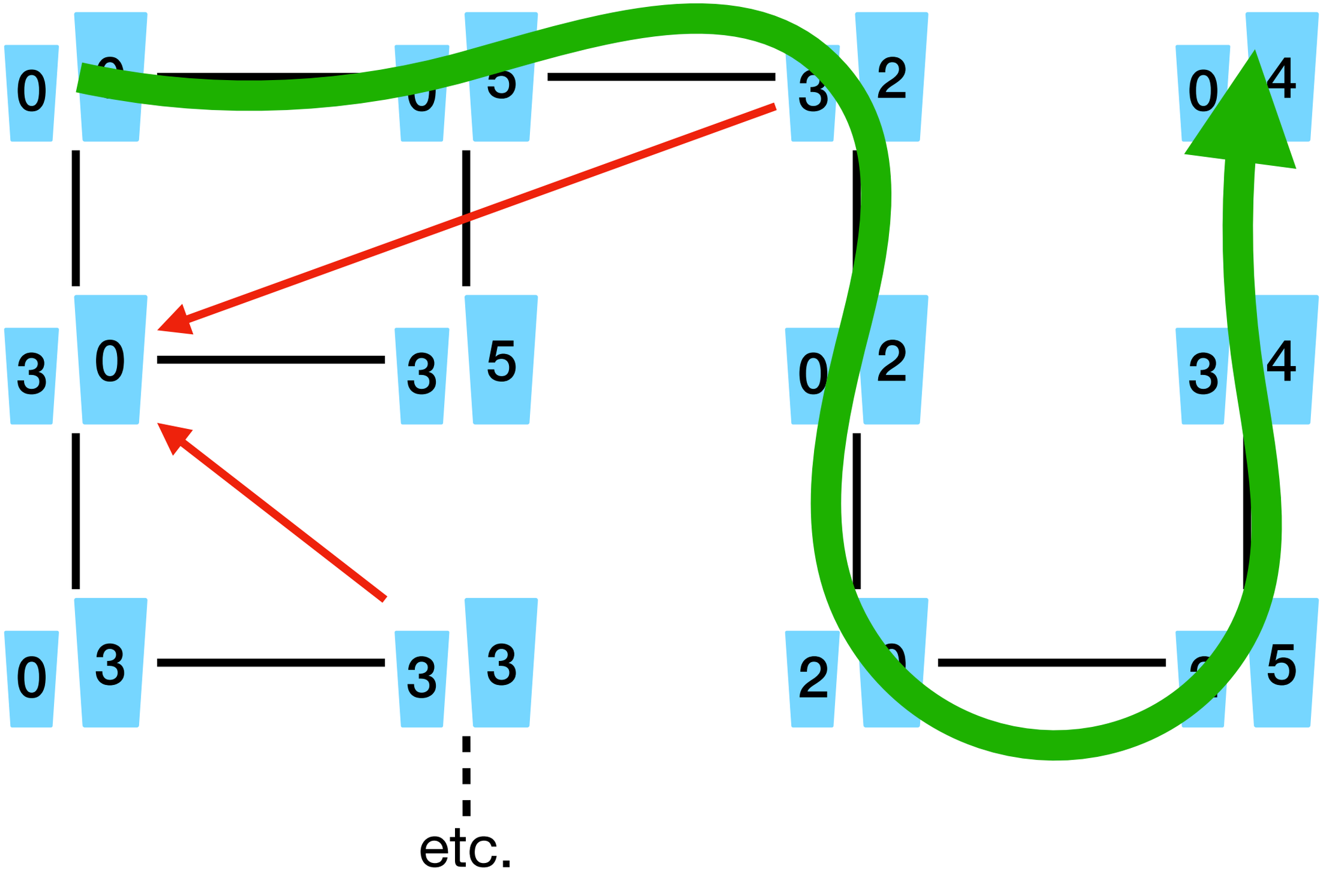
Enigme des récipients



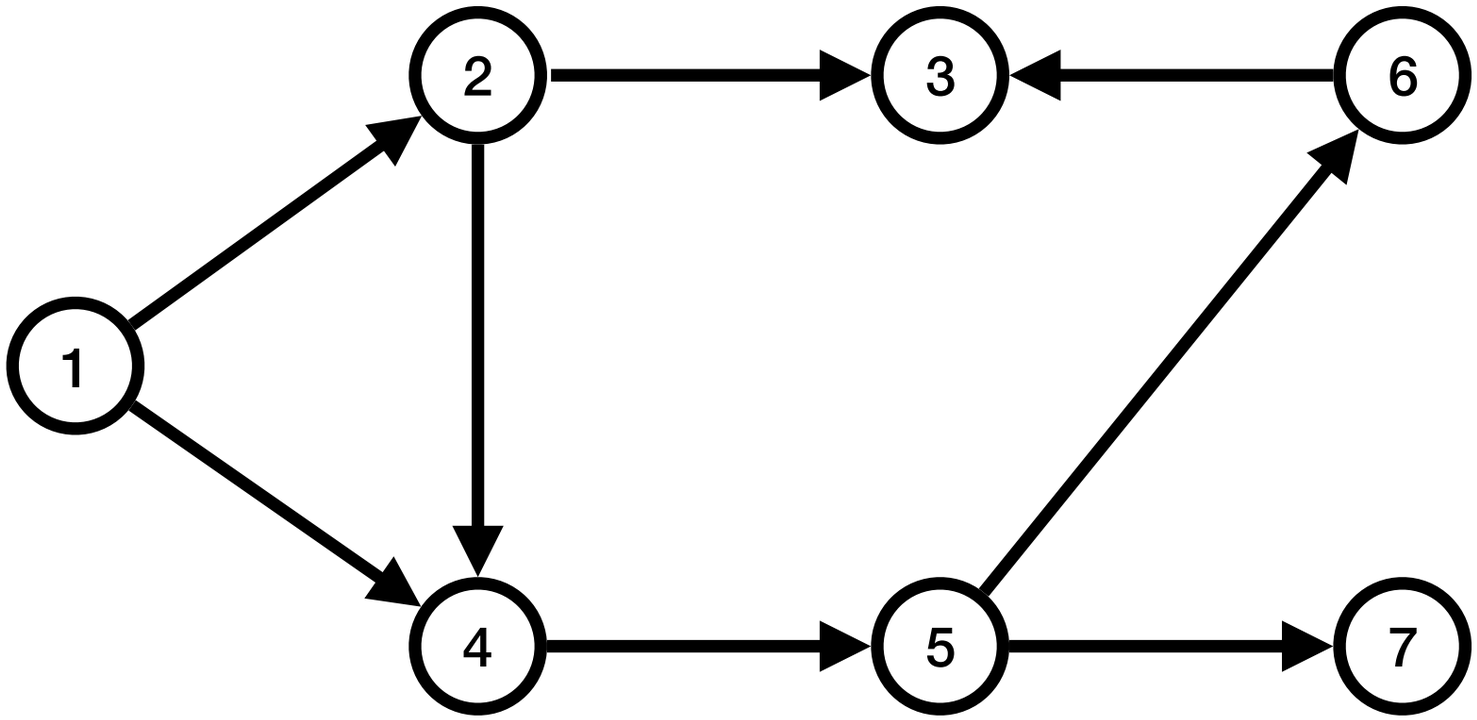
Graphe des configurations



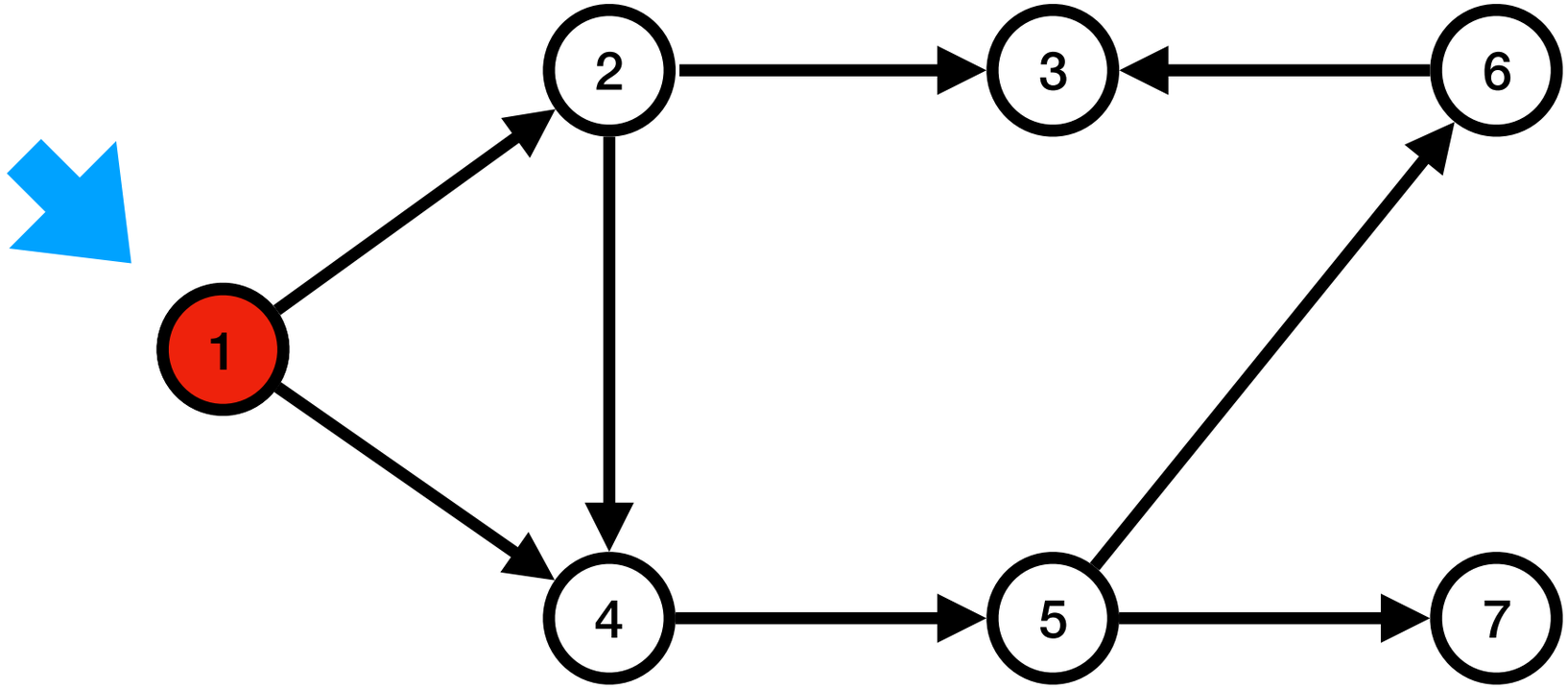
Graphe des configurations



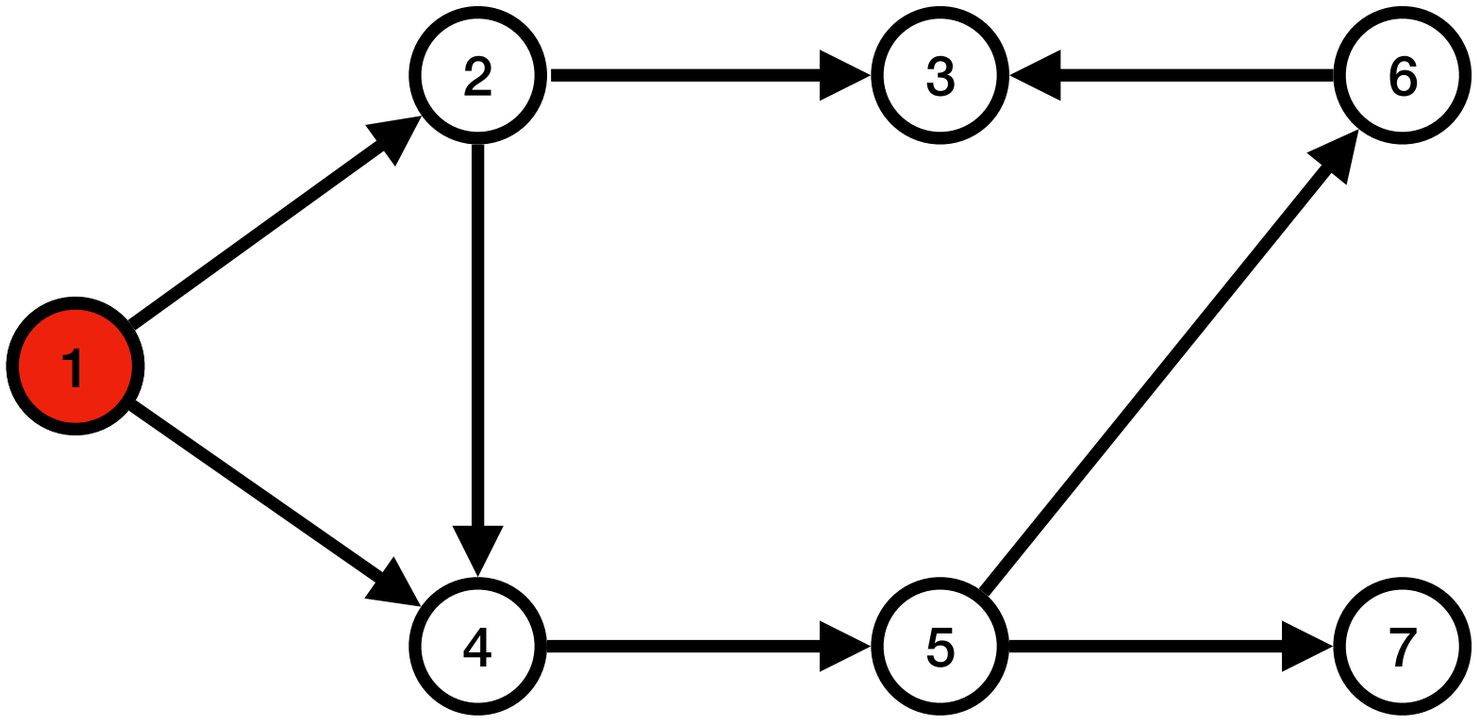
Parcours en largeur



Parcours en largeur

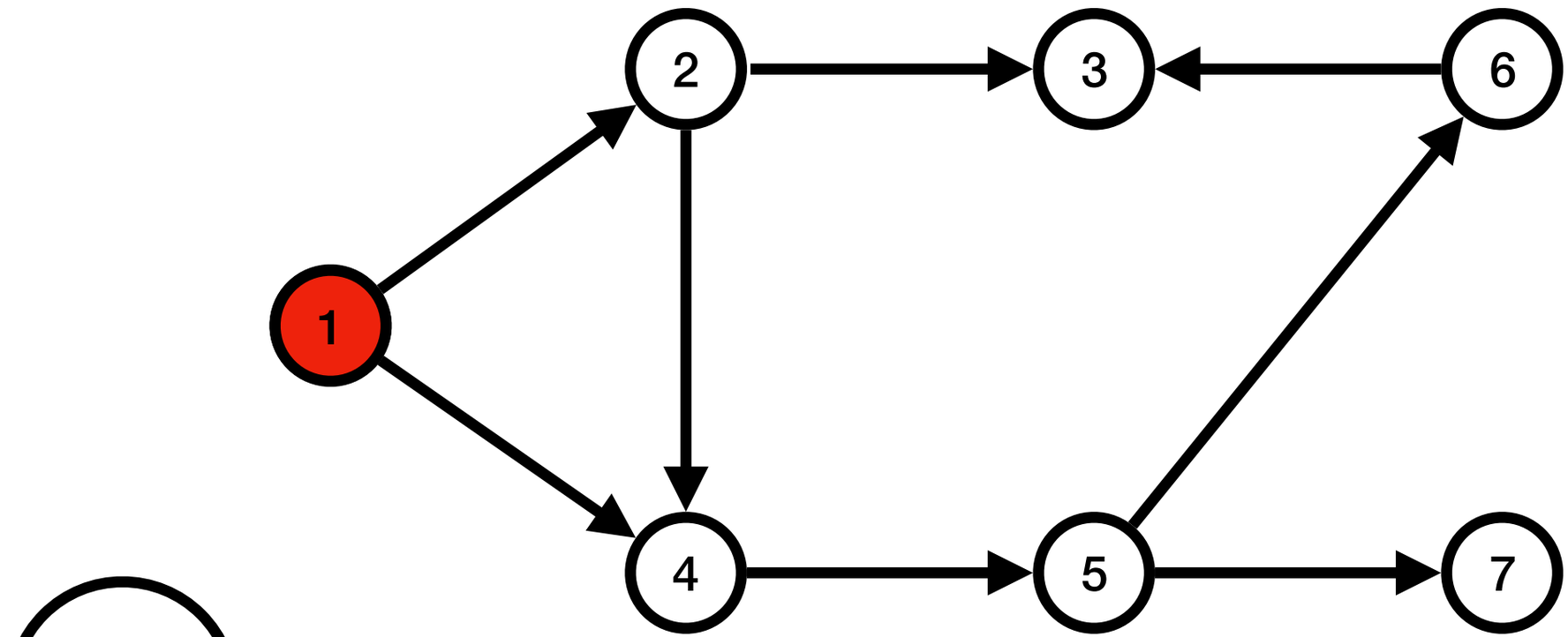


Parcours en largeur



File → 1

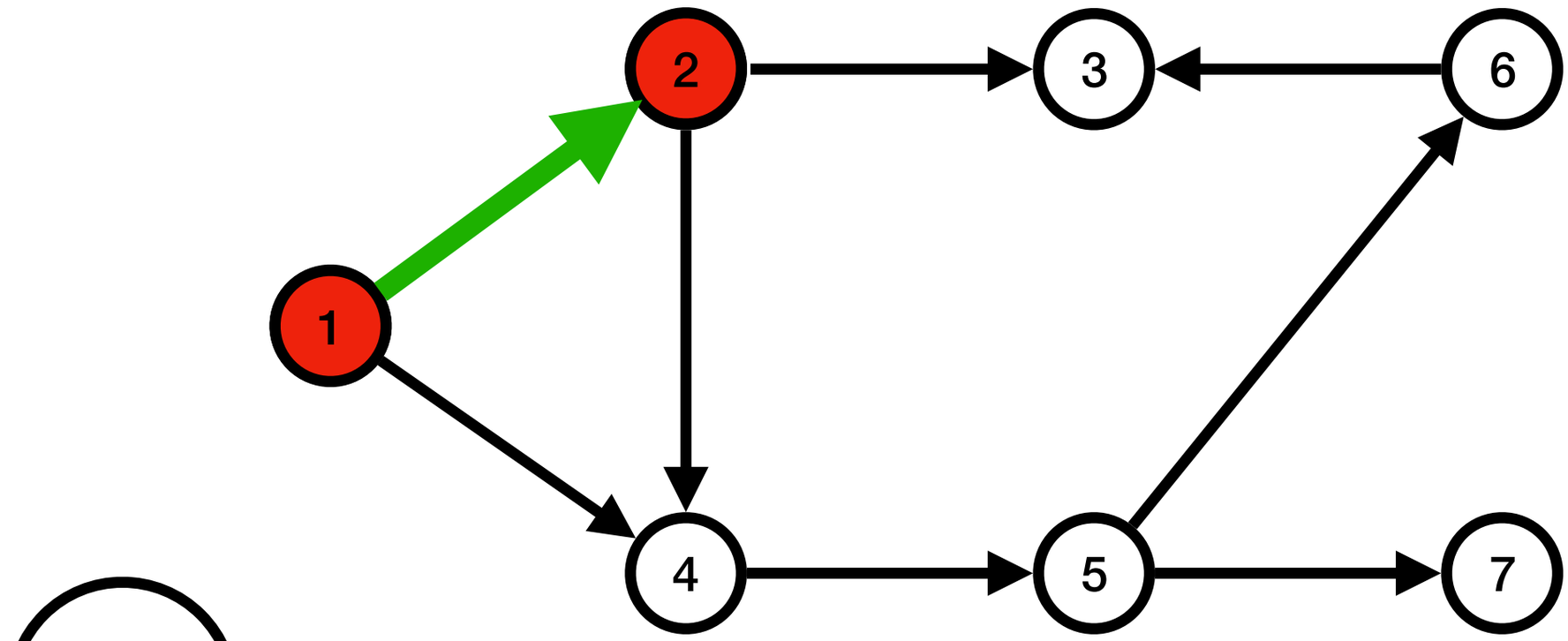
Parcours en largeur

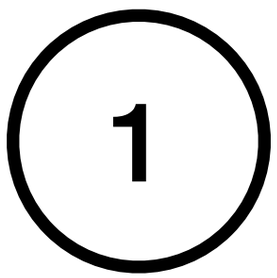


1

File →

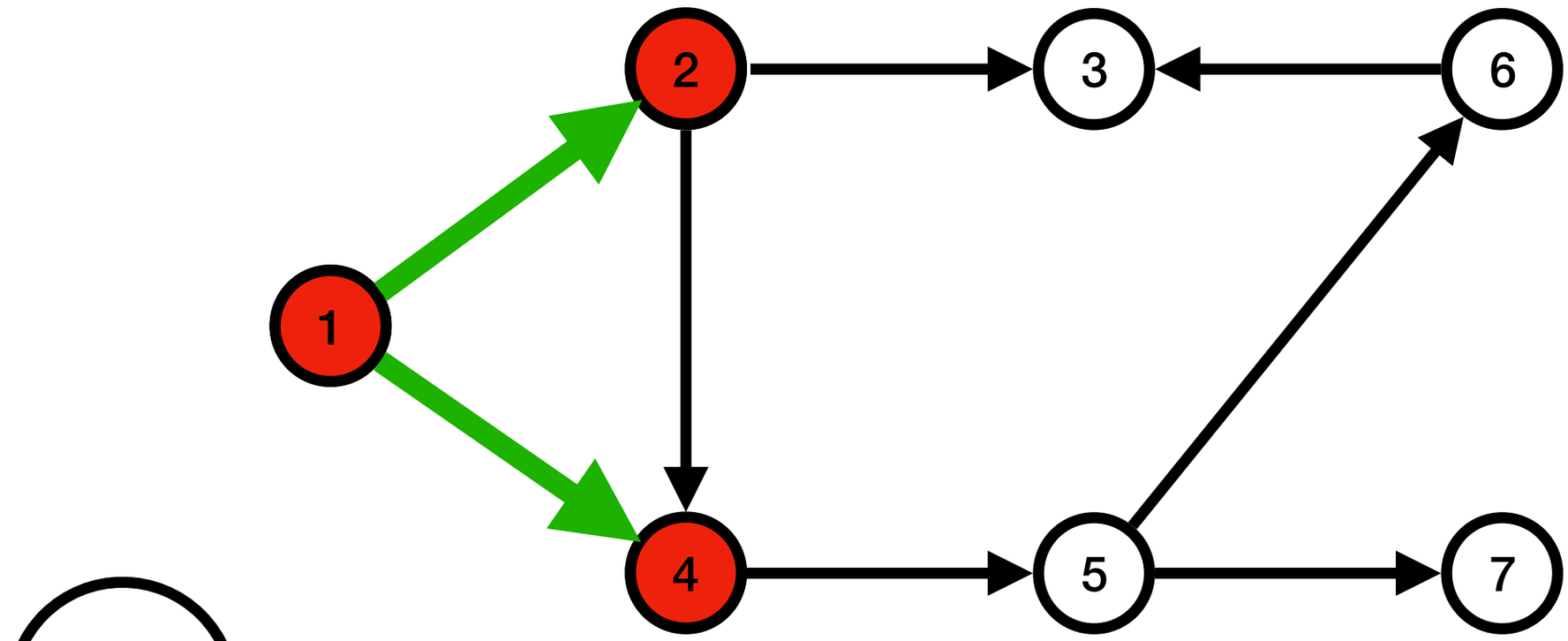
Parcours en largeur

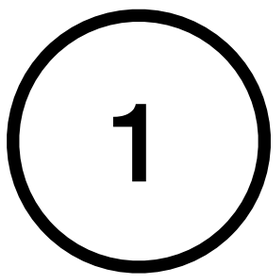




File → 

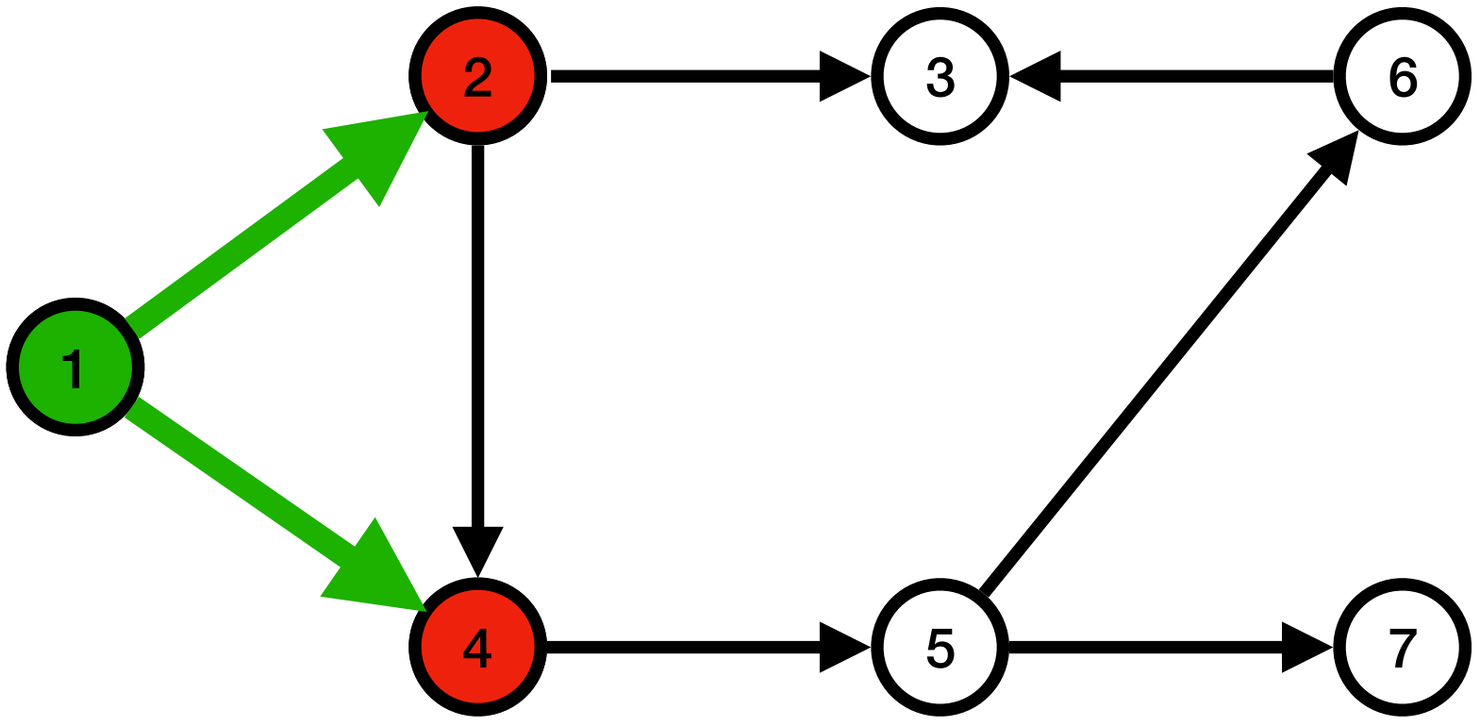
Parcours en largeur





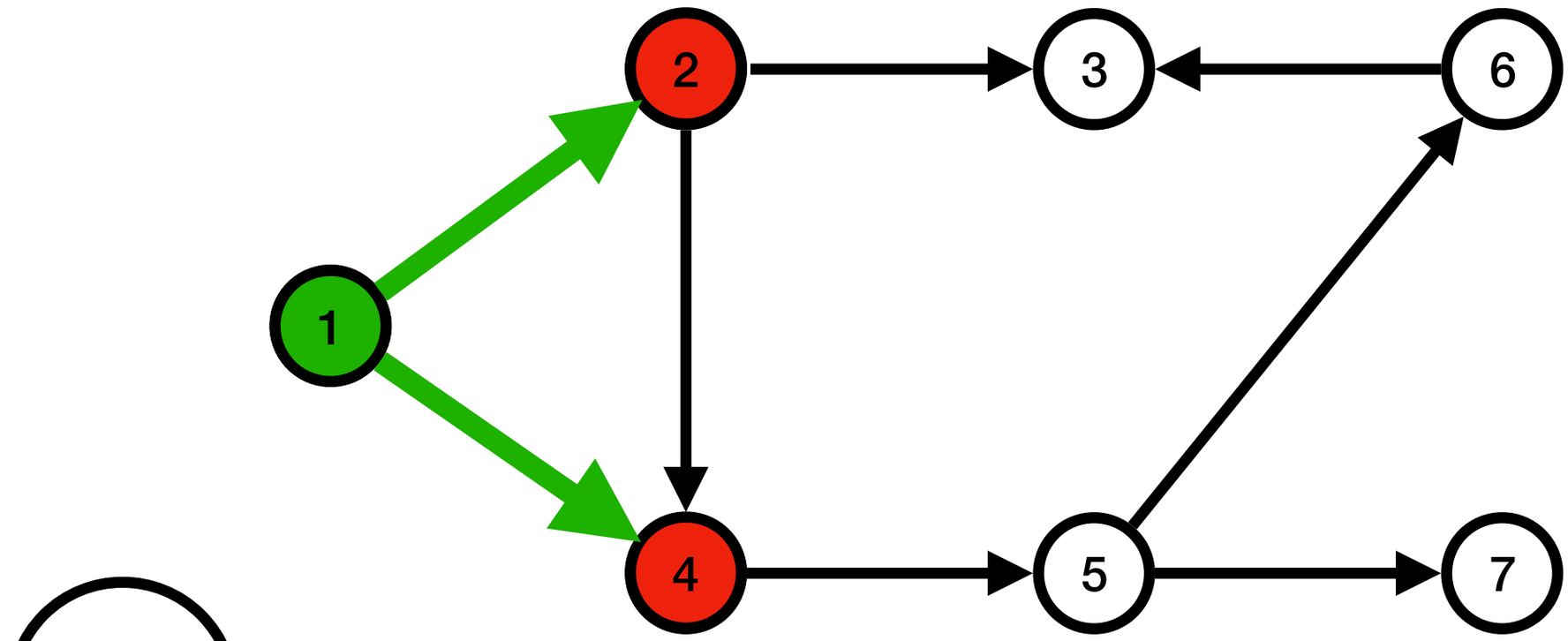
File → 2 4

Parcours en largeur



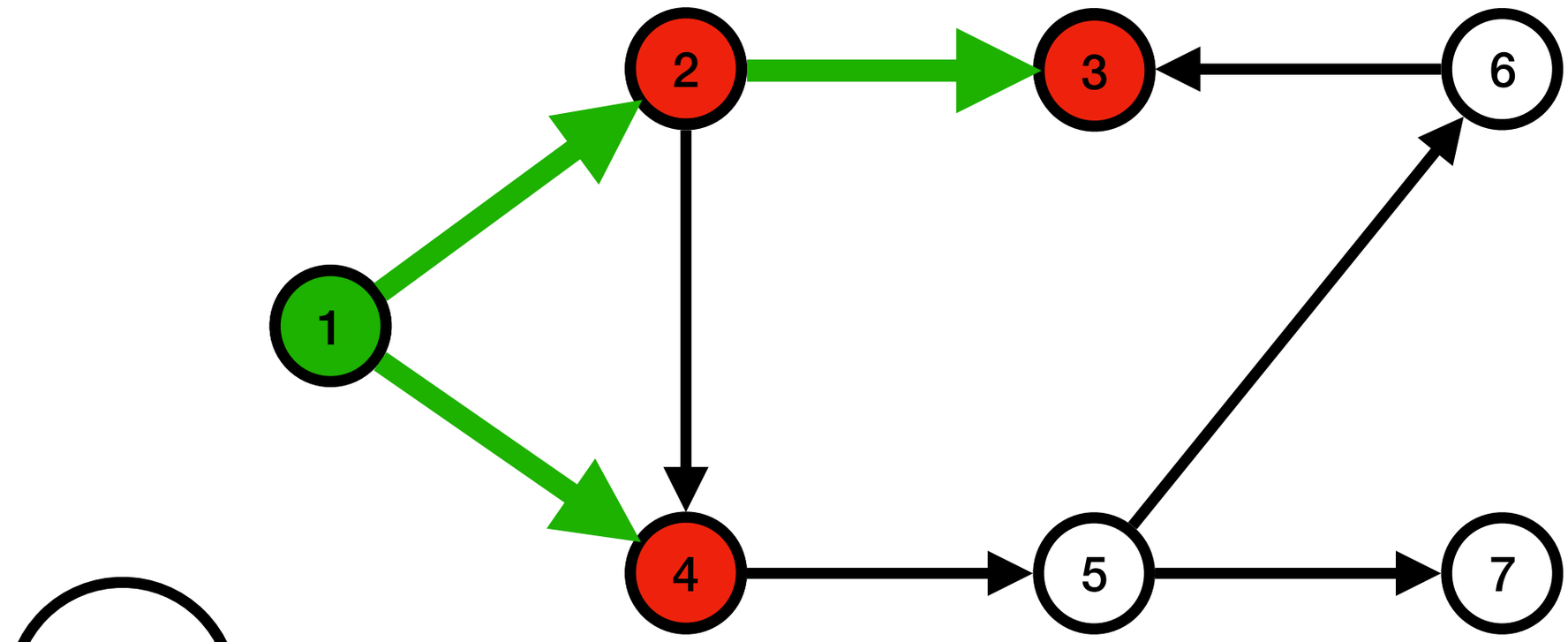
File → 2 4

Parcours en largeur



File → 

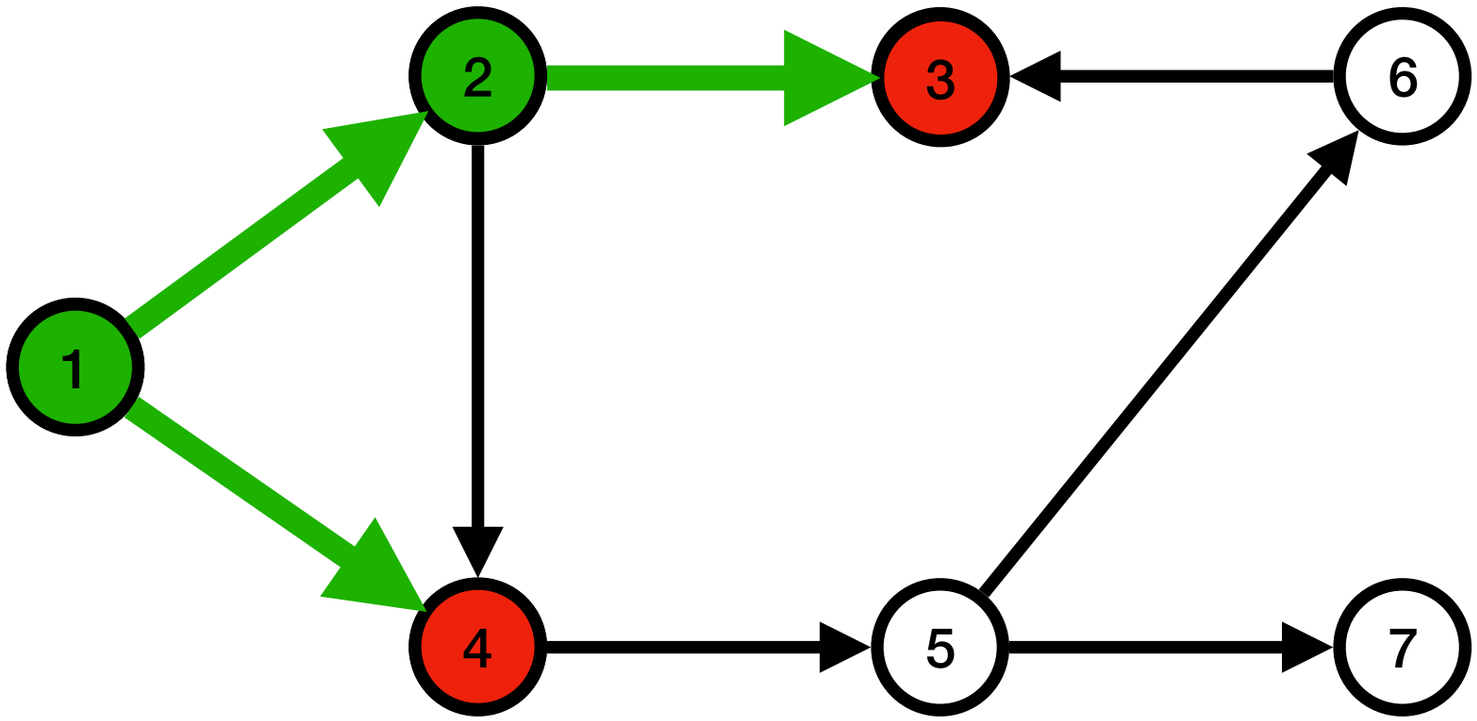
Parcours en largeur



2

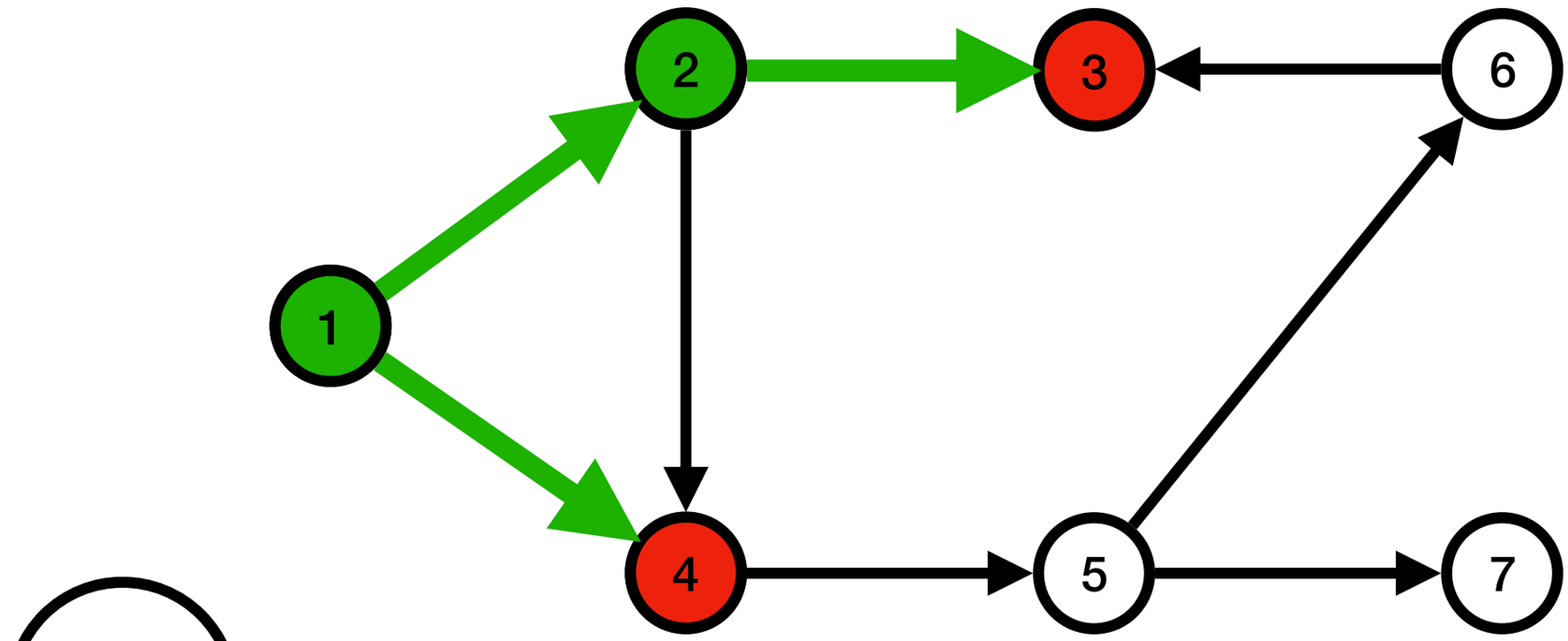
File → 4 3

Parcours en largeur



File → 4 3

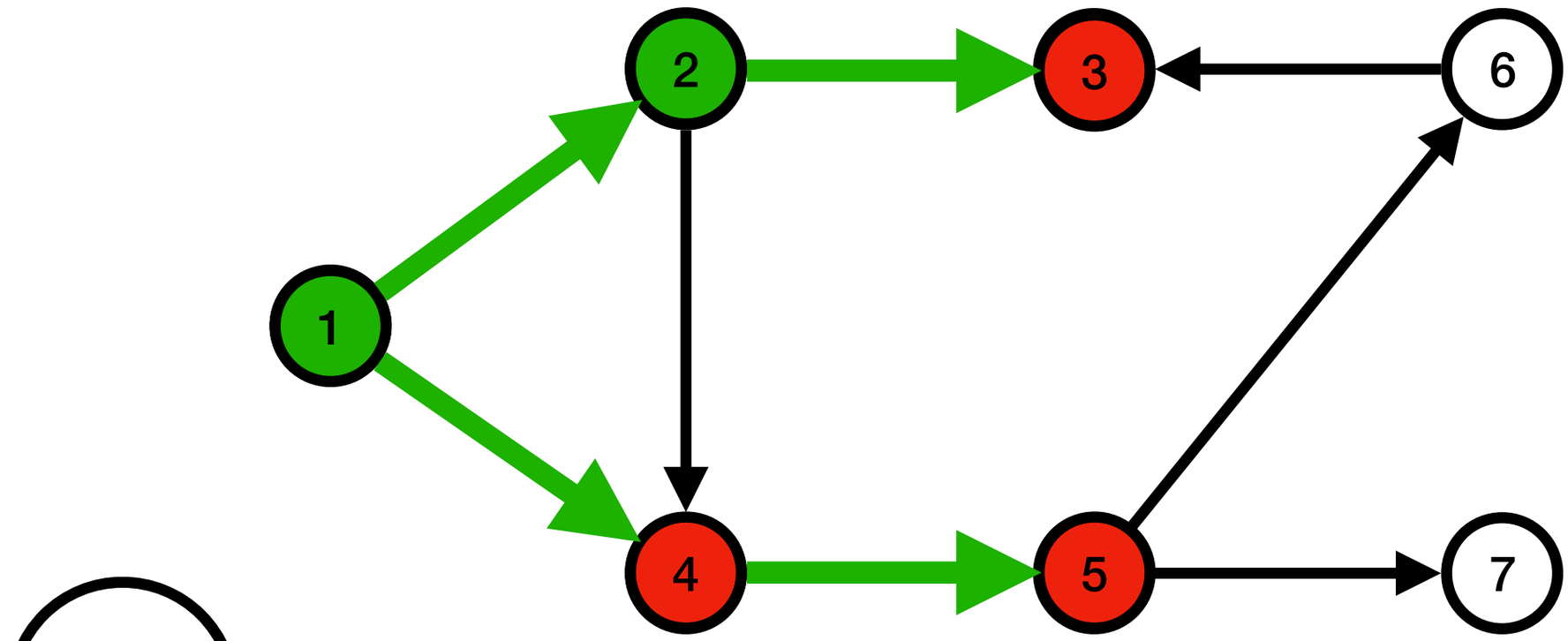
Parcours en largeur



4

File → 3

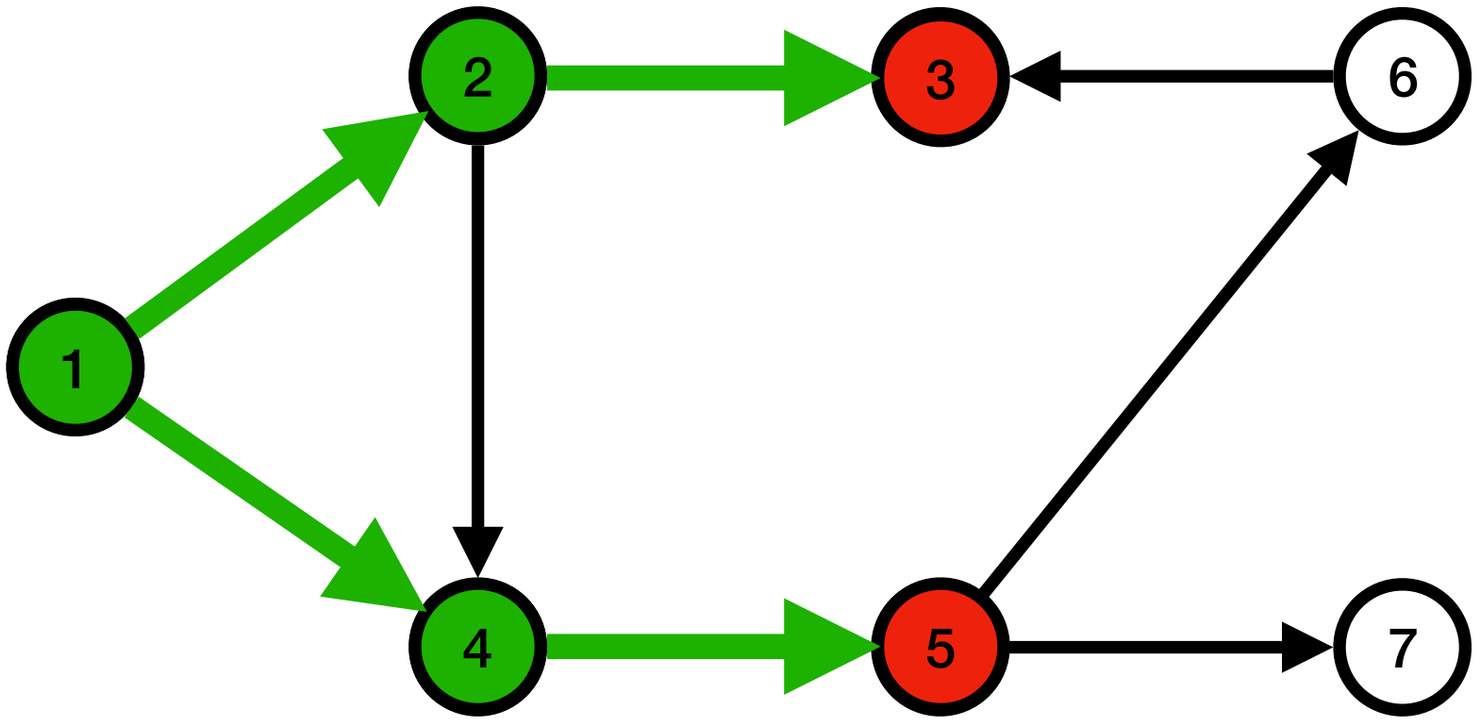
Parcours en largeur



4

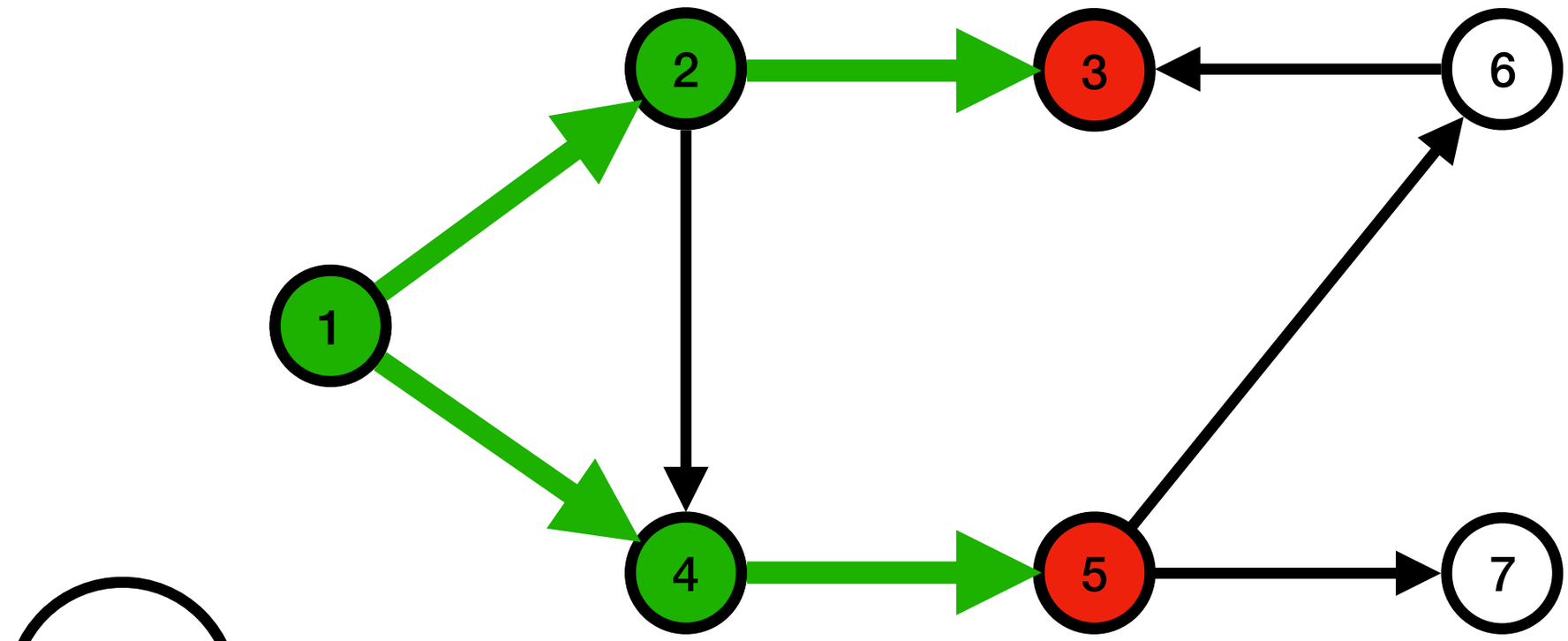
File → 3 5

Parcours en largeur



File → 3 5

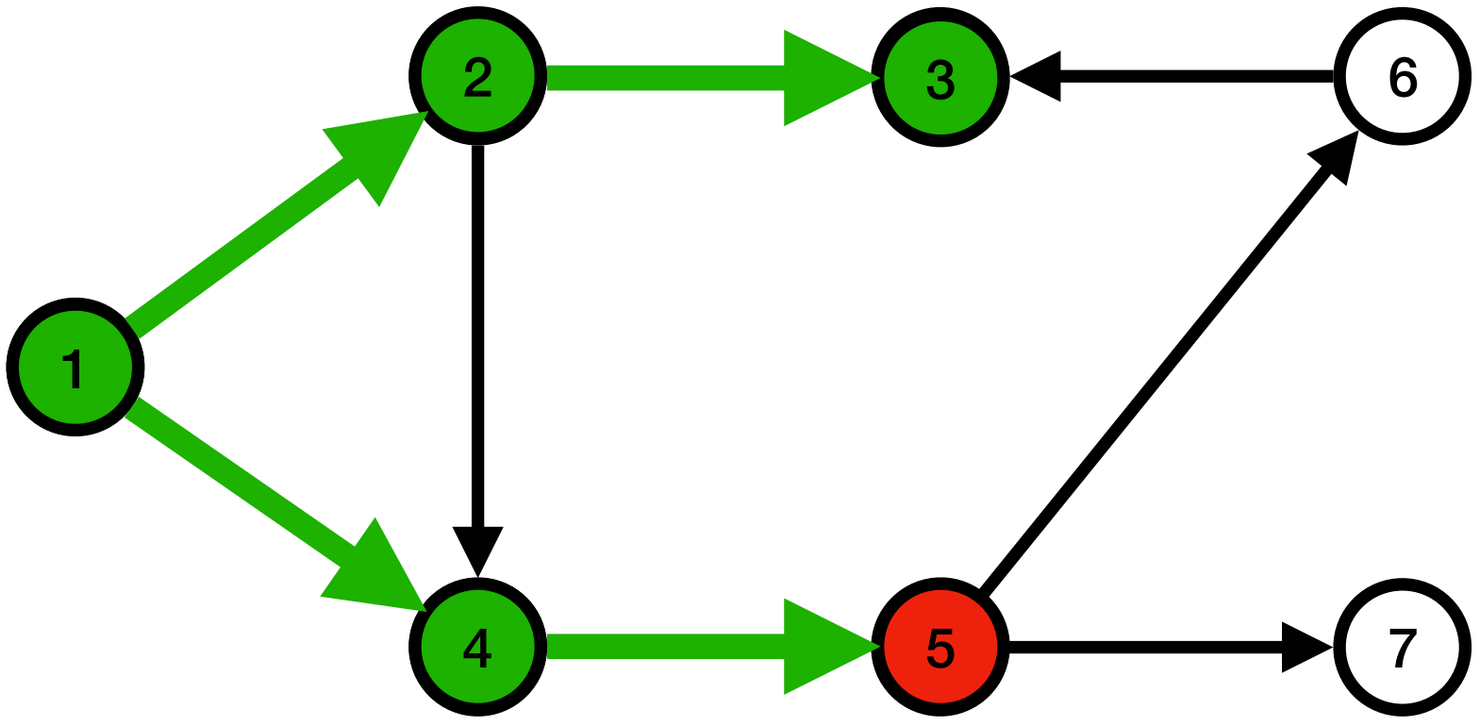
Parcours en largeur



3

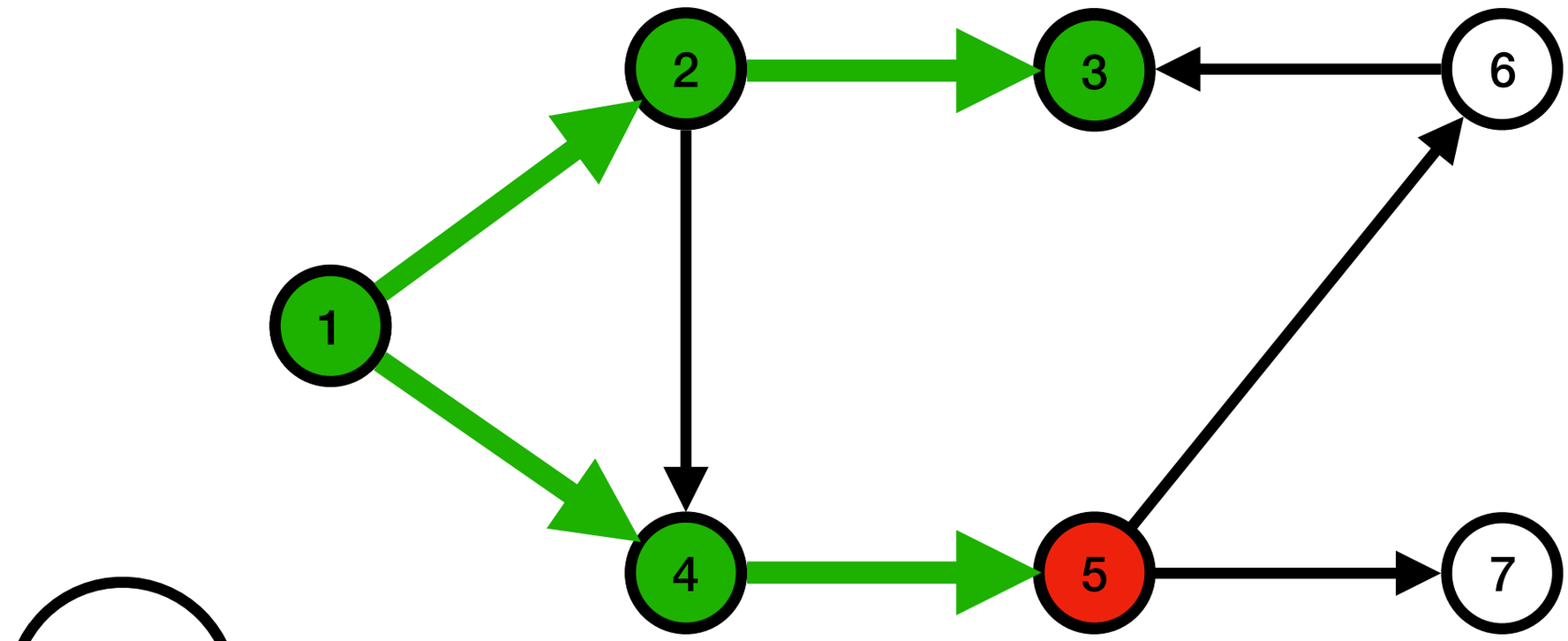
File → 5

Parcours en largeur



File → 

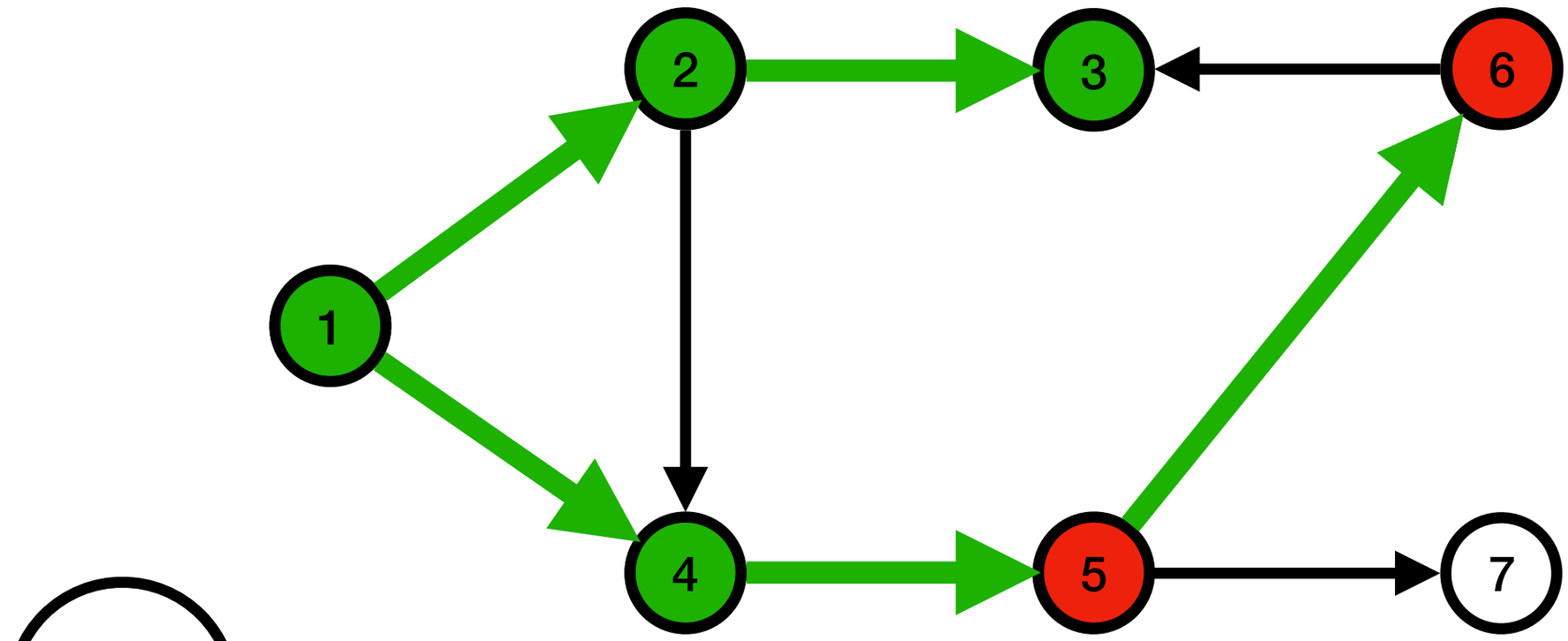
Parcours en largeur



5

File →

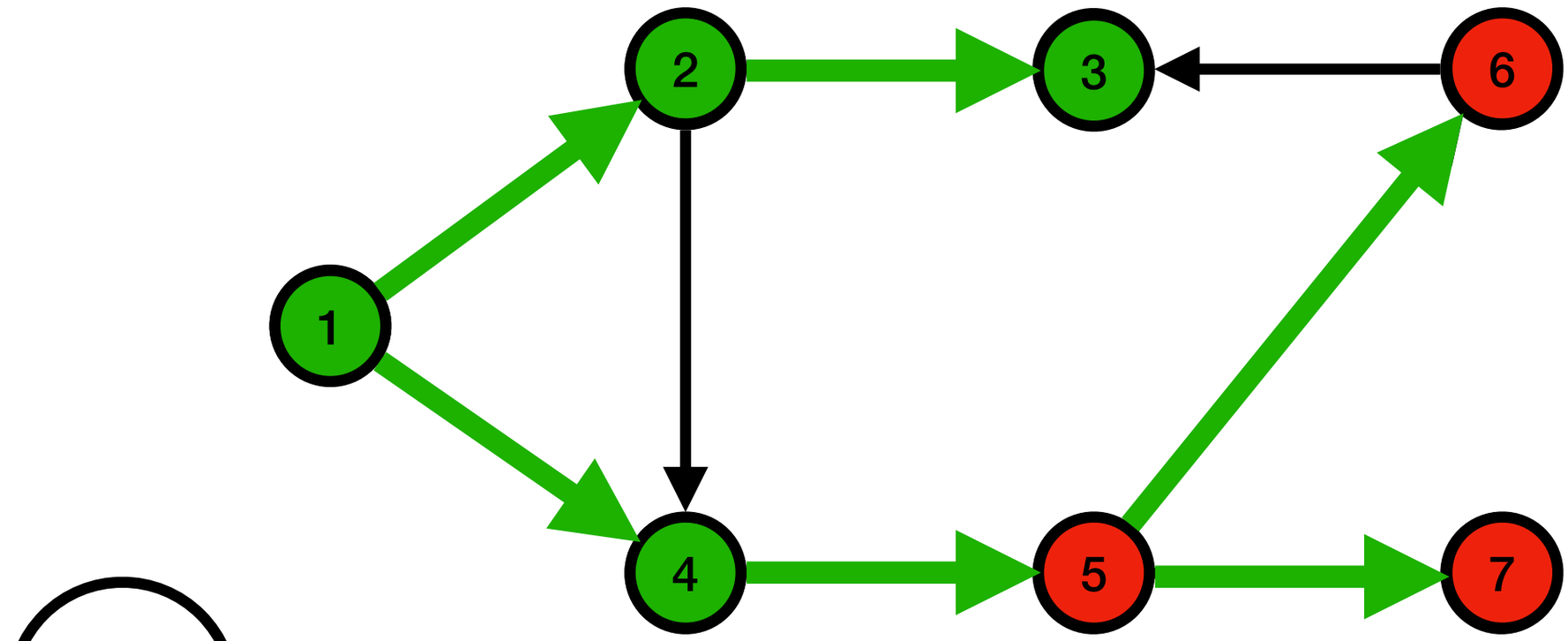
Parcours en largeur



5

File → 6

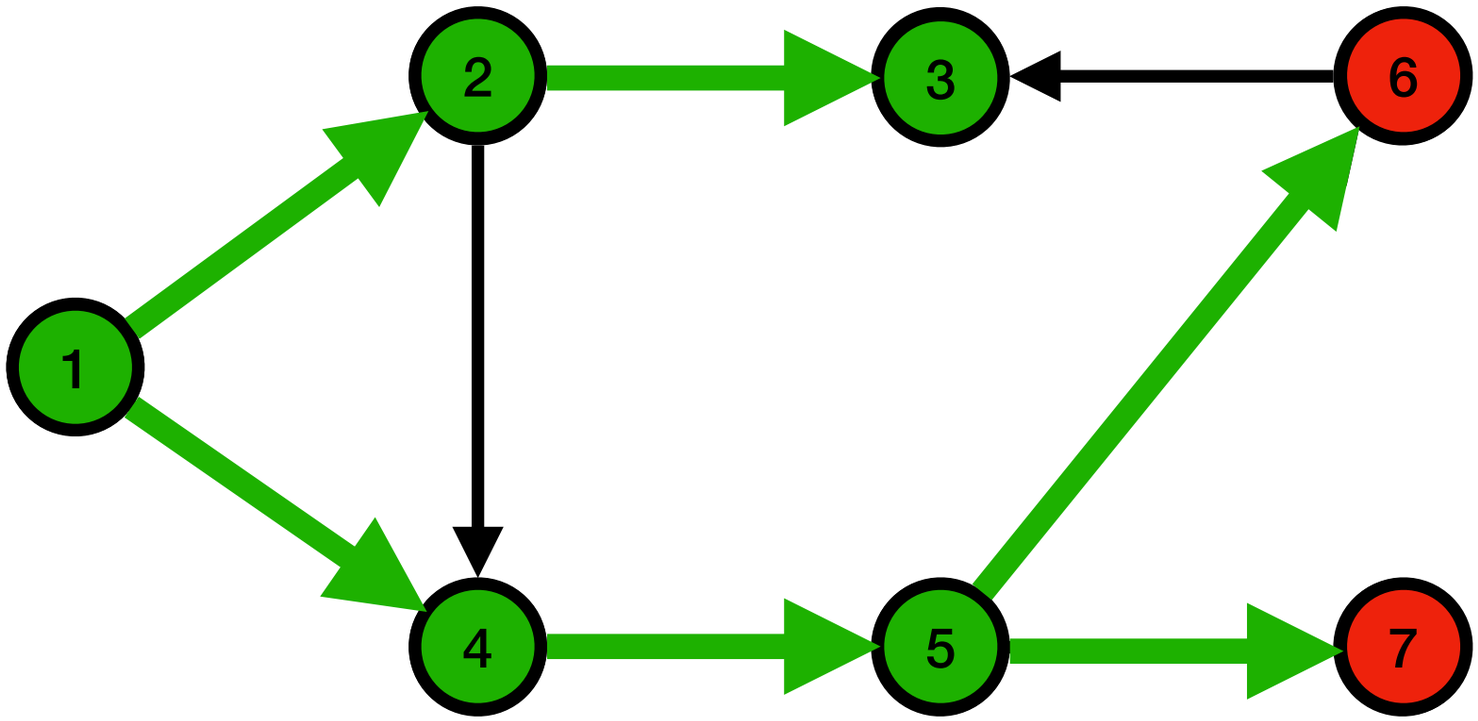
Parcours en largeur



5

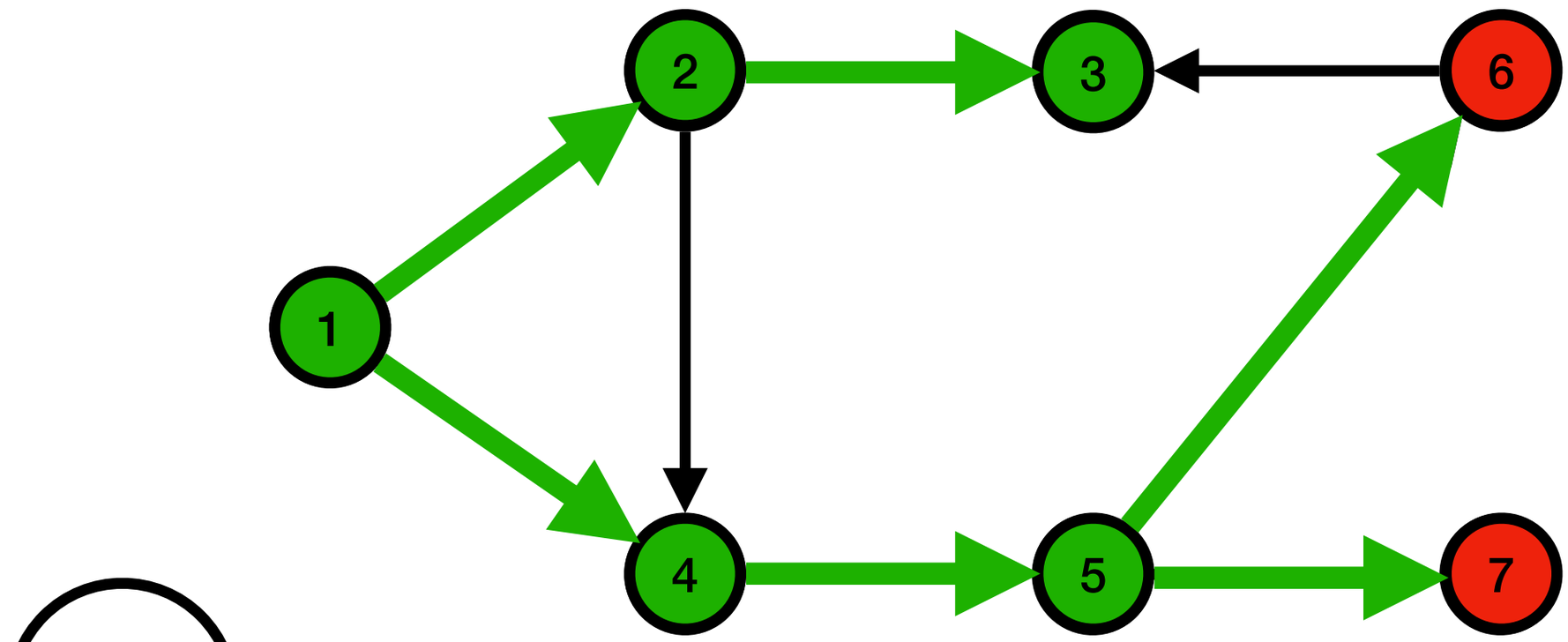
File → 6 7

Parcours en largeur



File → 6 7

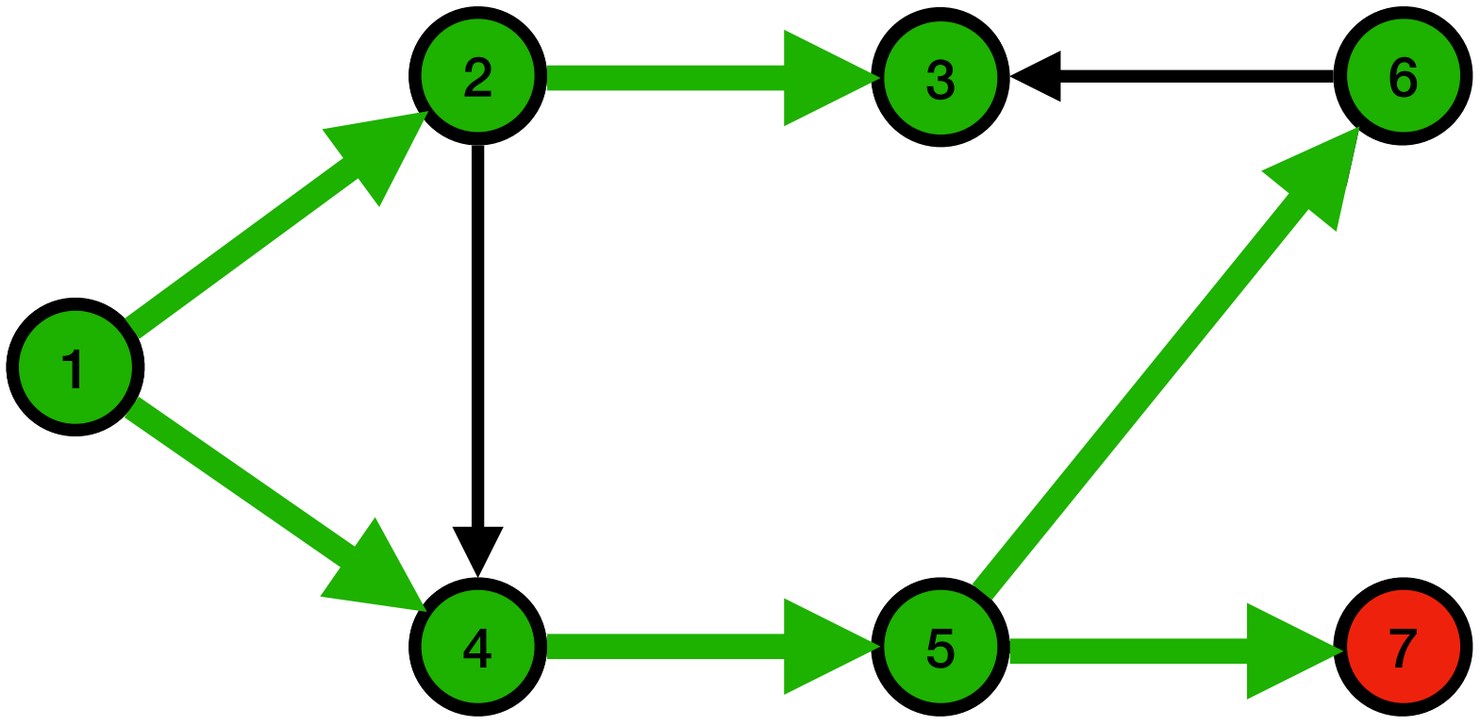
Parcours en largeur



6

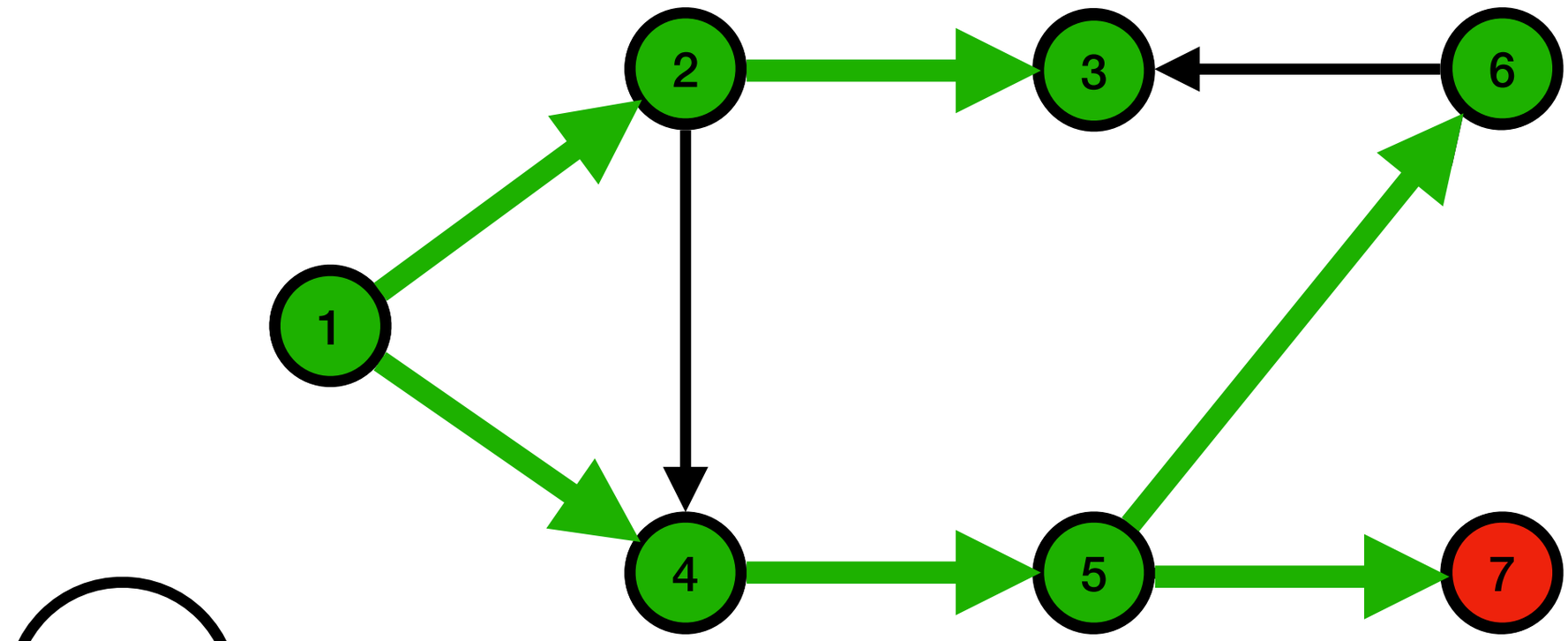
File → 7

Parcours en largeur



File → 7

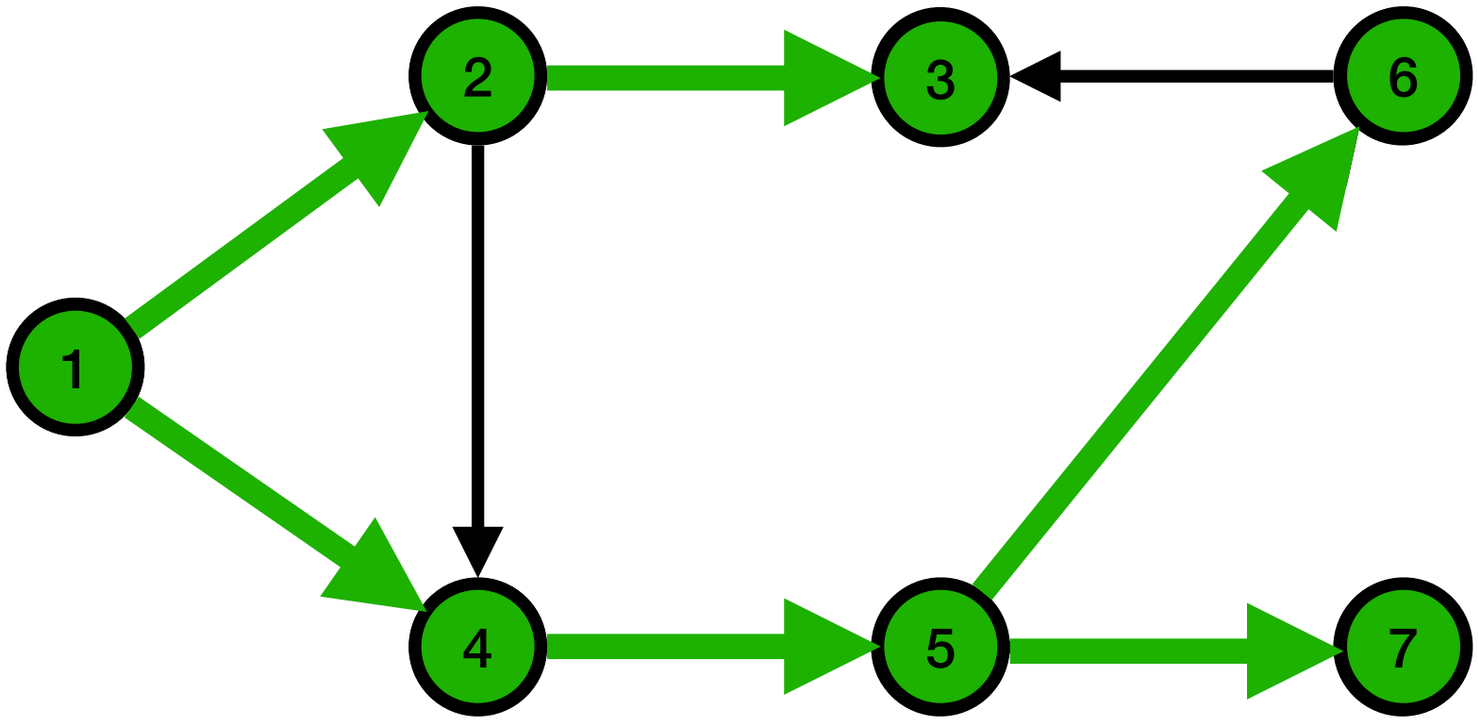
Parcours en largeur



7

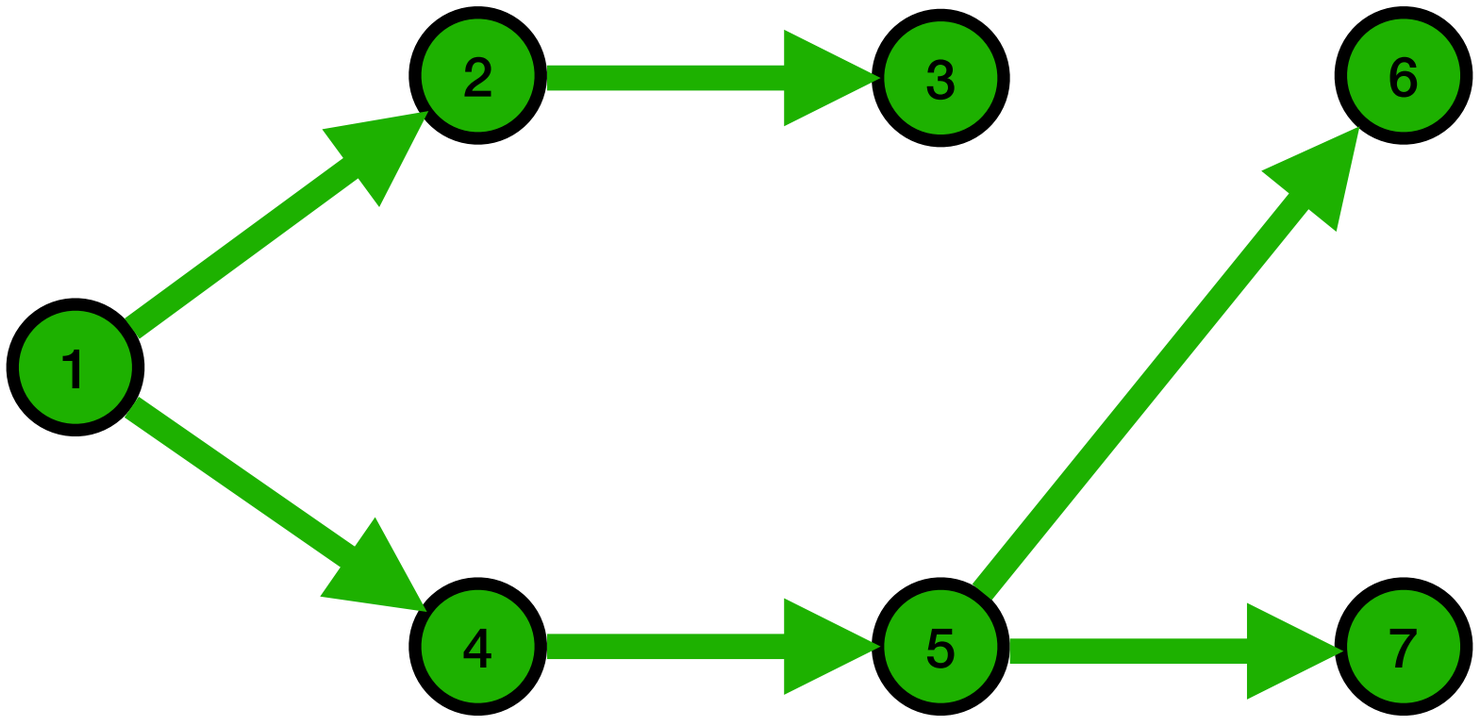
File →

Parcours en largeur



File →

Parcours en largeur



Structure de données « file »

$$F := \varnothing$$

Structure de données « file »

$F := \varnothing$

$F \rightarrow$

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 ≔ défiler(F)

F → 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 ≔ défiler(F)

F → 3 4 5

Structure de données « file »

x ≔ défiler(F)

F → 4 5

Structure de données « file »

x ≔ défiler(F)

F → 5

Structure de données « file »

x ≔ défiler(F)

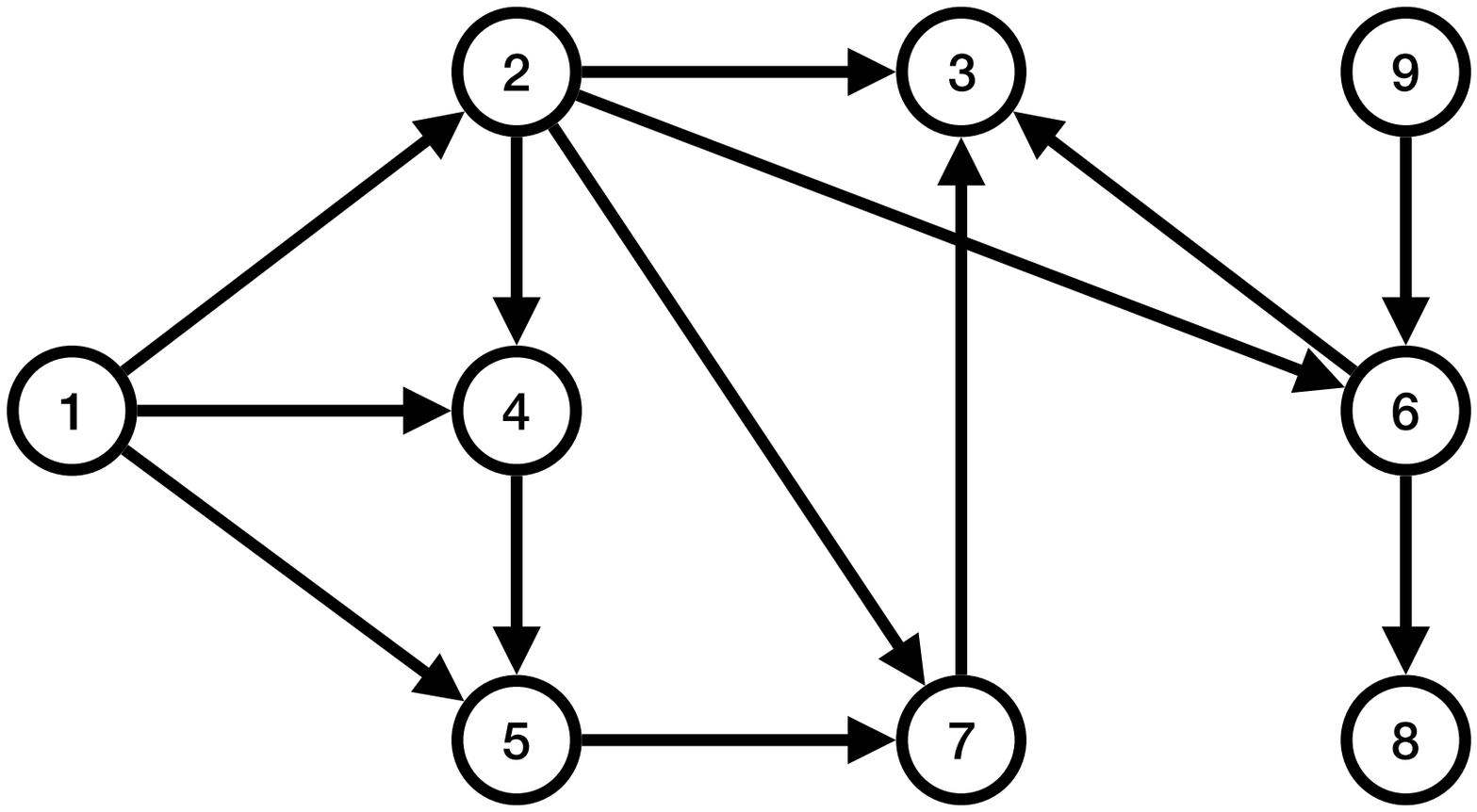
F →

Structure de données « file »

F →

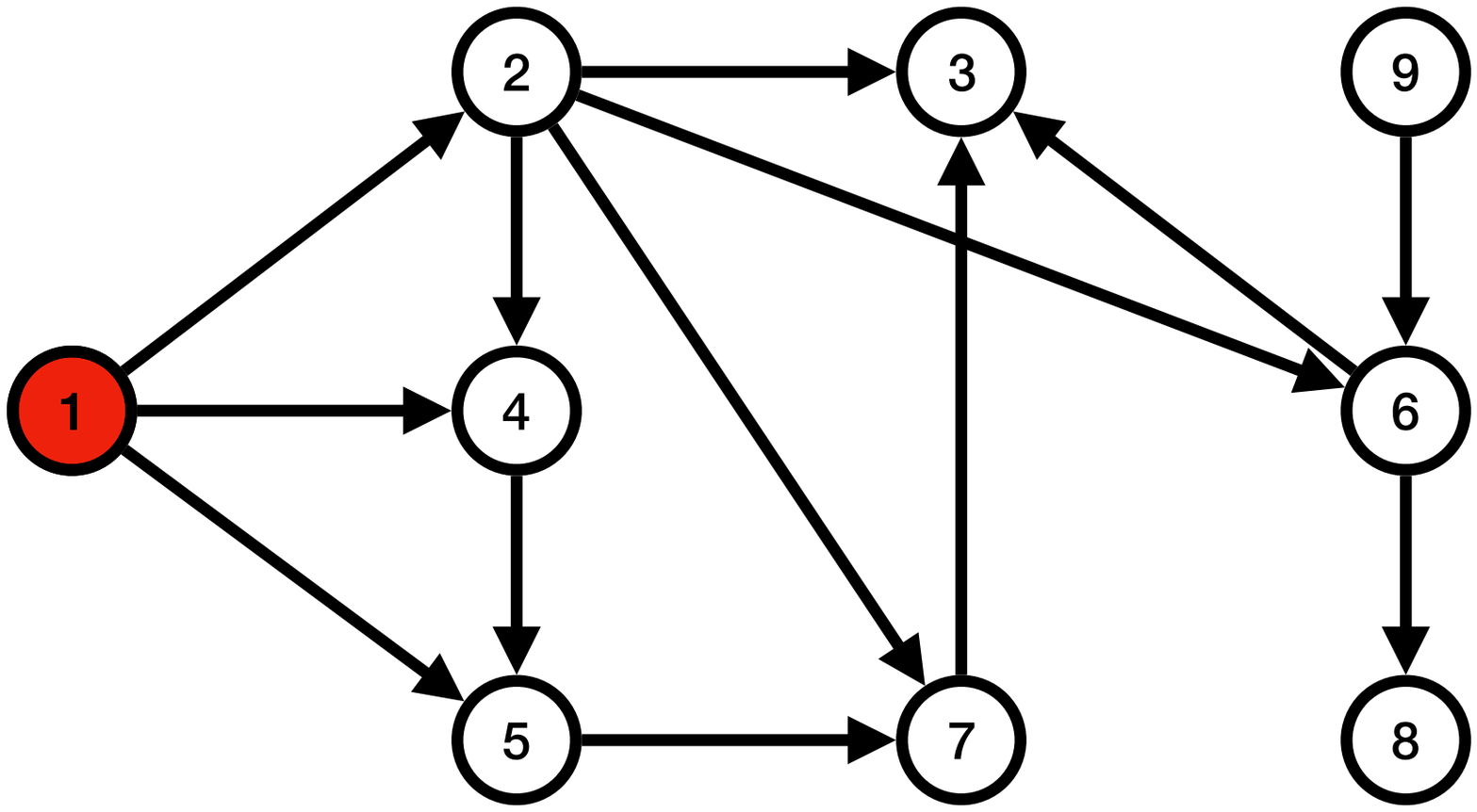
```
fonction parcours-en-largeur(G, s)
    n ≔ |G|                 (nombre de sommets)
    H ≔ graphe(n)           (graphe vide)
    F ≔ ∅                   (file vide)
    couleur[s] ≔ rouge
    enfiler(F, s)
    tant que F ≠ ∅ faire
        u ≔ défiler(F)
        pour v ≔ 0 jusqu'à n – 1 faire
   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

Un autre exemple



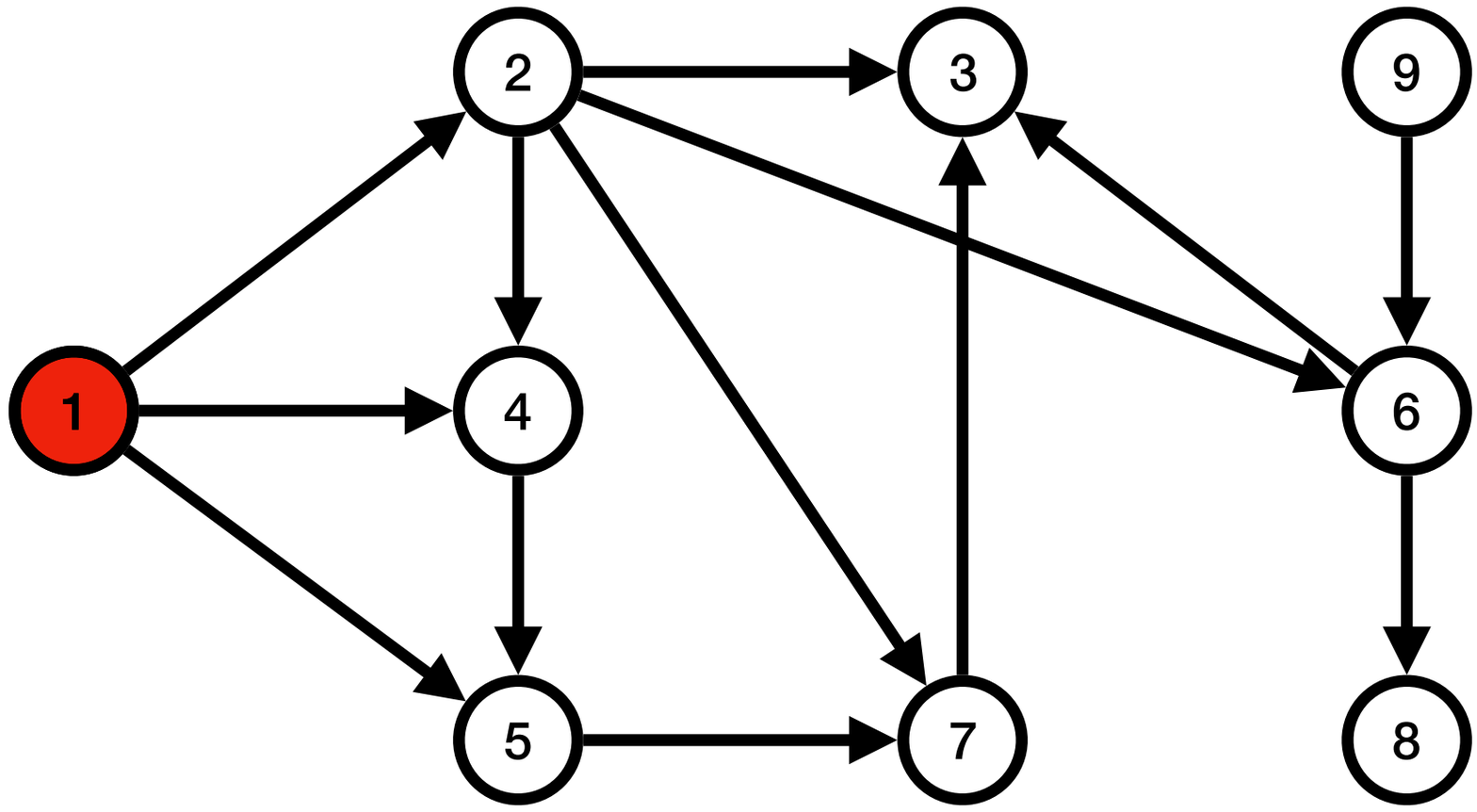
File →

Un autre exemple



File → 1

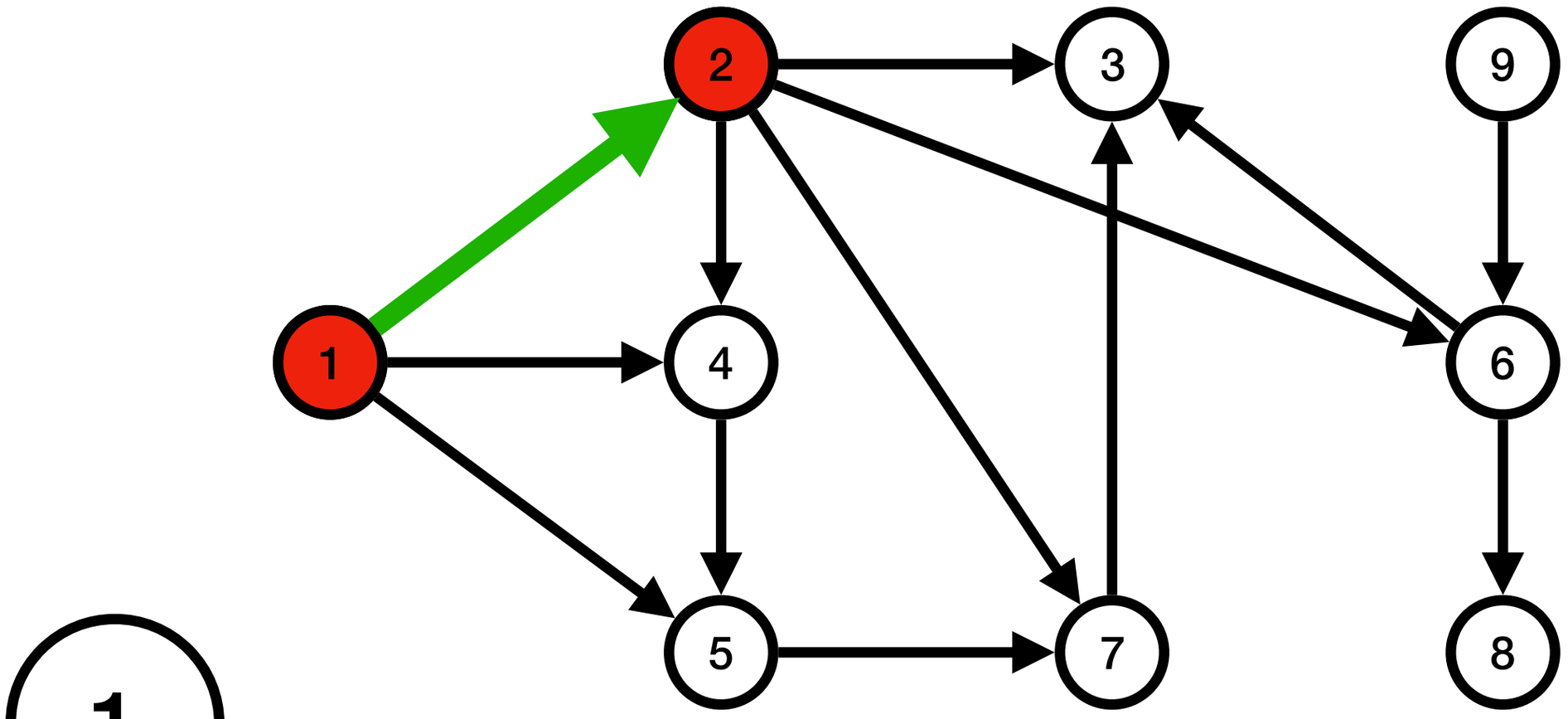
Un autre exemple



1

File →

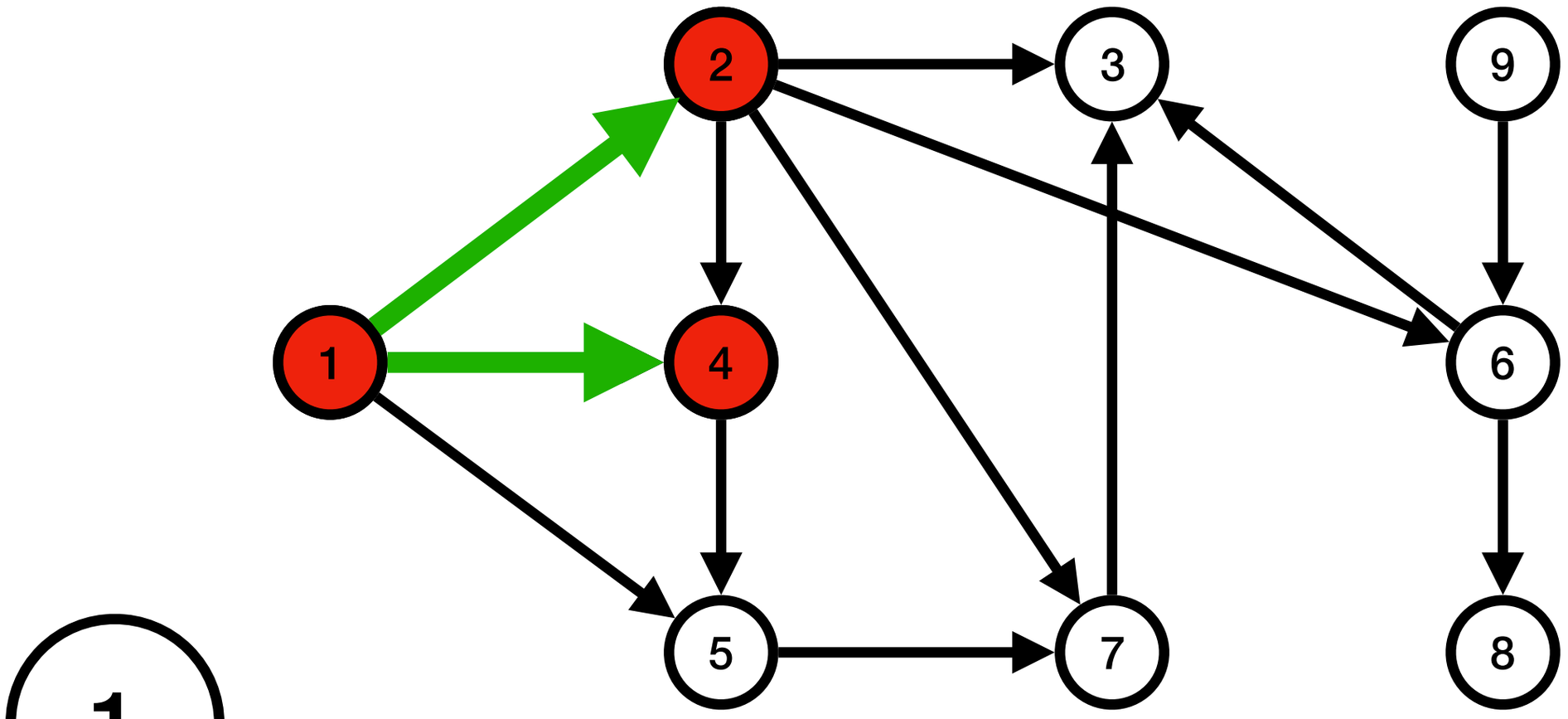
Un autre exemple



1

File → 2

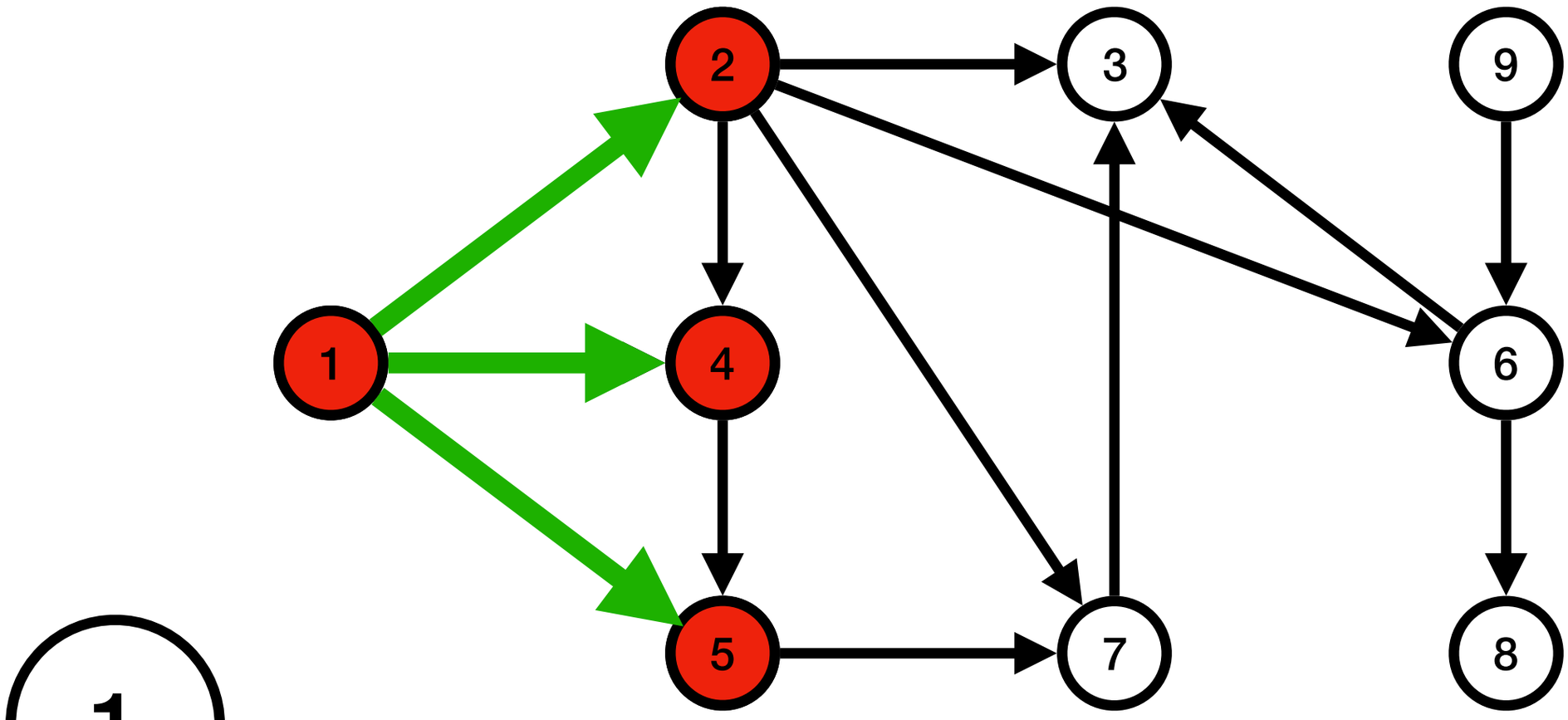
Un autre exemple



1

File → 2 4

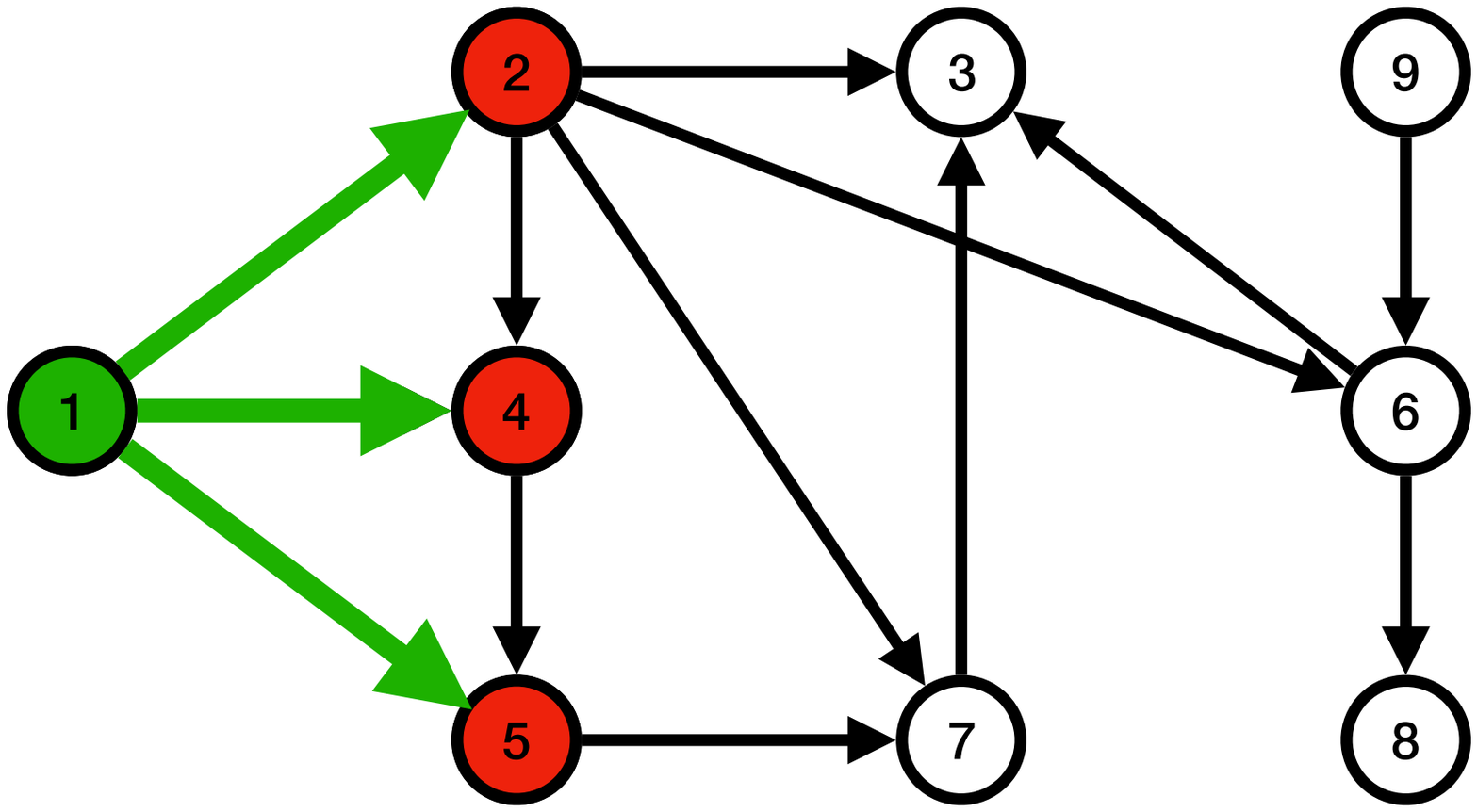
Un autre exemple



1

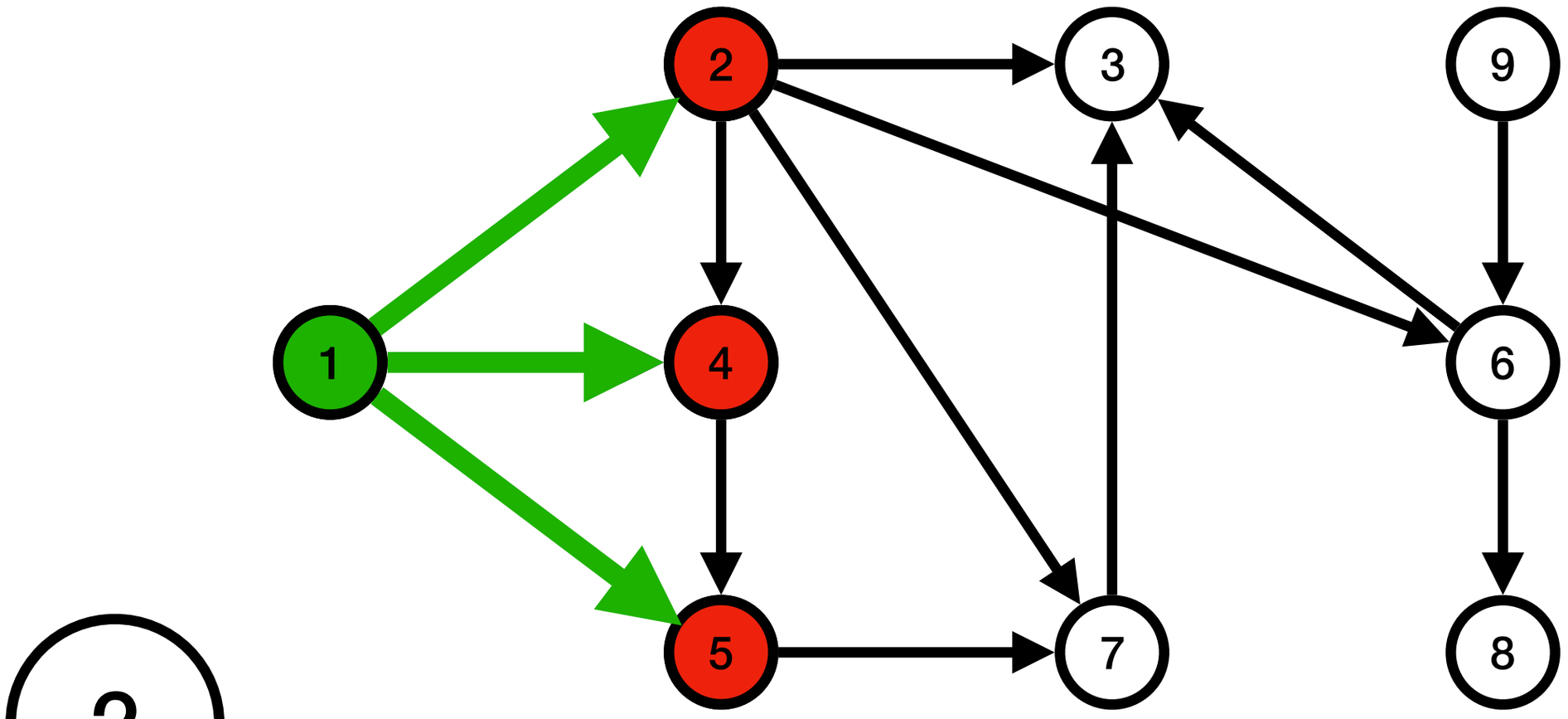
File → 2 4 5

Un autre exemple



File → 2 4 5

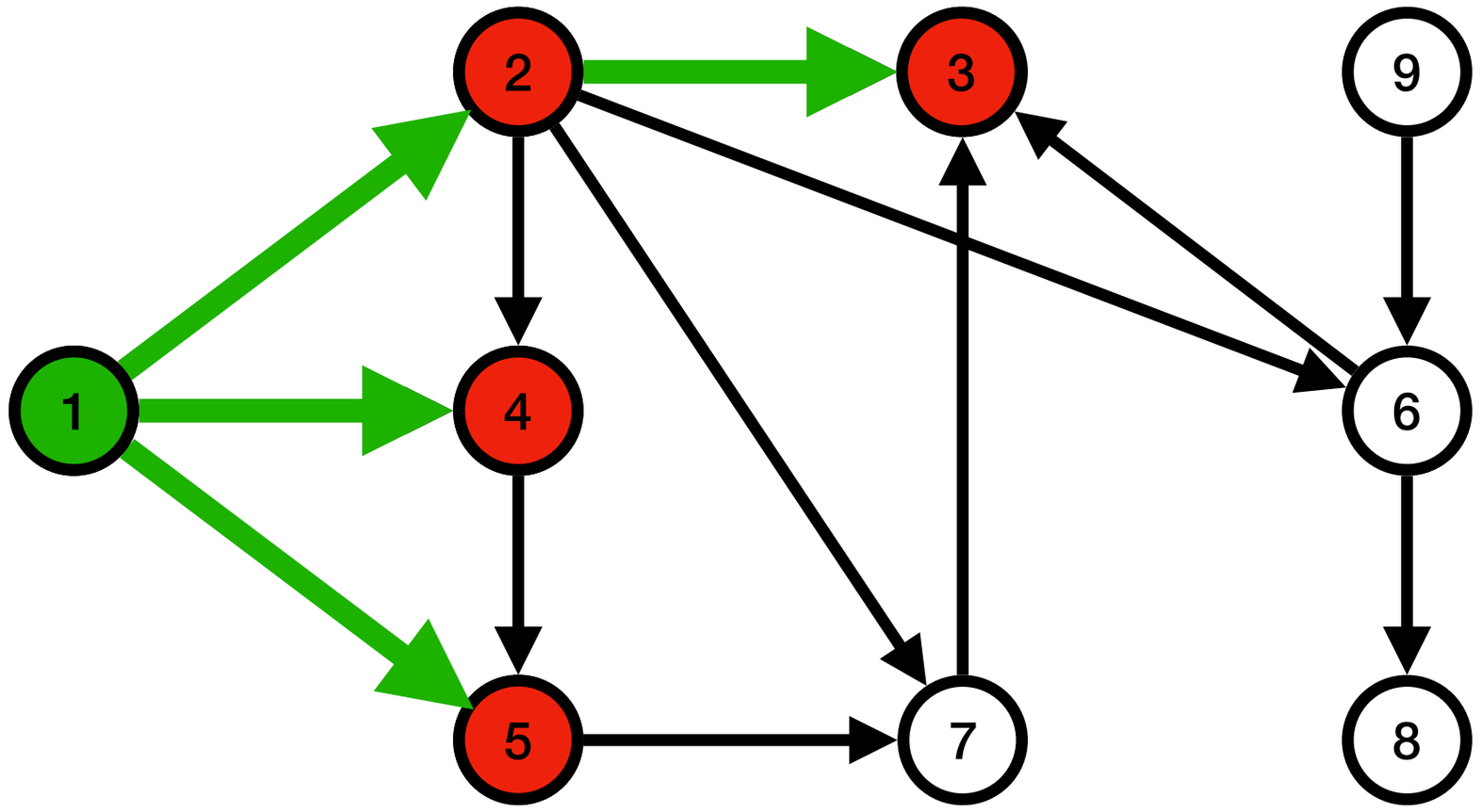
Un autre exemple



2

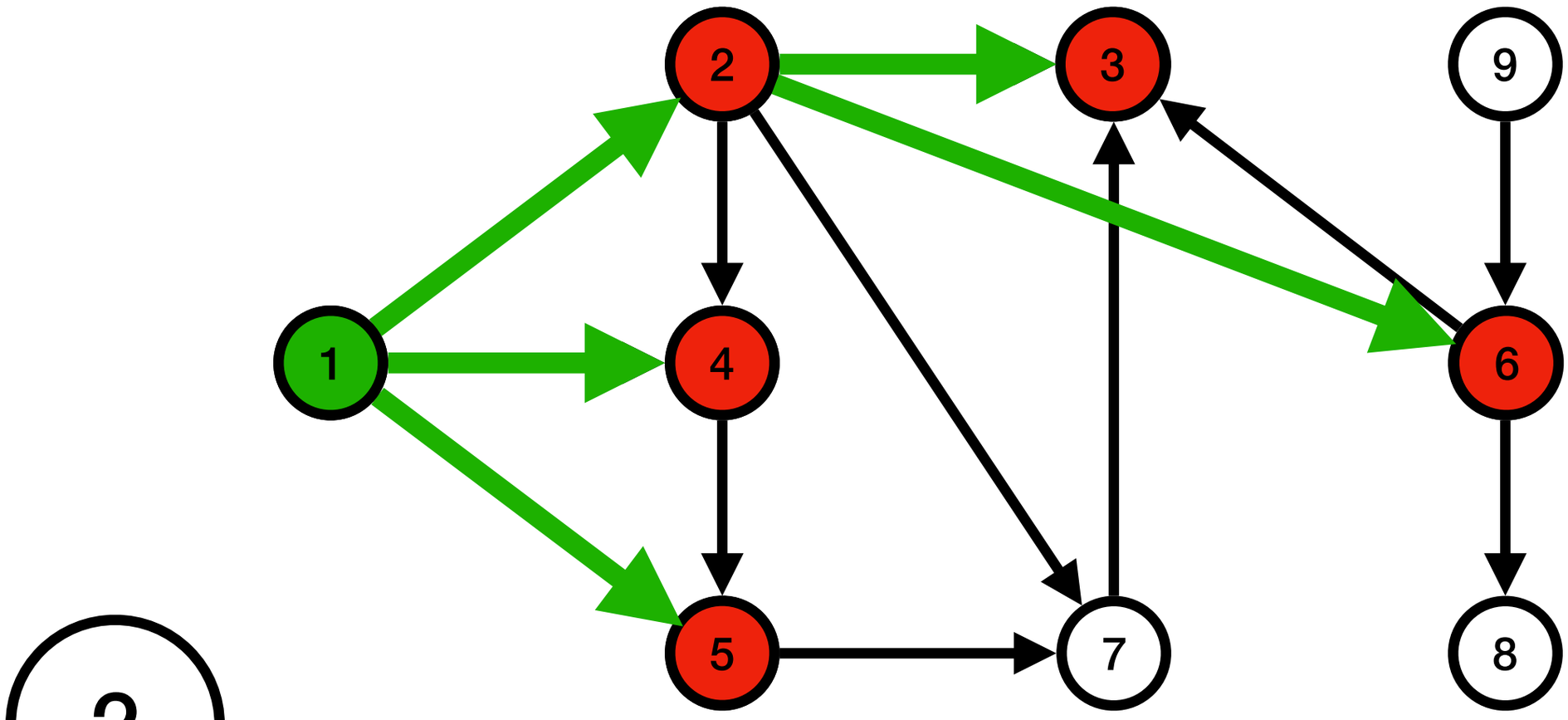
File → 4 5

Un autre exemple



File → 

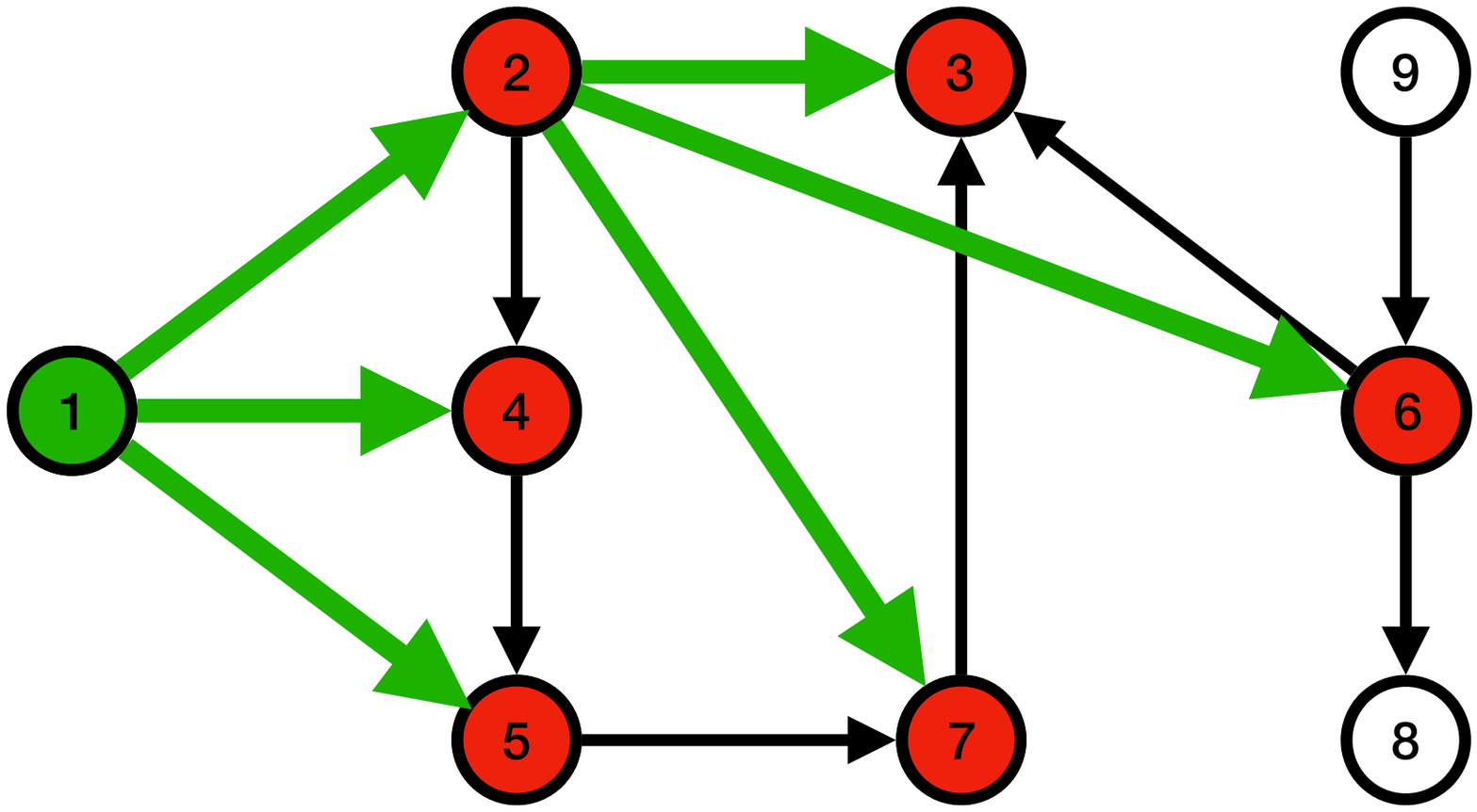
Un autre exemple



2

File → 4 5 3 6

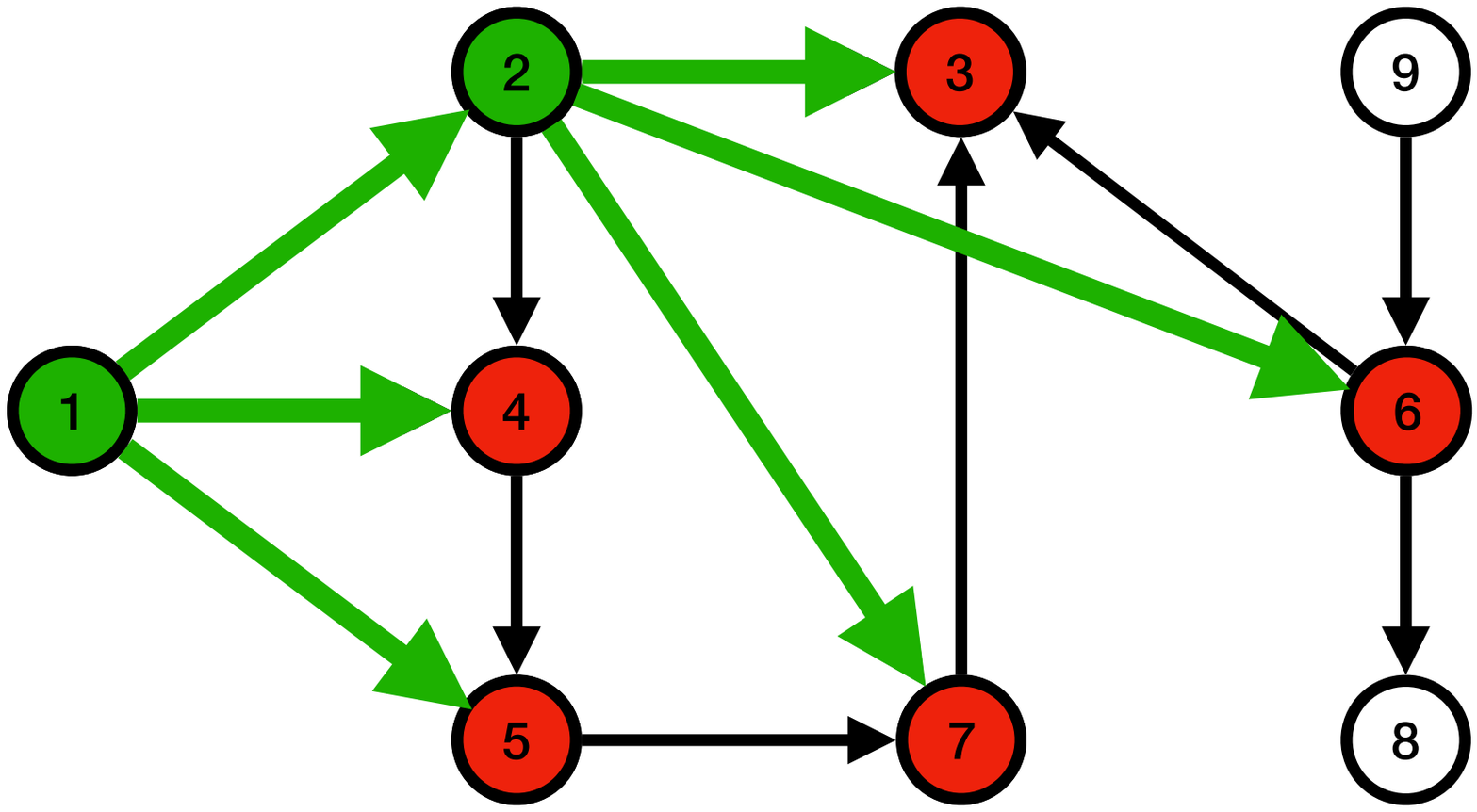
Un autre exemple



2

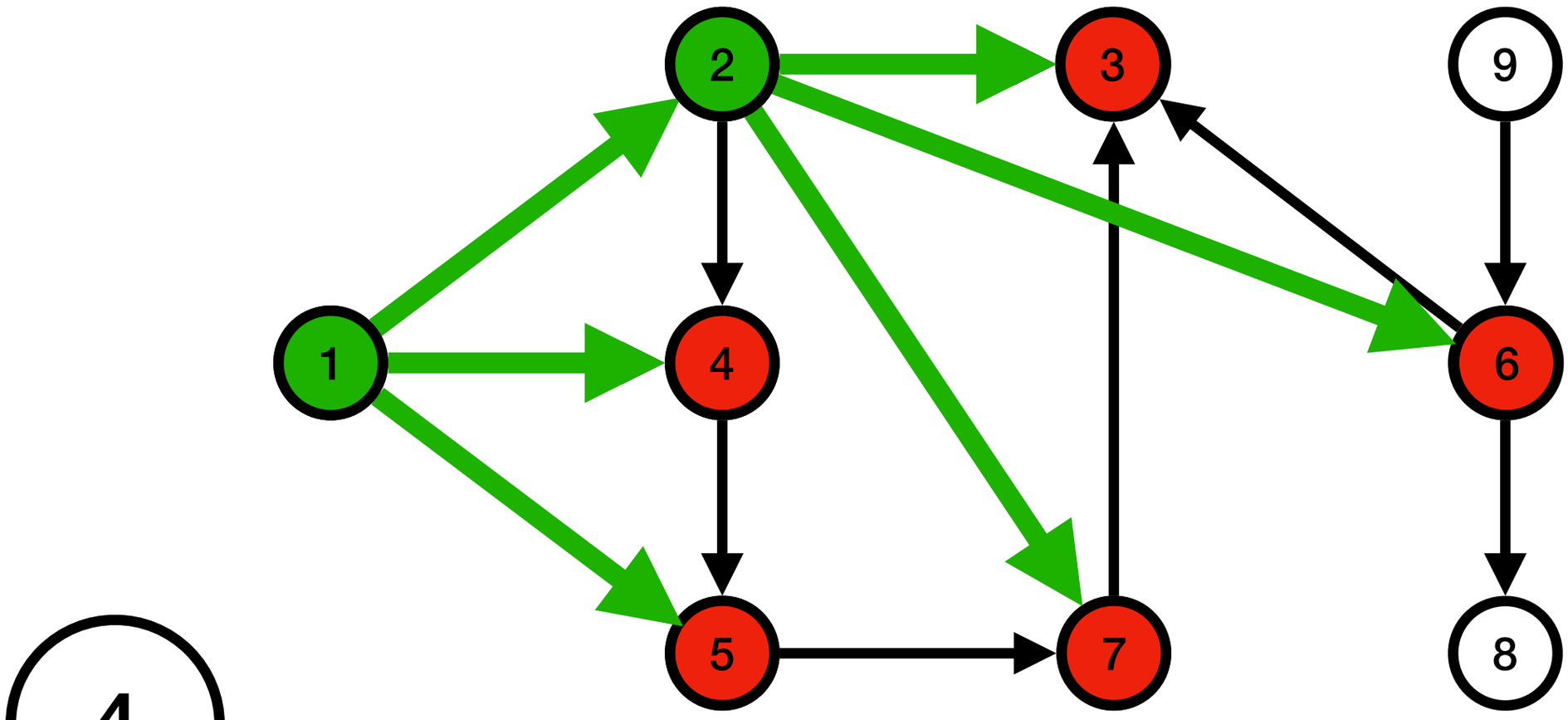
File → 4 5 3 6 7

Un autre exemple



File → 4 5 3 6 7

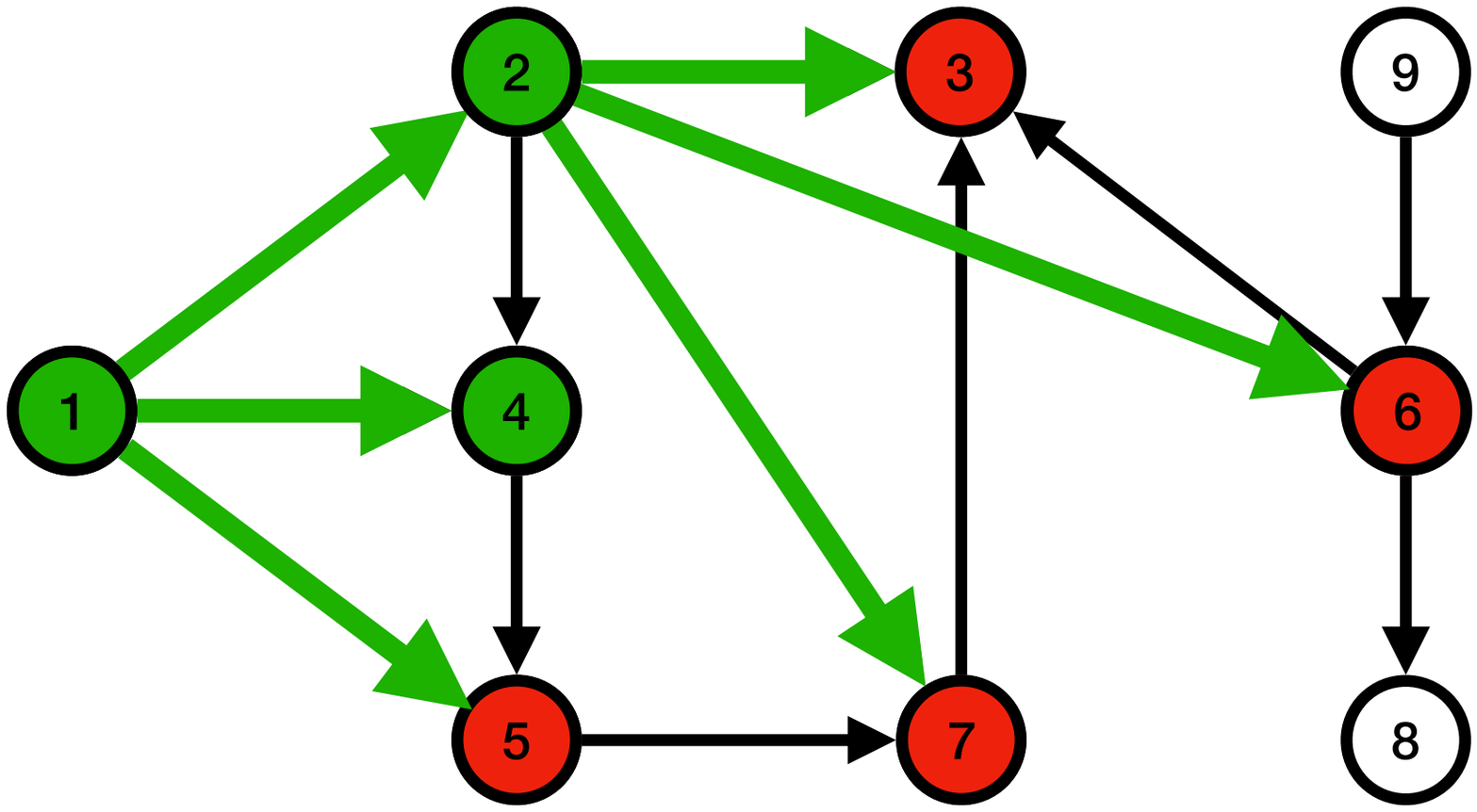
Un autre exemple



4

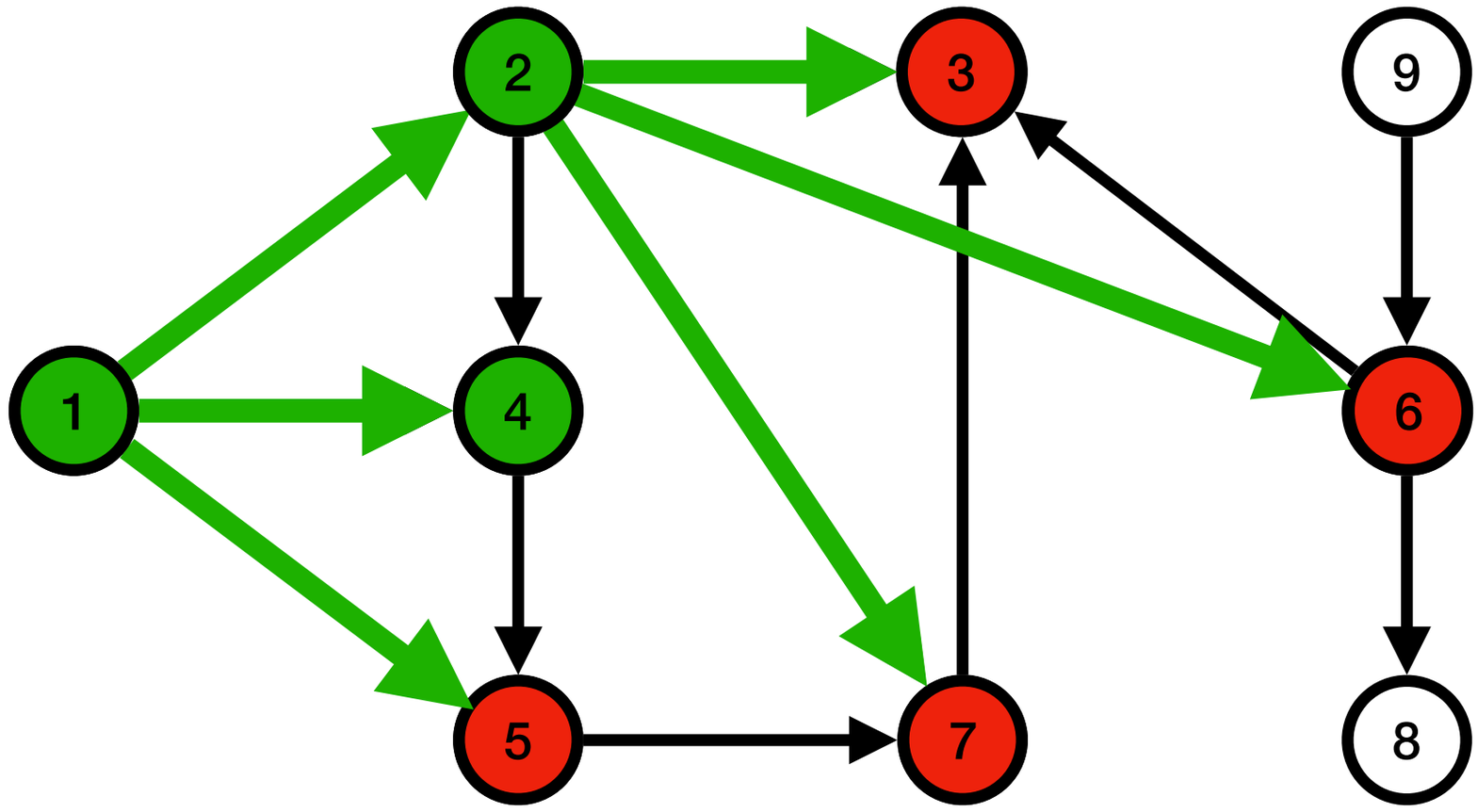
File → 5 3 6 7

Un autre exemple



File → 5 3 6 7

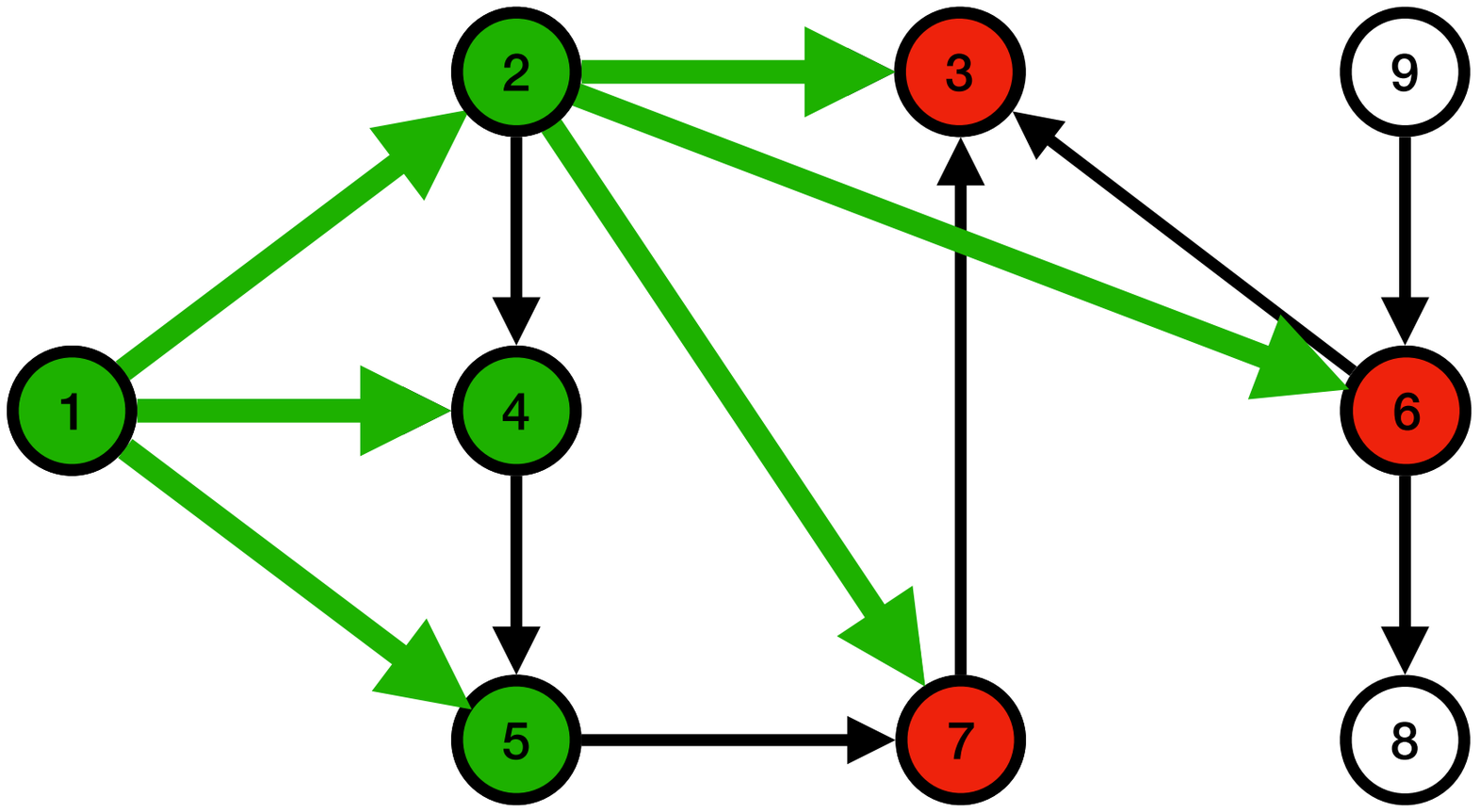
Un autre exemple



5

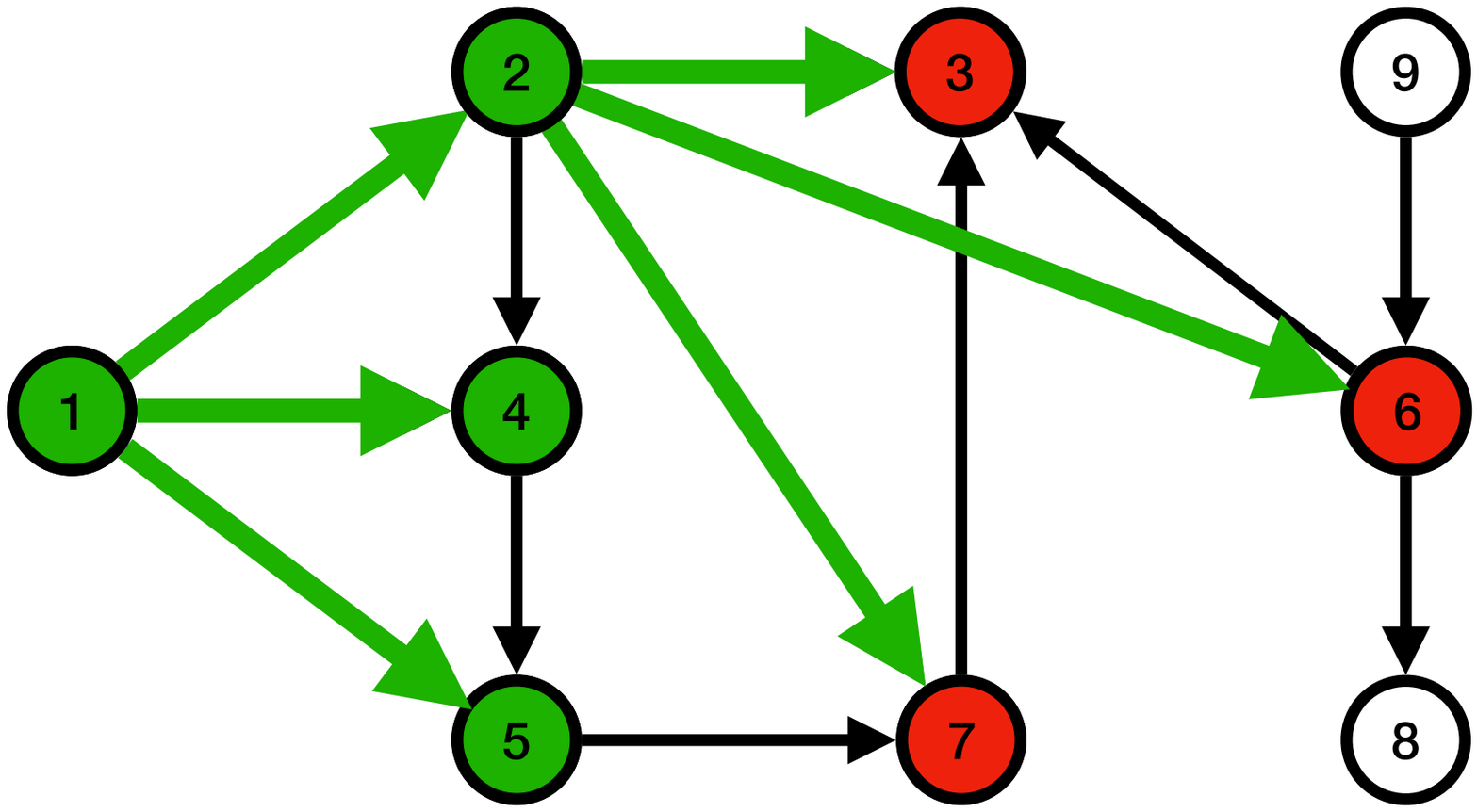
File → 3 6 7

Un autre exemple



File → 3 6 7

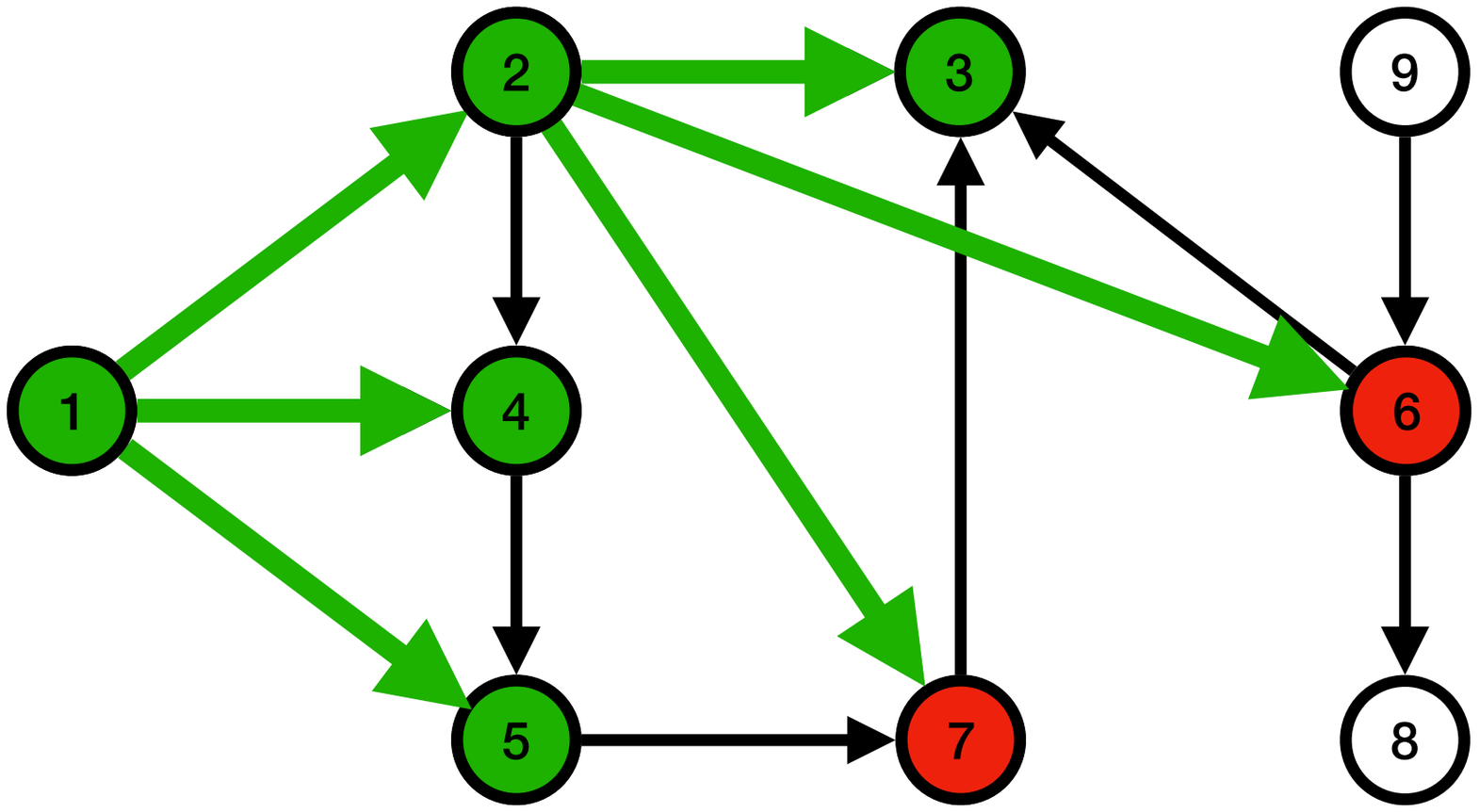
Un autre exemple



3

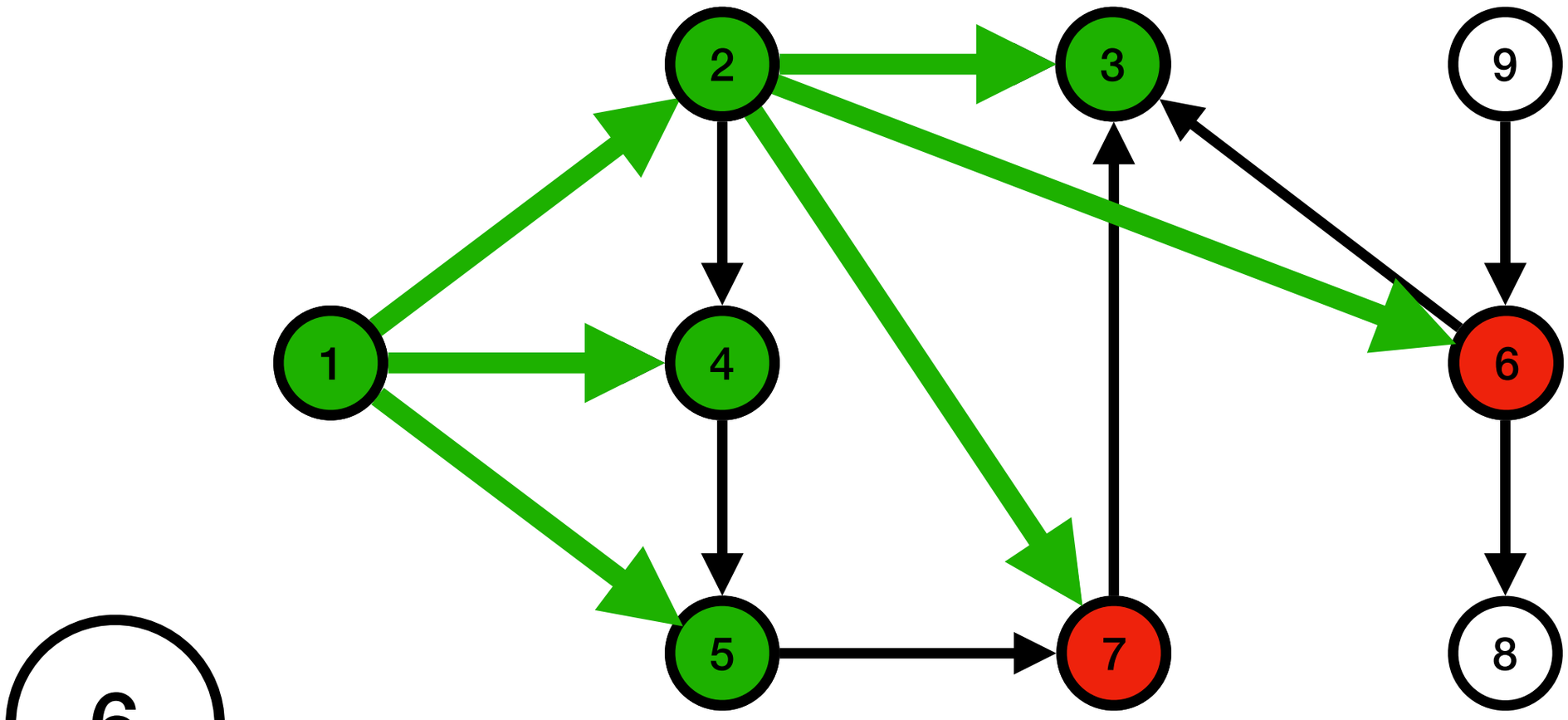
File → 6 7

Un autre exemple



File → 6 7

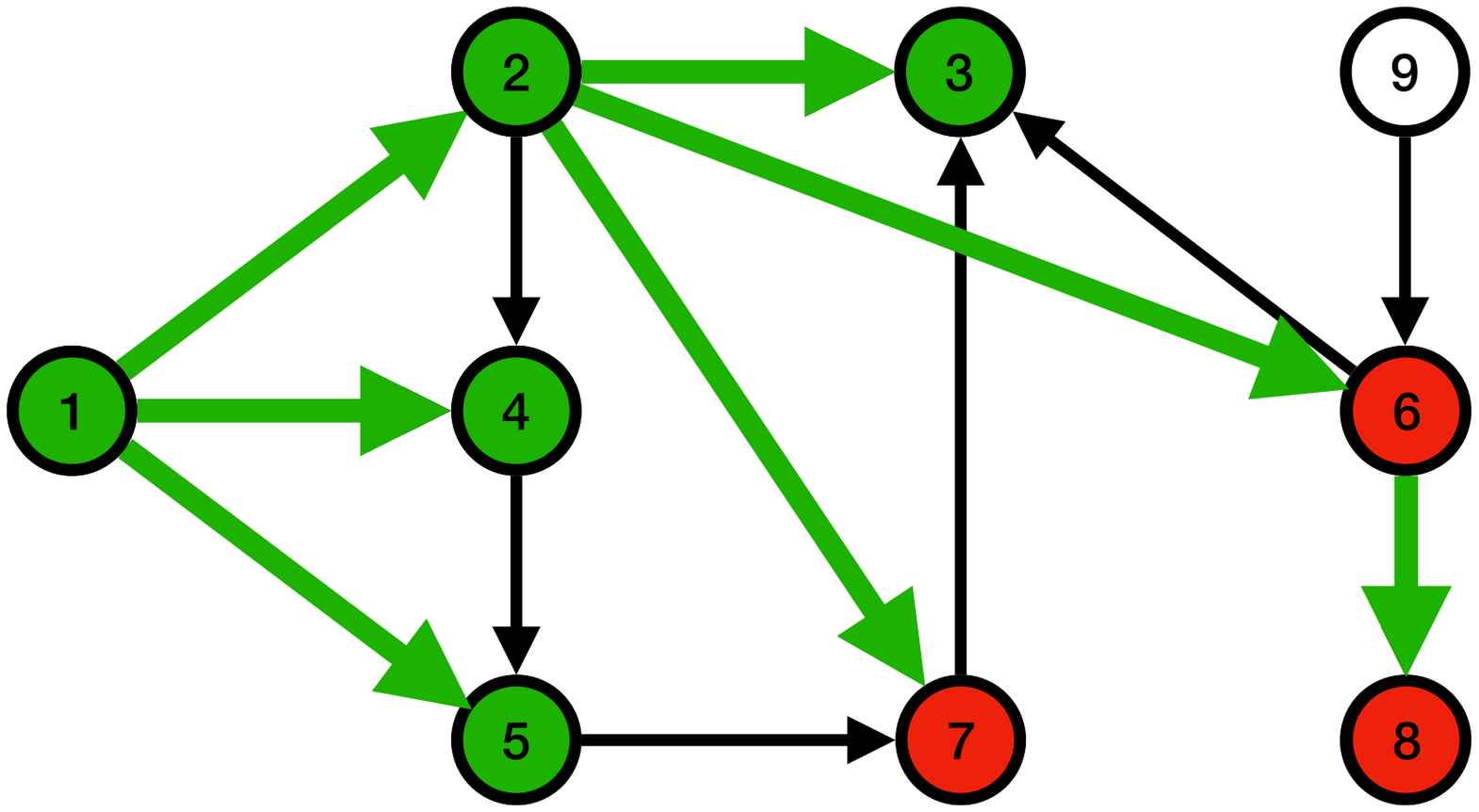
Un autre exemple



6

File → 7

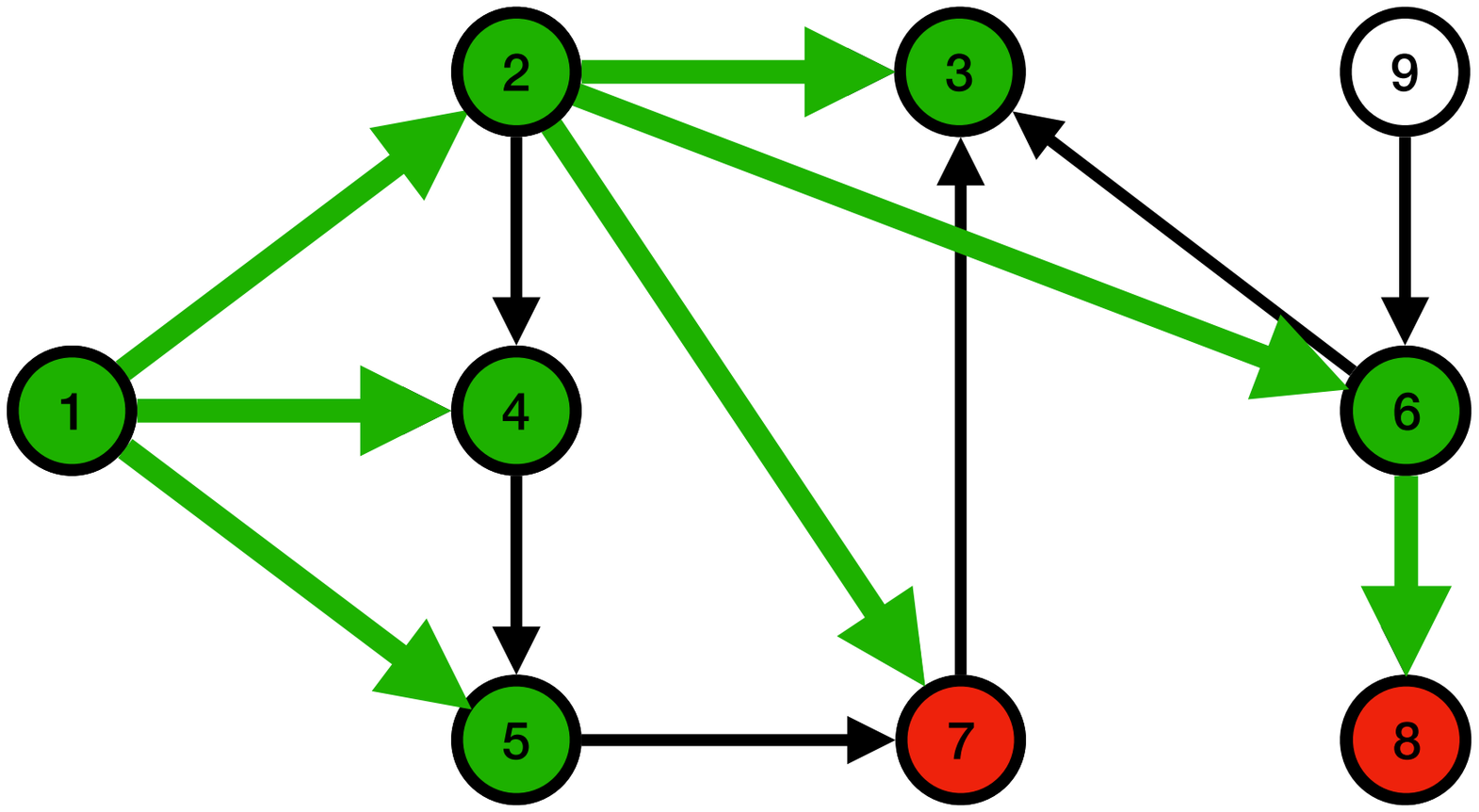
Un autre exemple



6

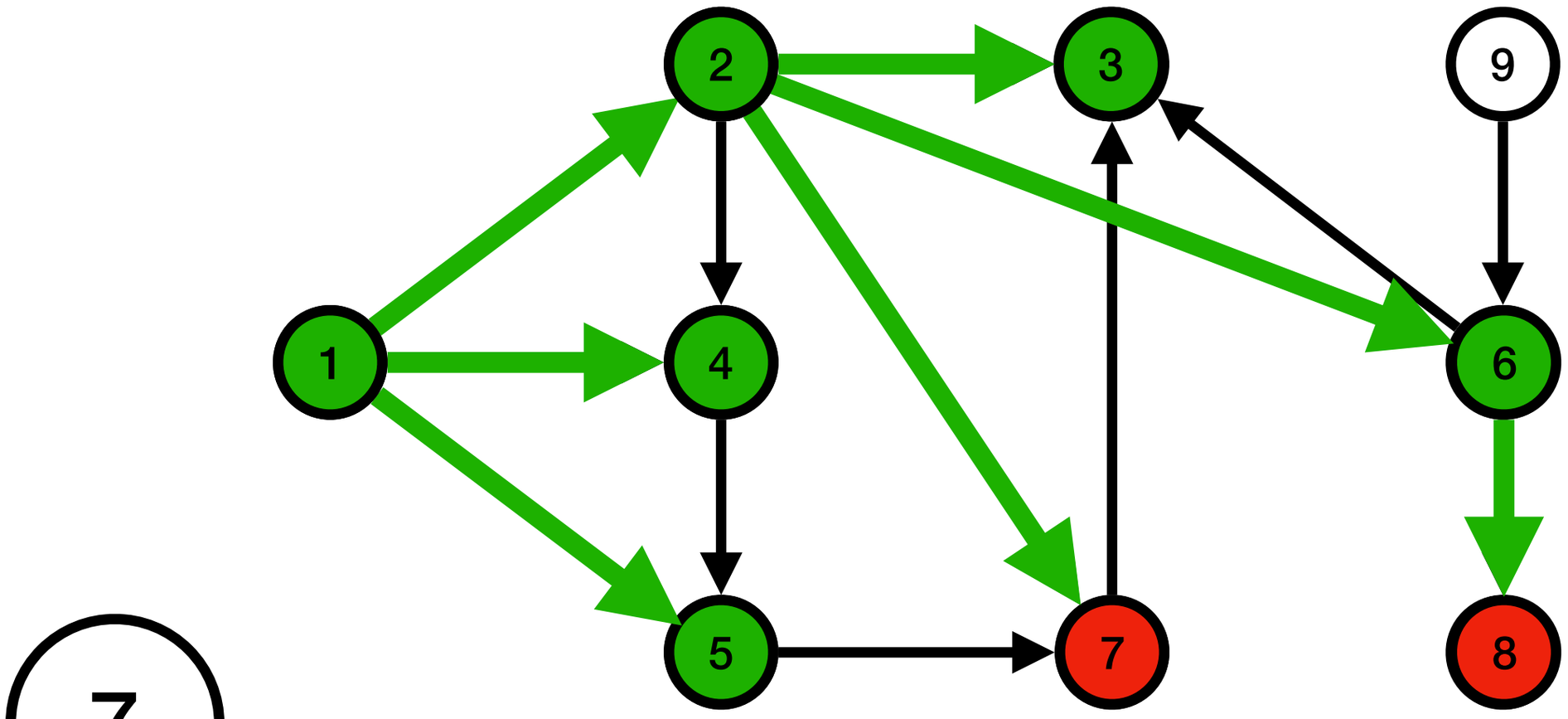
File → 7 8

Un autre exemple



File → 7 8

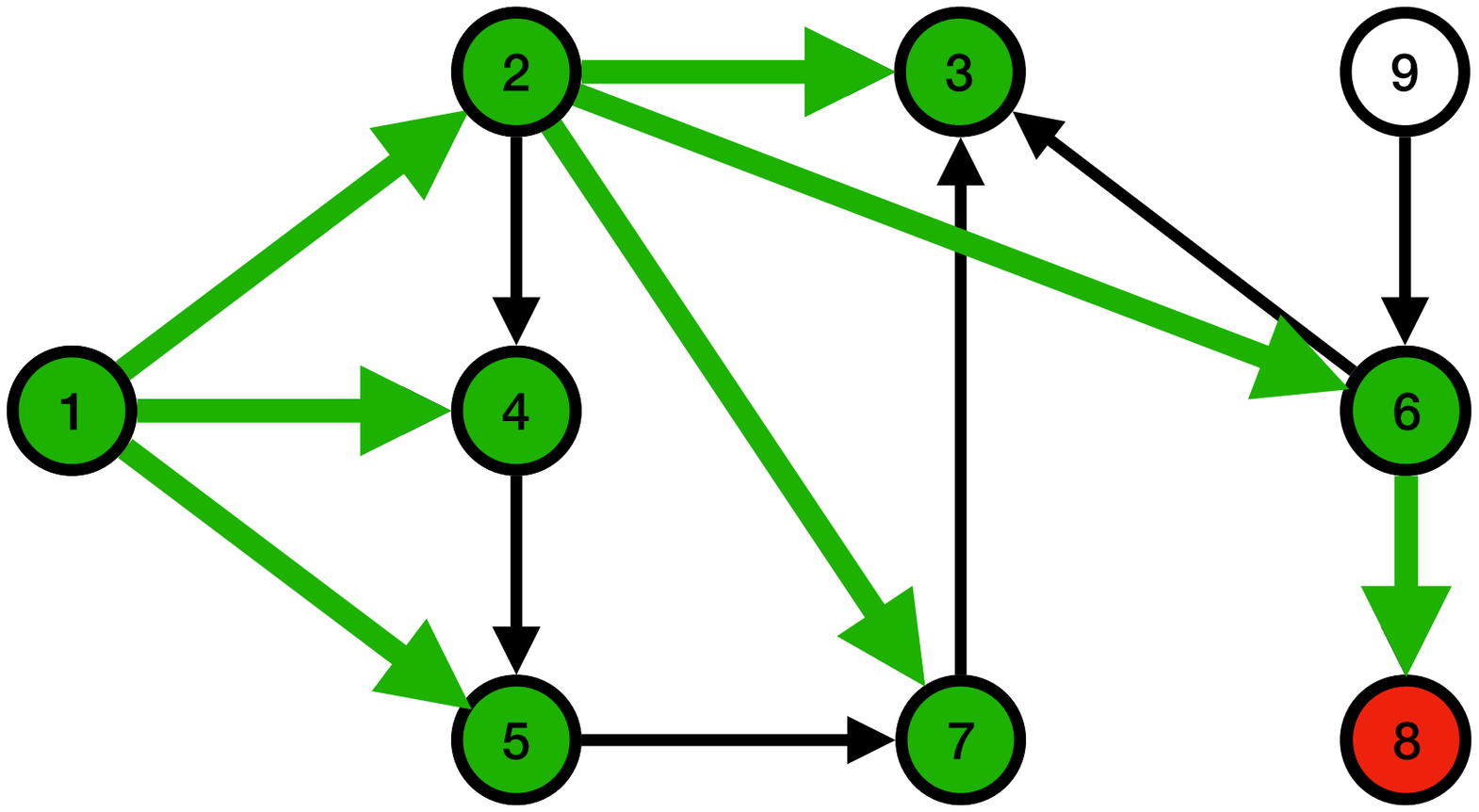
Un autre exemple



7

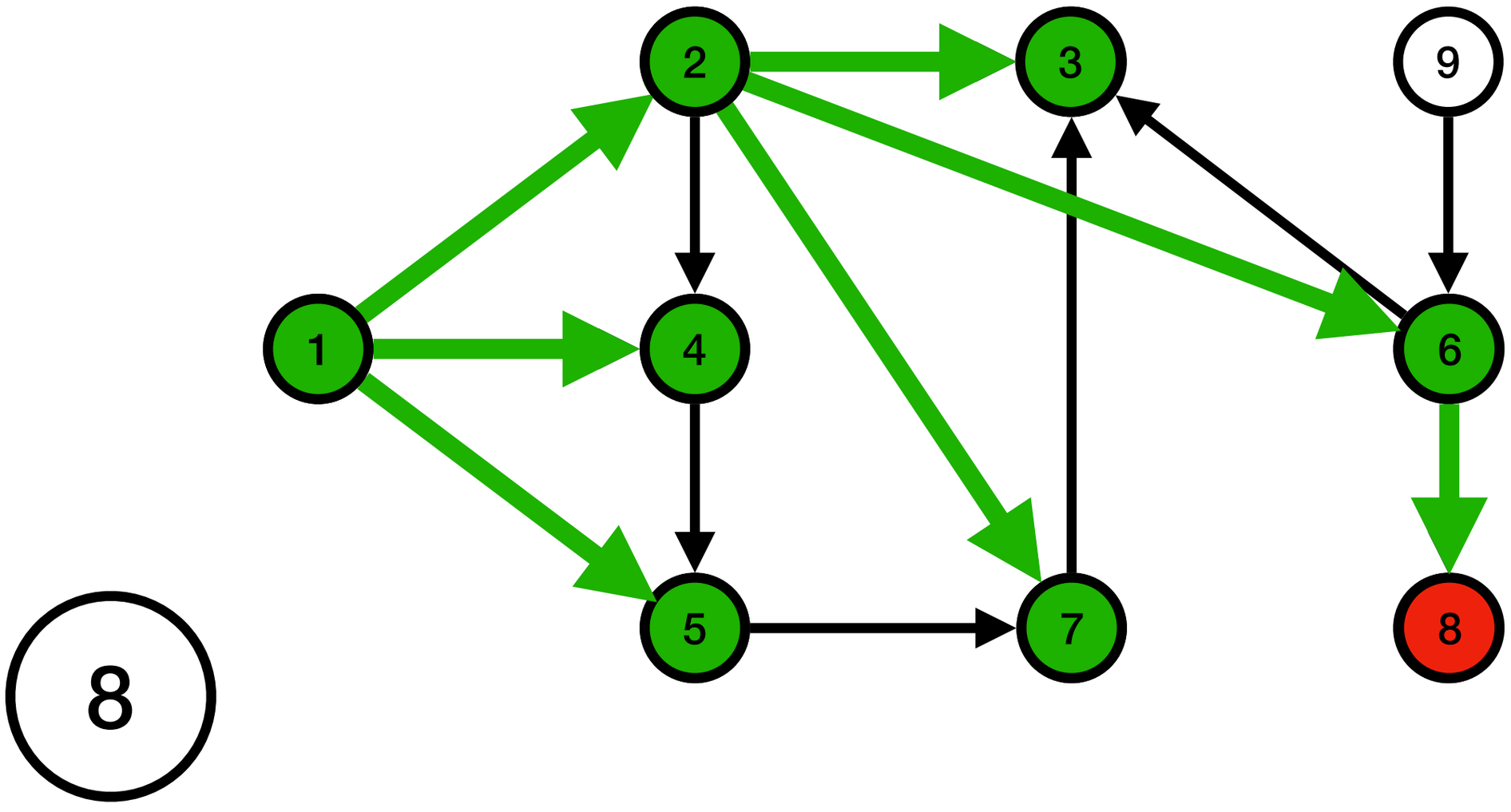
File → 8

Un autre exemple



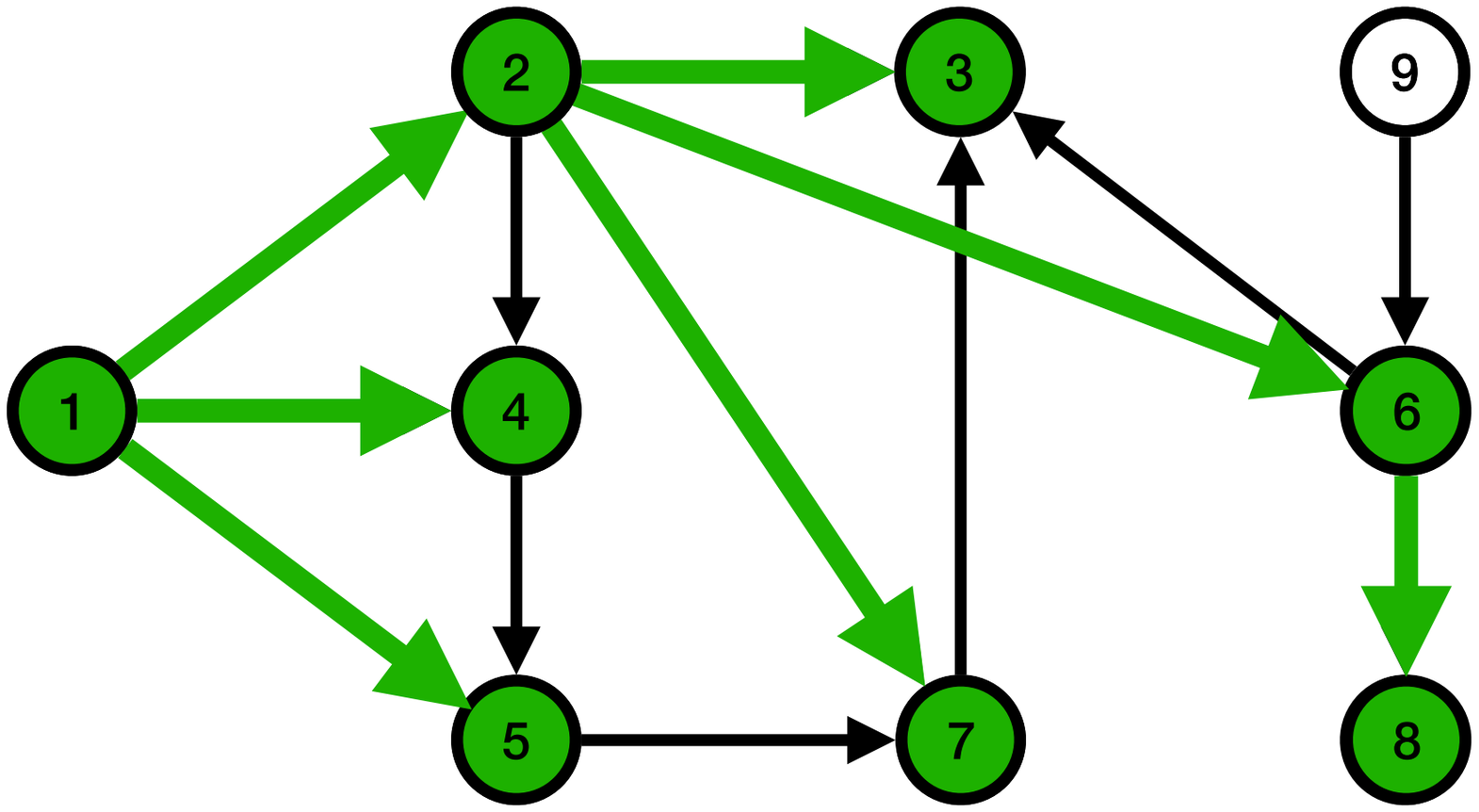
File → 8

Un autre exemple



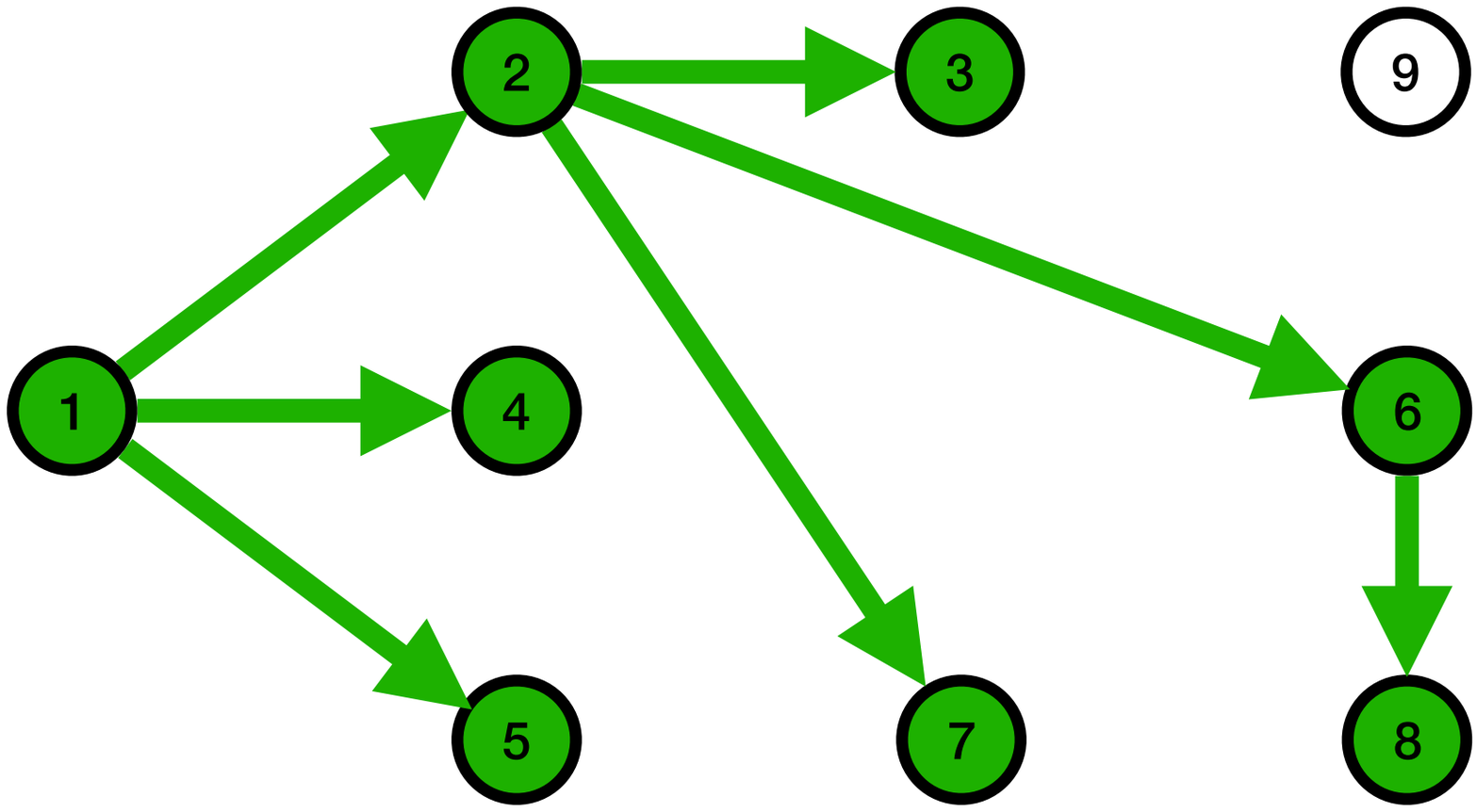
File →

Un autre exemple

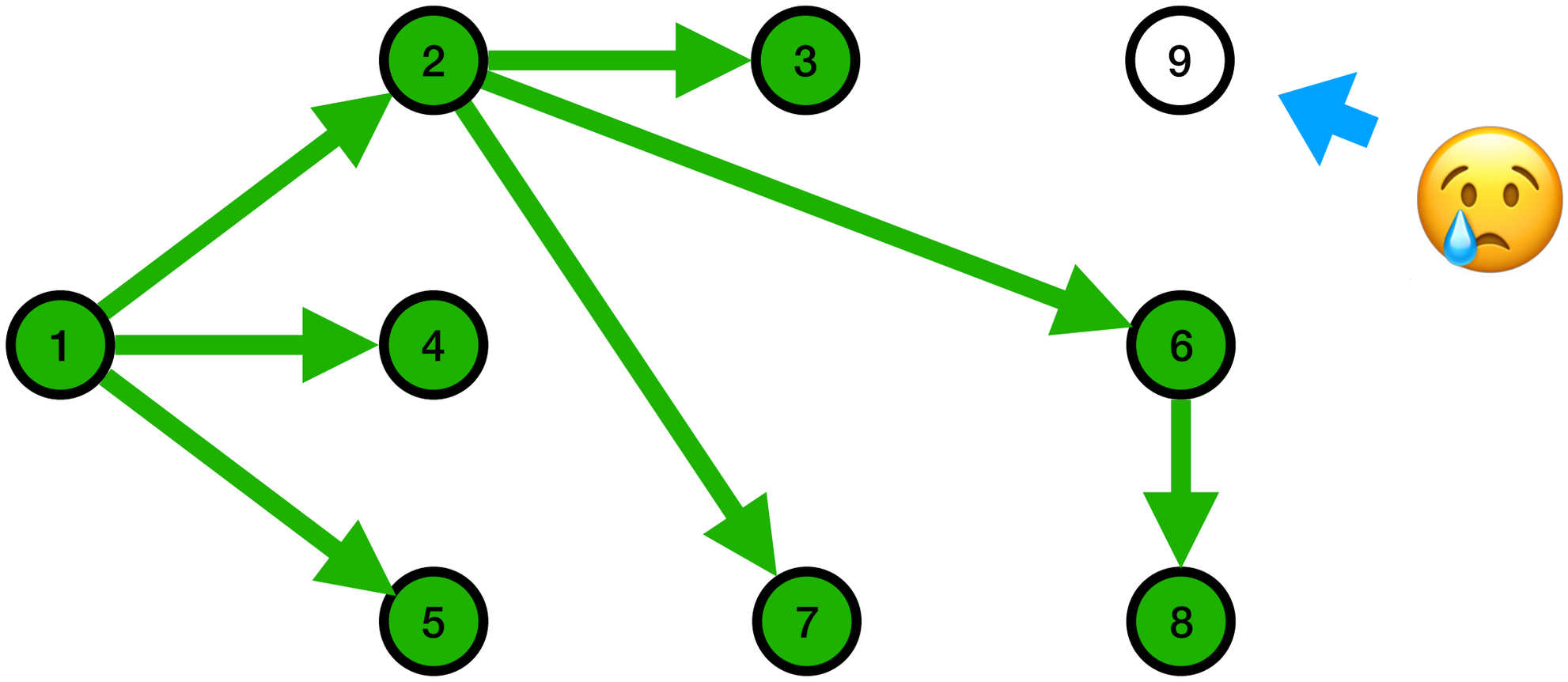


File →

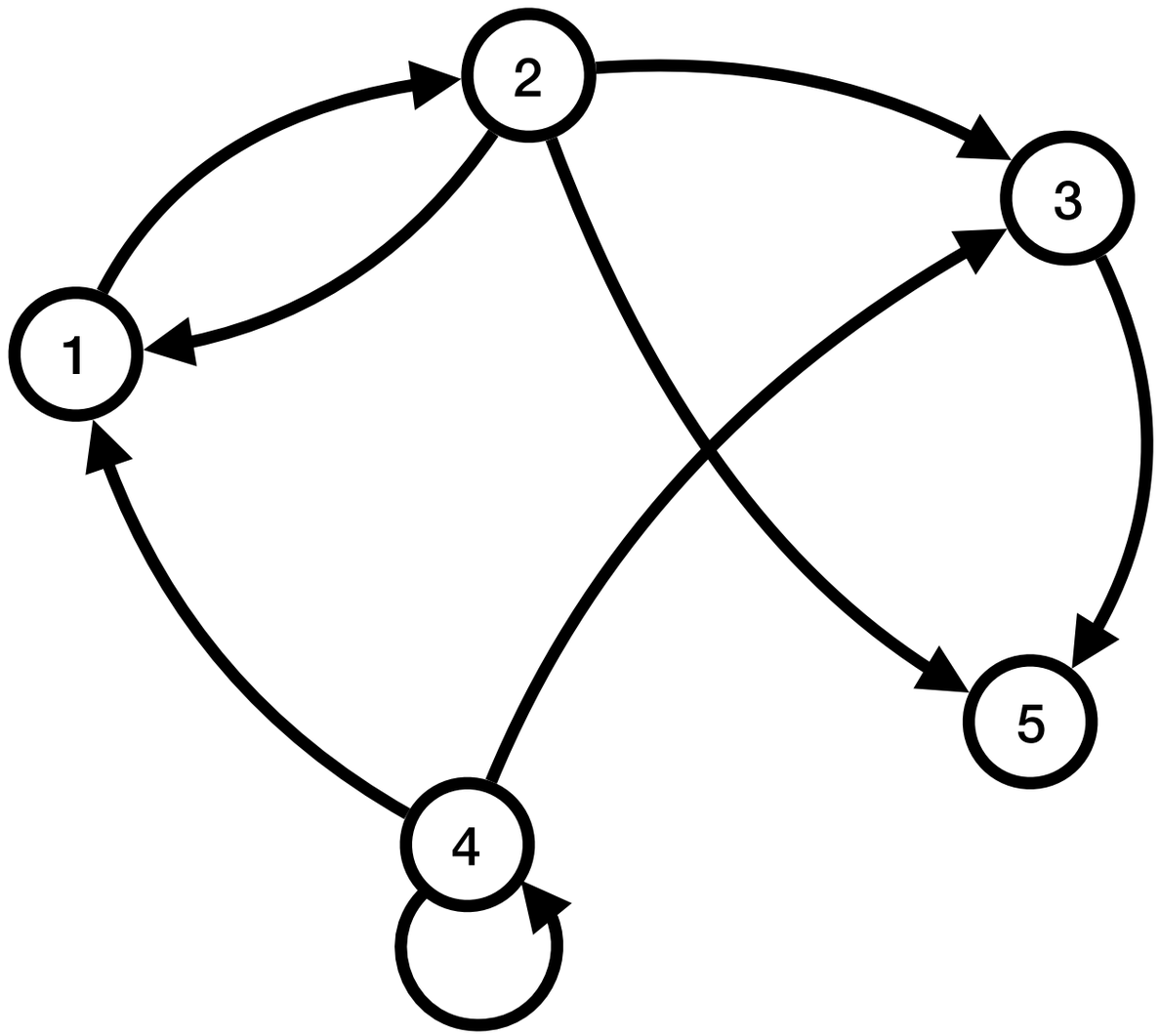
Un autre exemple



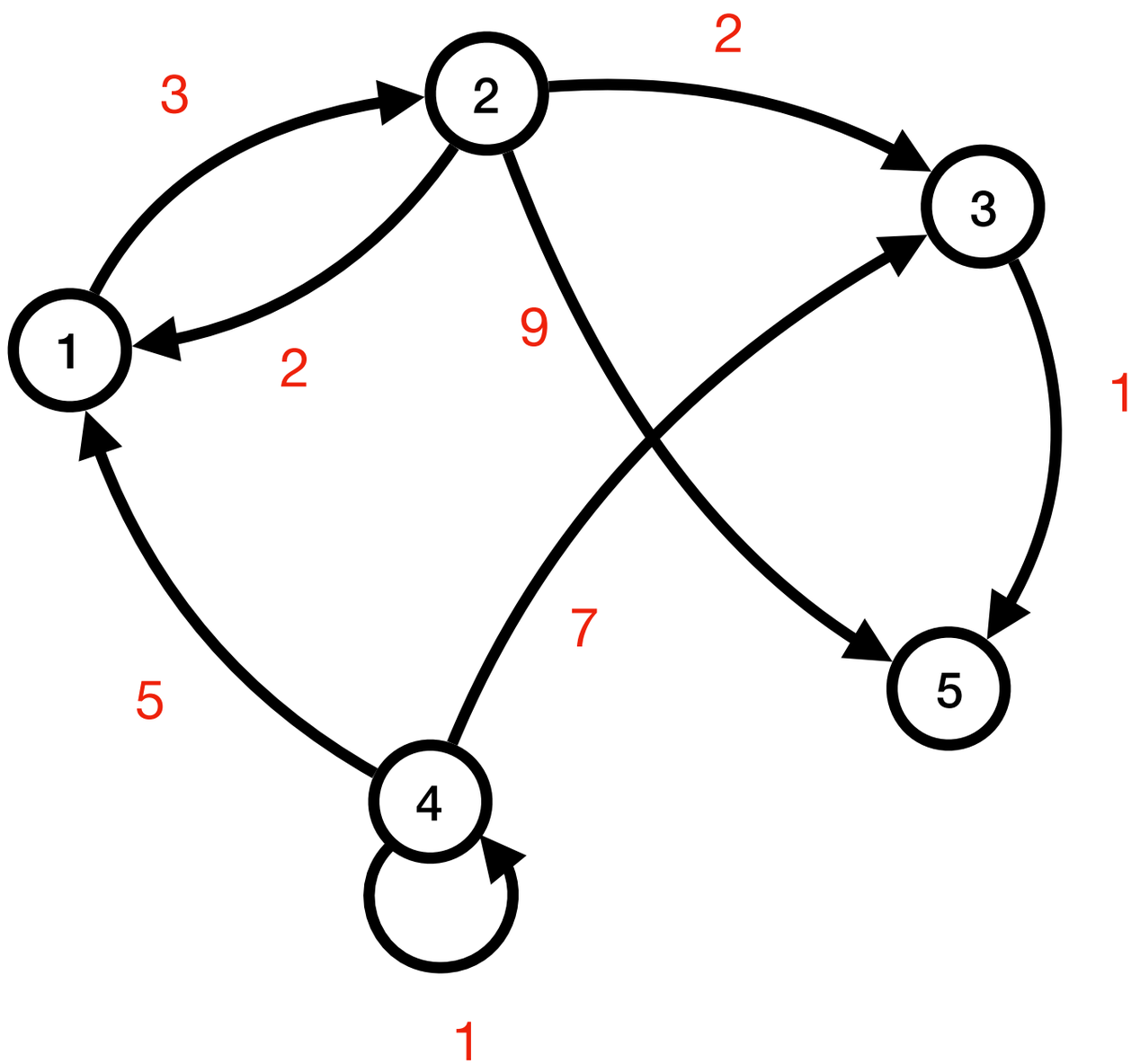
Un autre exemple



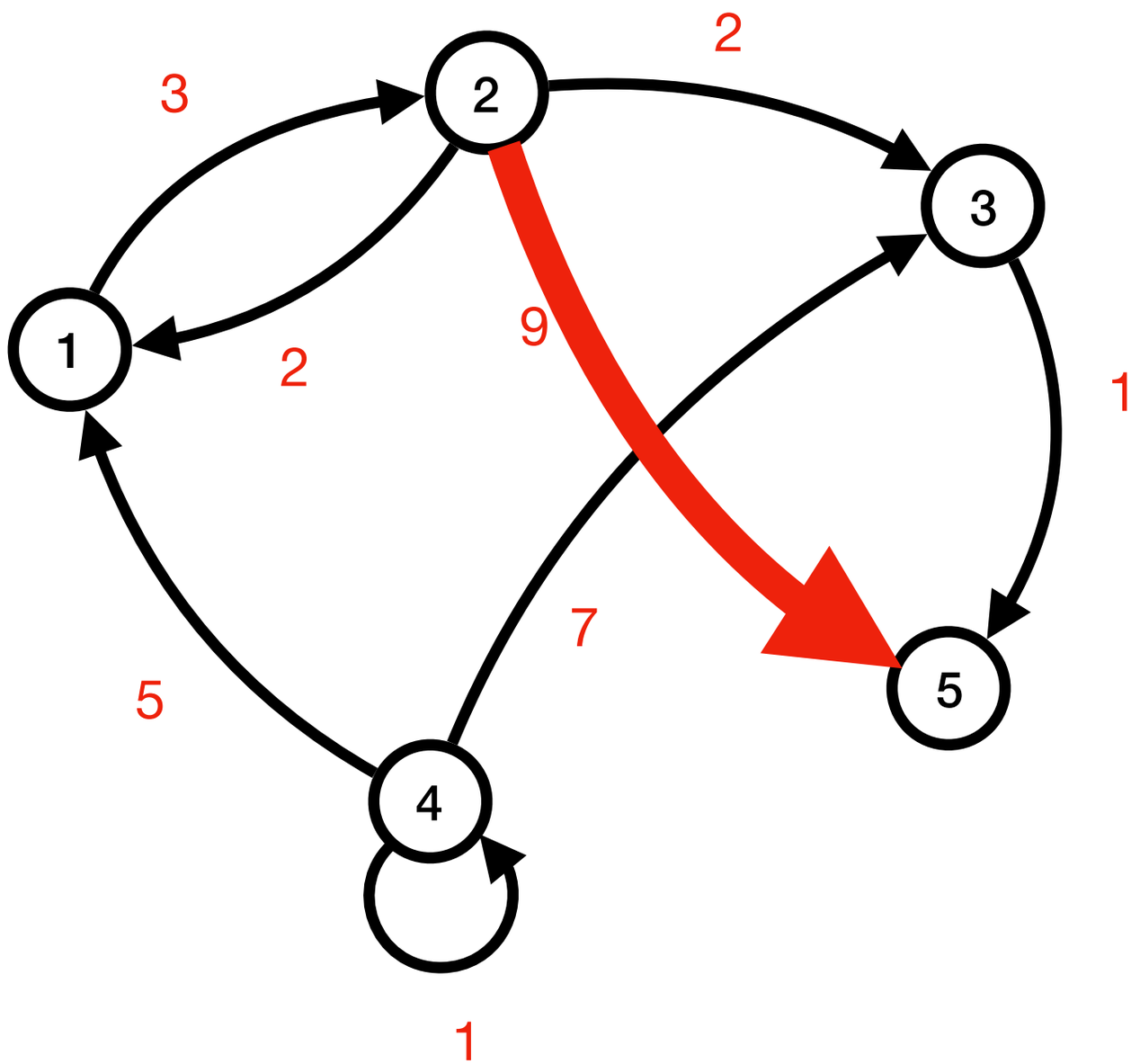
Graphes pondérés



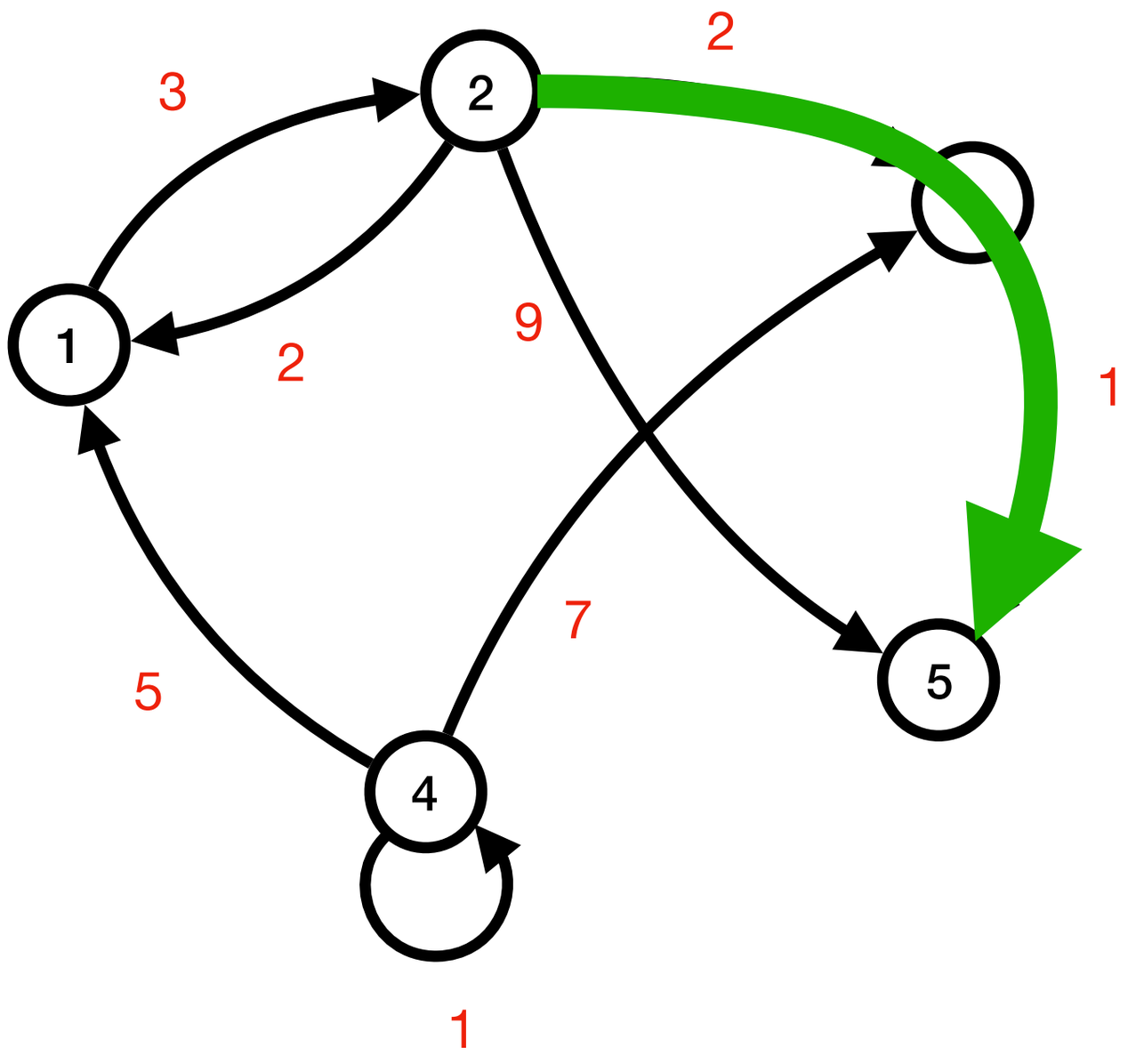
Graphes pondérés



Graphes pondérés

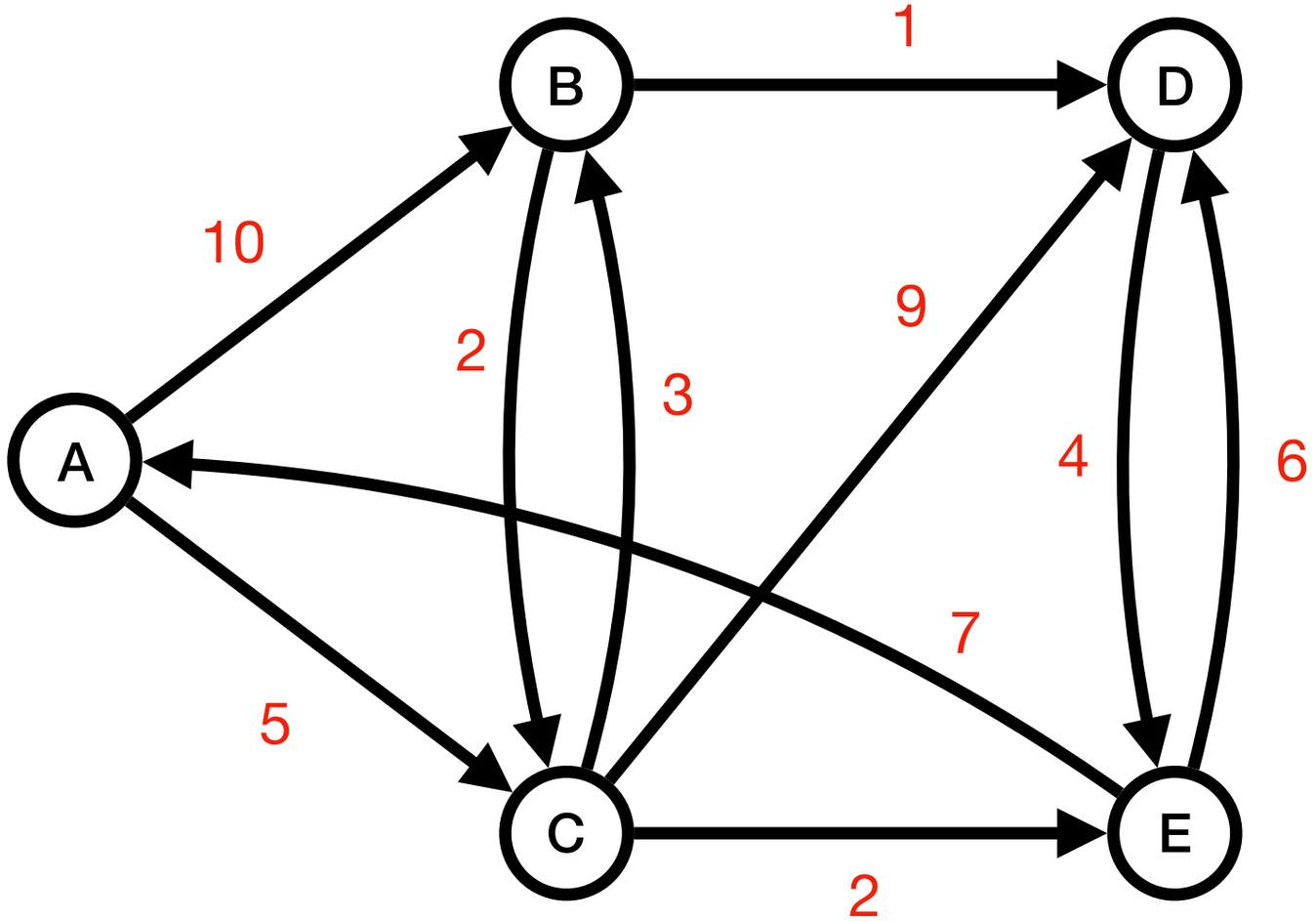


Graphes pondérés

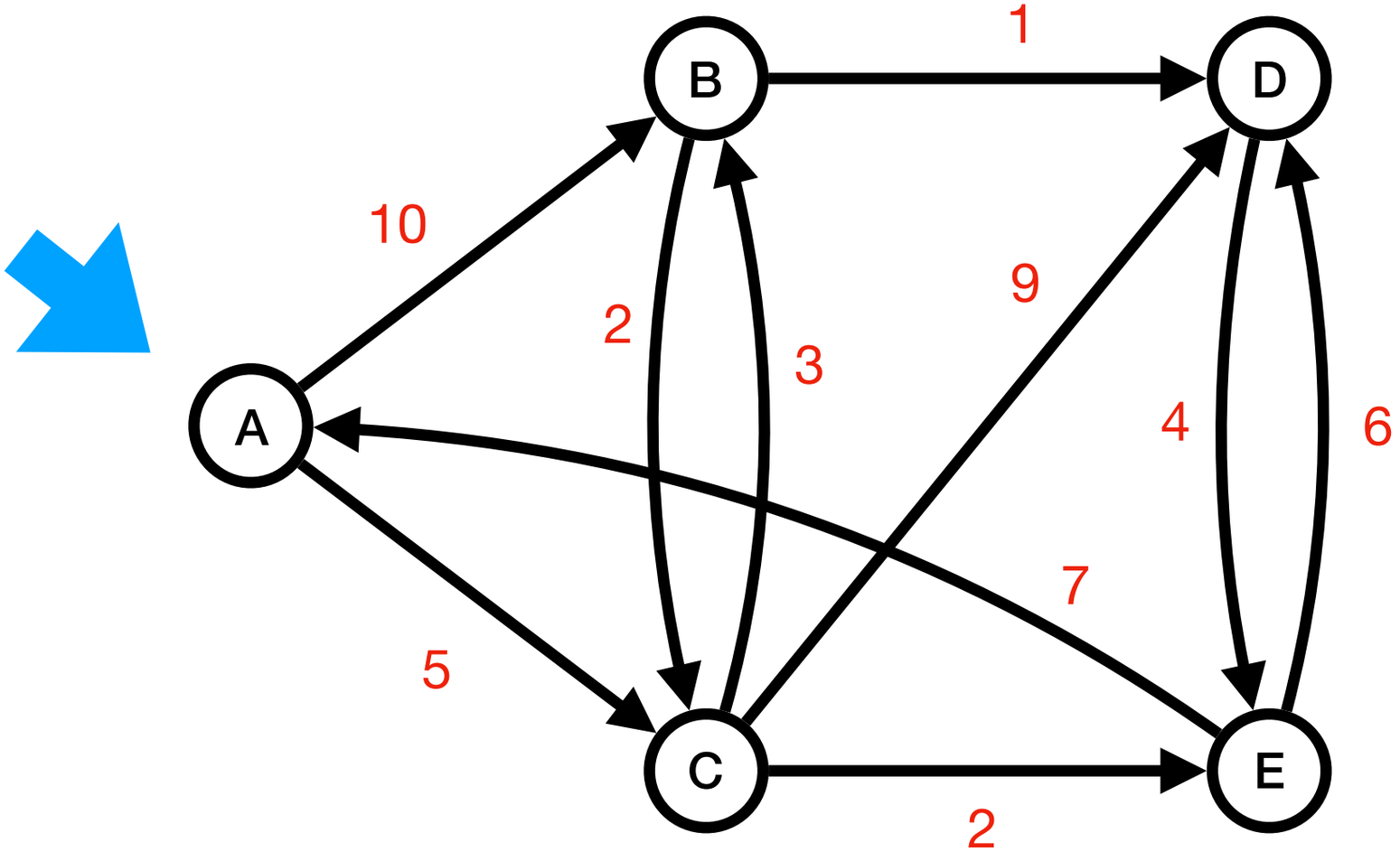


L’algorithme de parcours en largeur ne garantit pas d’obtenir un chemin minimal sur un graphe pondéré

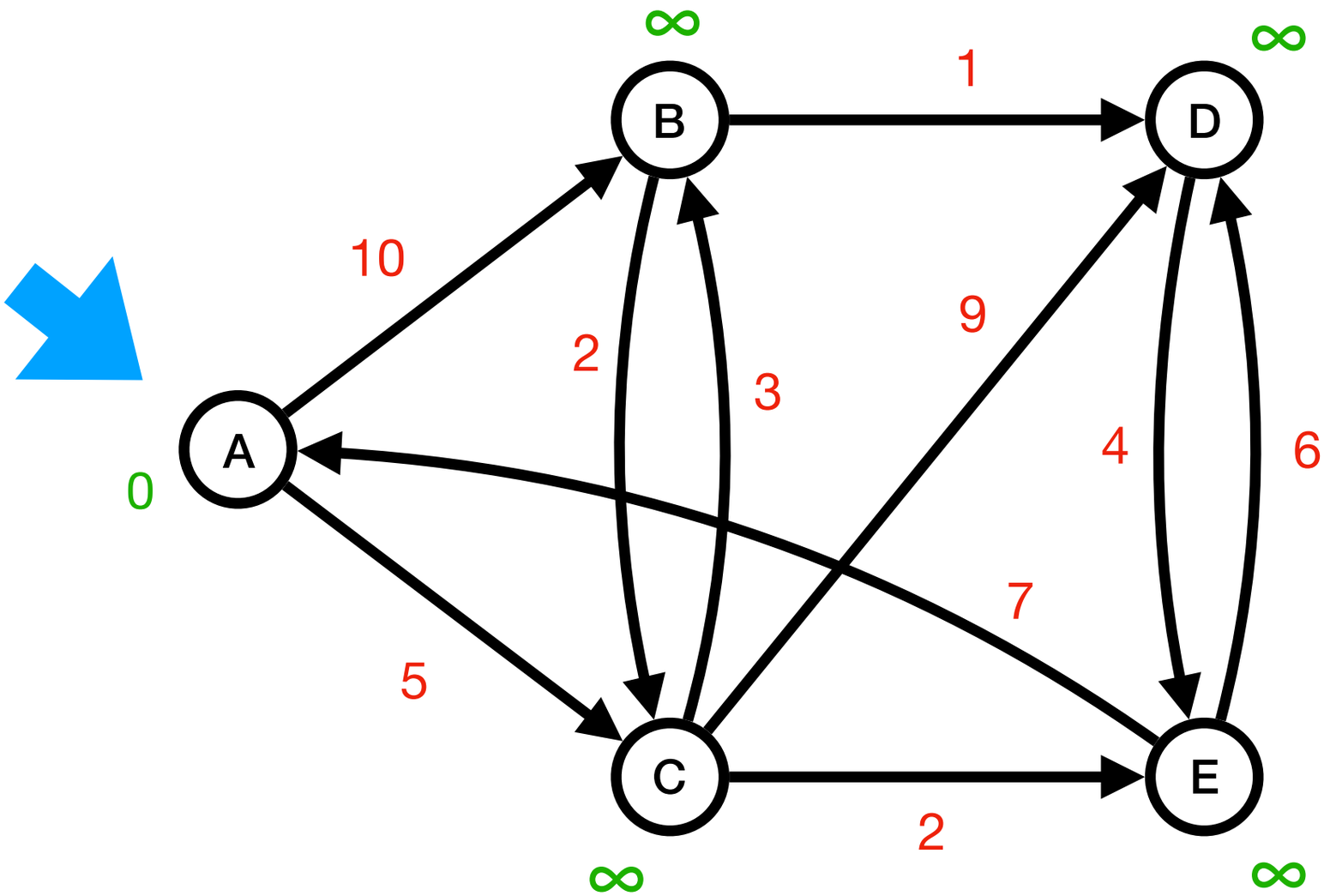
Algorithme de Dijkstra



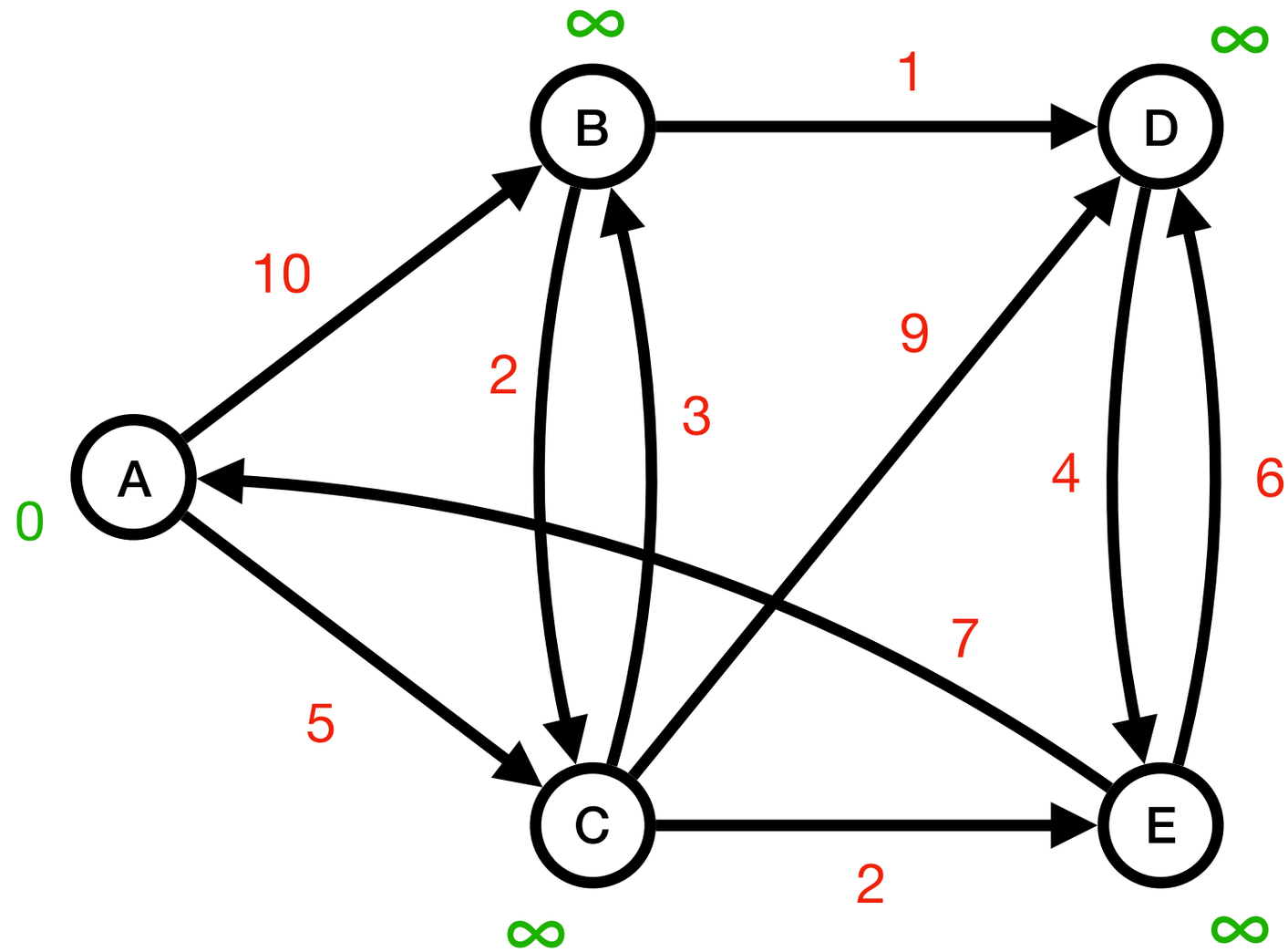
Algorithme de Dijkstra



Algorithme de Dijkstra



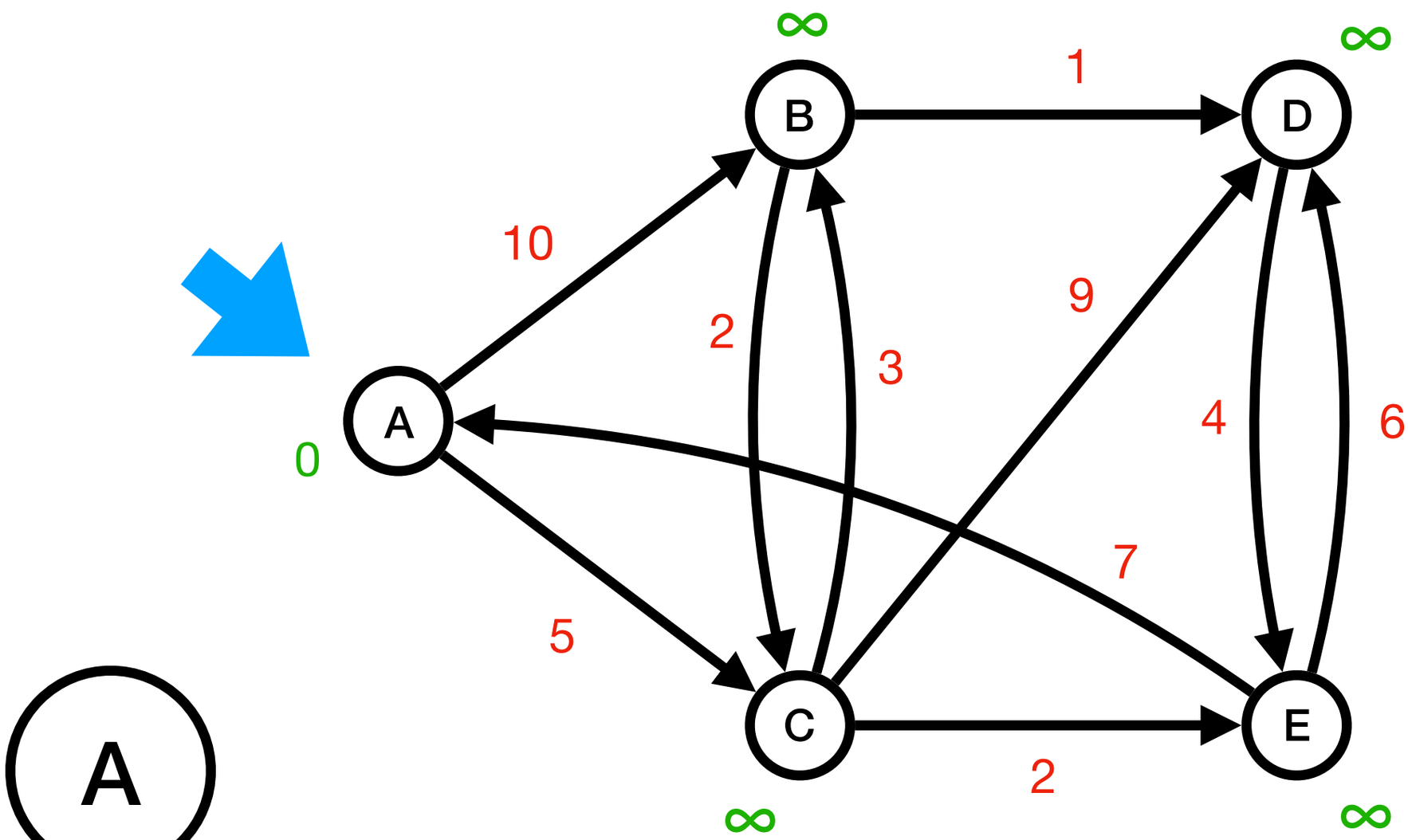
Algorithme de Dijkstra



File de priorité → A

0

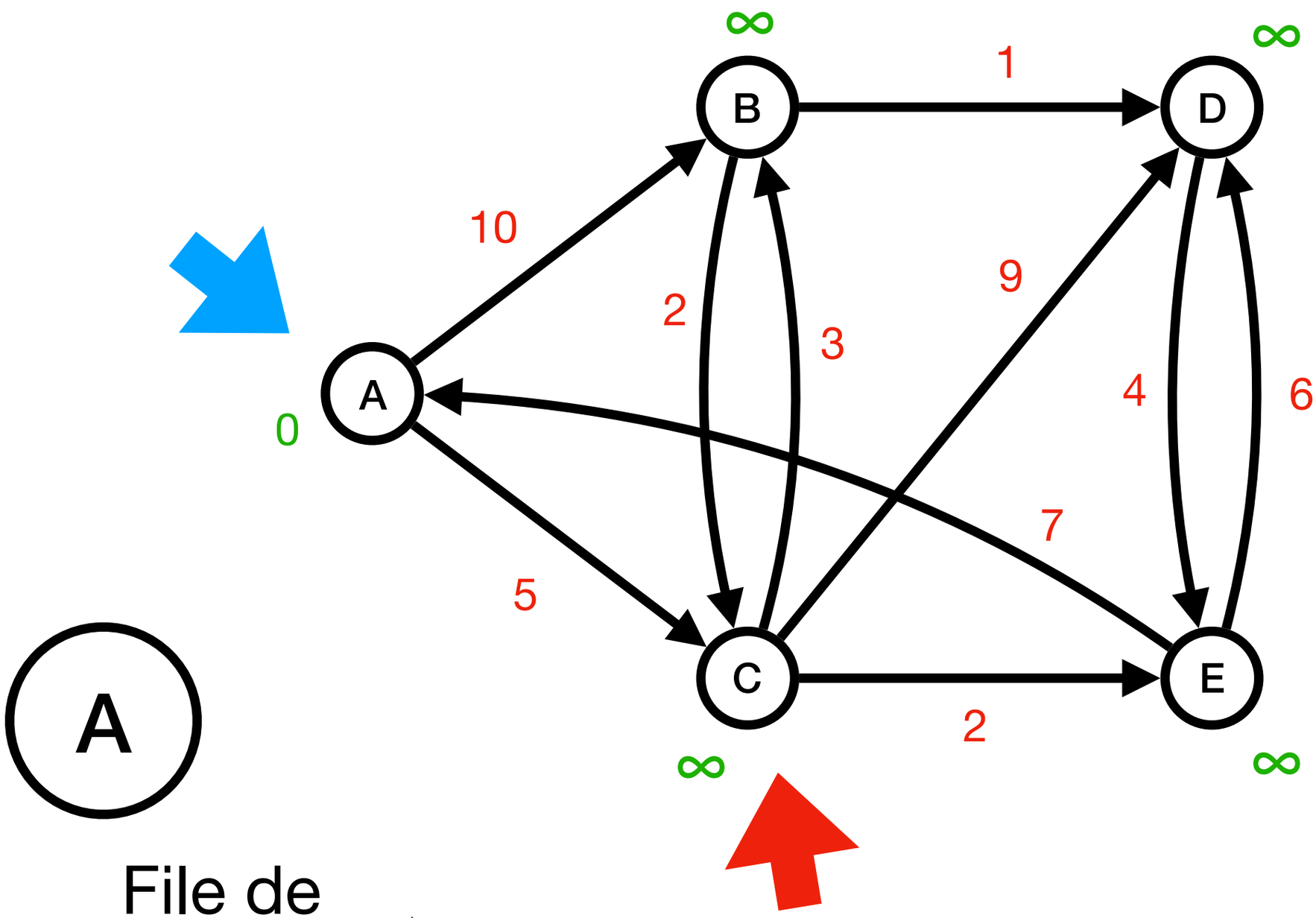
Algorithme de Dijkstra



A

File de priorité →

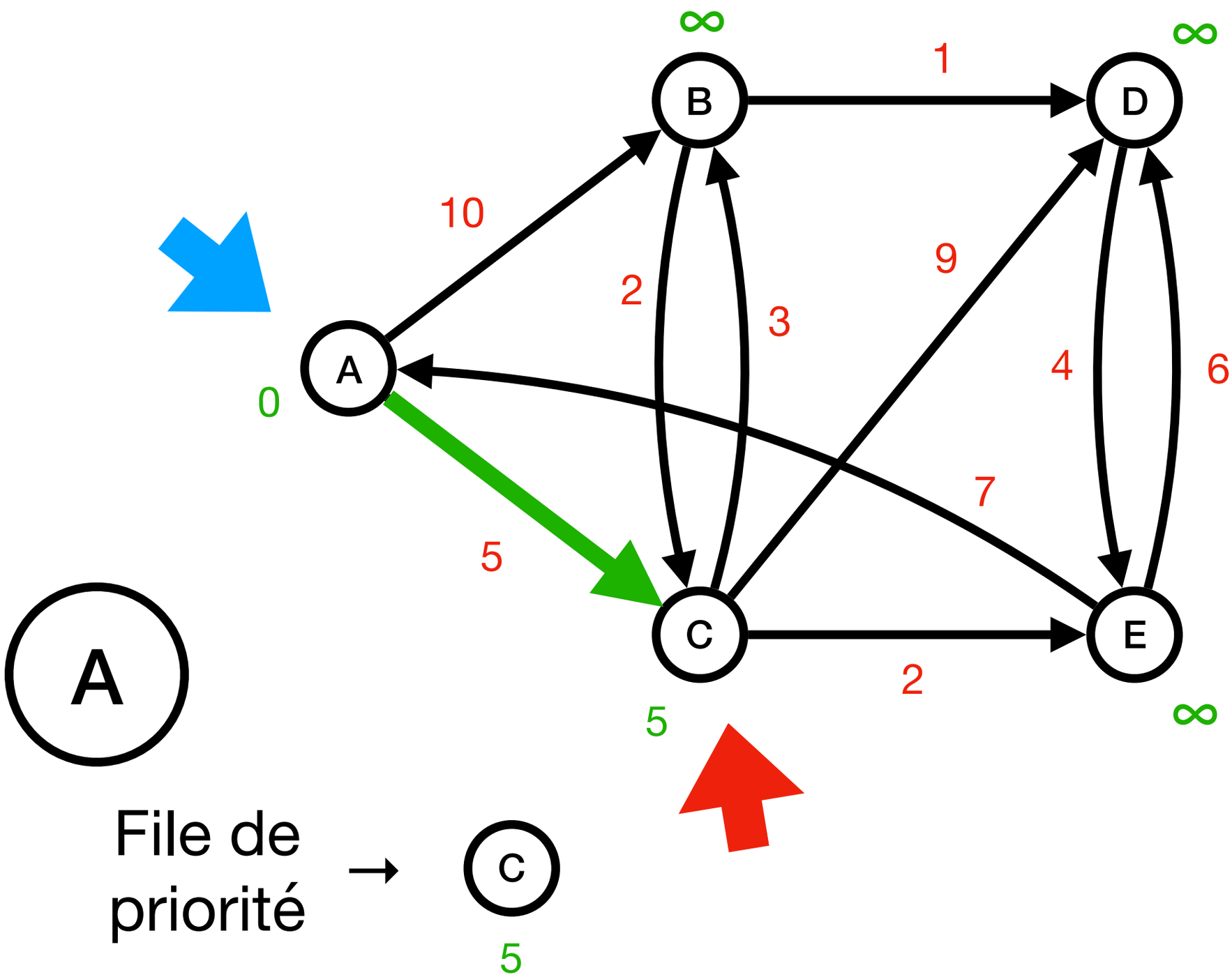
Algorithme de Dijkstra



A

File de priorité →

Algorithme de Dijkstra

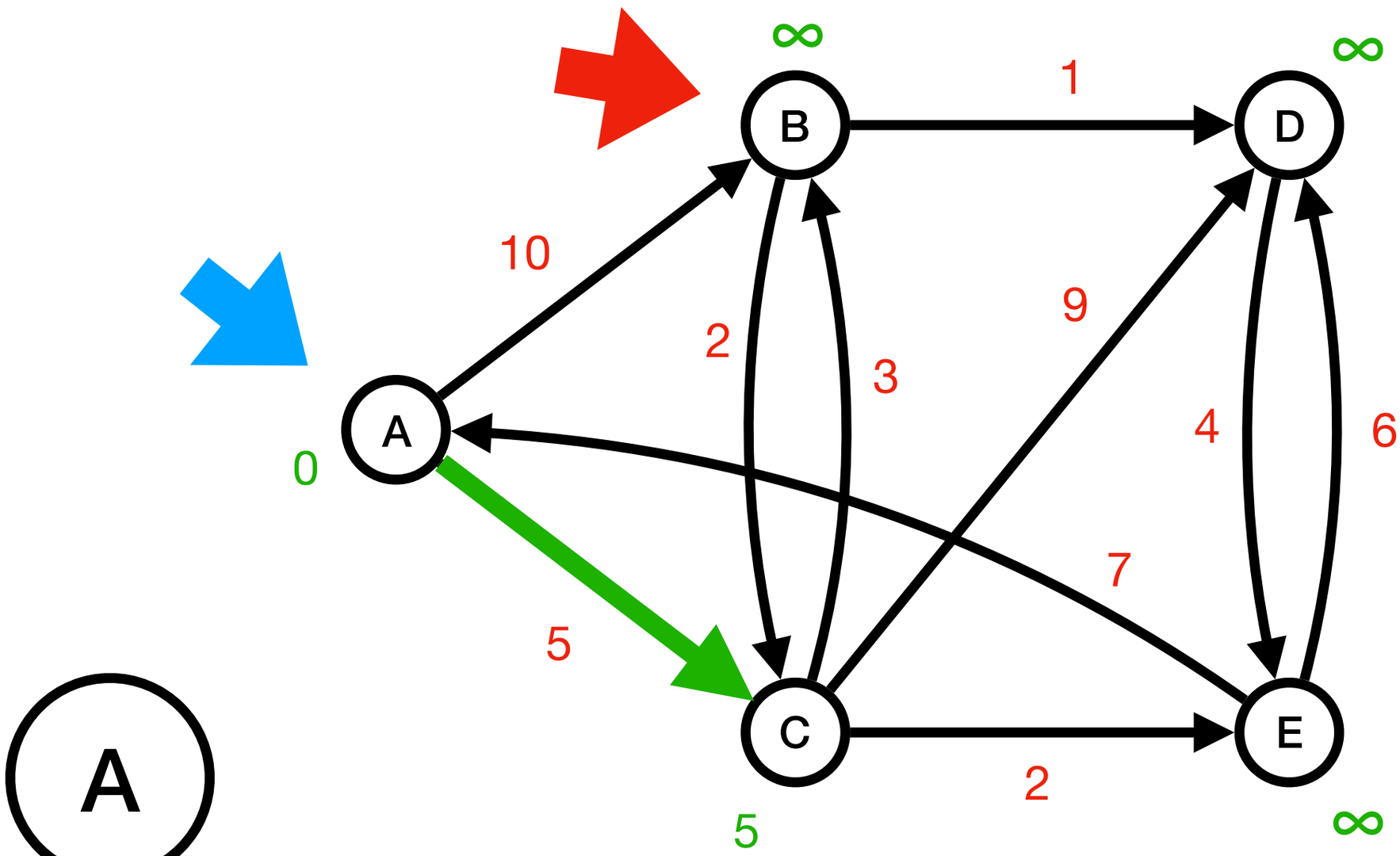


A

File de priorité → C

5

Algorithme de Dijkstra

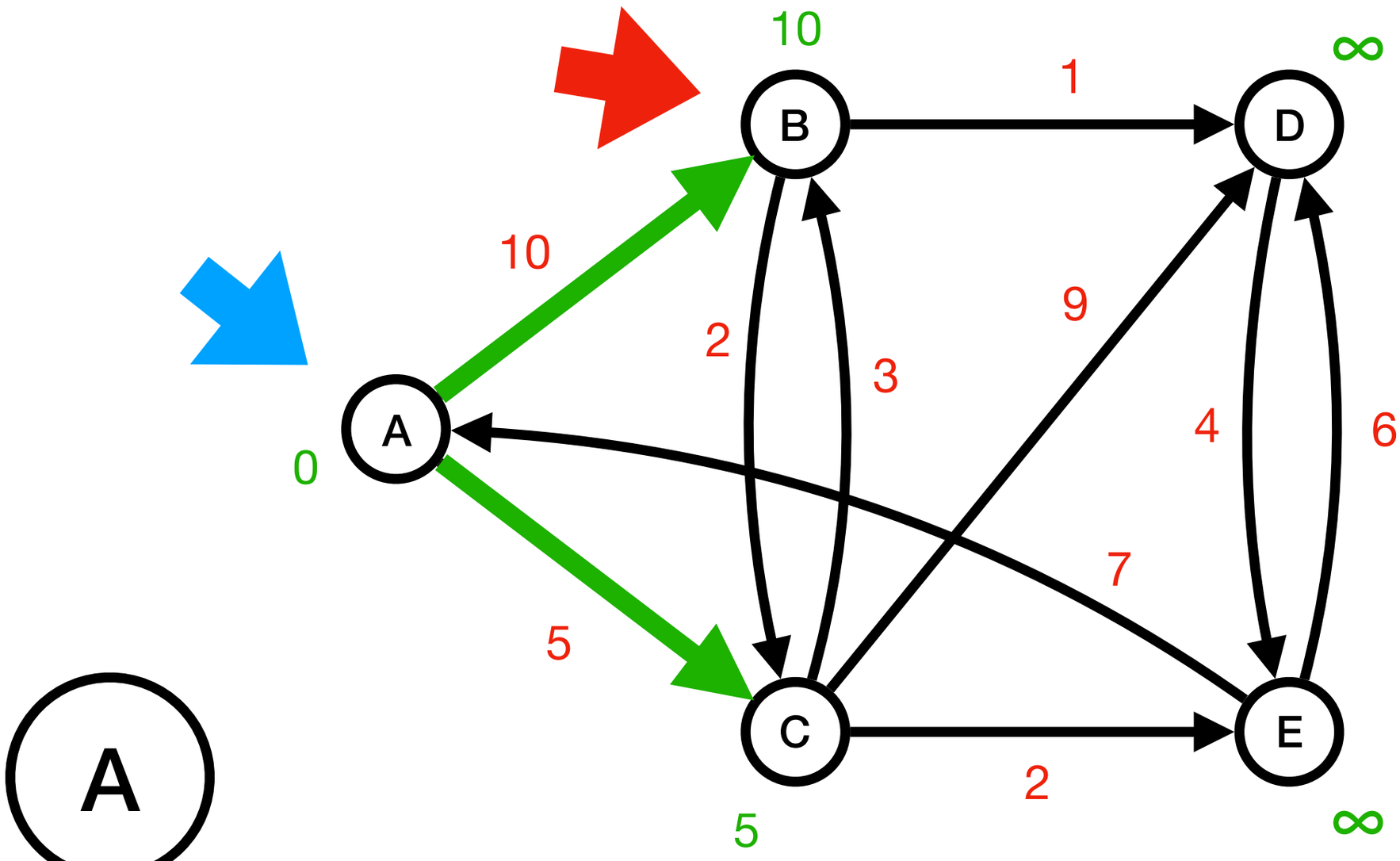


A

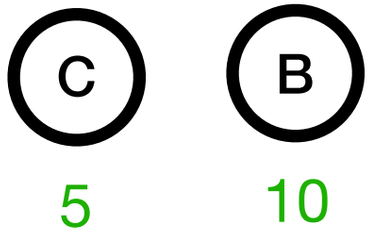
File de priorité → C

5

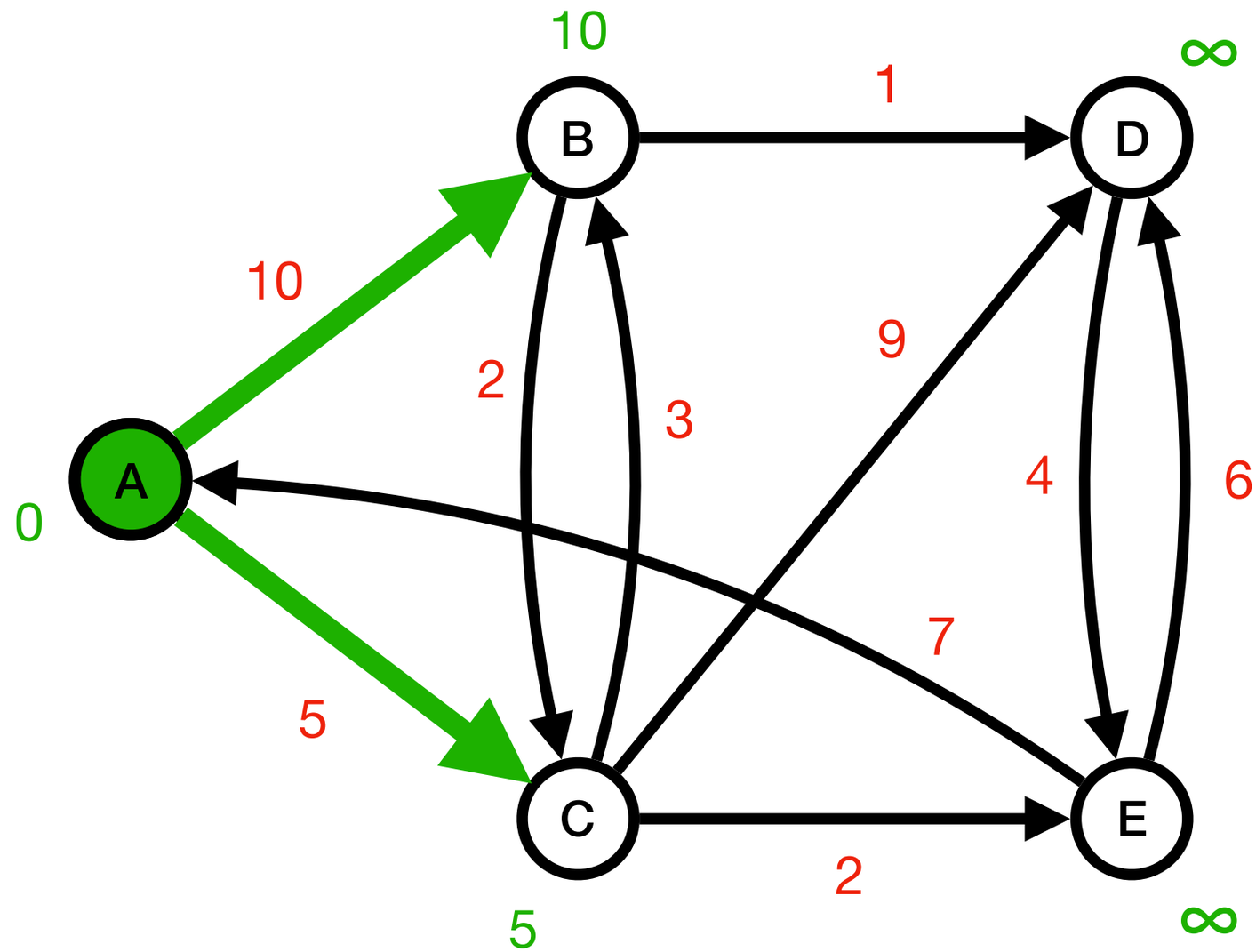
Algorithme de Dijkstra

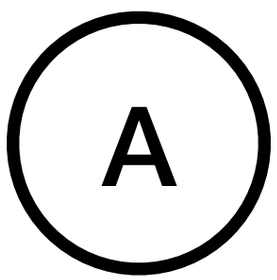


File de priorité →

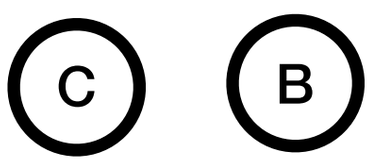


Algorithme de Dijkstra

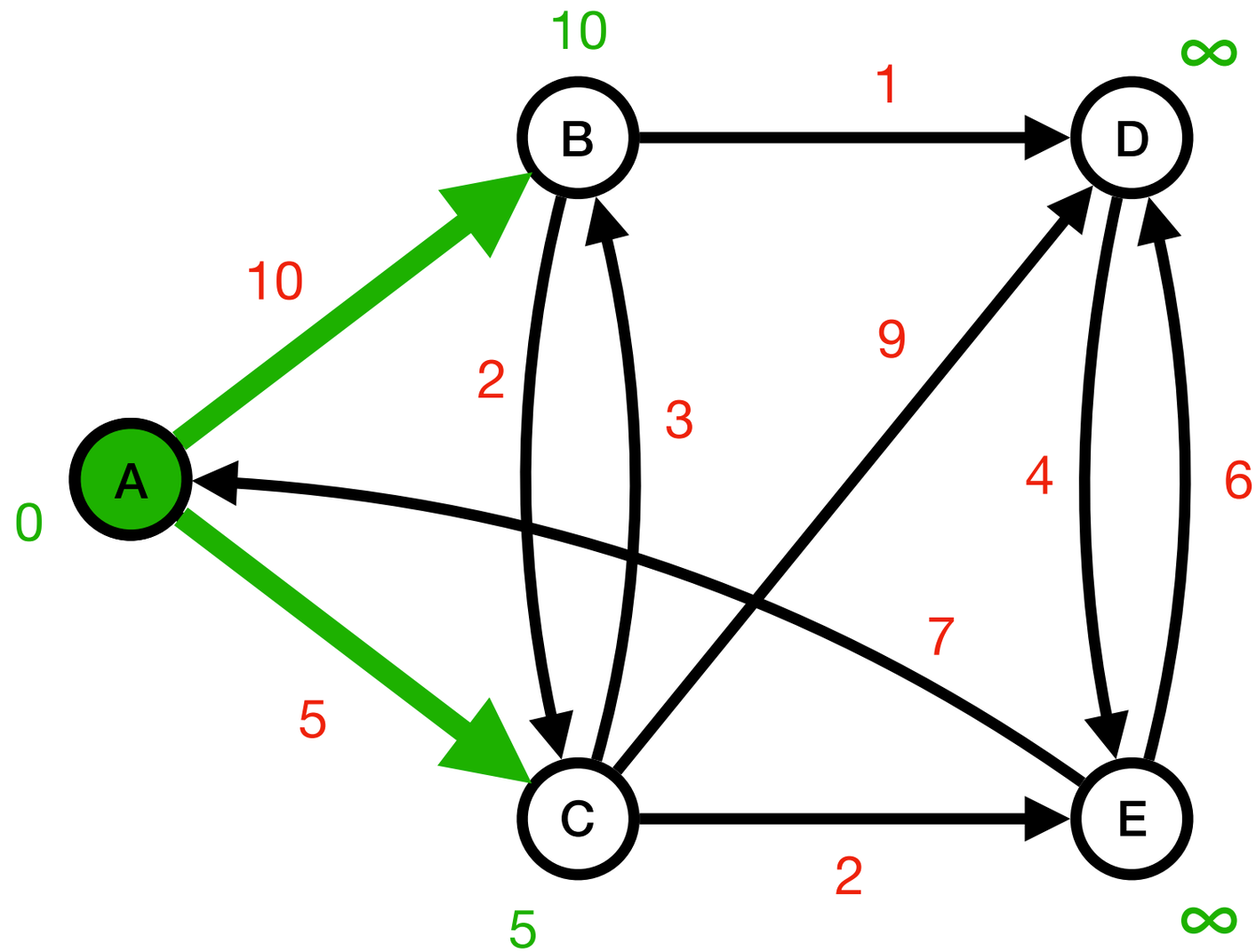




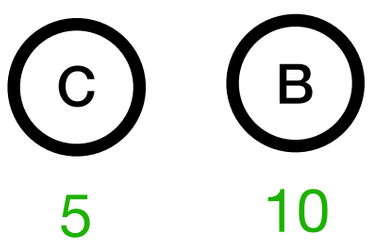
File de priorité →



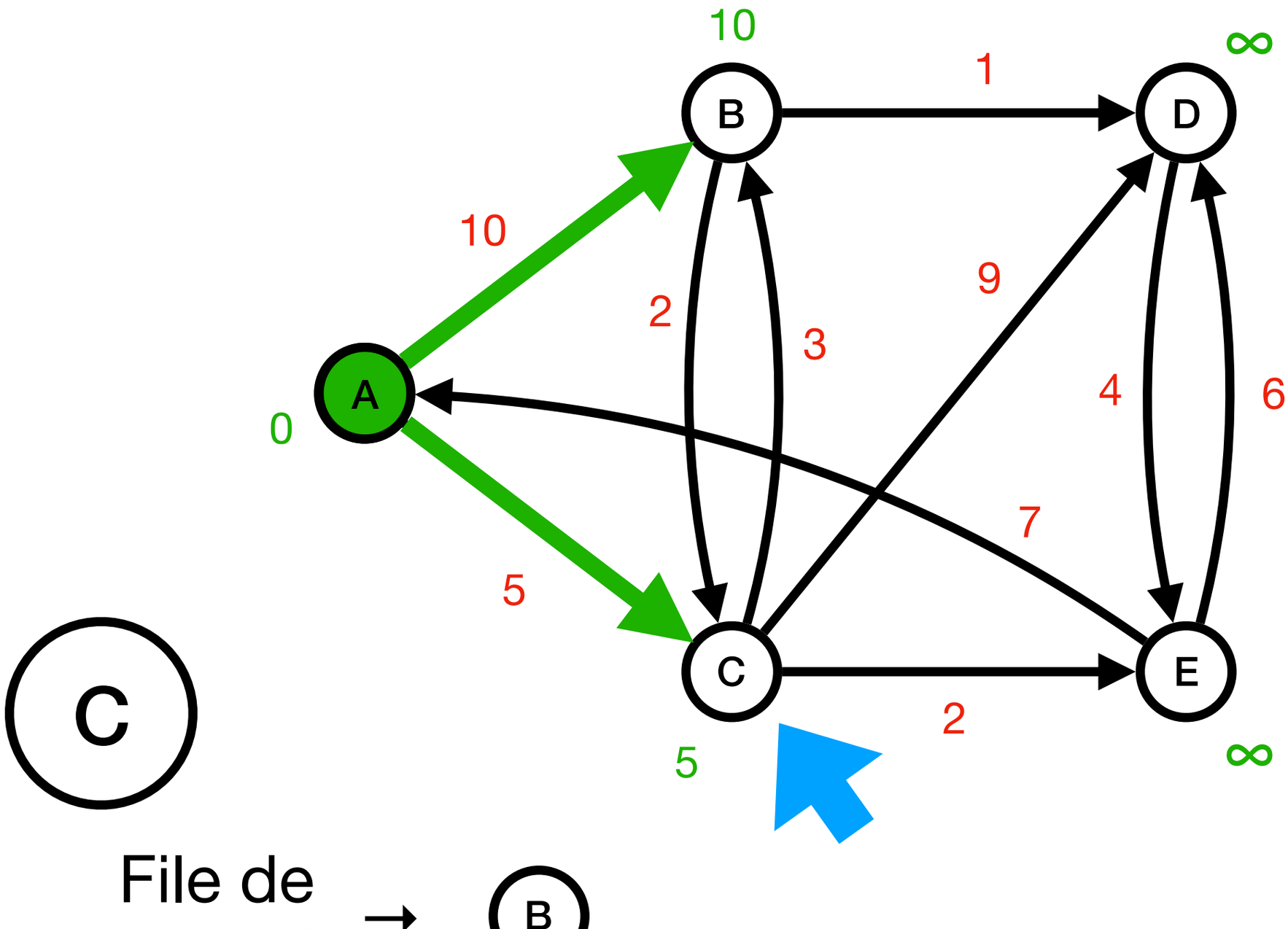
Algorithme de Dijkstra



File de priorité →



Algorithme de Dijkstra

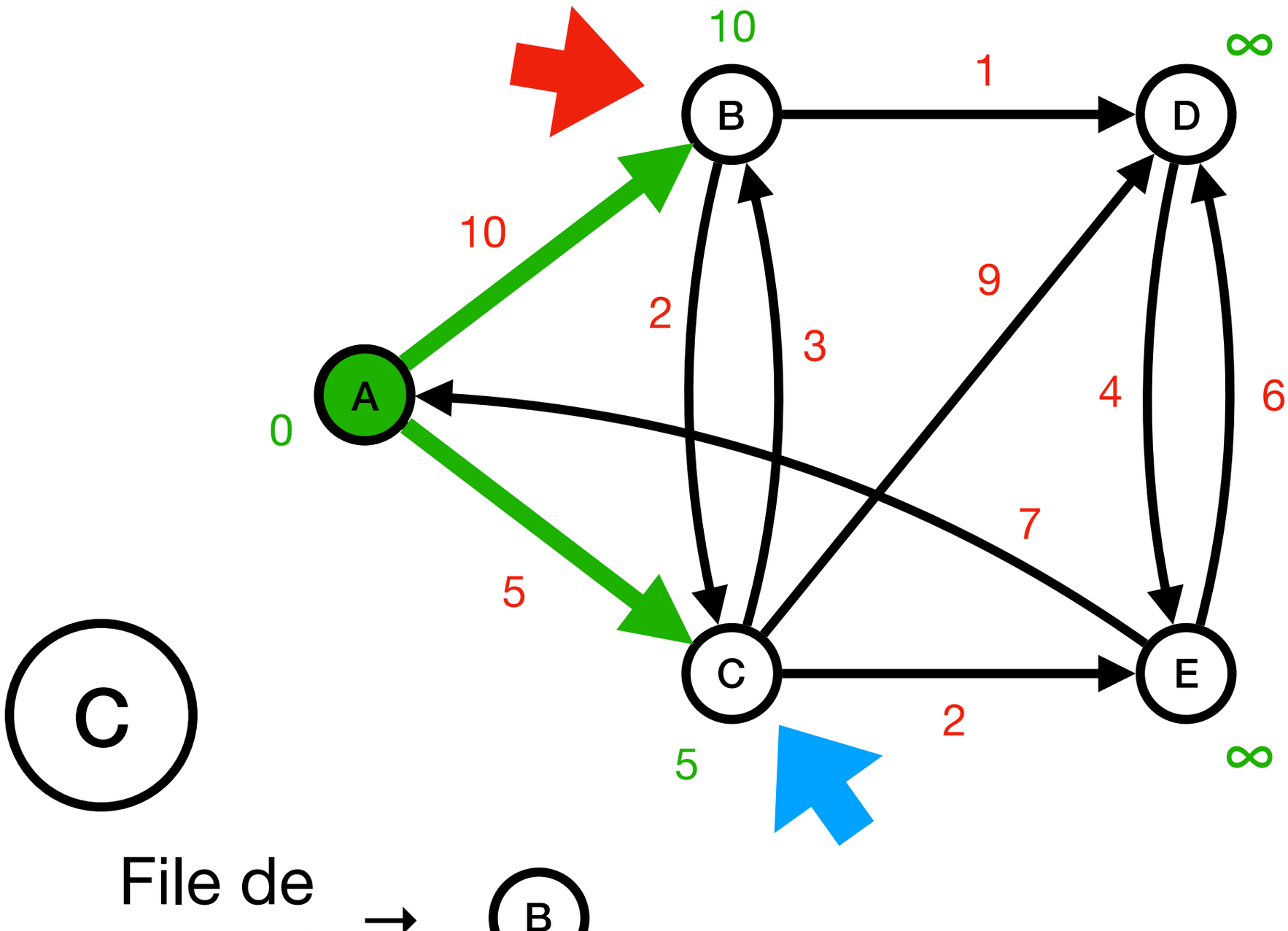


C

File de priorité → B

10

Algorithme de Dijkstra

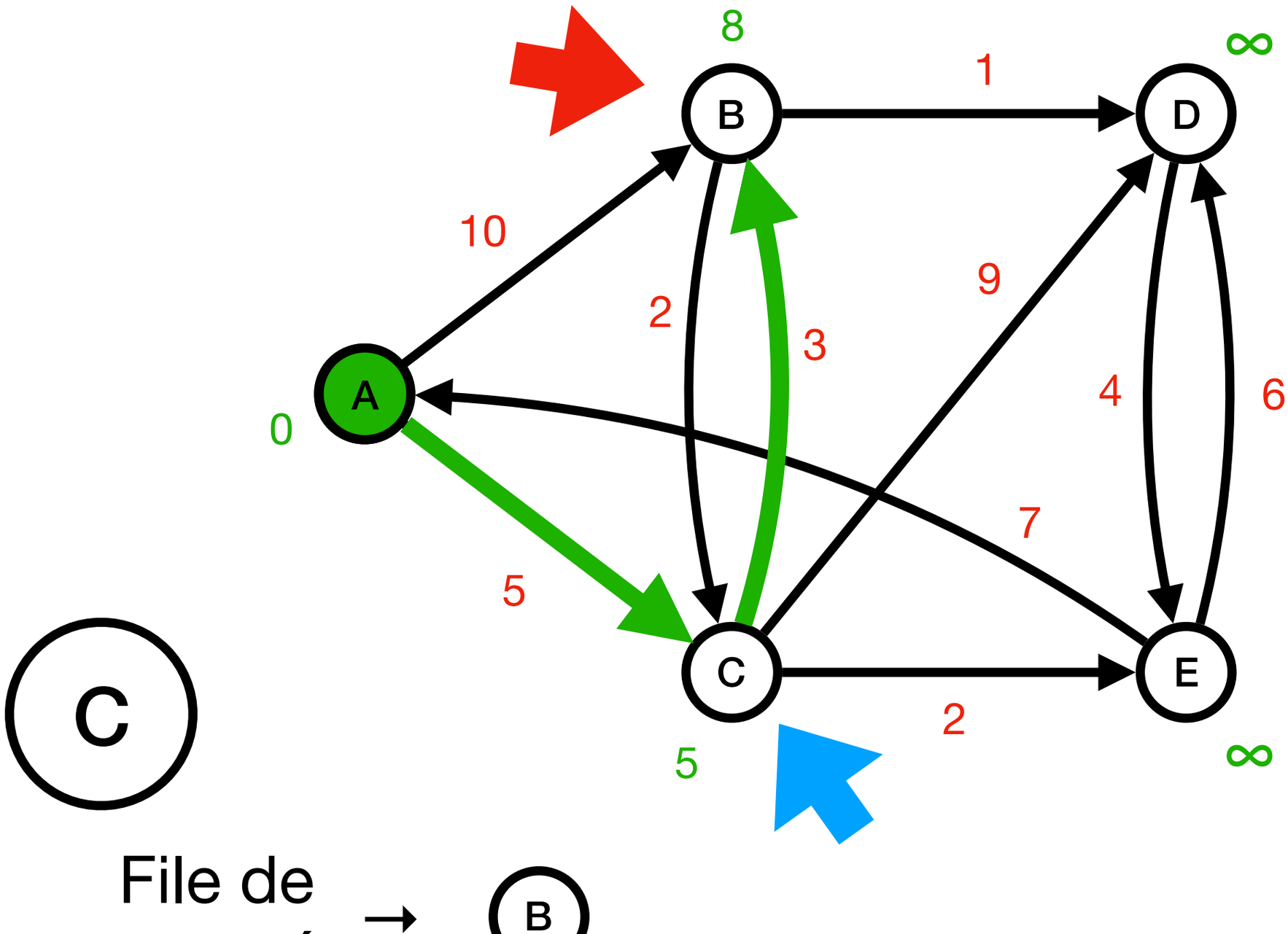


C

File de priorité → B

10

Algorithme de Dijkstra

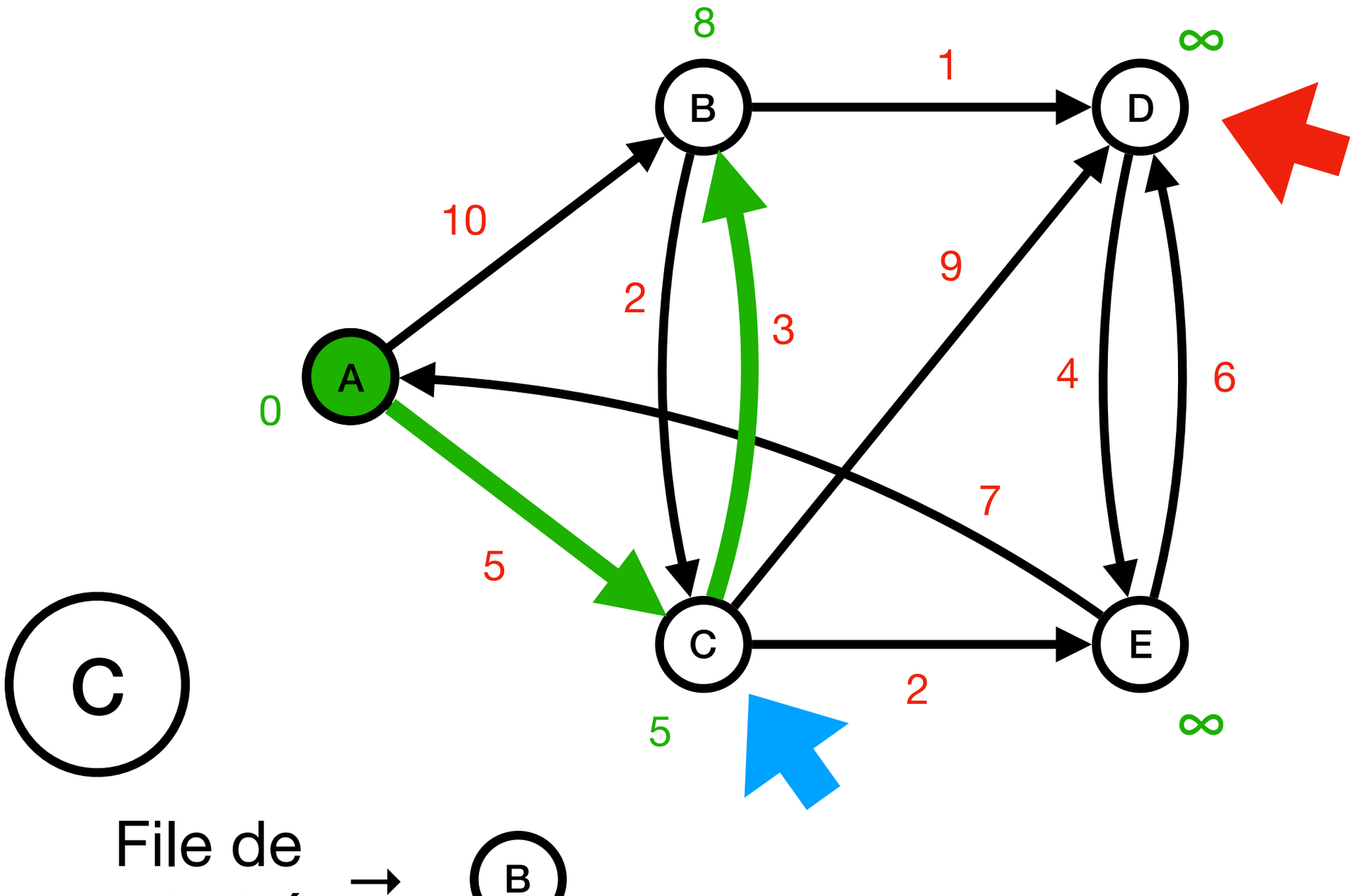


C

File de priorité → B

8

Algorithme de Dijkstra

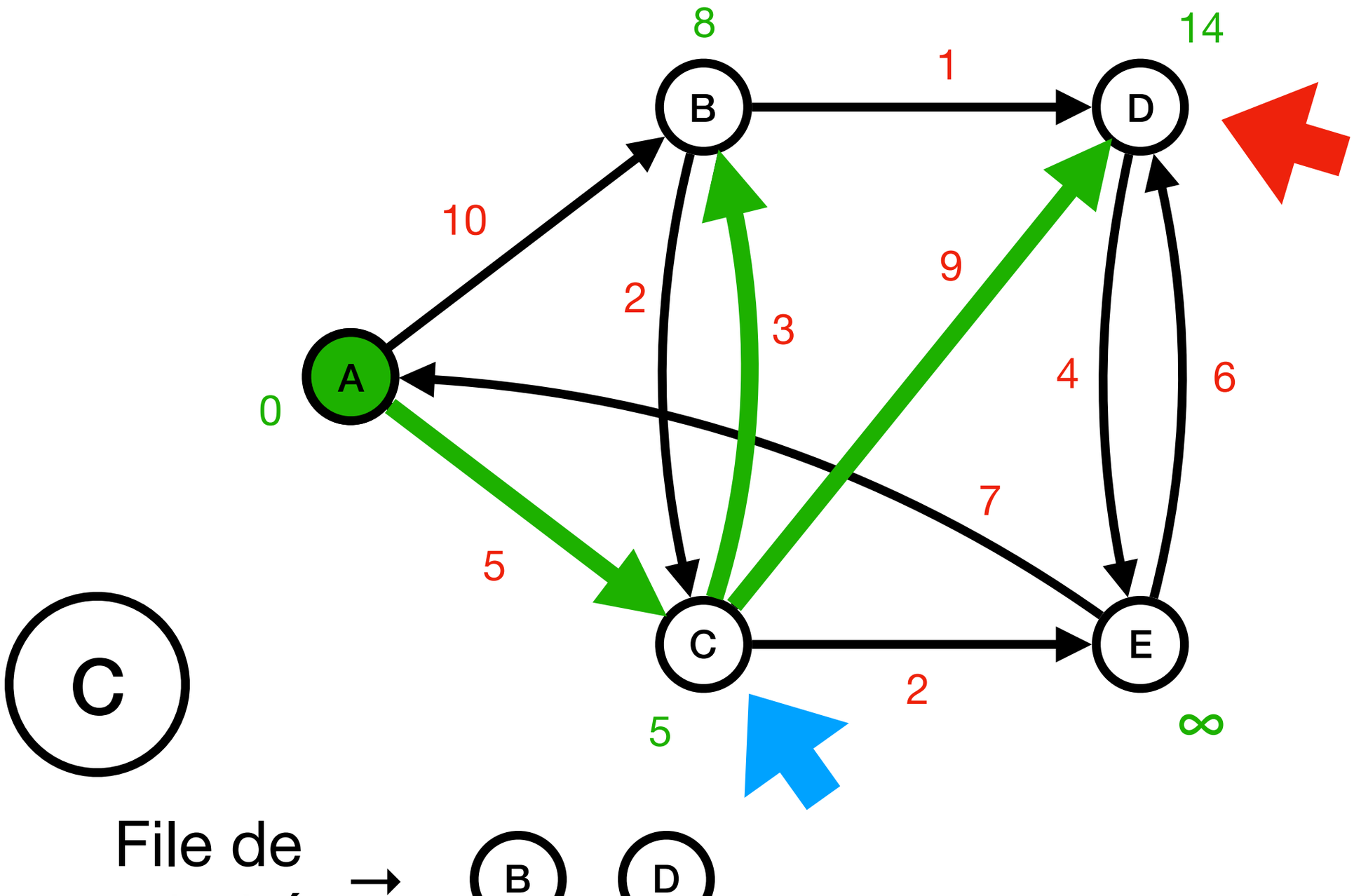


C

File de priorité → B

8

Algorithme de Dijkstra

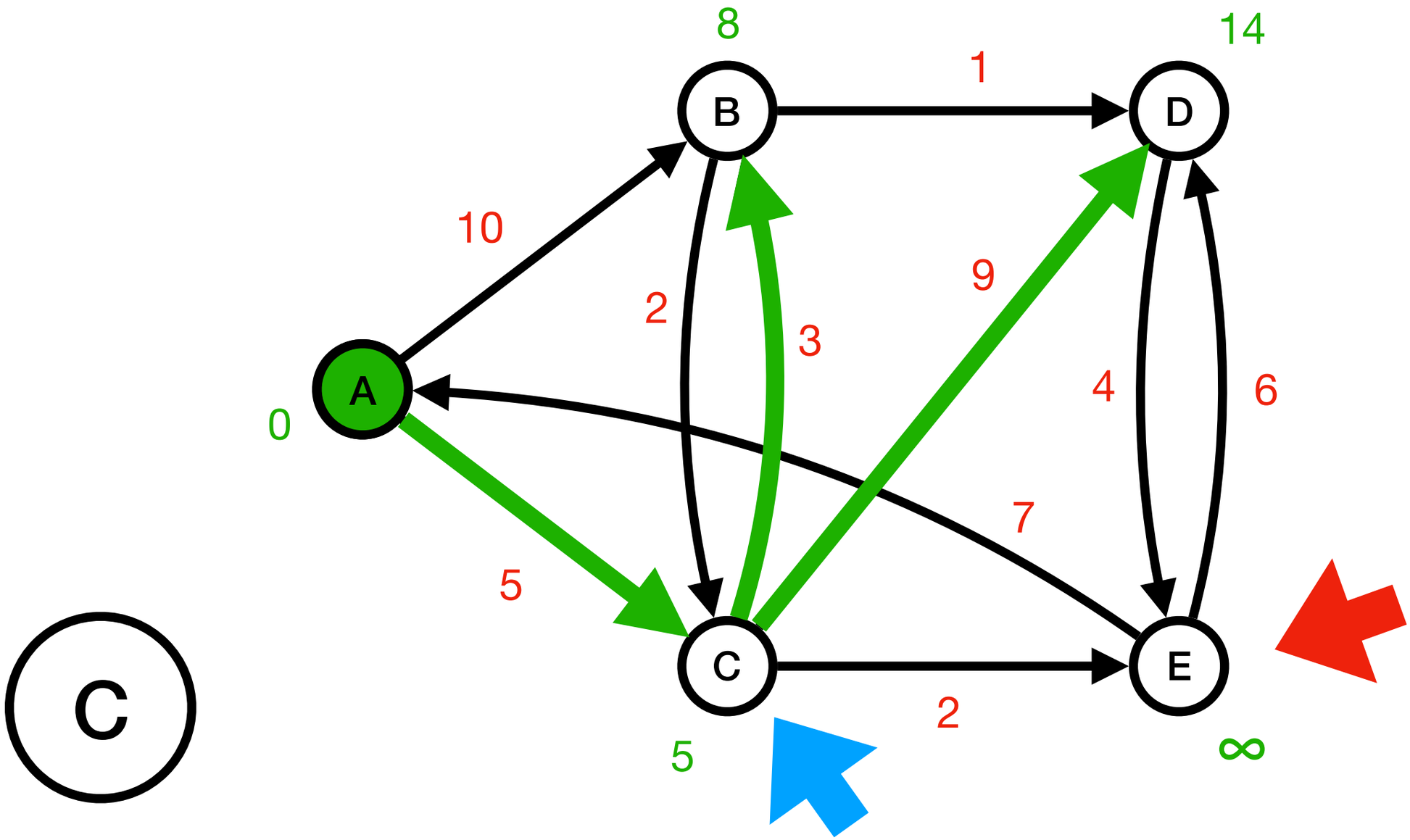


C

File de priorité → B D

8 14

Algorithme de Dijkstra

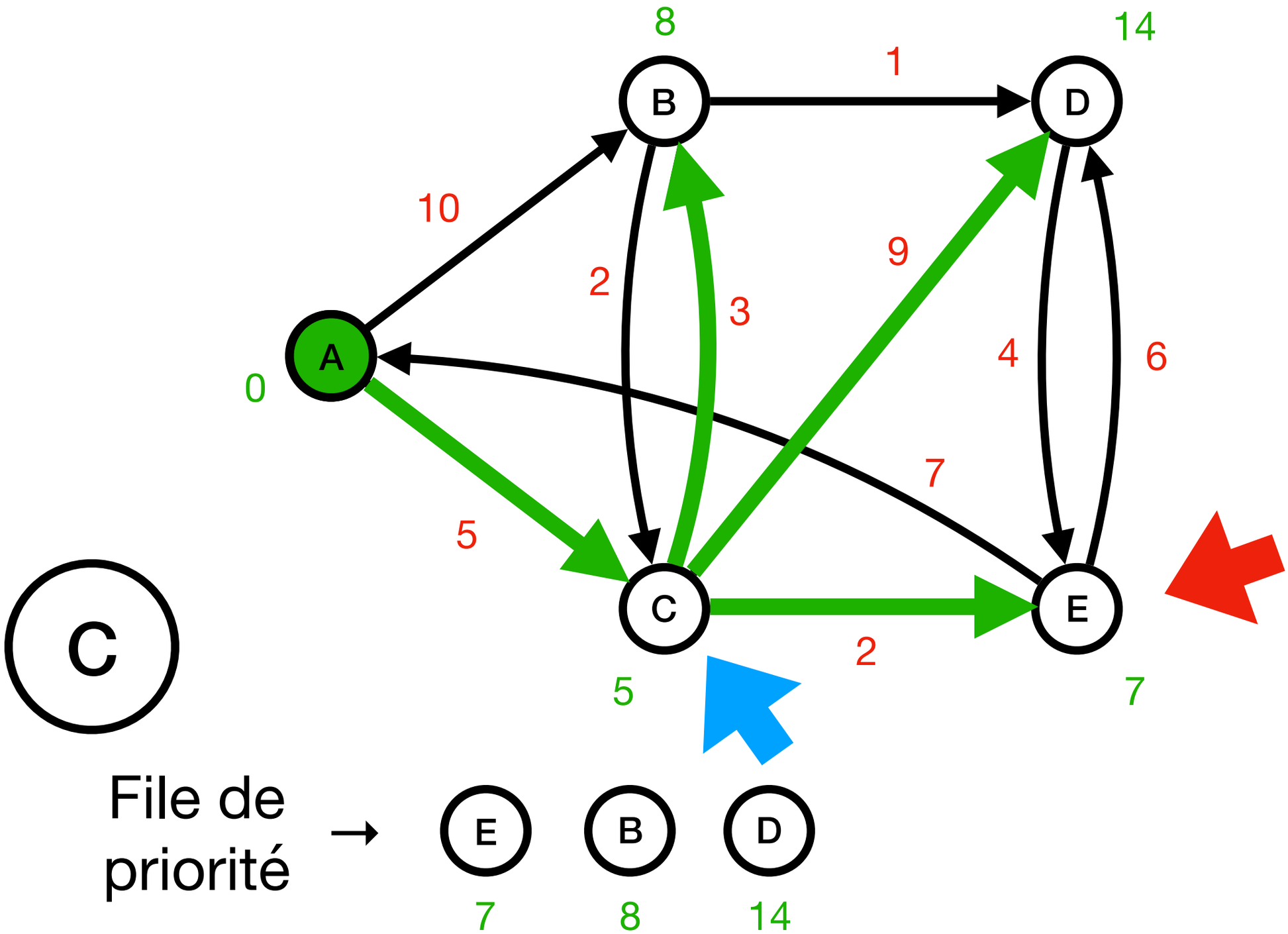


C

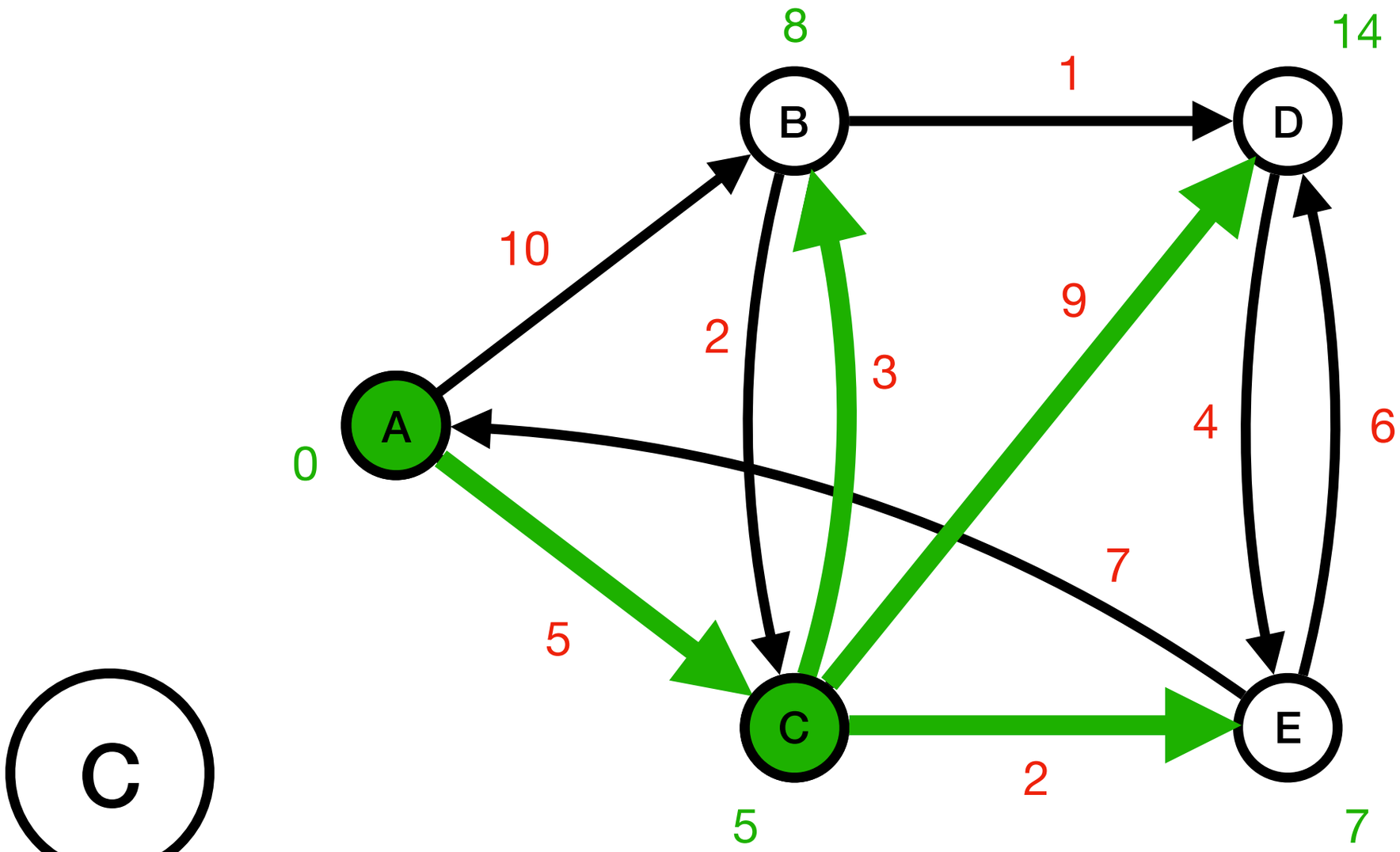
File de priorité → B D

8 14

Algorithme de Dijkstra



Algorithme de Dijkstra

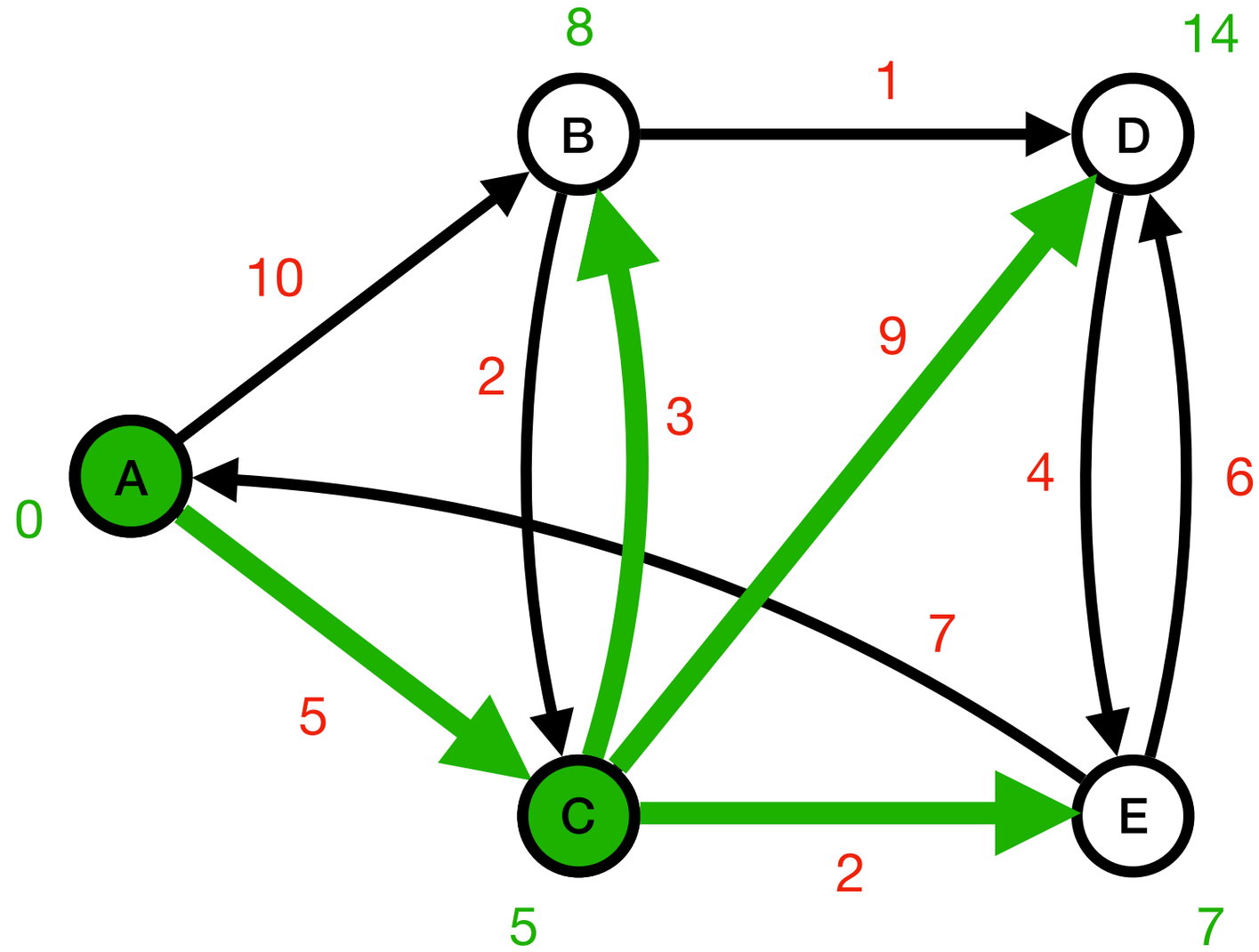


C

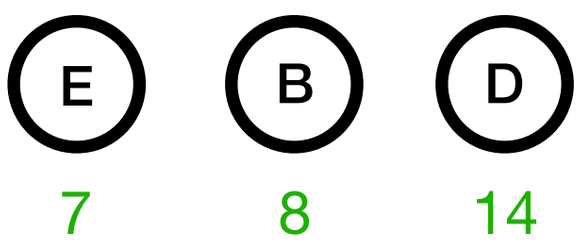
File de priorité → E B D

7 8 14

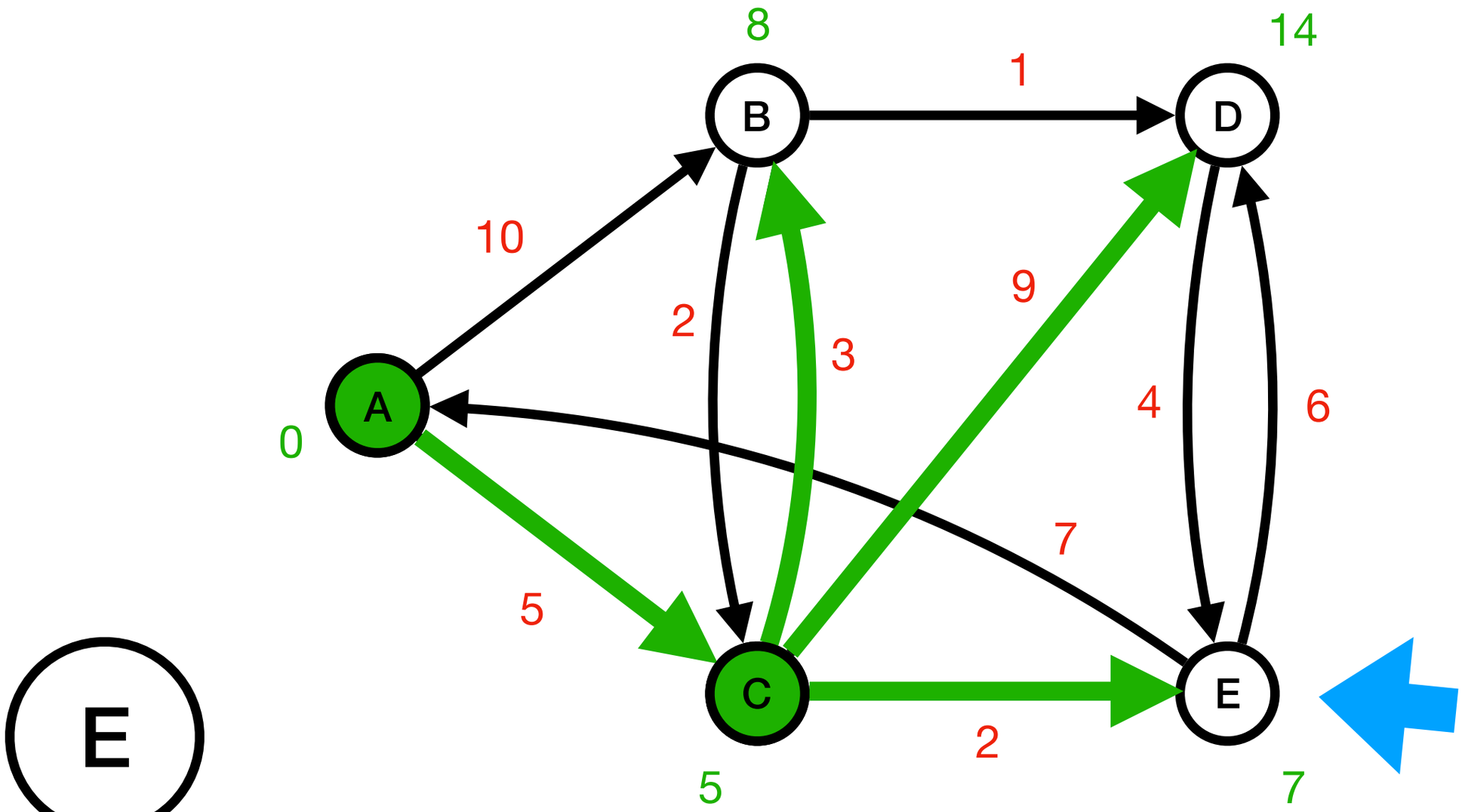
Algorithme de Dijkstra



File de priorité →



Algorithme de Dijkstra

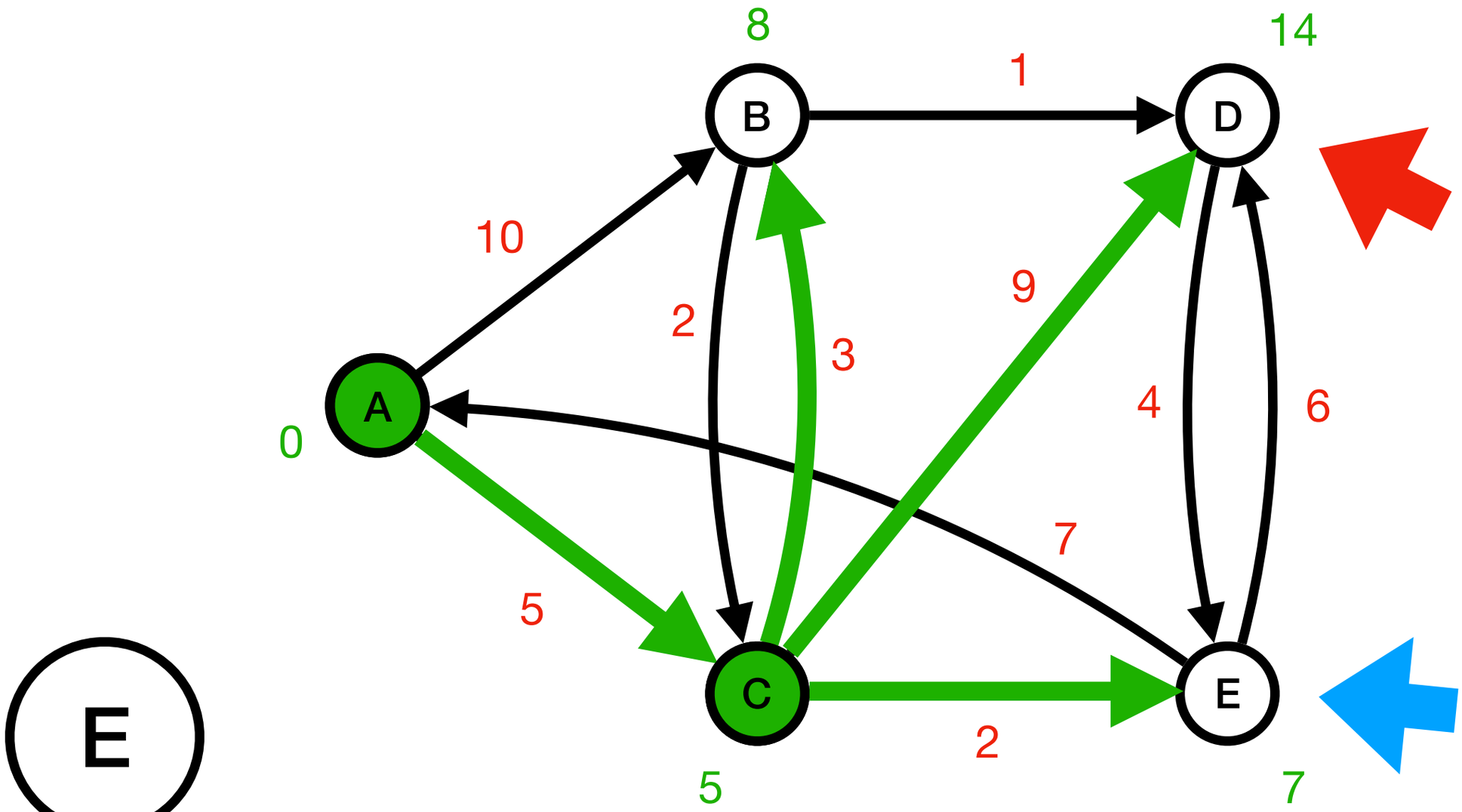


E

File de priorité → B D

8 14

Algorithme de Dijkstra

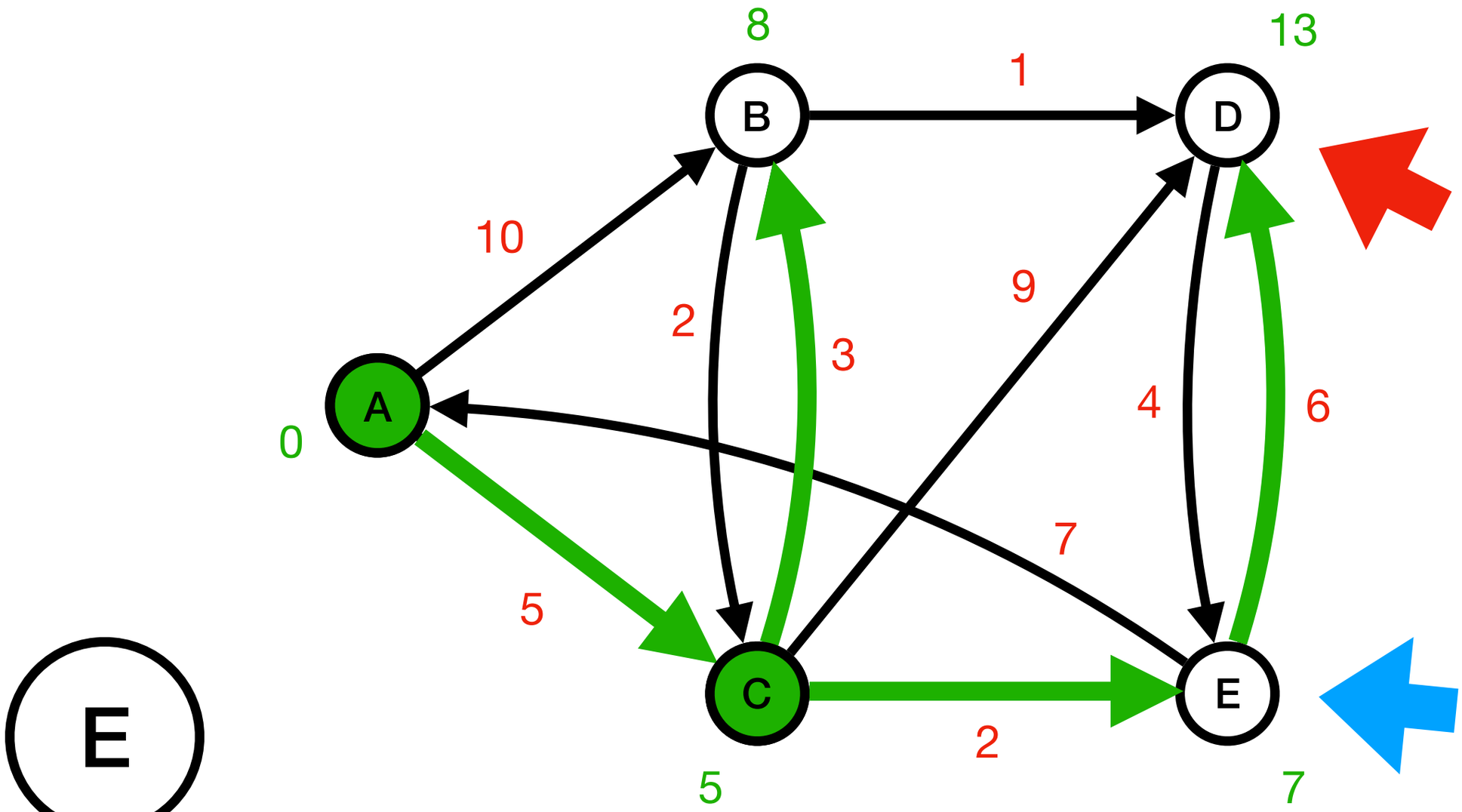


E

File de priorité → B D

8 14

Algorithme de Dijkstra

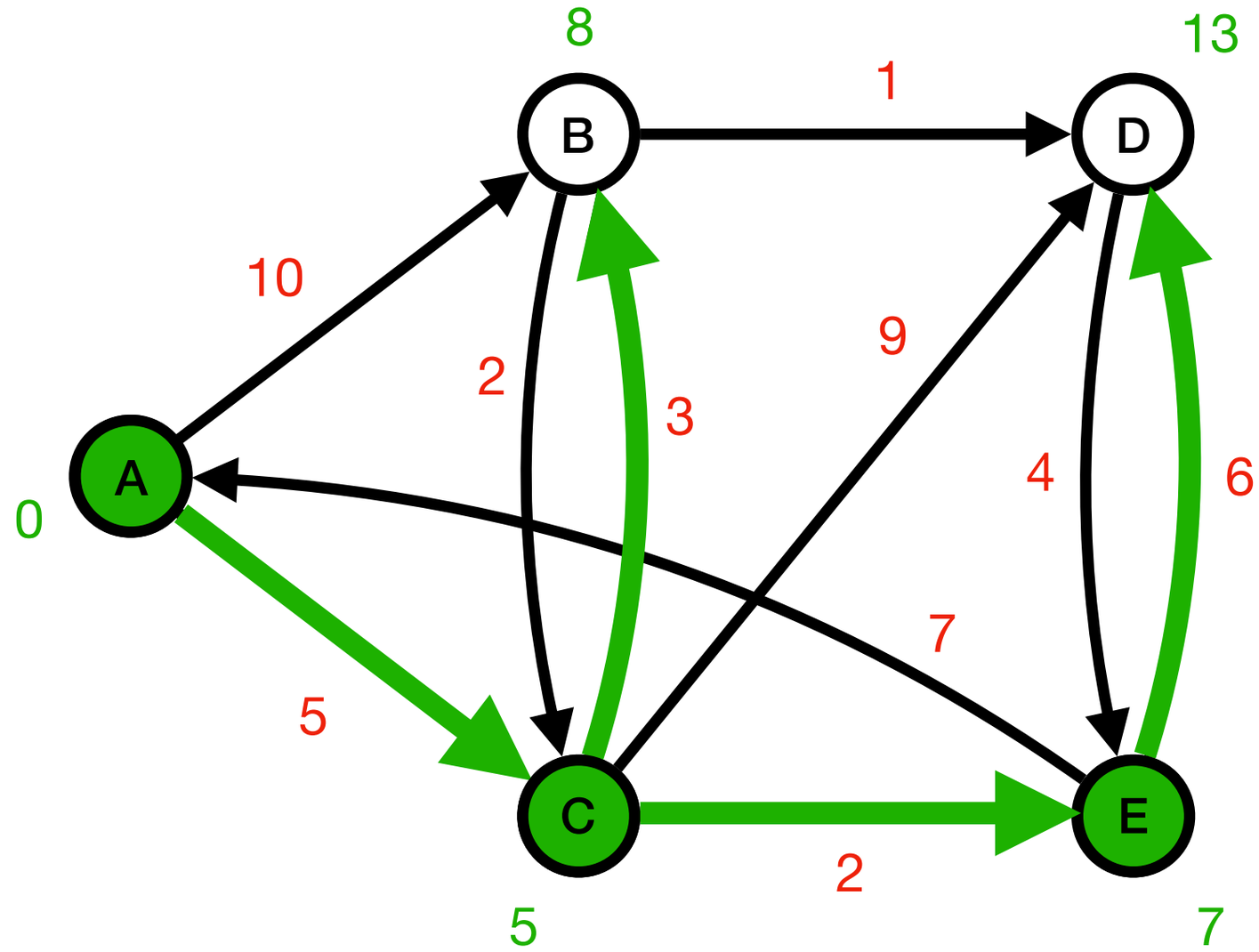


E

File de priorité → B D

8 13

Algorithme de Dijkstra

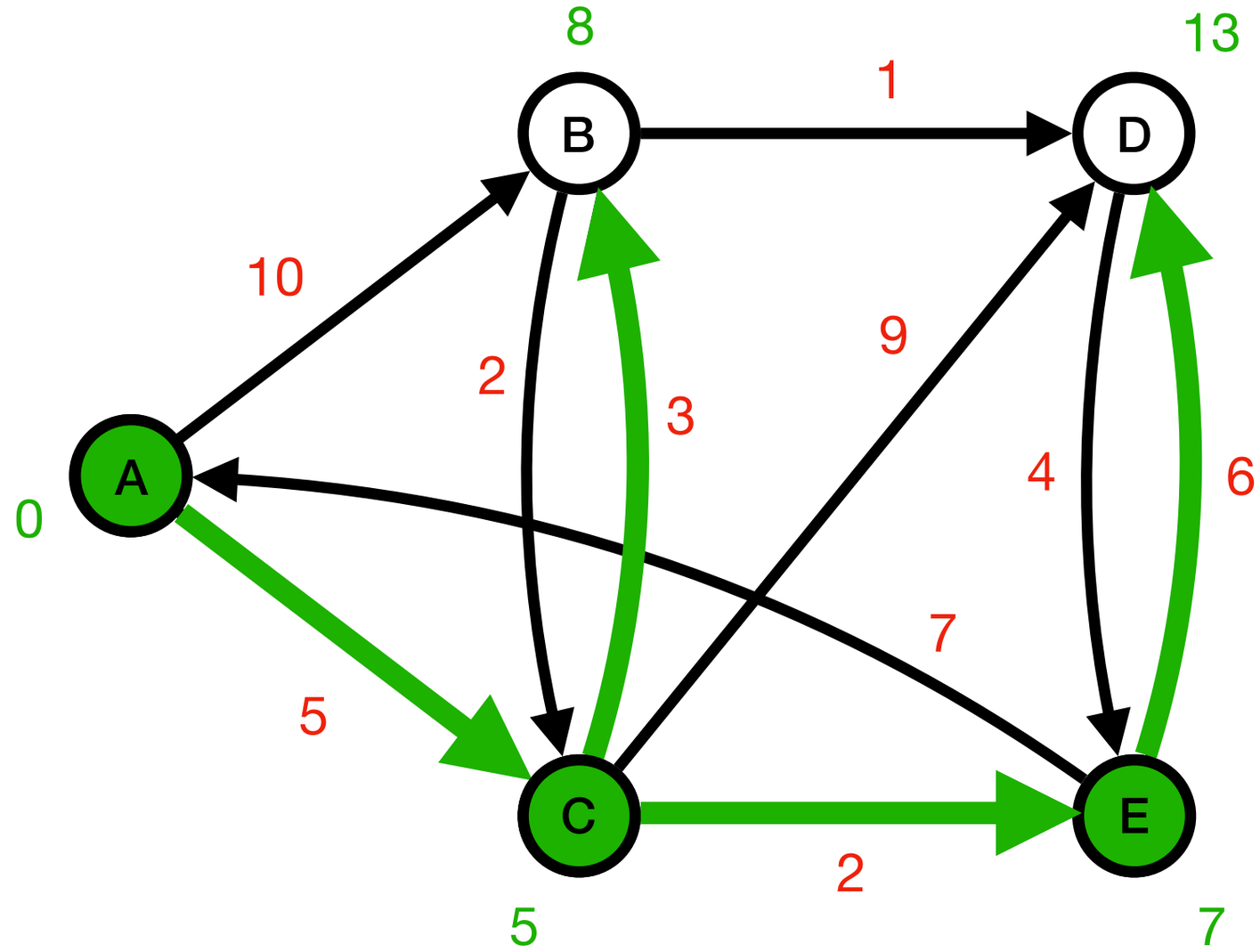


E

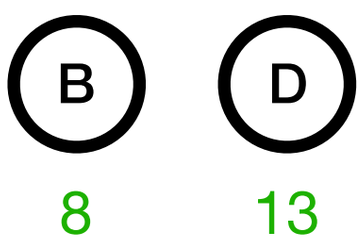
File de priorité → B D

8 13

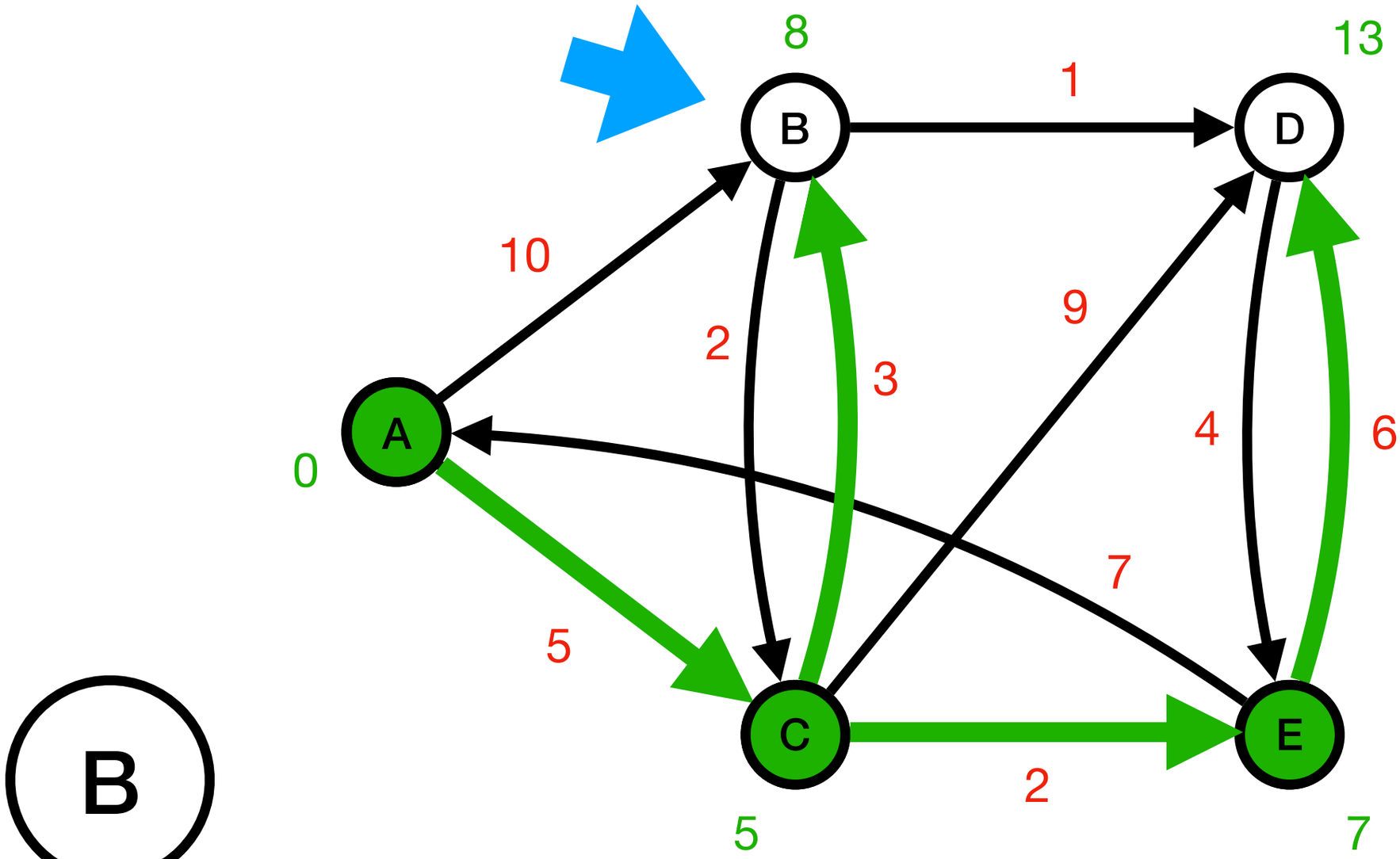
Algorithme de Dijkstra



File de priorité →



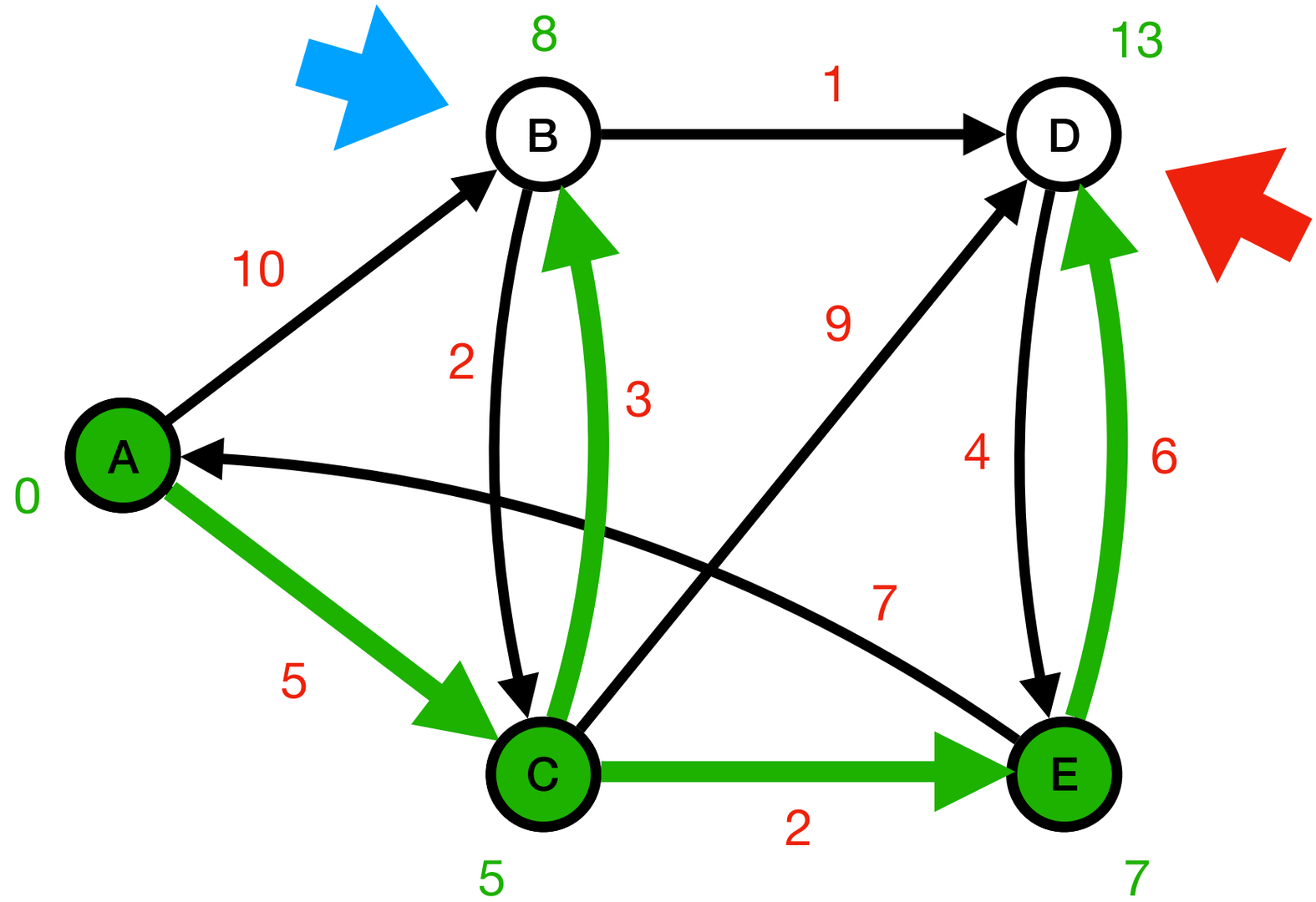
Algorithme de Dijkstra

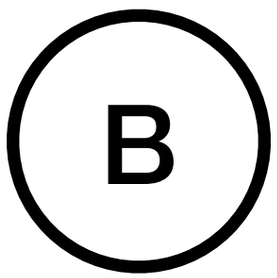


File de priorité → D

13

Algorithme de Dijkstra

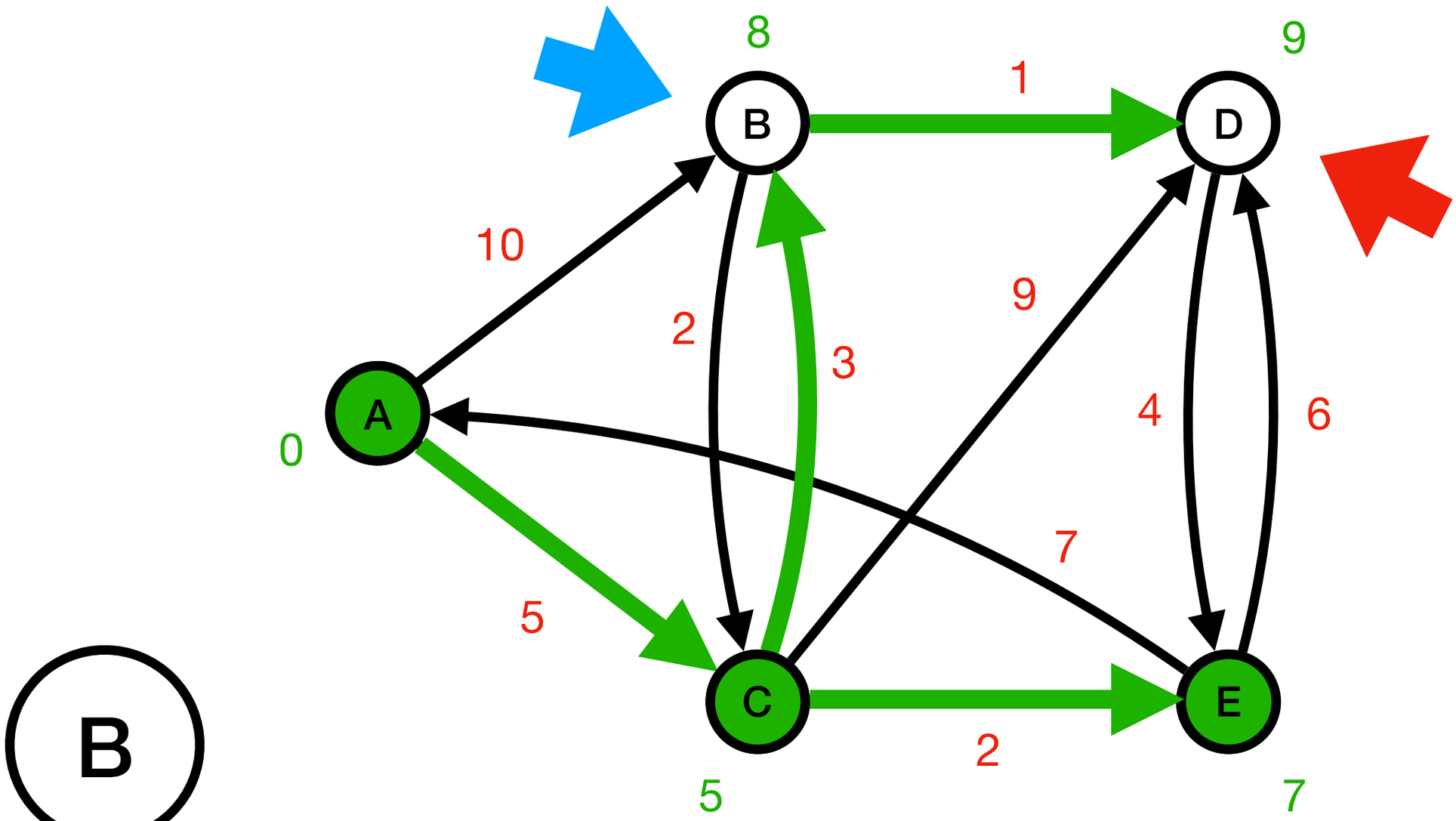


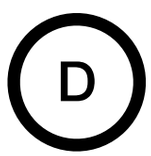


File de priorité → D

13

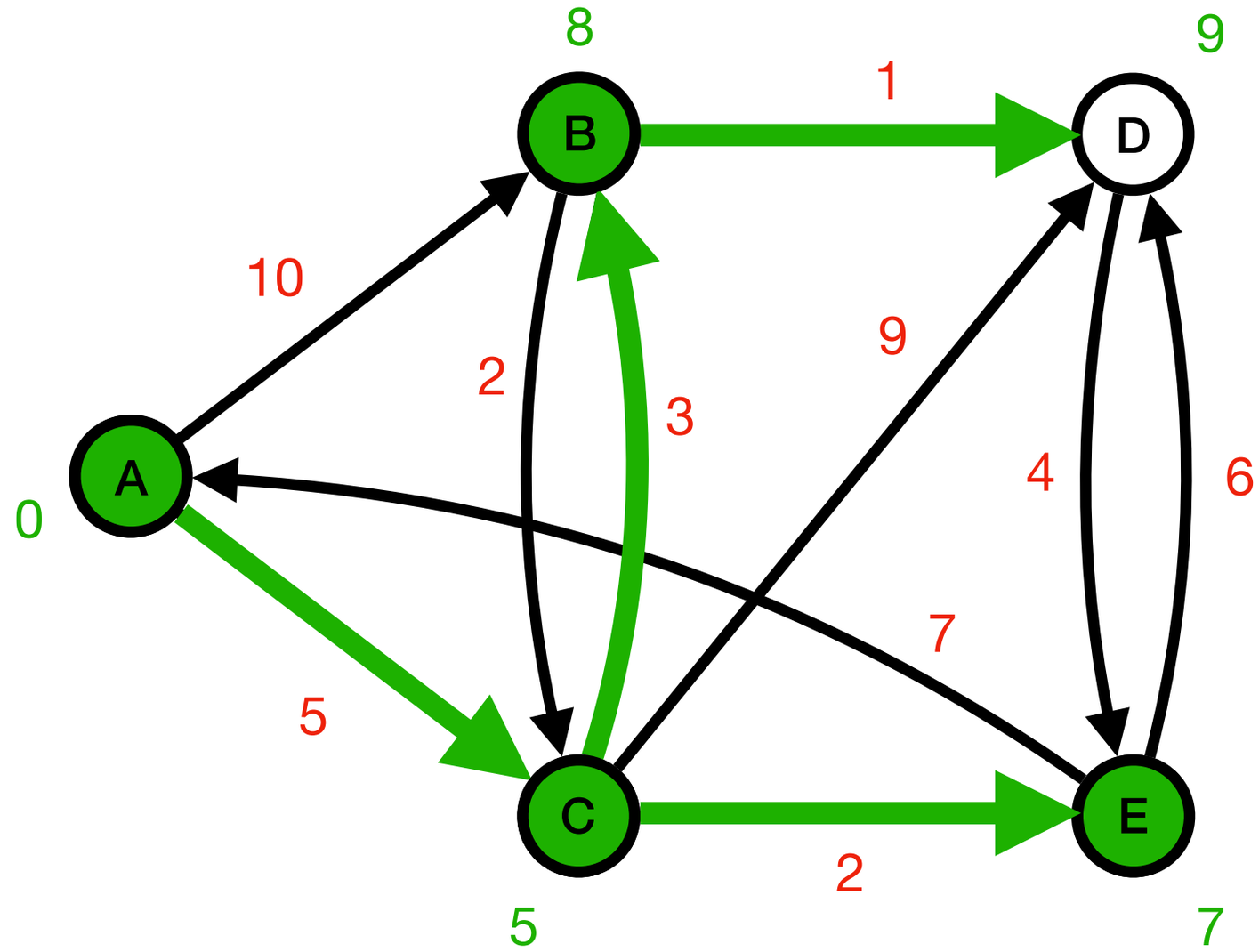
Algorithme de Dijkstra



File de priorité → 

9

Algorithme de Dijkstra

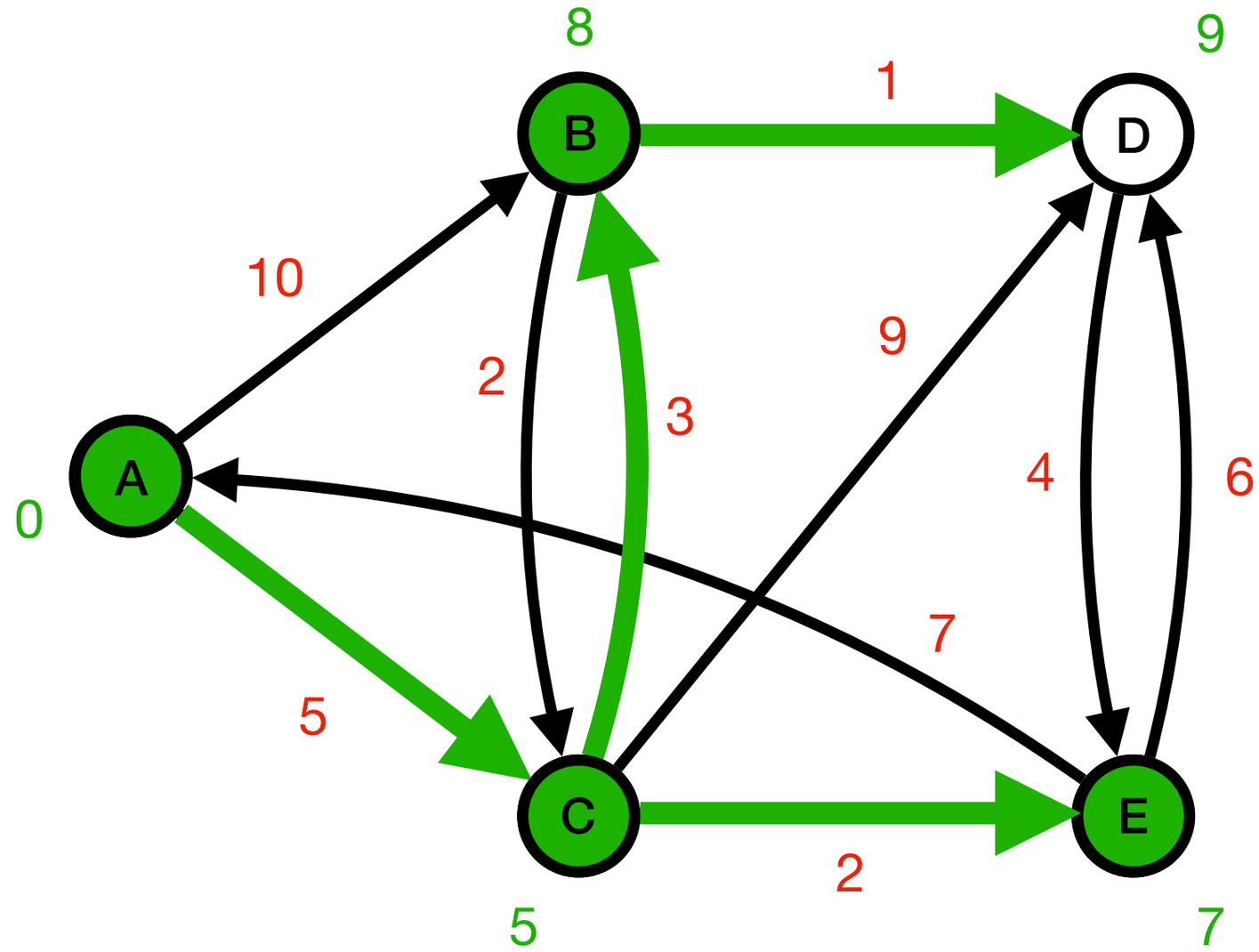


B

File de priorité → D

9

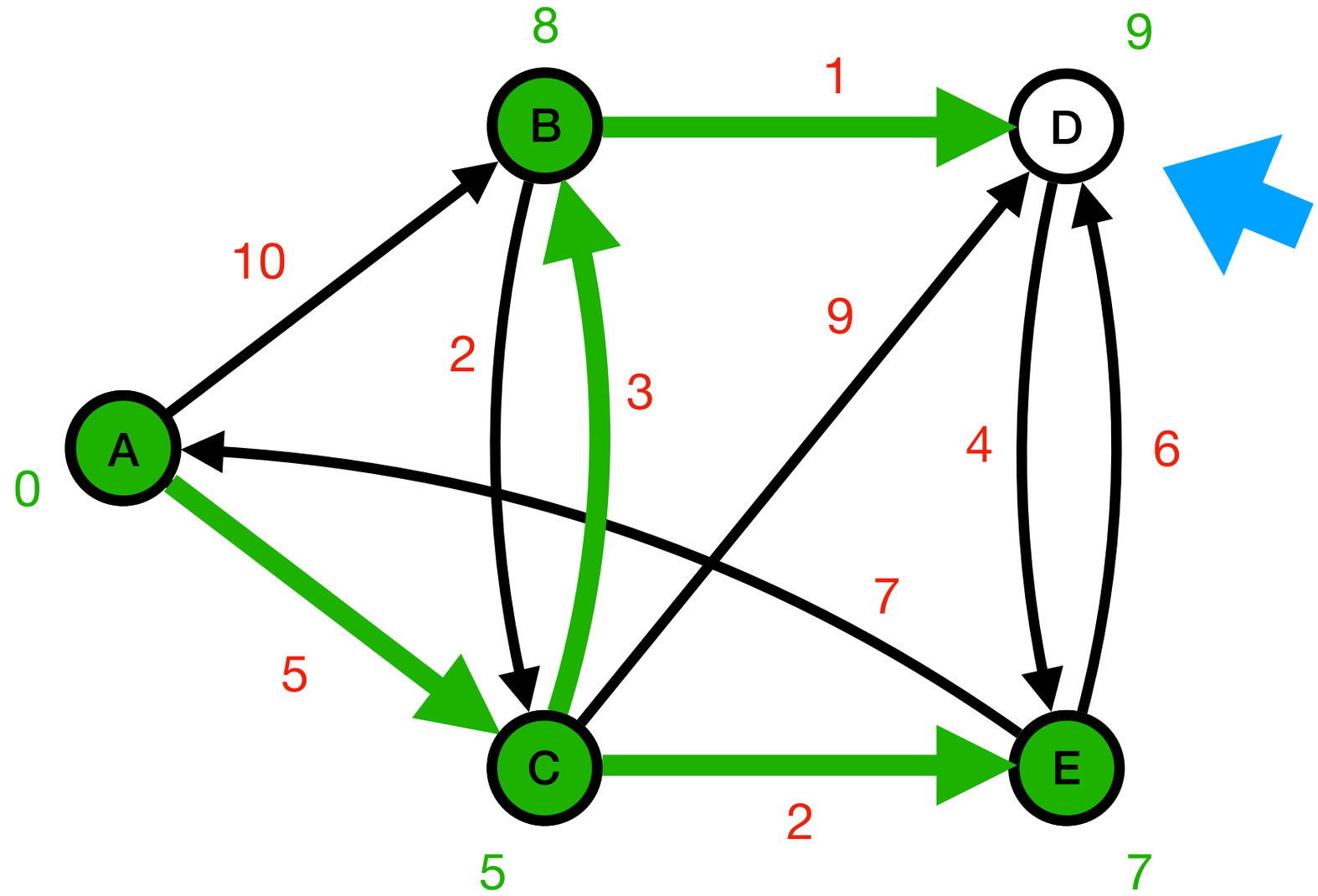
Algorithme de Dijkstra



File de priorité → D

9

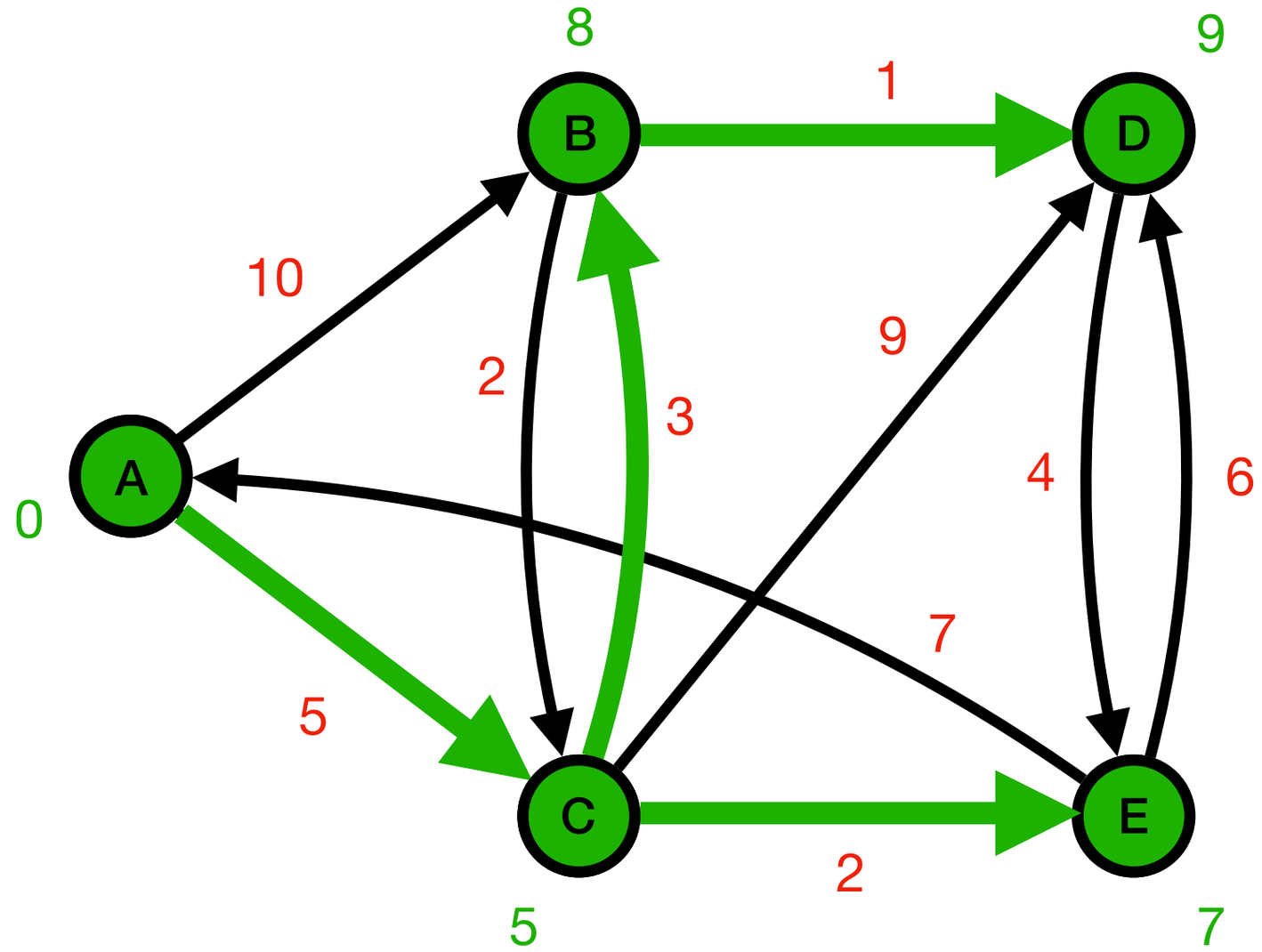
Algorithme de Dijkstra



D

File de priorité →

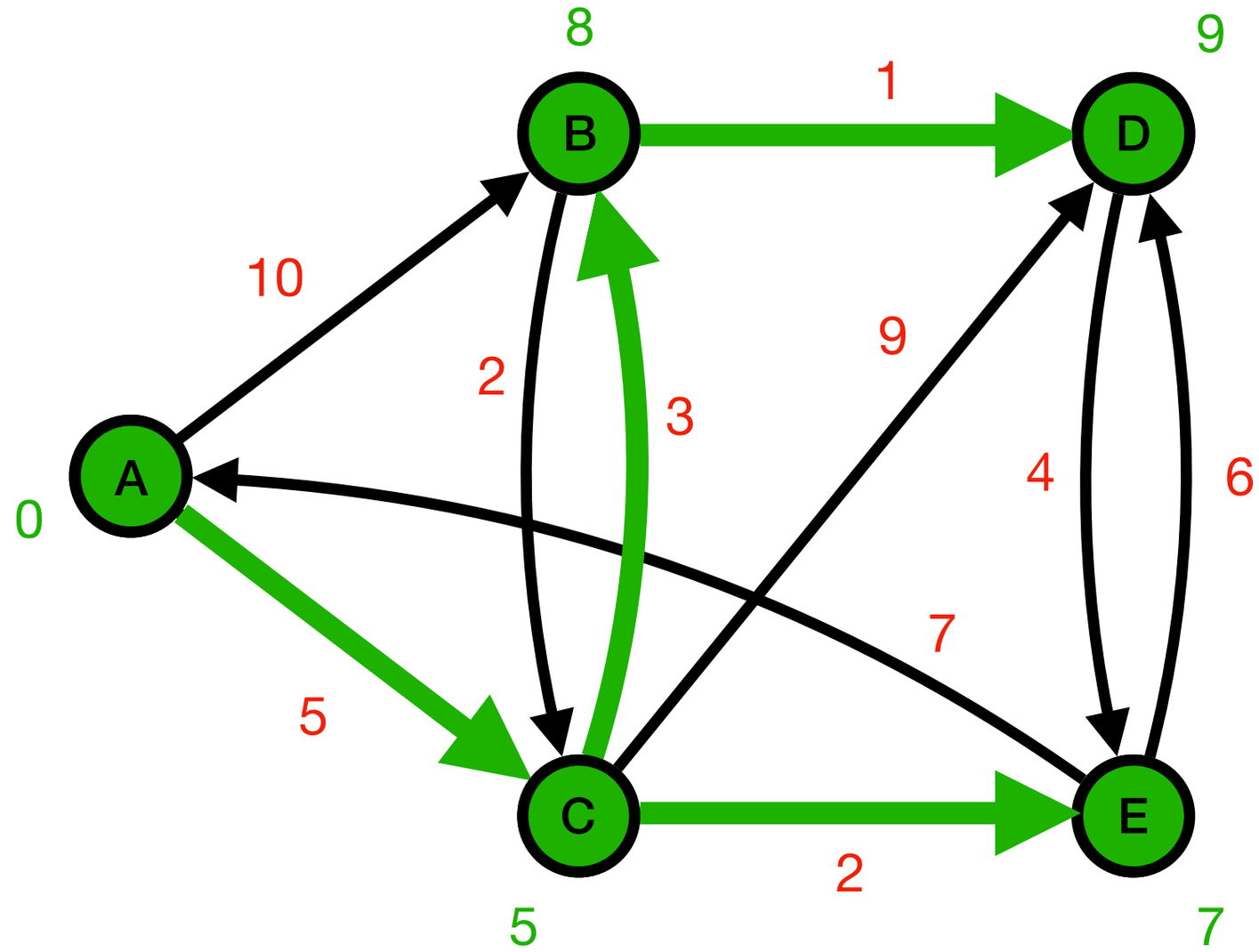
Algorithme de Dijkstra



D

File de priorité →

Algorithme de Dijkstra



File de priorité →

Algorithme de Dijkstra