

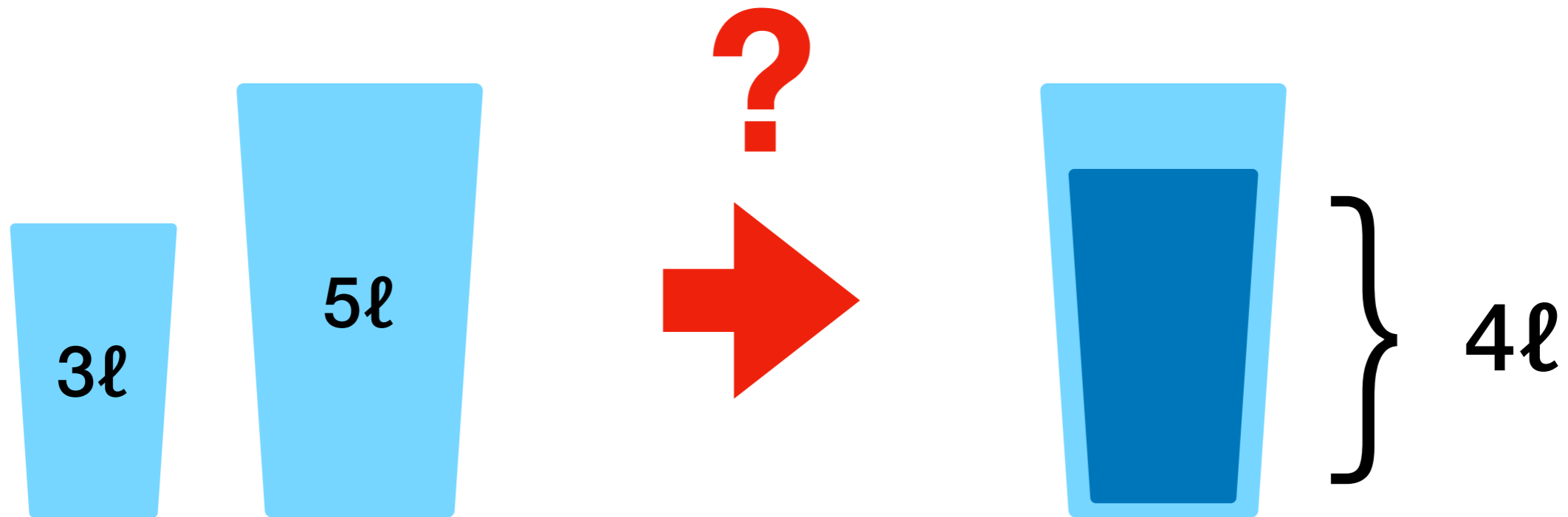
Science informatique

CM4

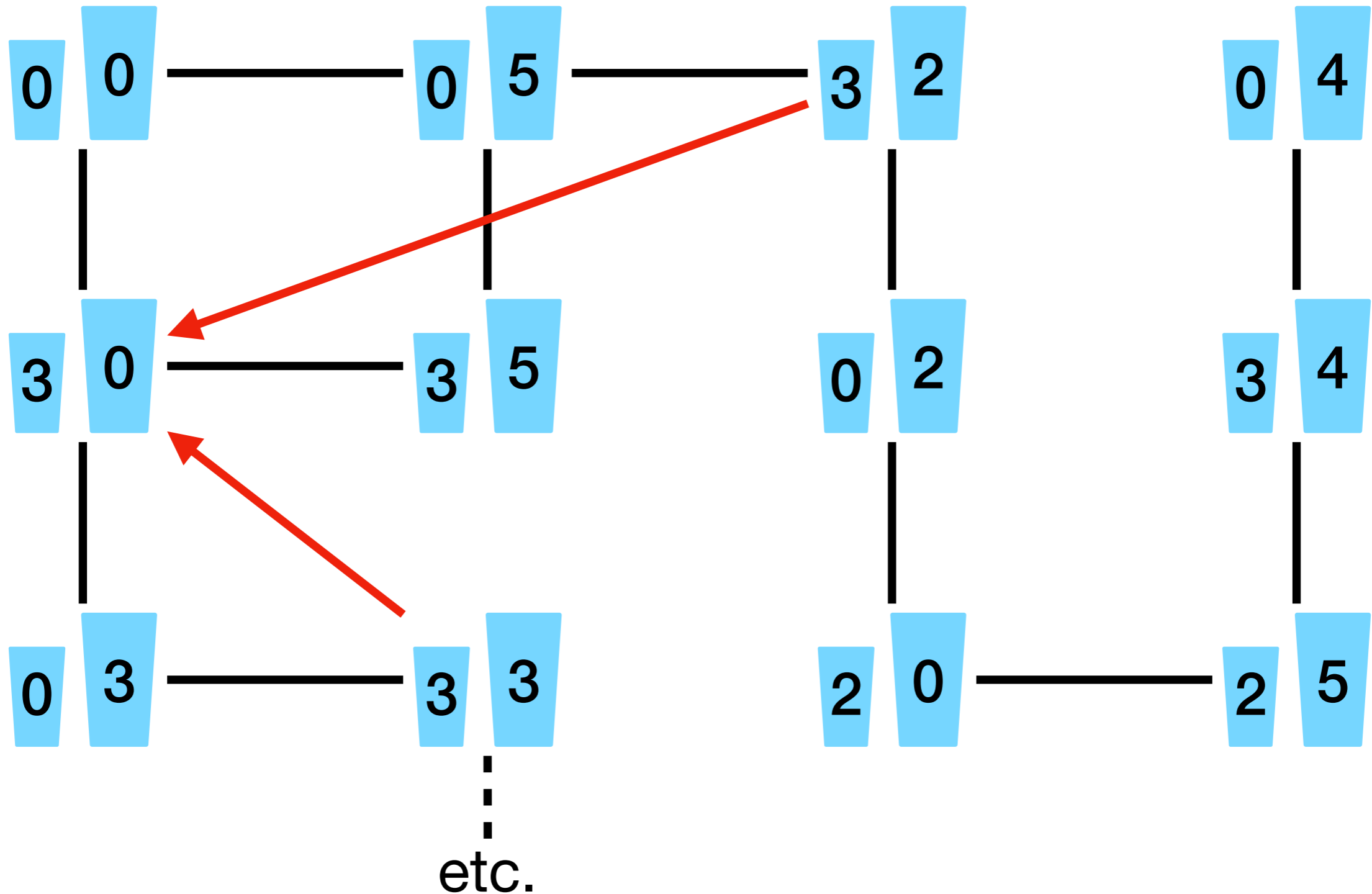
Antonio E. Porreca

aeporreca.org/scienceinfo

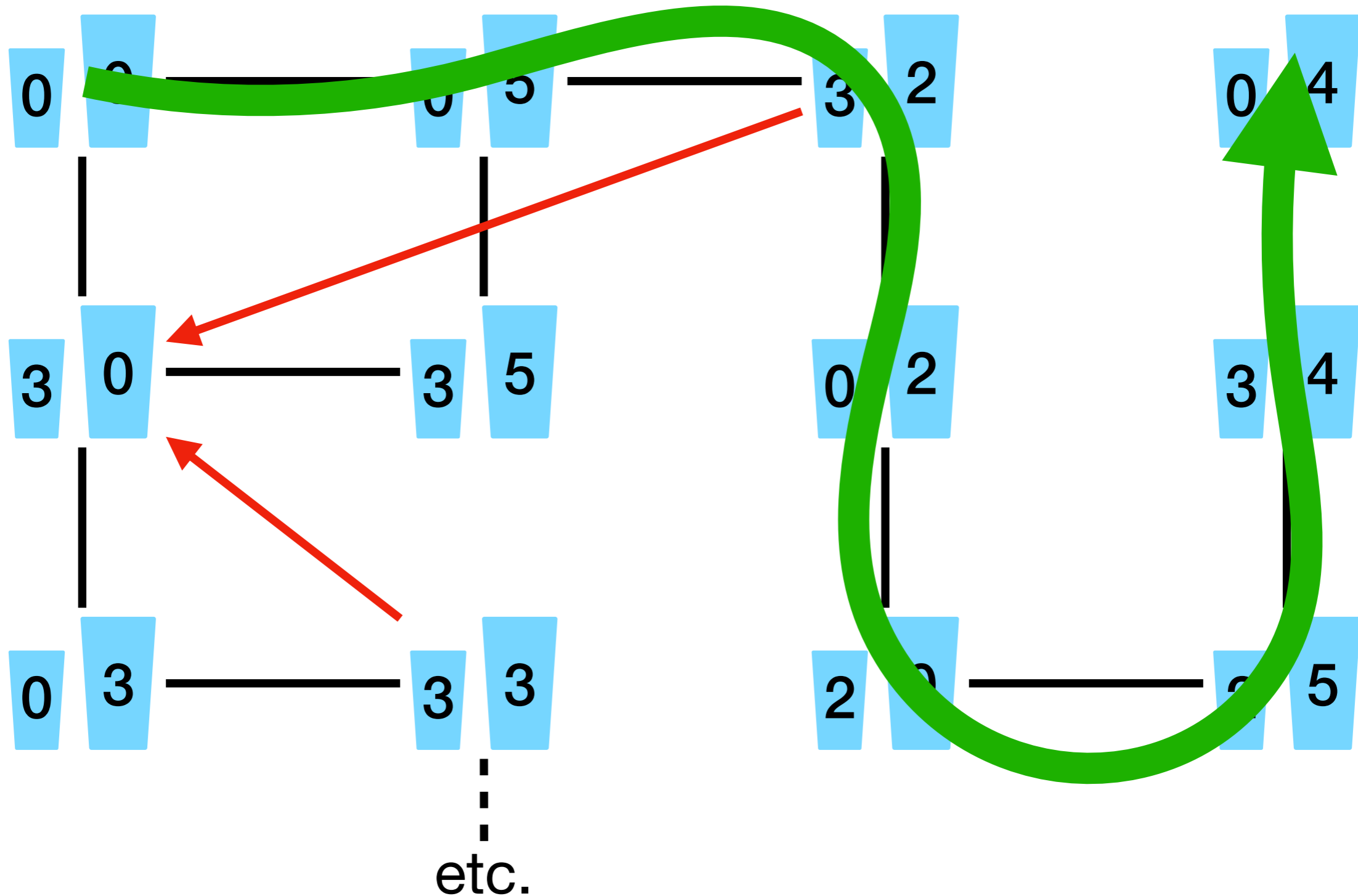
Enigme des récipients



Graphe des configurations



Graphe des configurations



Parcours de graphes

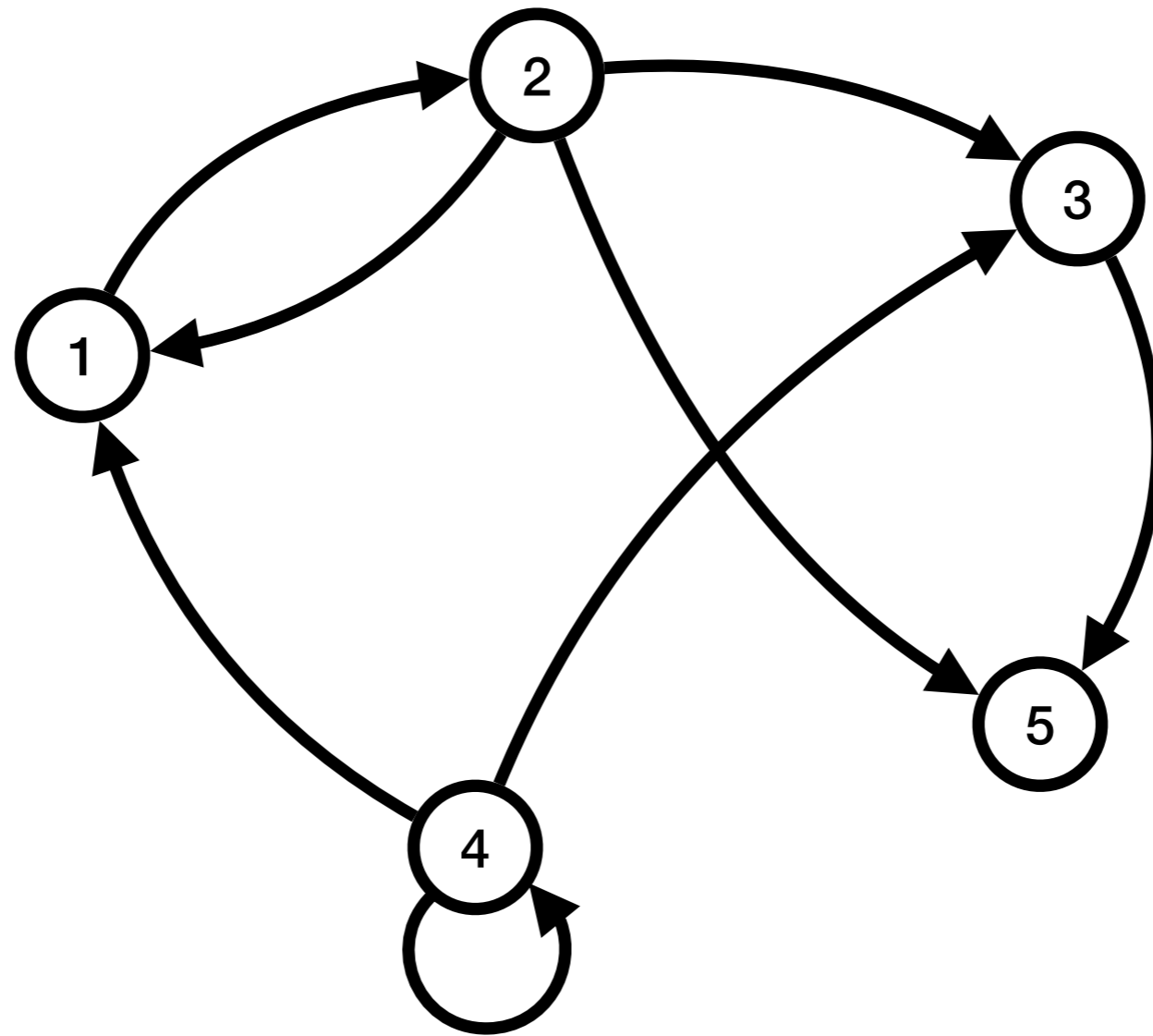
Parcours en largeur

```
def parcours_en_largeur(G, s):
    n = len(G)
    H = nouveau_graphe(n)           # graphe vide
    F = nouvelle_file()           # file vide
    couleur = ['blanc'] * n       # n cases blanches
    couleur[s] = 'rouge'
    enfiler(F, s)
    while not est_vide(F):
        u = défiler(F)
        for v in range(n):
            if G[u][v] == 1 and couleur[v] == 'blanc':
                couleur[v] = 'rouge'
                enfiler(F, v)
                H[u][v] = 1
        couleur[u] = 'vert'
    return H
```

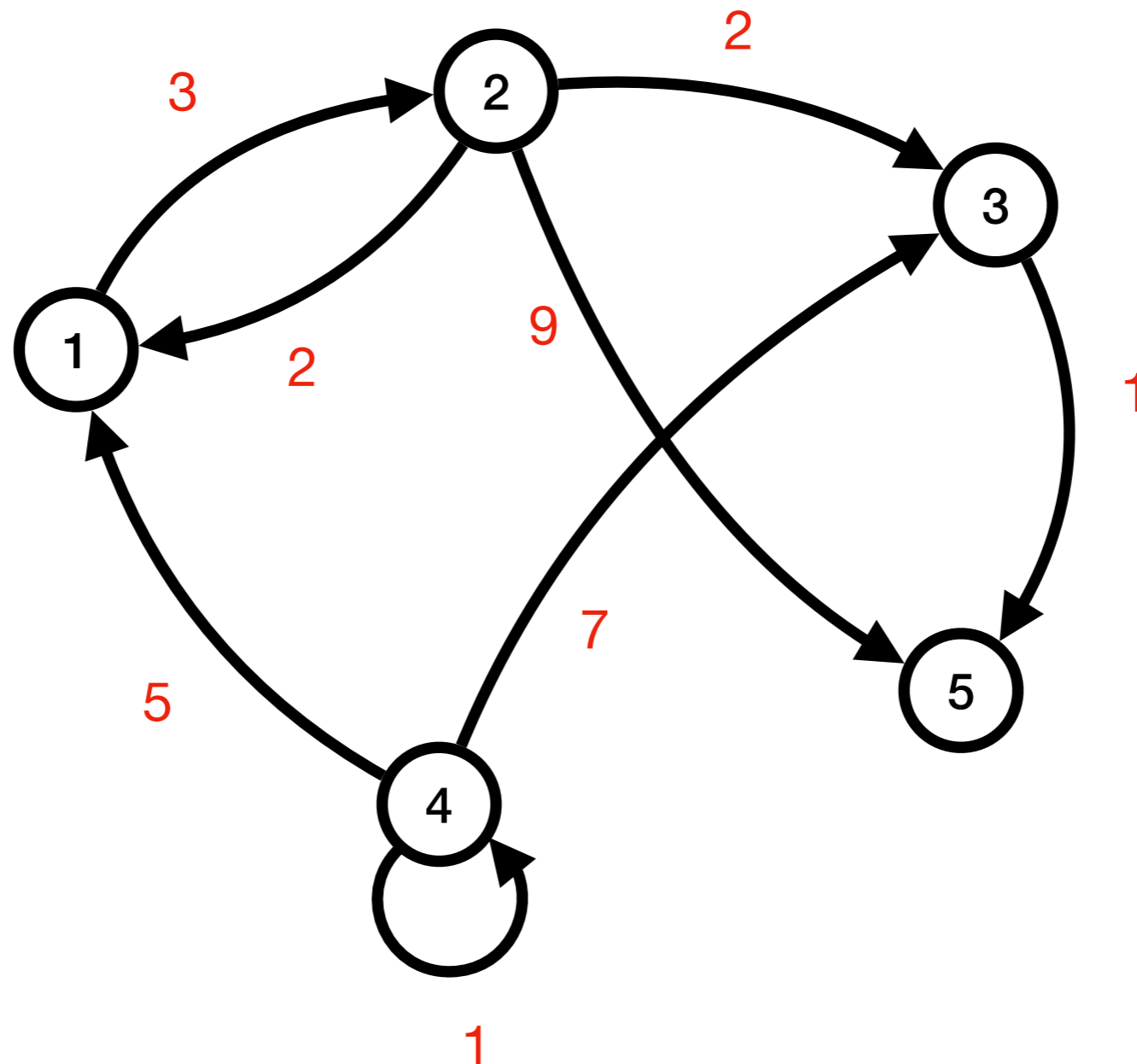
Exercice 4 du TD3

Le plus court chemin

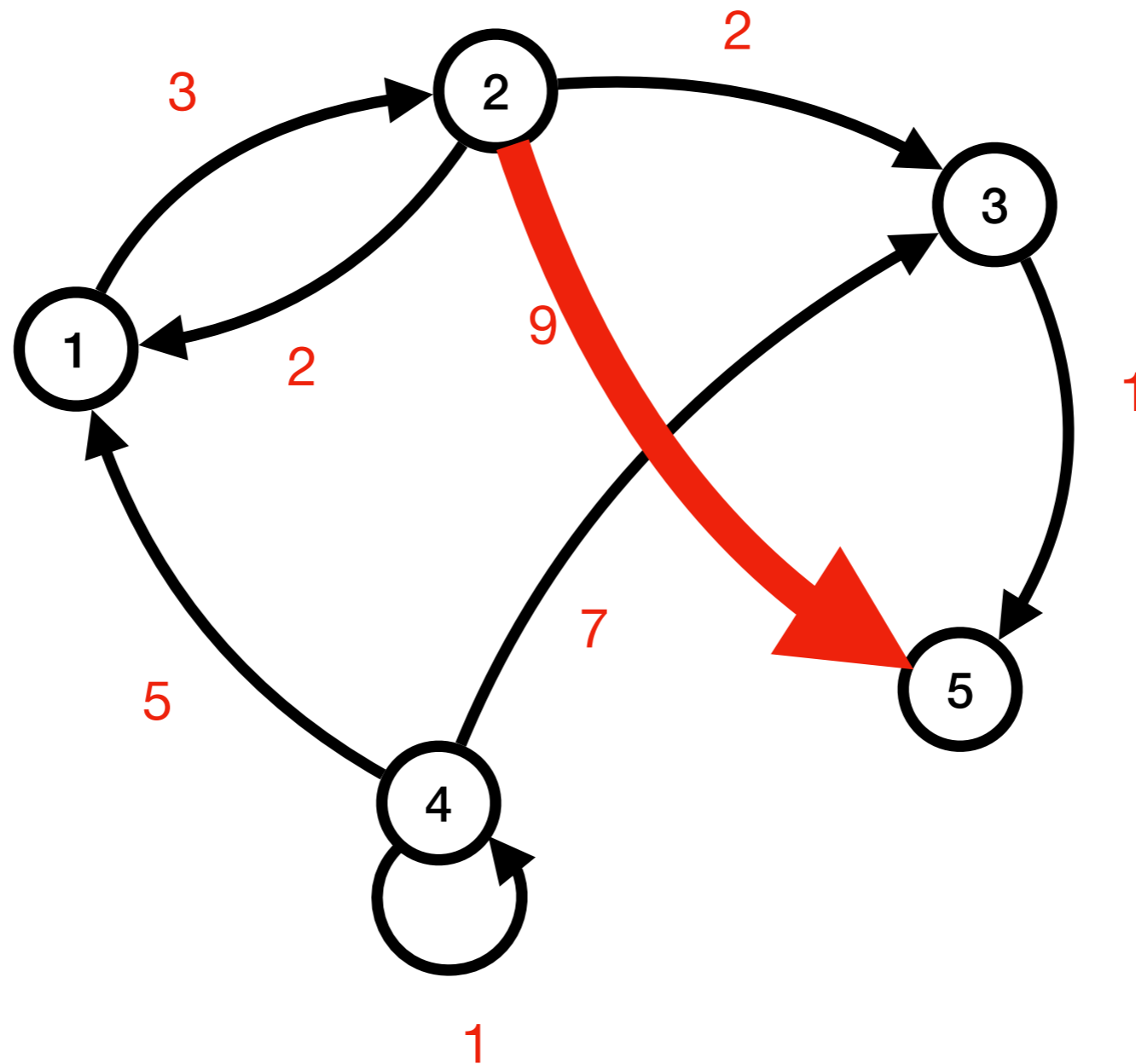
Graphes pondérés



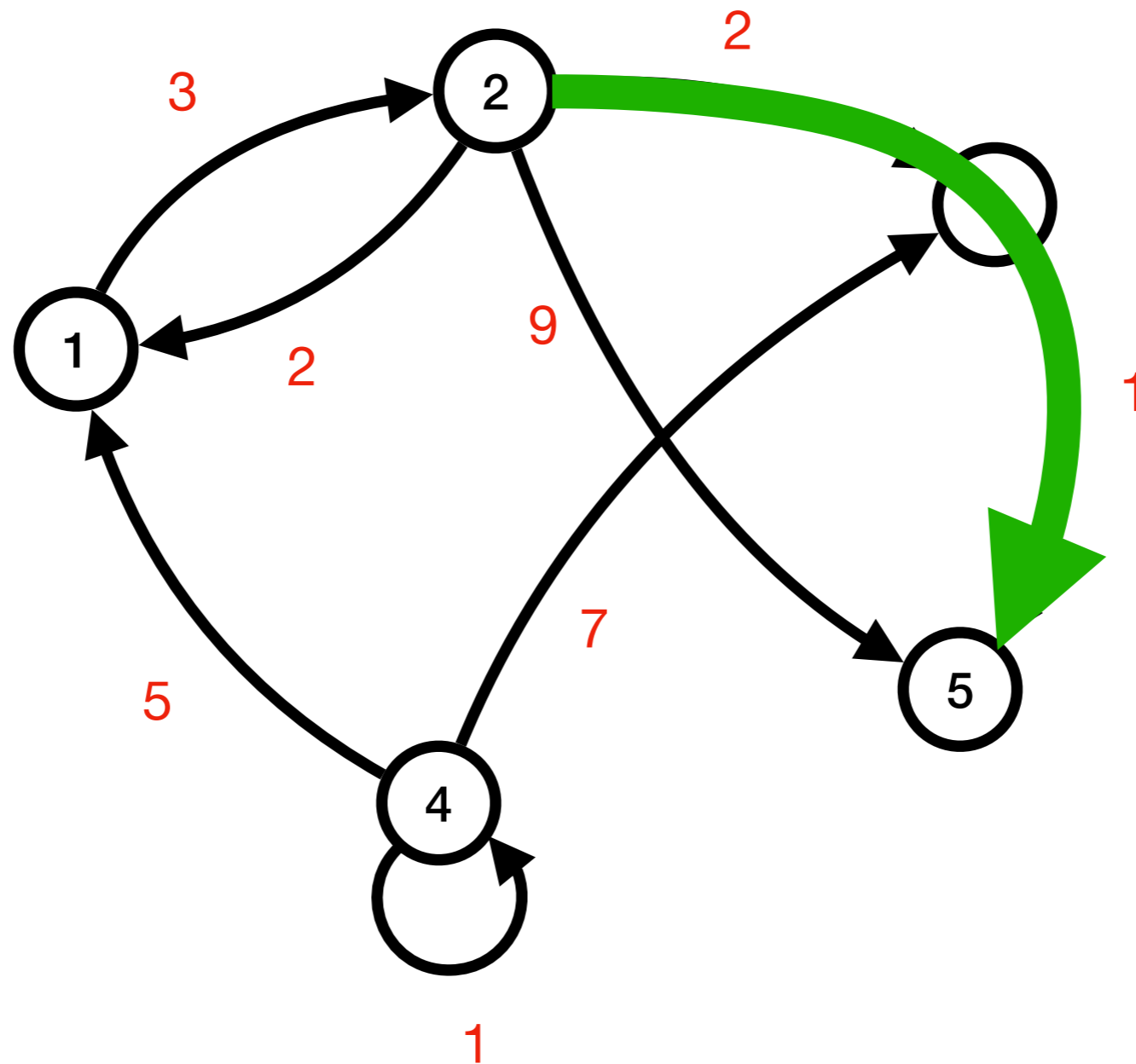
Graphes pondérés



Graphes pondérés

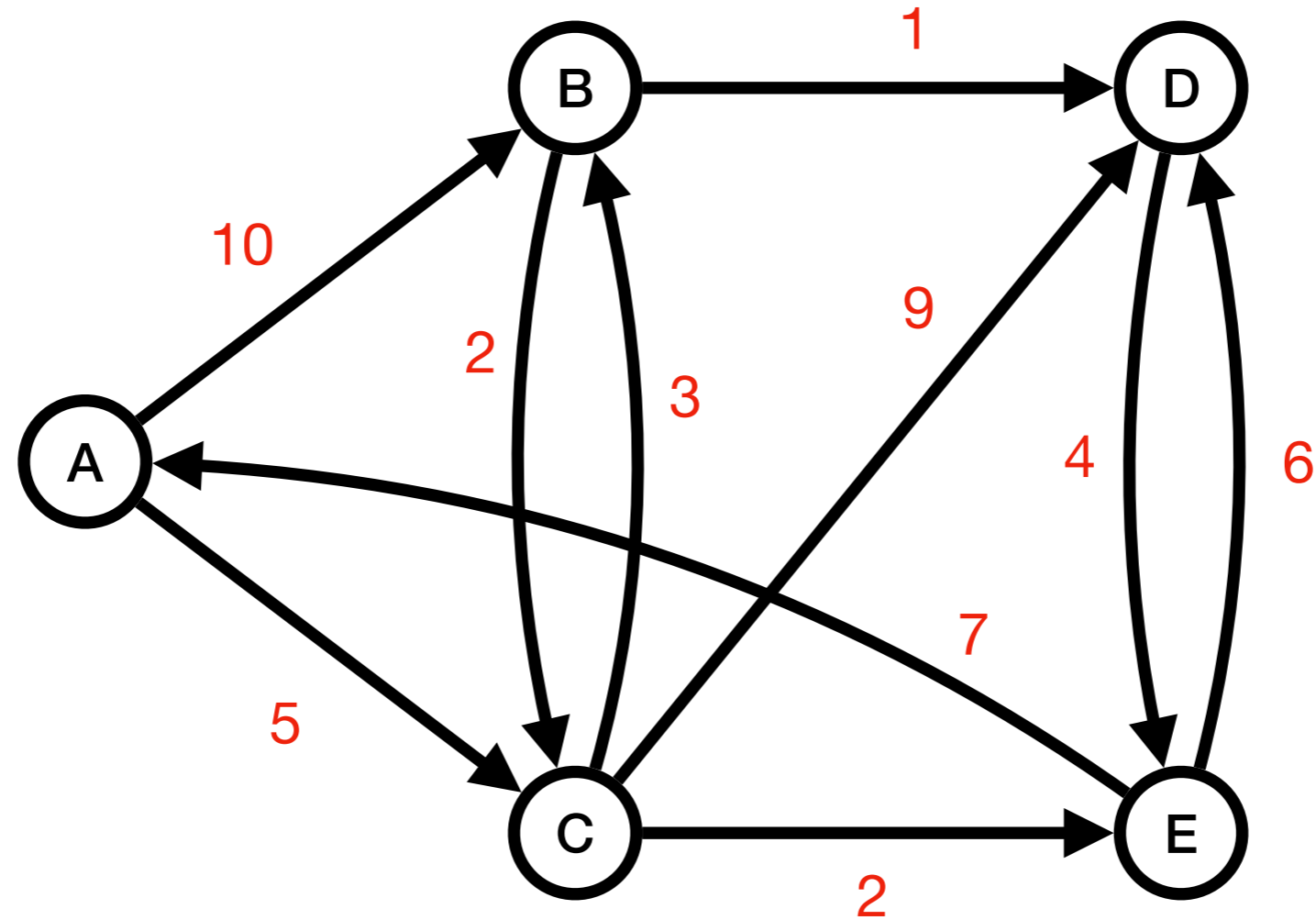


Graphes pondérés

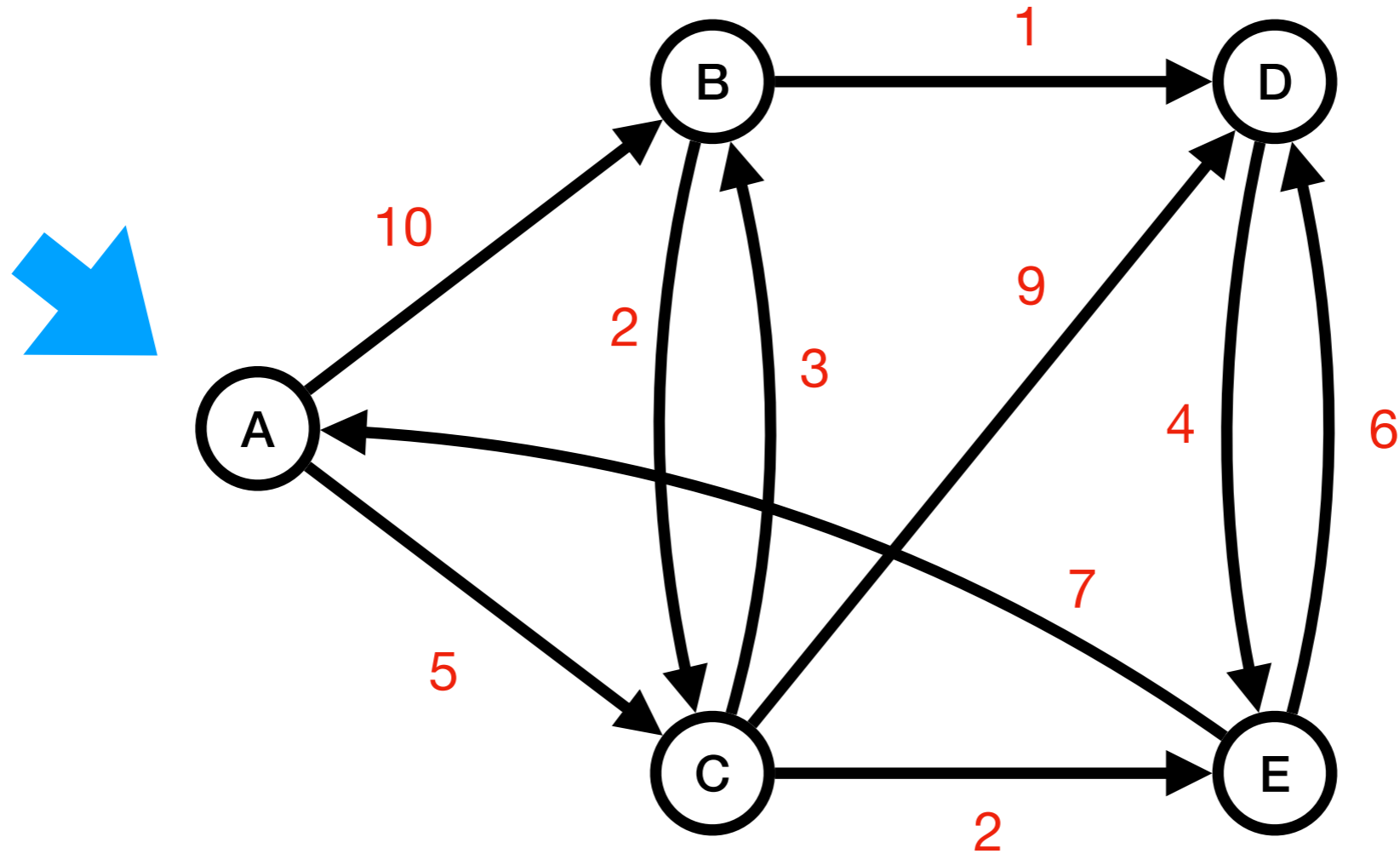


**L'algorithme de parcours
en largeur ne garantit pas
d'obtenir un chemin minimal
sur un graphe pondéré**

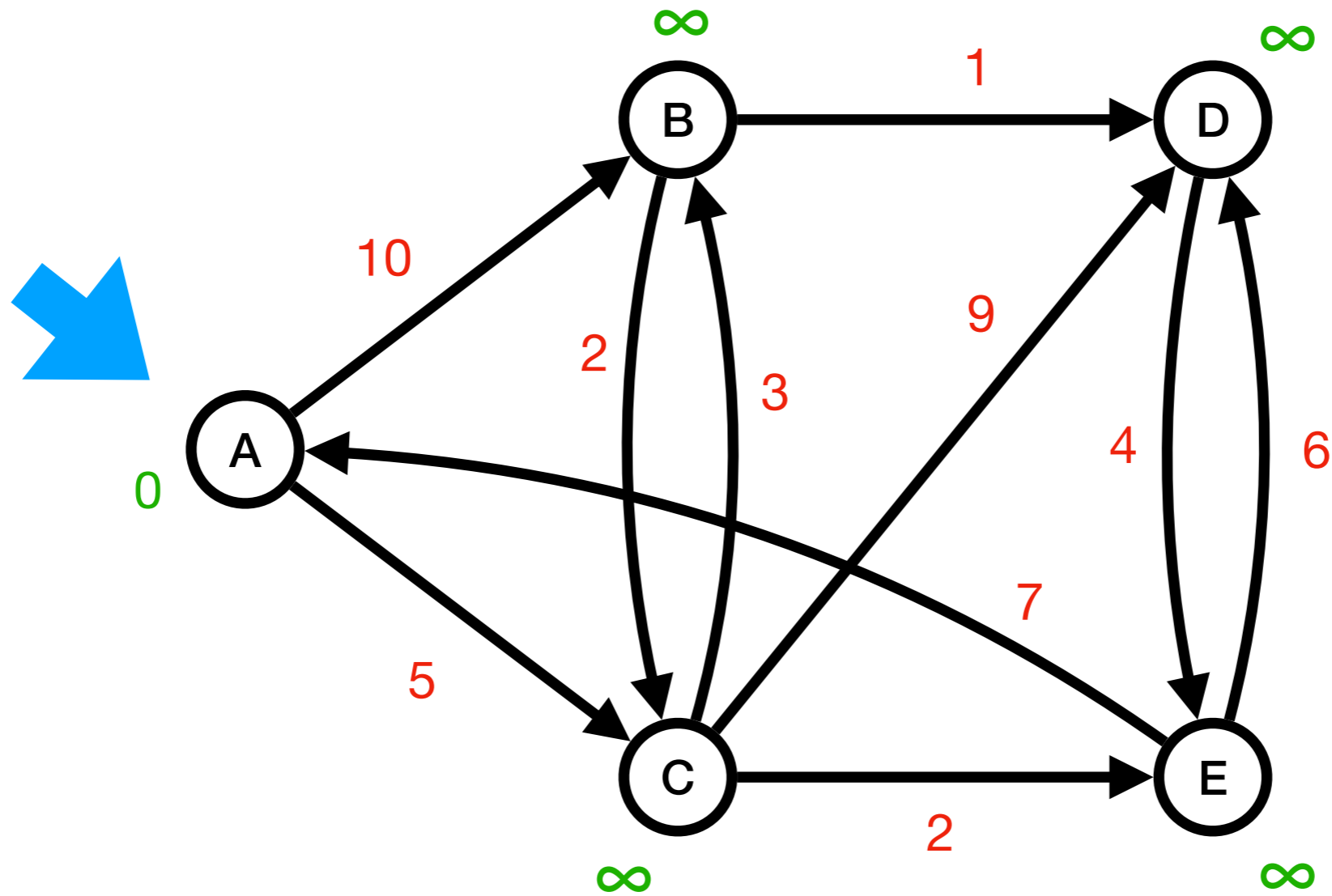
Algorithme de Dijkstra



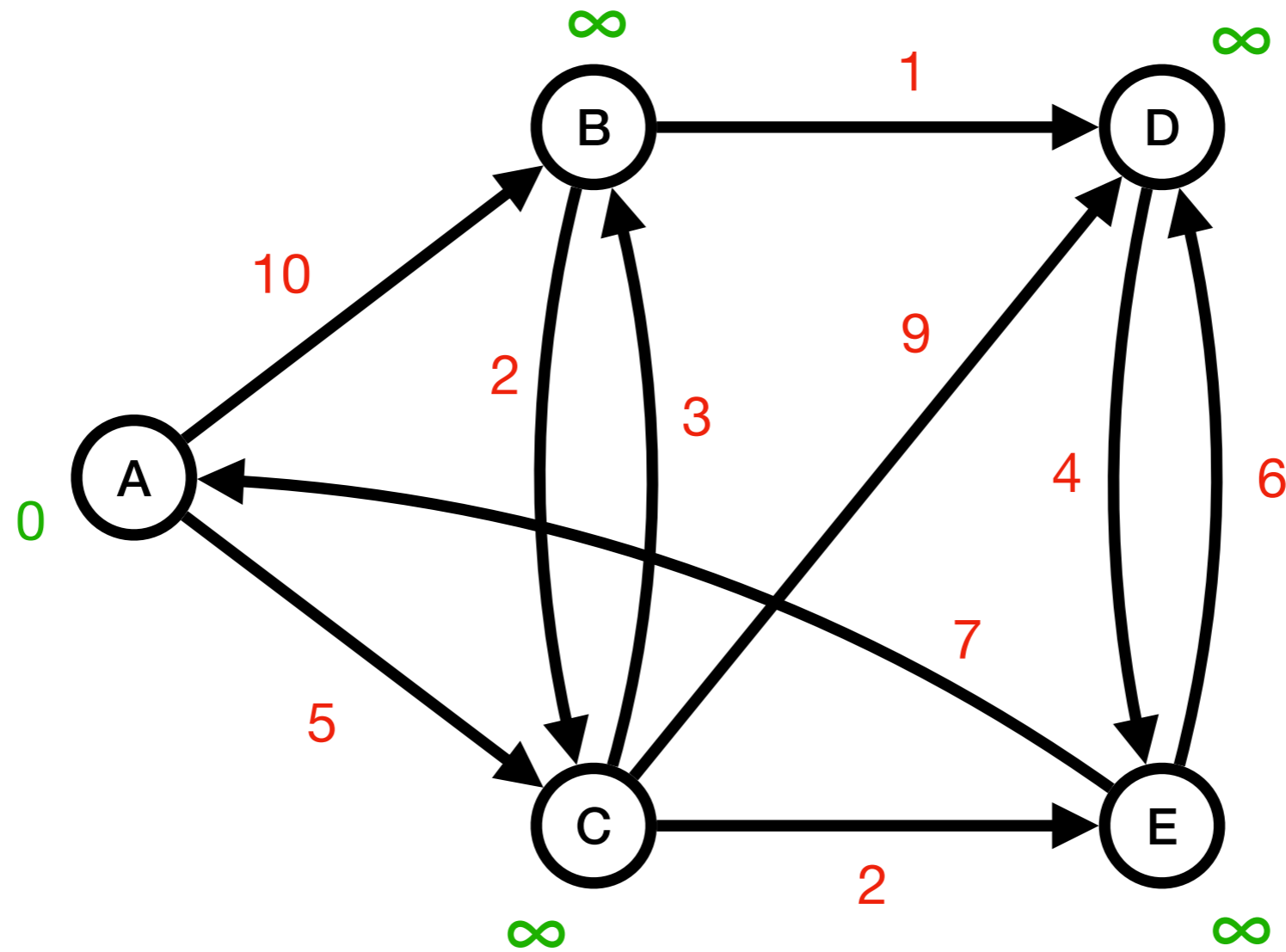
Algorithme de Dijkstra



Algorithme de Dijkstra



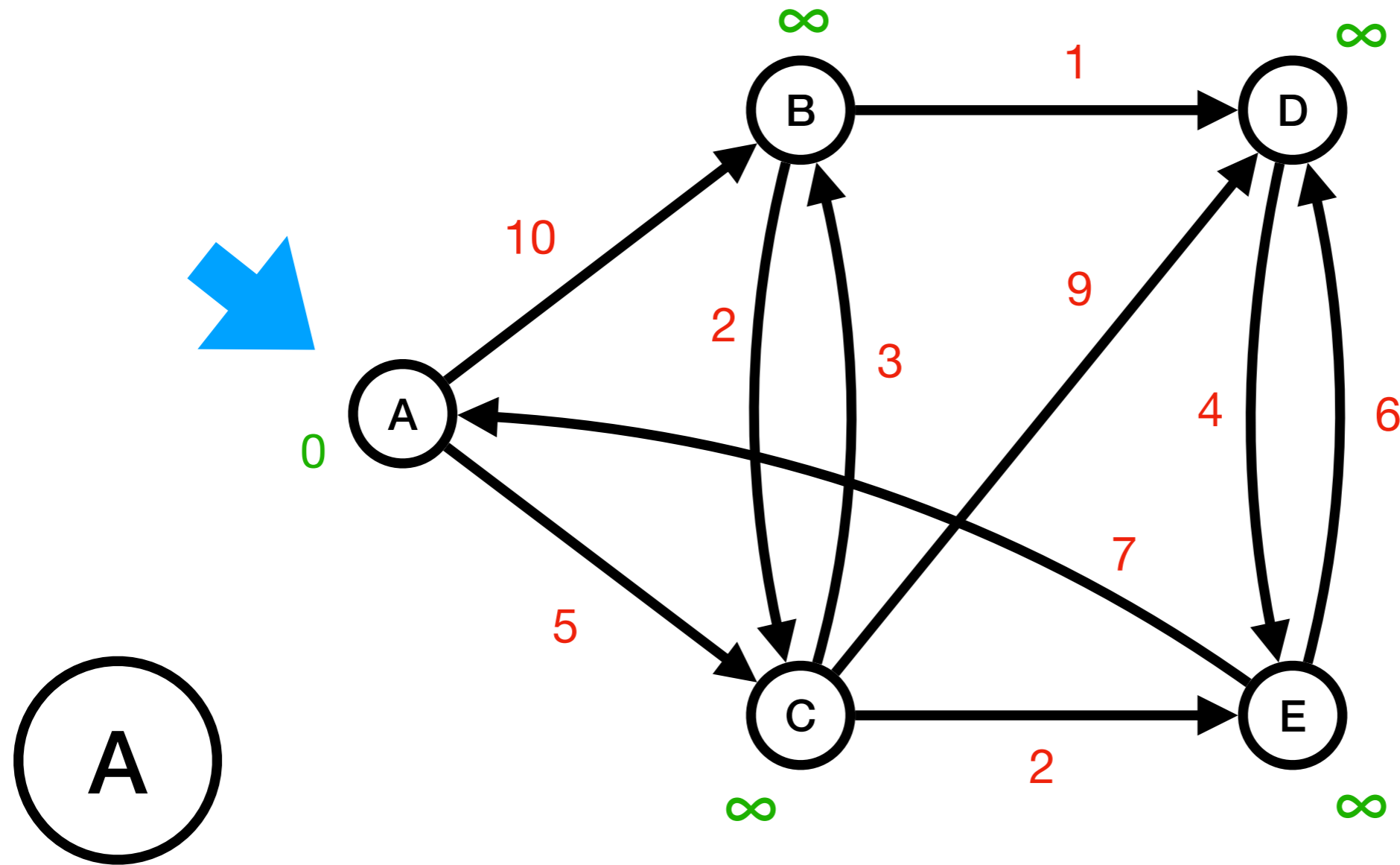
Algorithme de Dijkstra



File de priorité →

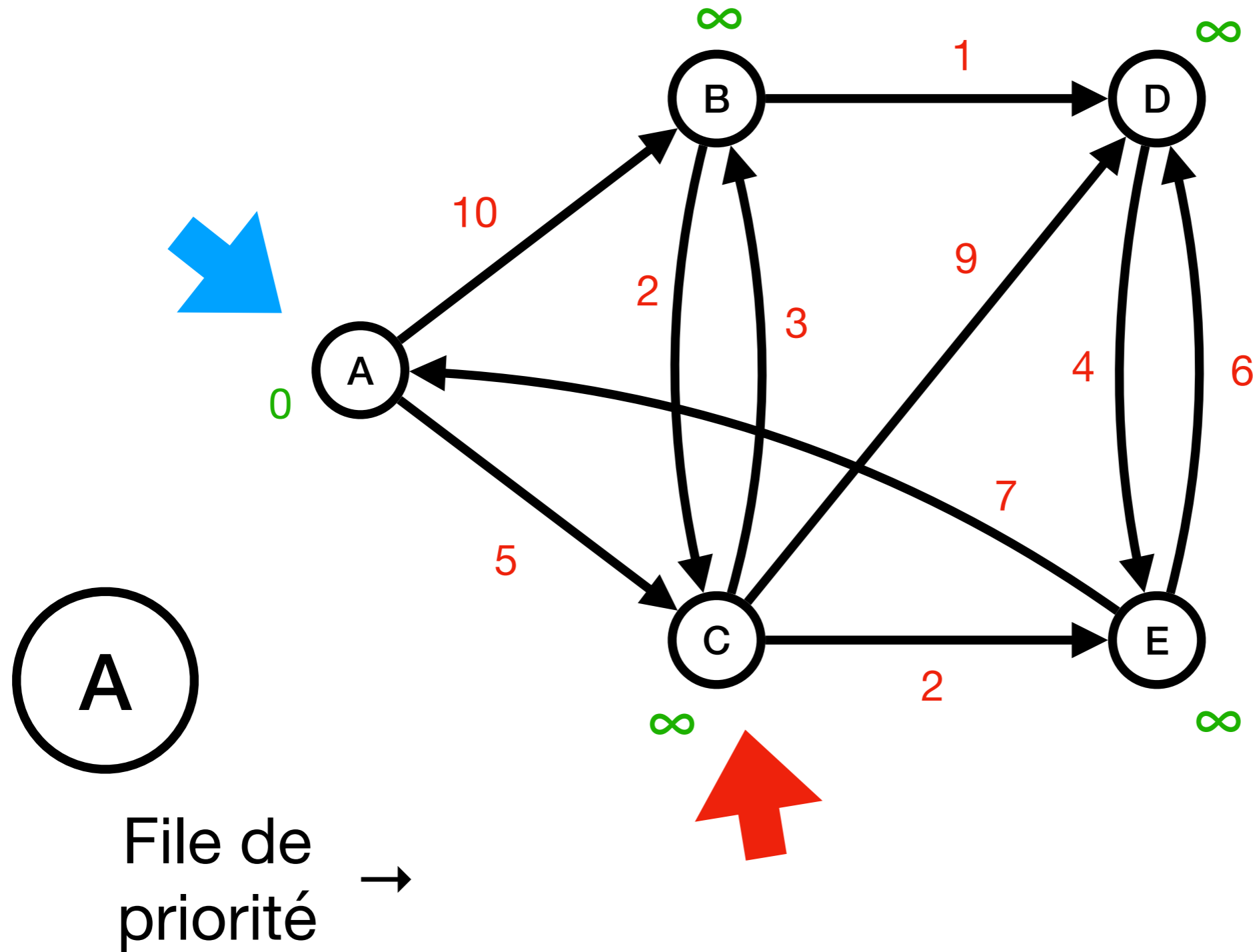


Algorithme de Dijkstra

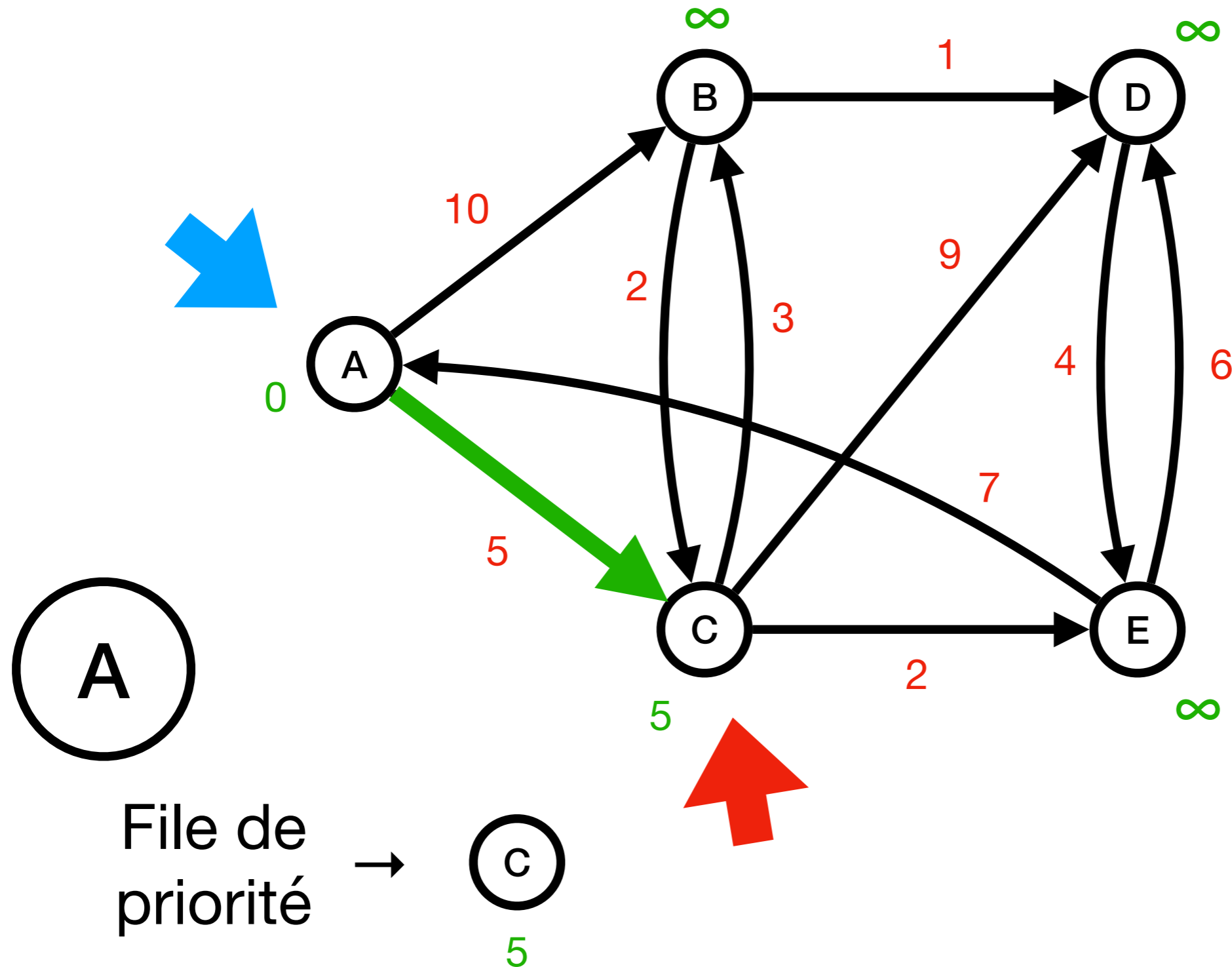


File de priorité →

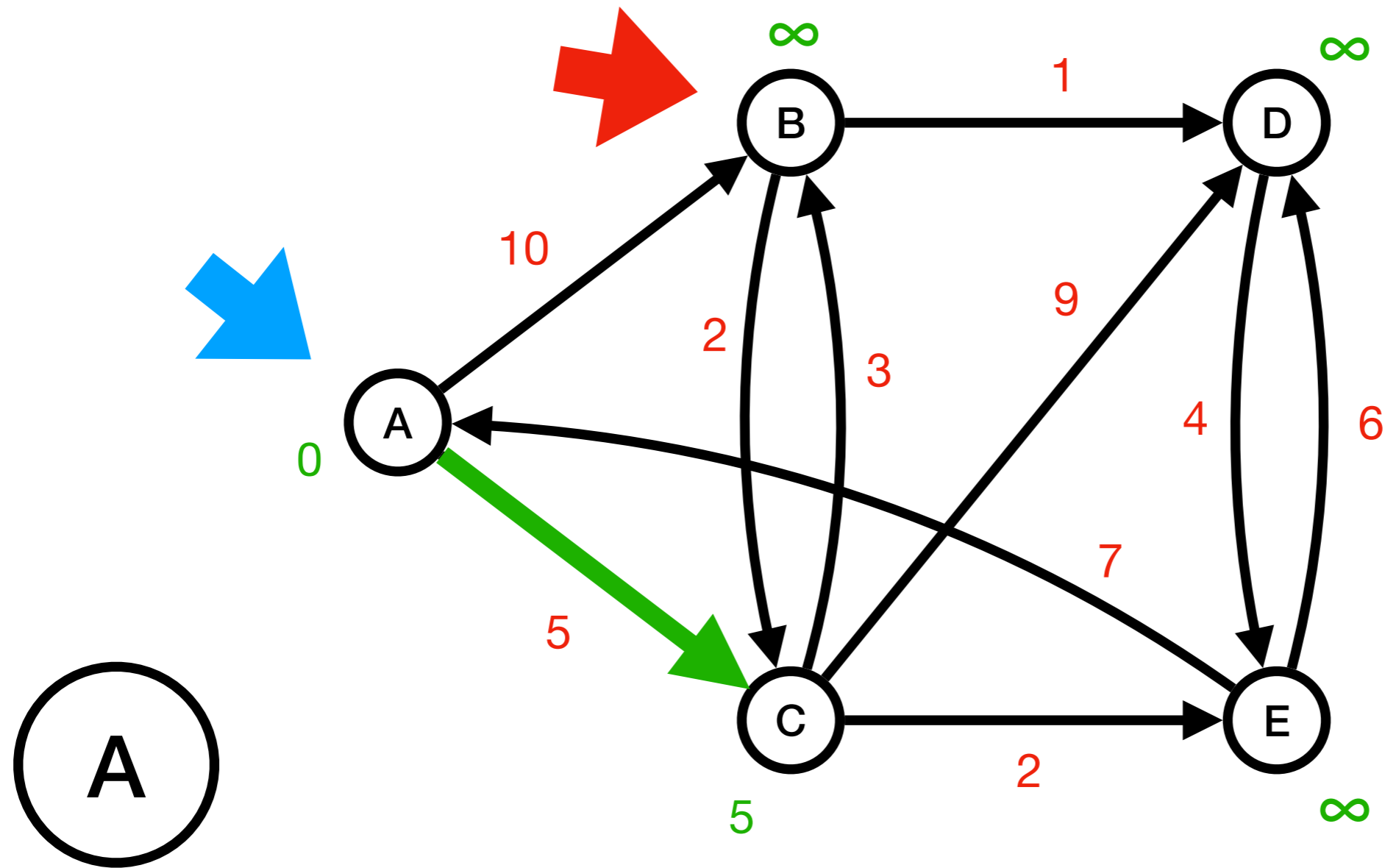
Algorithme de Dijkstra



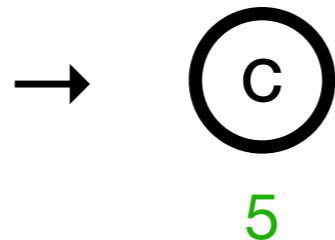
Algorithme de Dijkstra



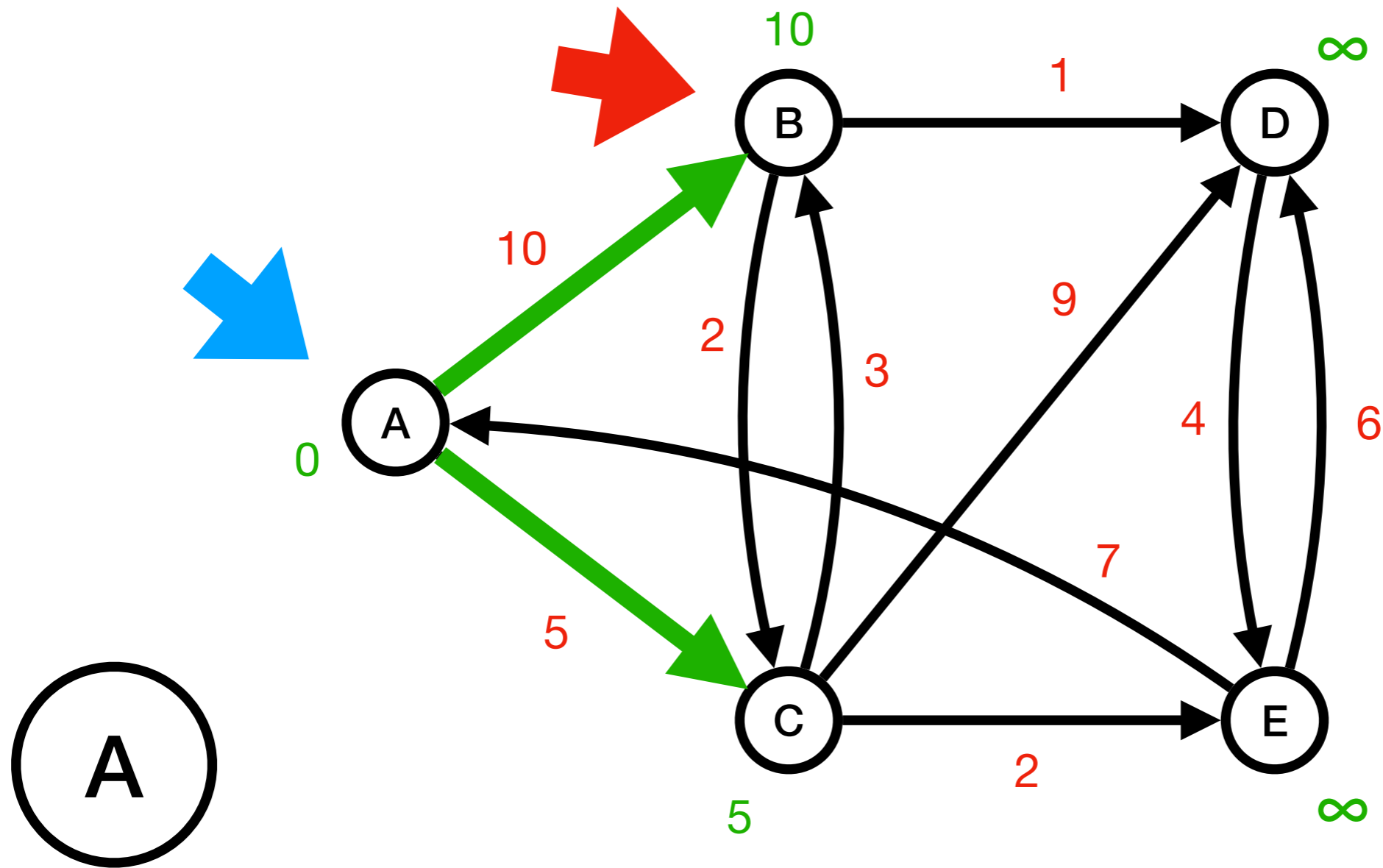
Algorithme de Dijkstra



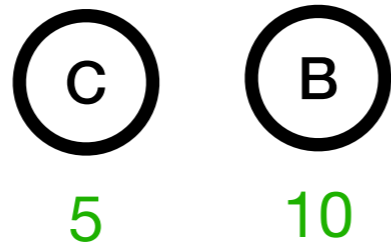
File de
priorité



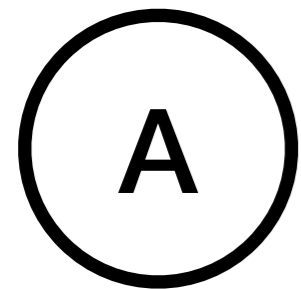
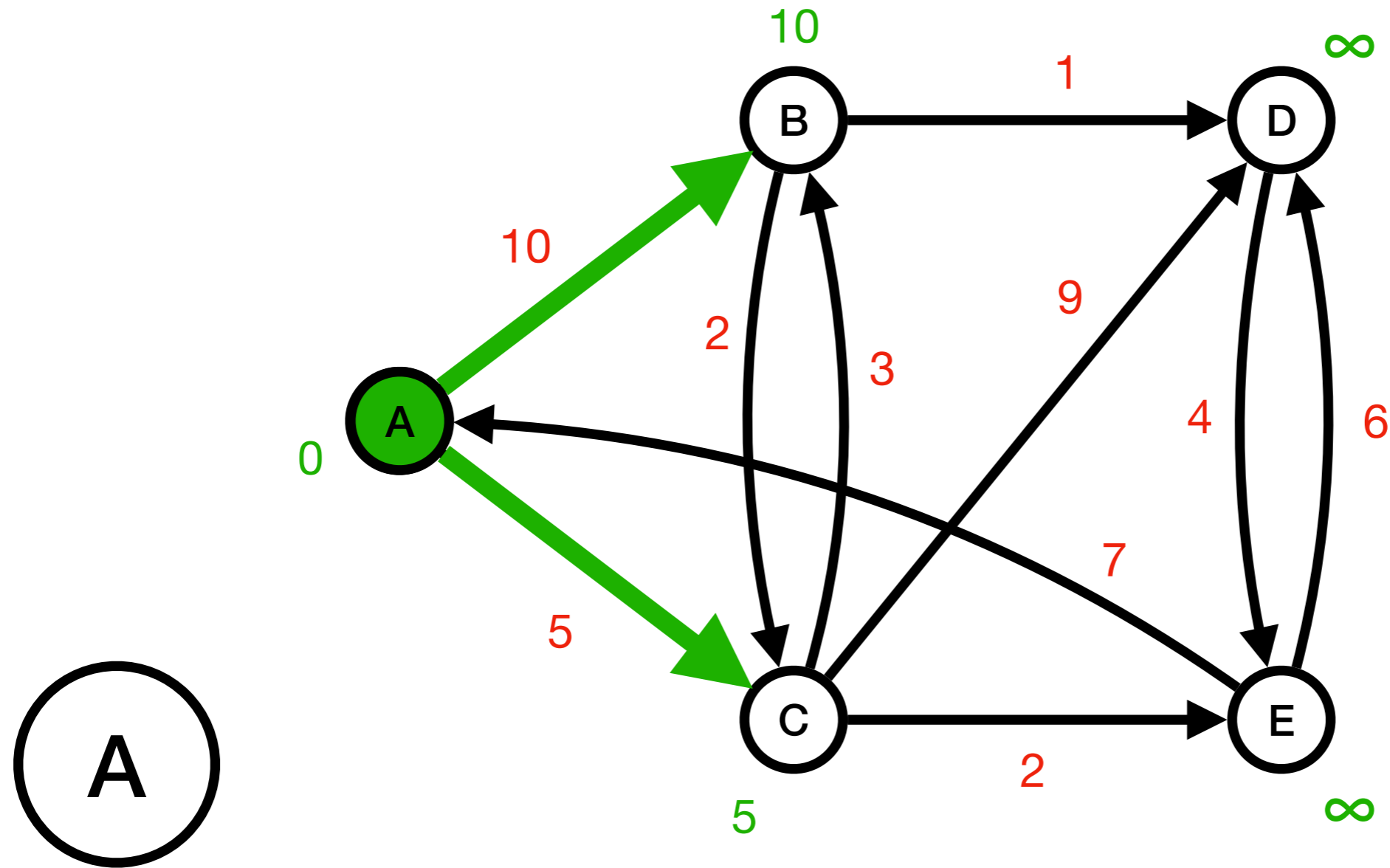
Algorithme de Dijkstra



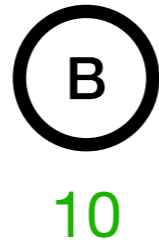
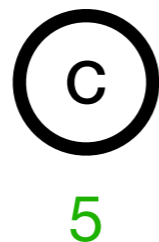
File de priorité →



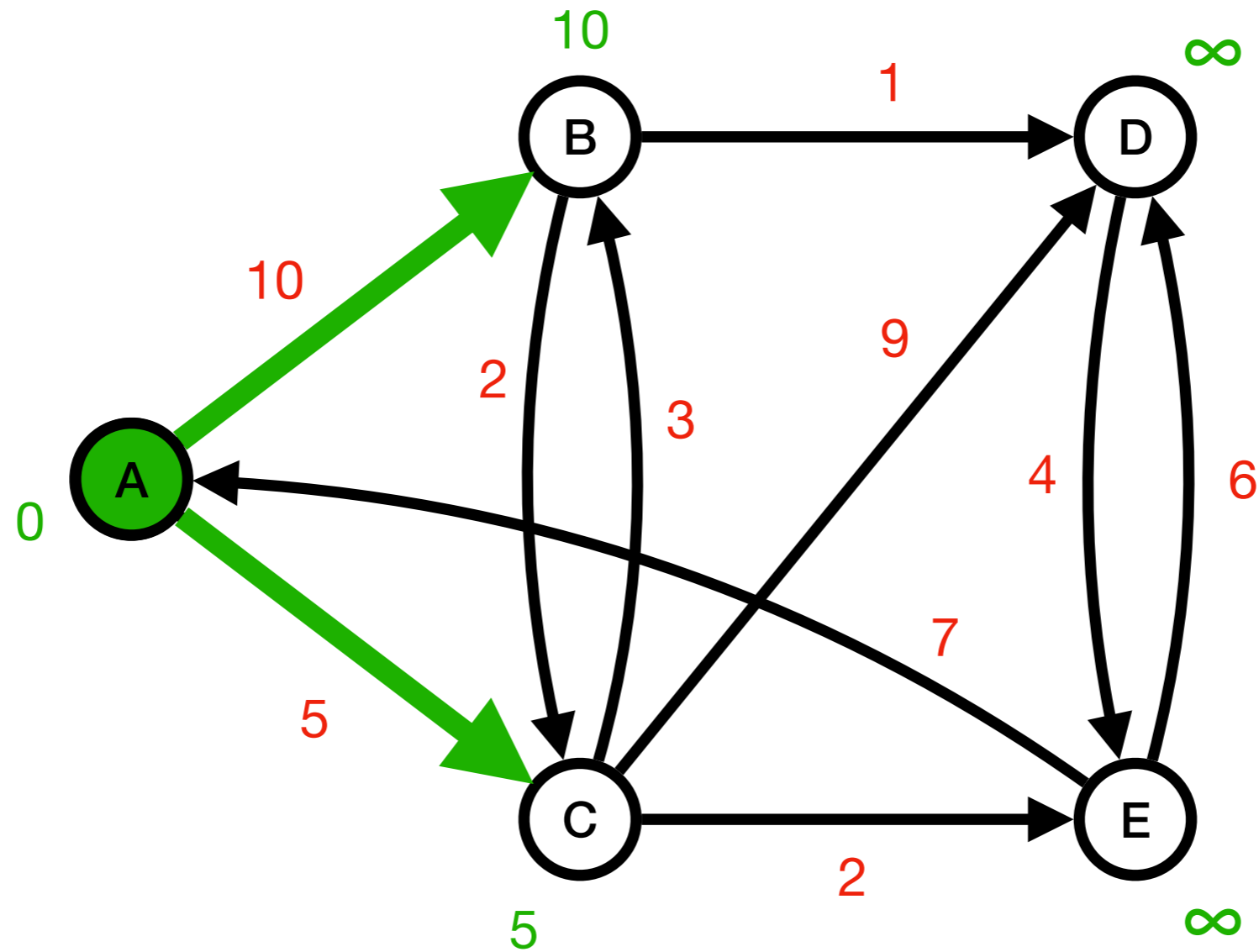
Algorithme de Dijkstra



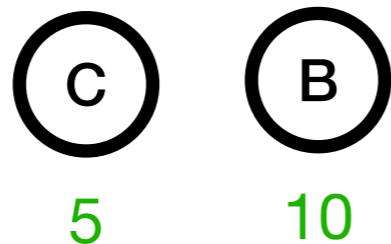
File de
priorité →



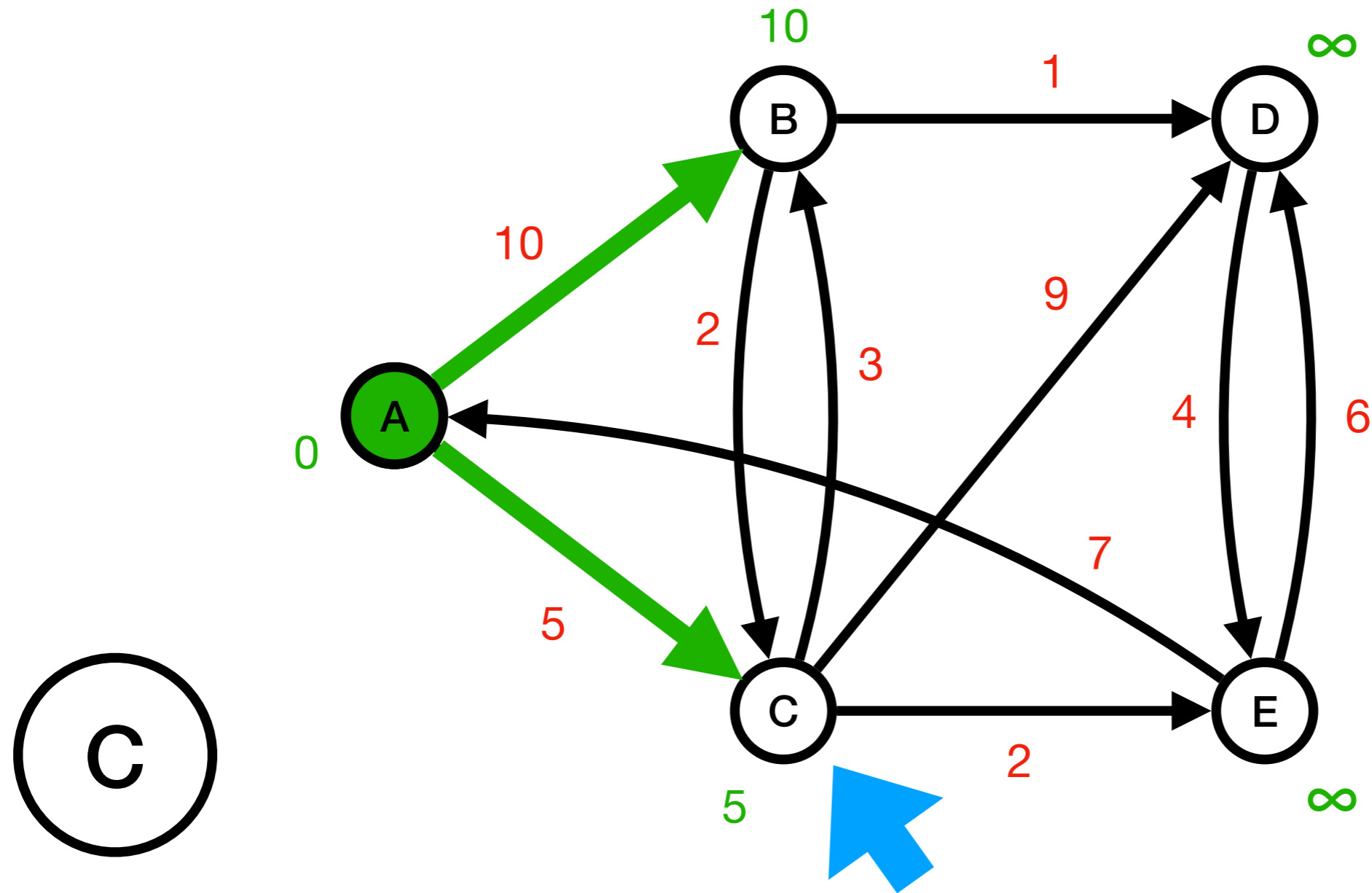
Algorithme de Dijkstra



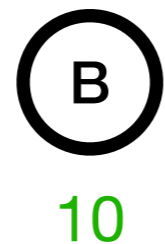
File de priorité →



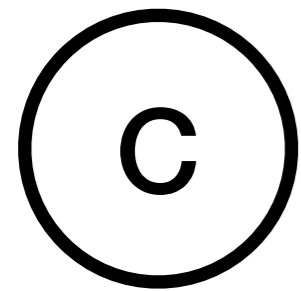
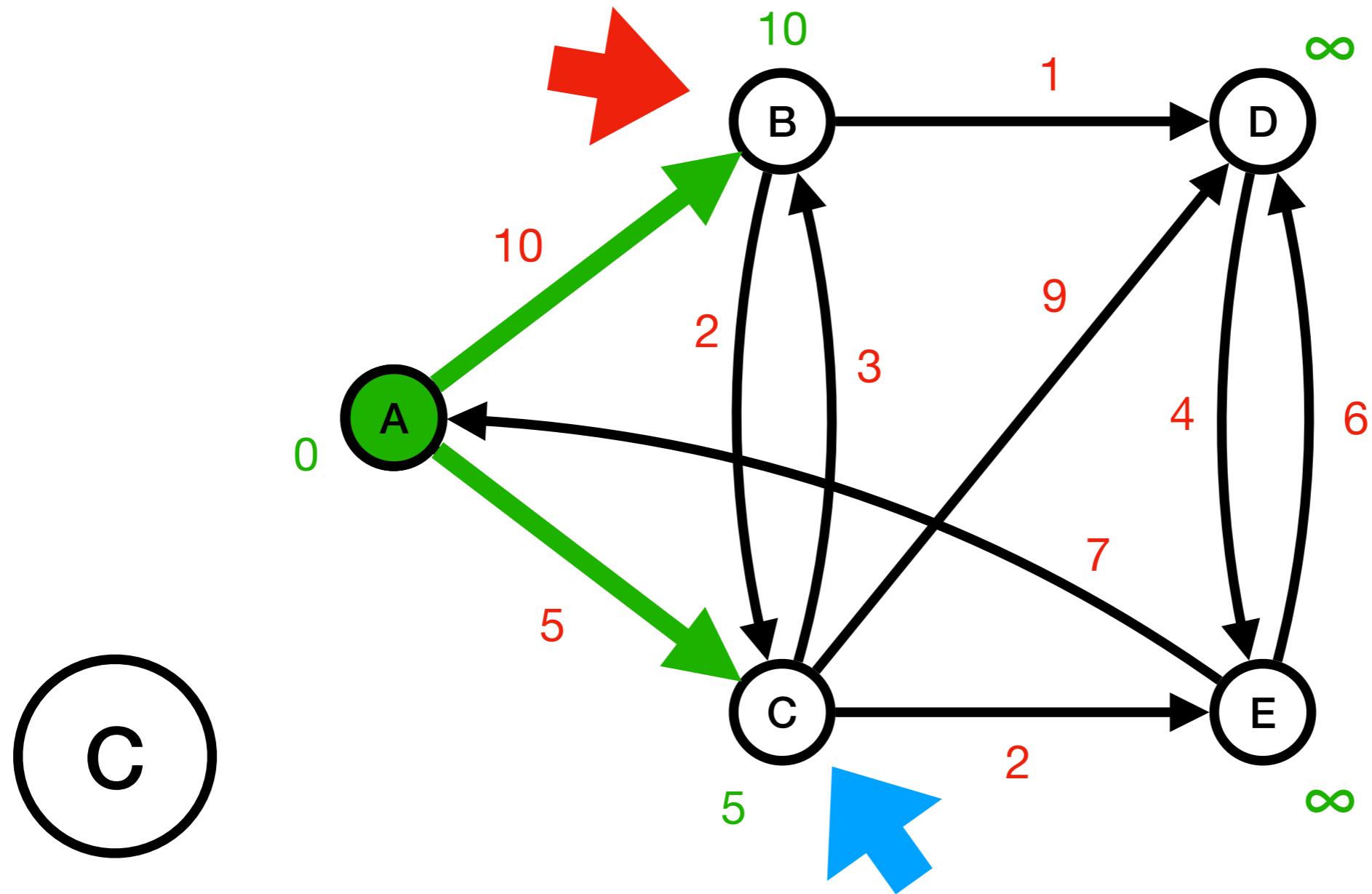
Algorithme de Dijkstra



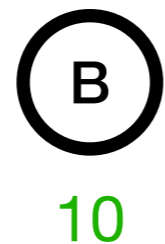
File de priorité →



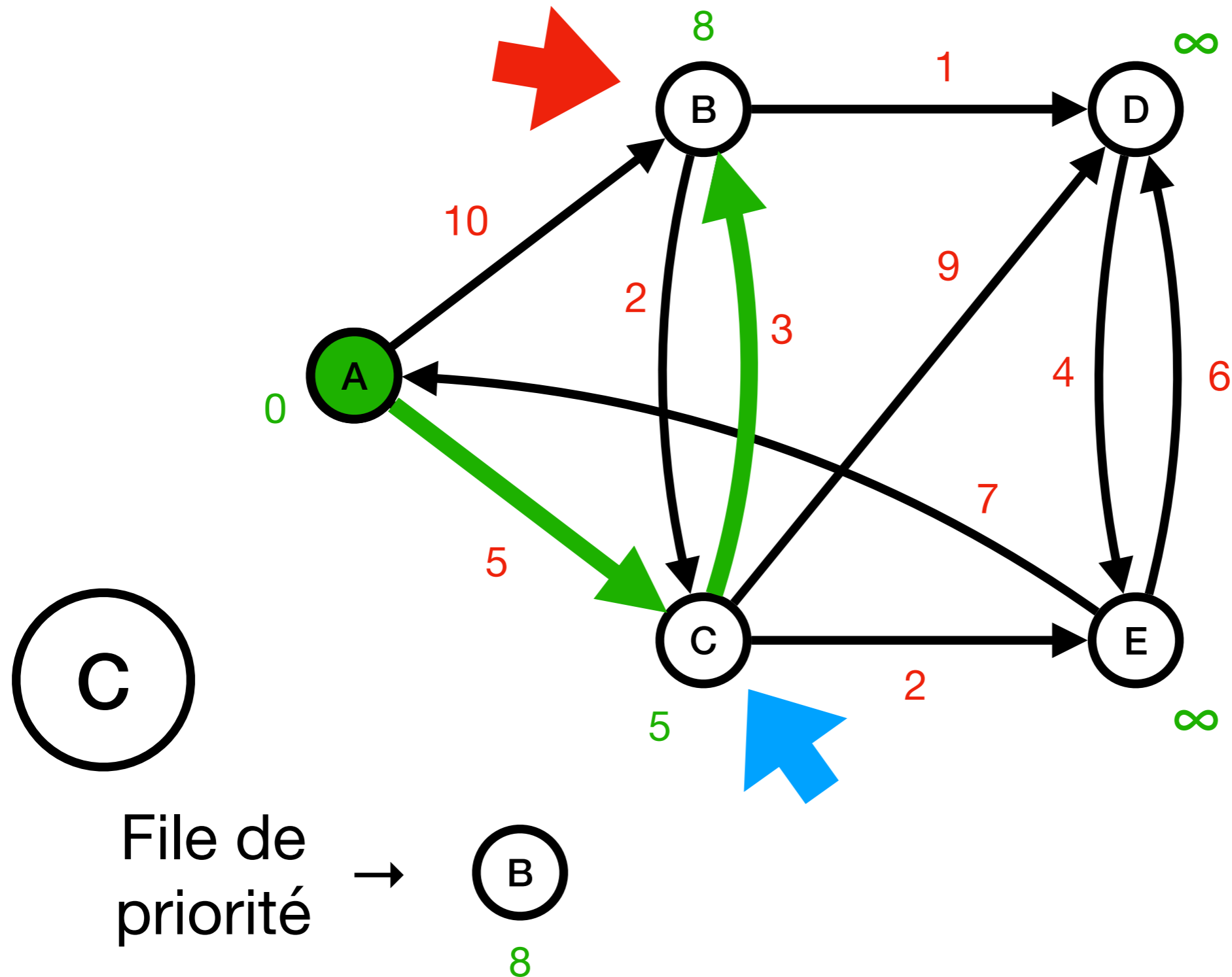
Algorithme de Dijkstra



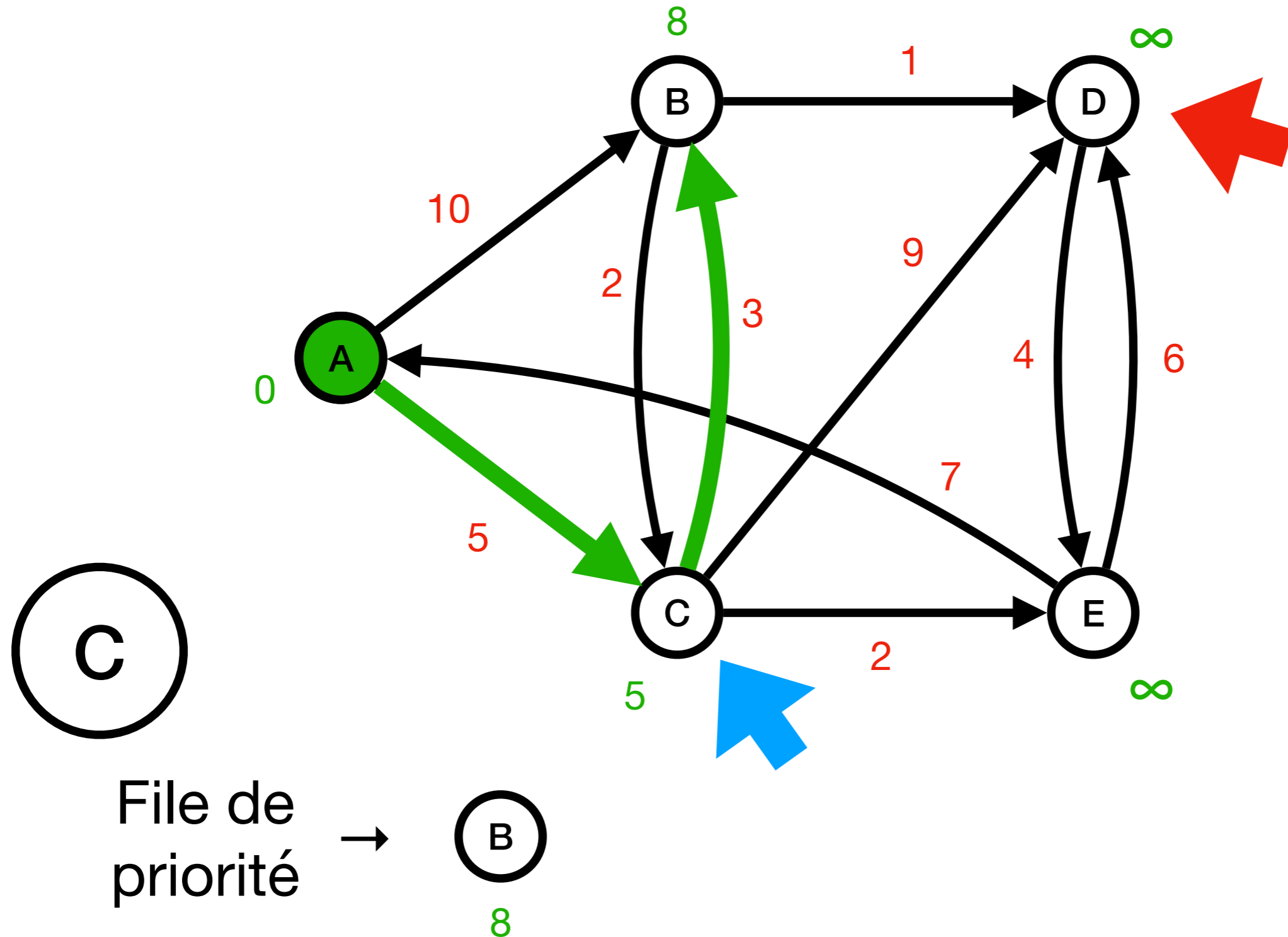
File de
priorité →



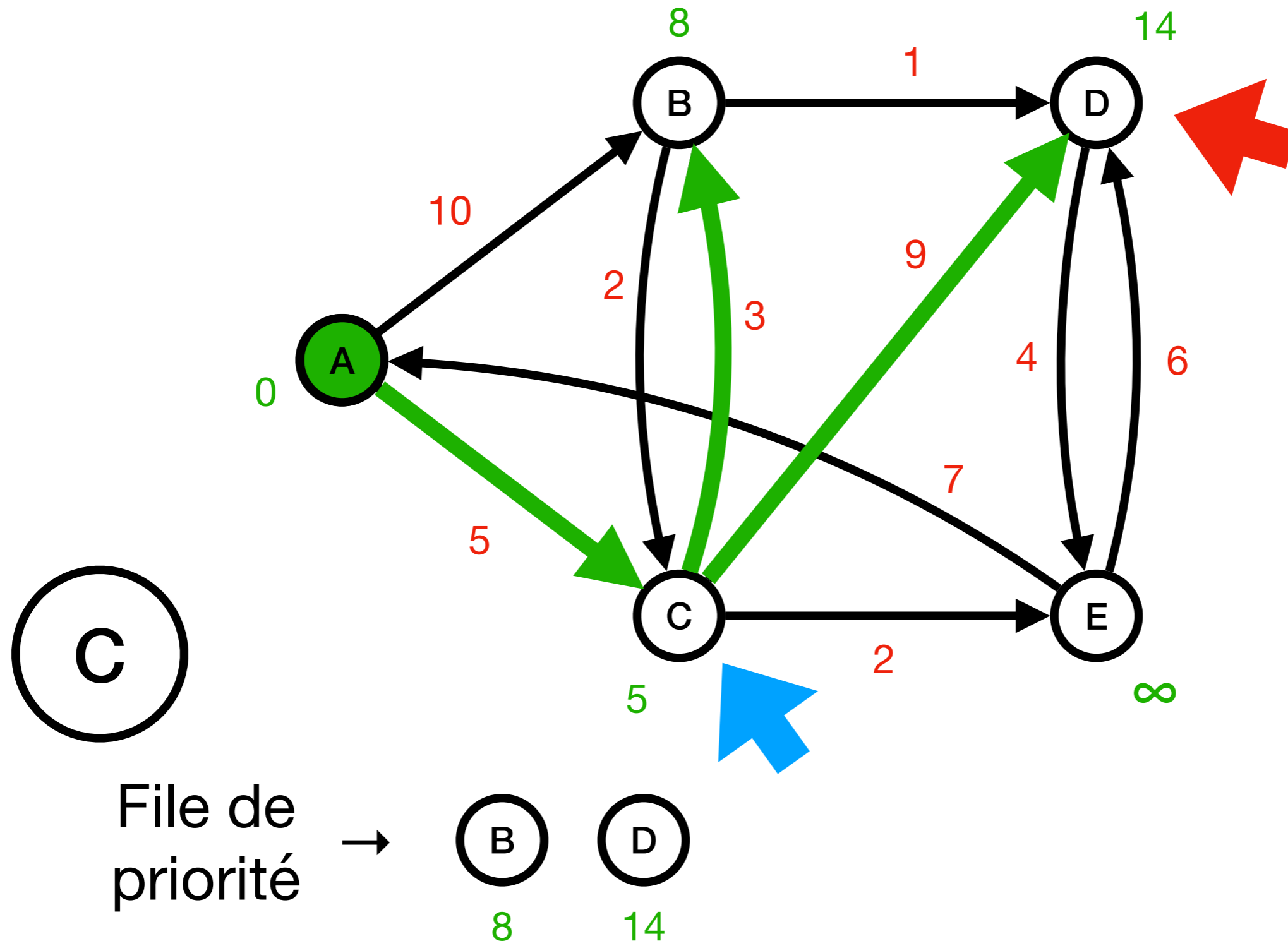
Algorithme de Dijkstra



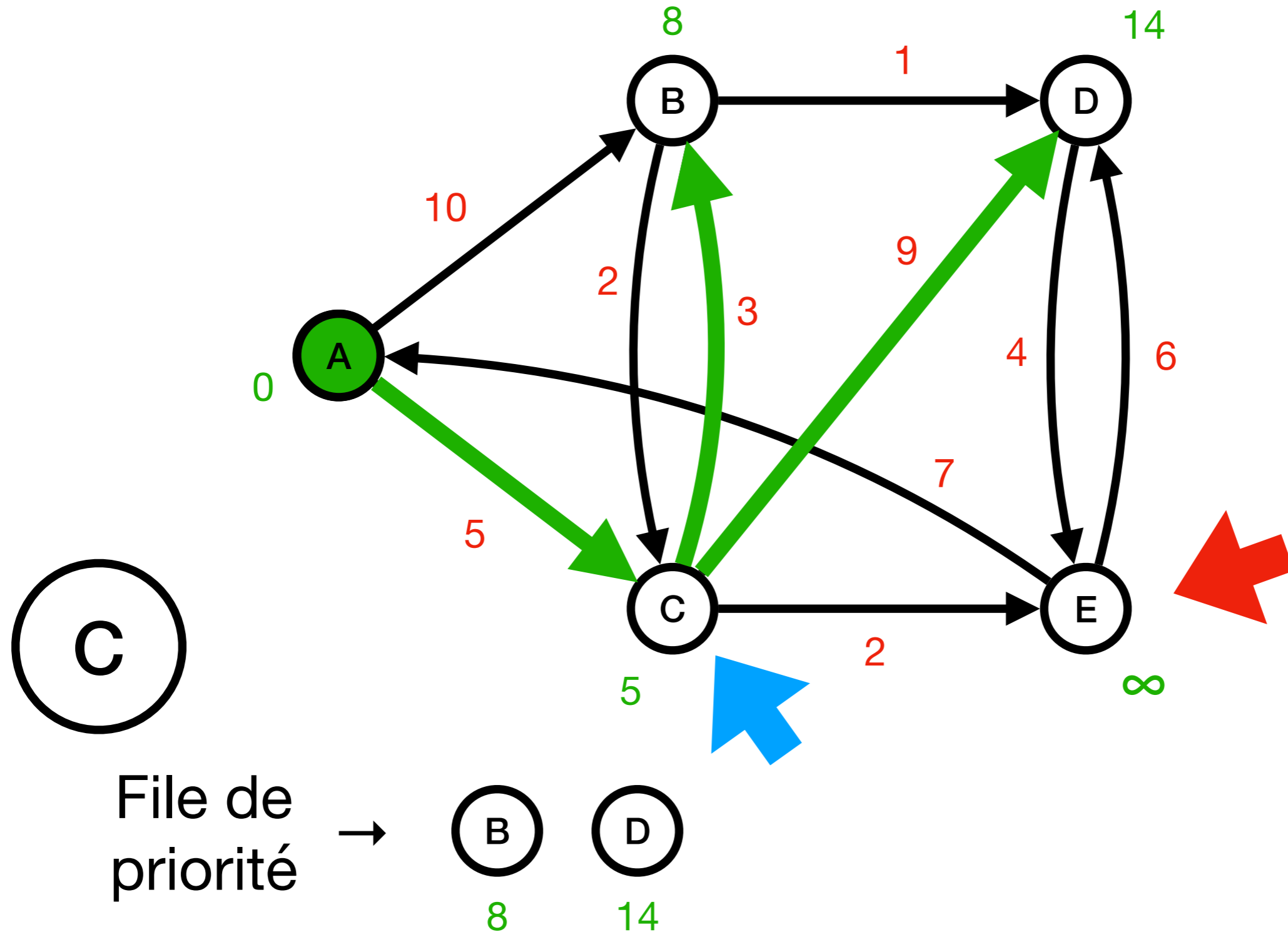
Algorithme de Dijkstra



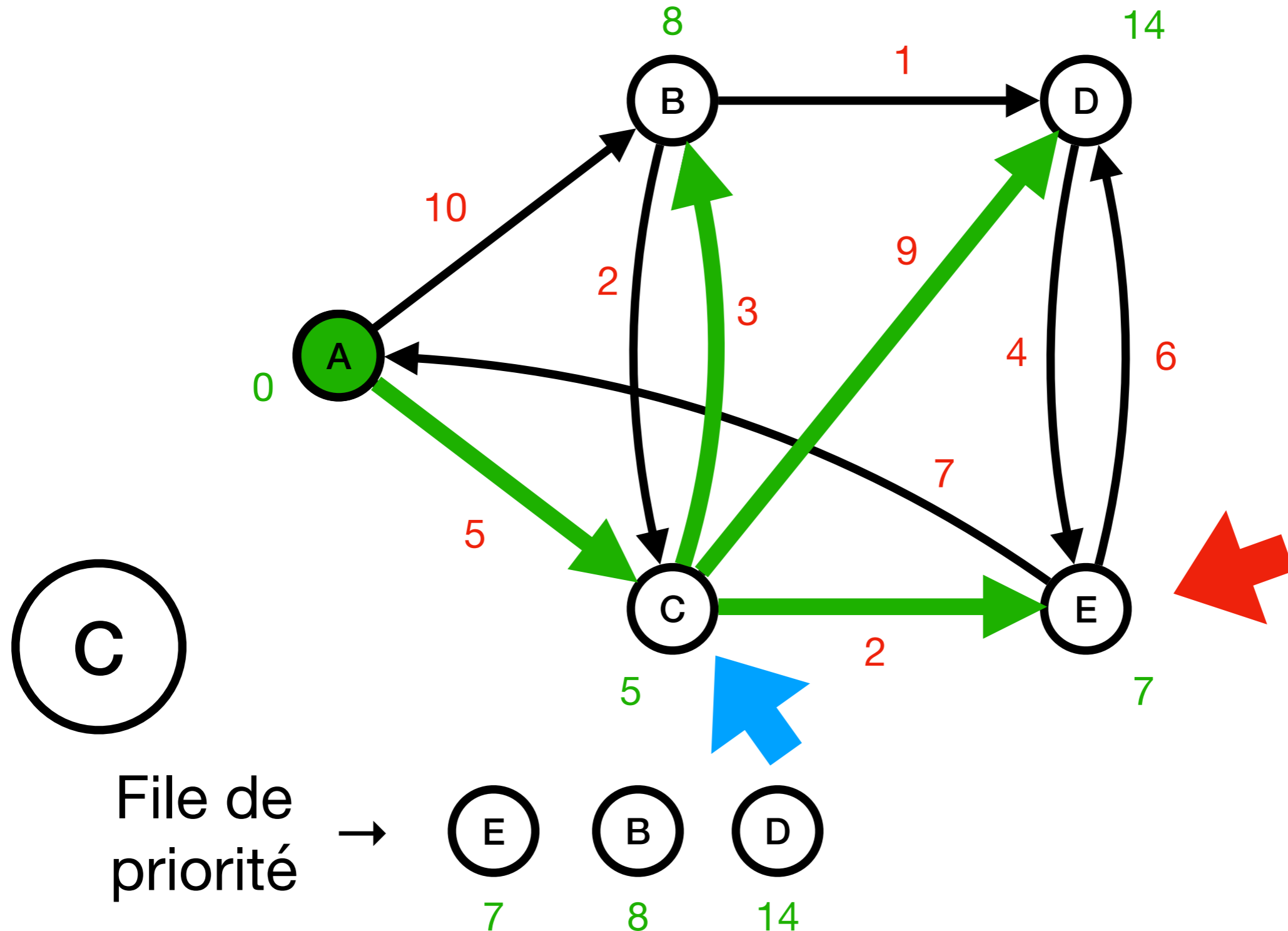
Algorithme de Dijkstra



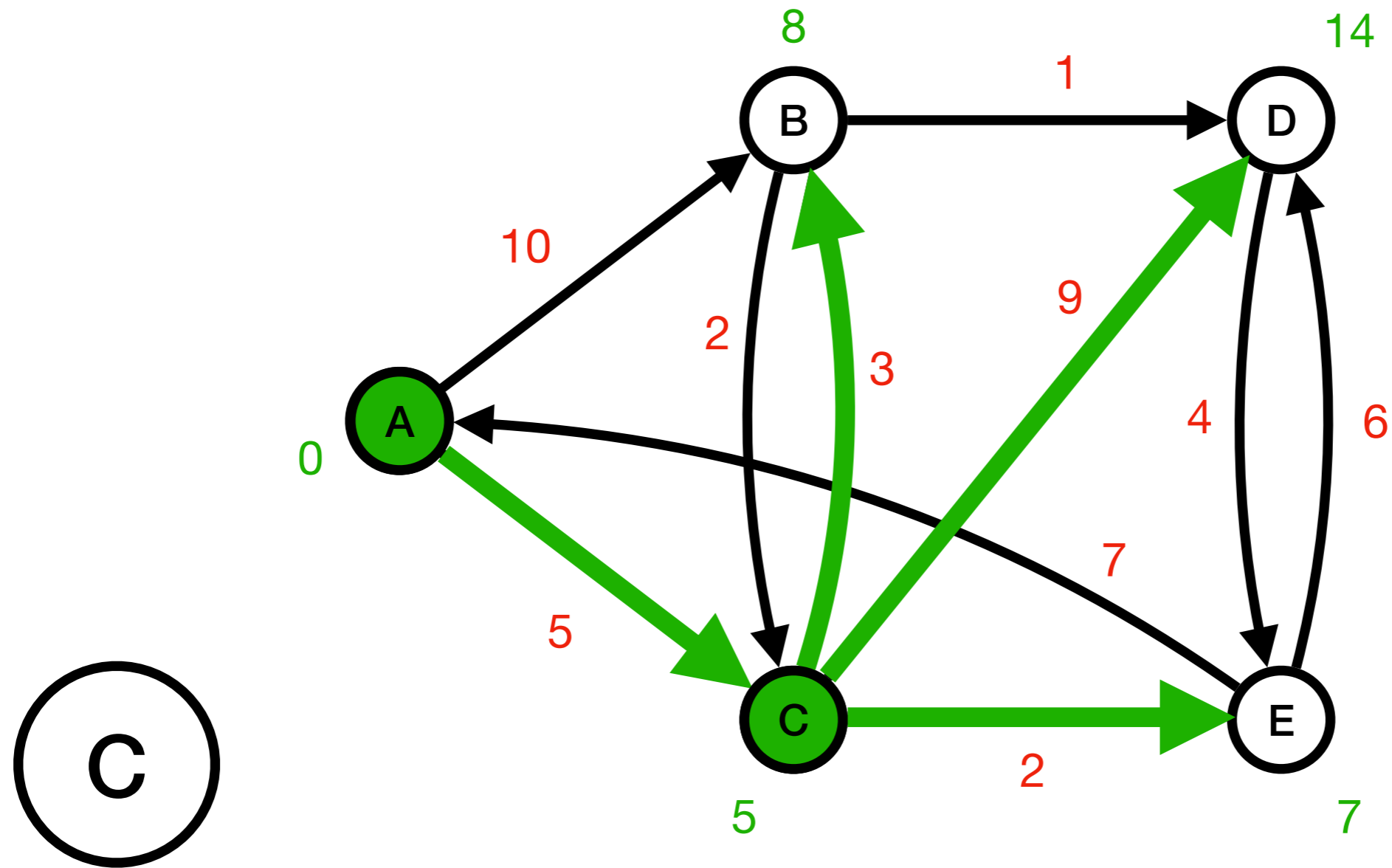
Algorithme de Dijkstra



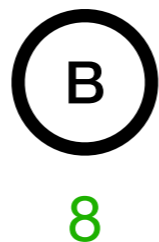
Algorithme de Dijkstra



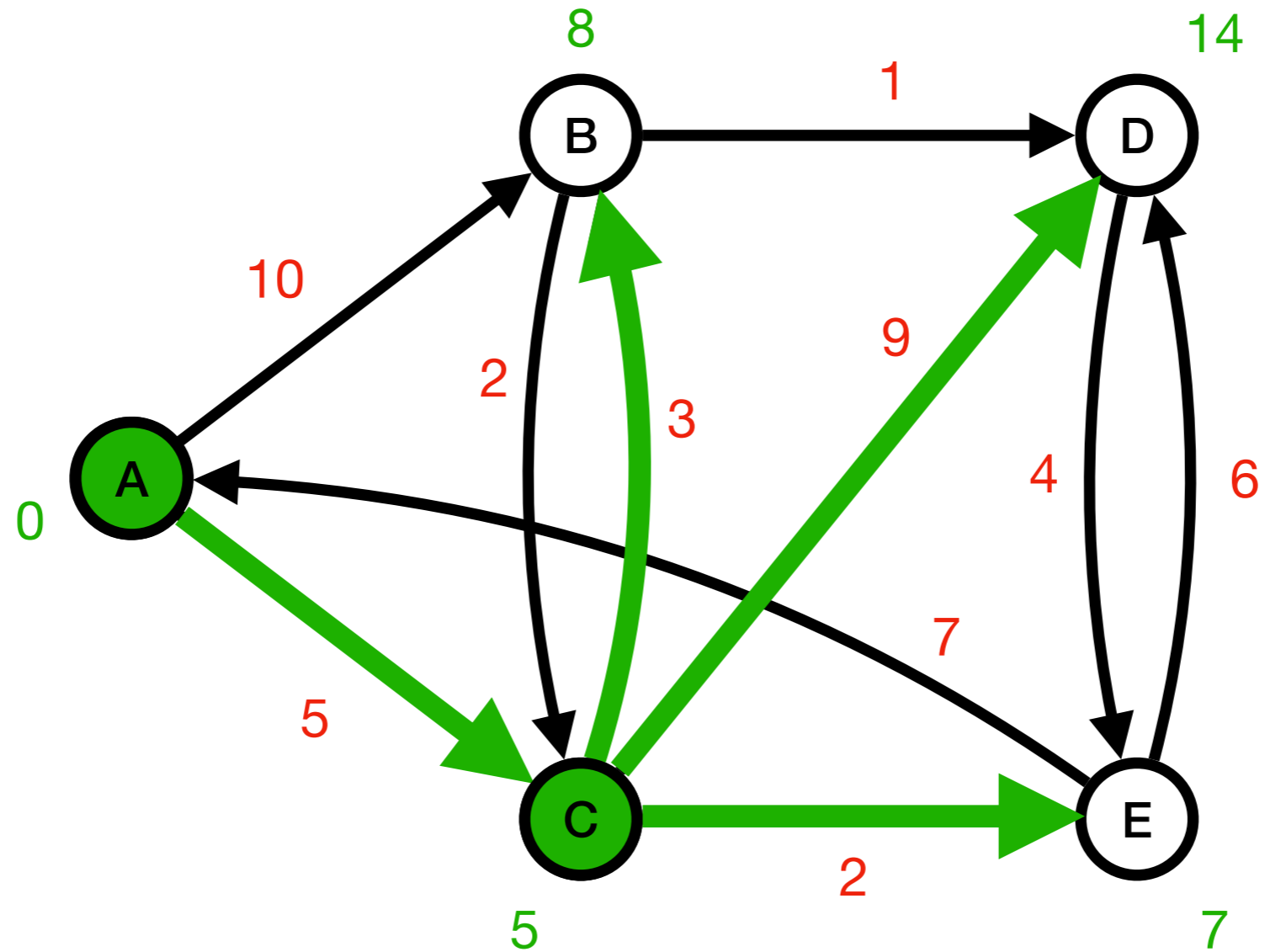
Algorithme de Dijkstra



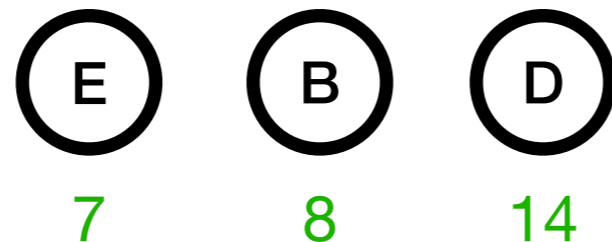
File de priorité →



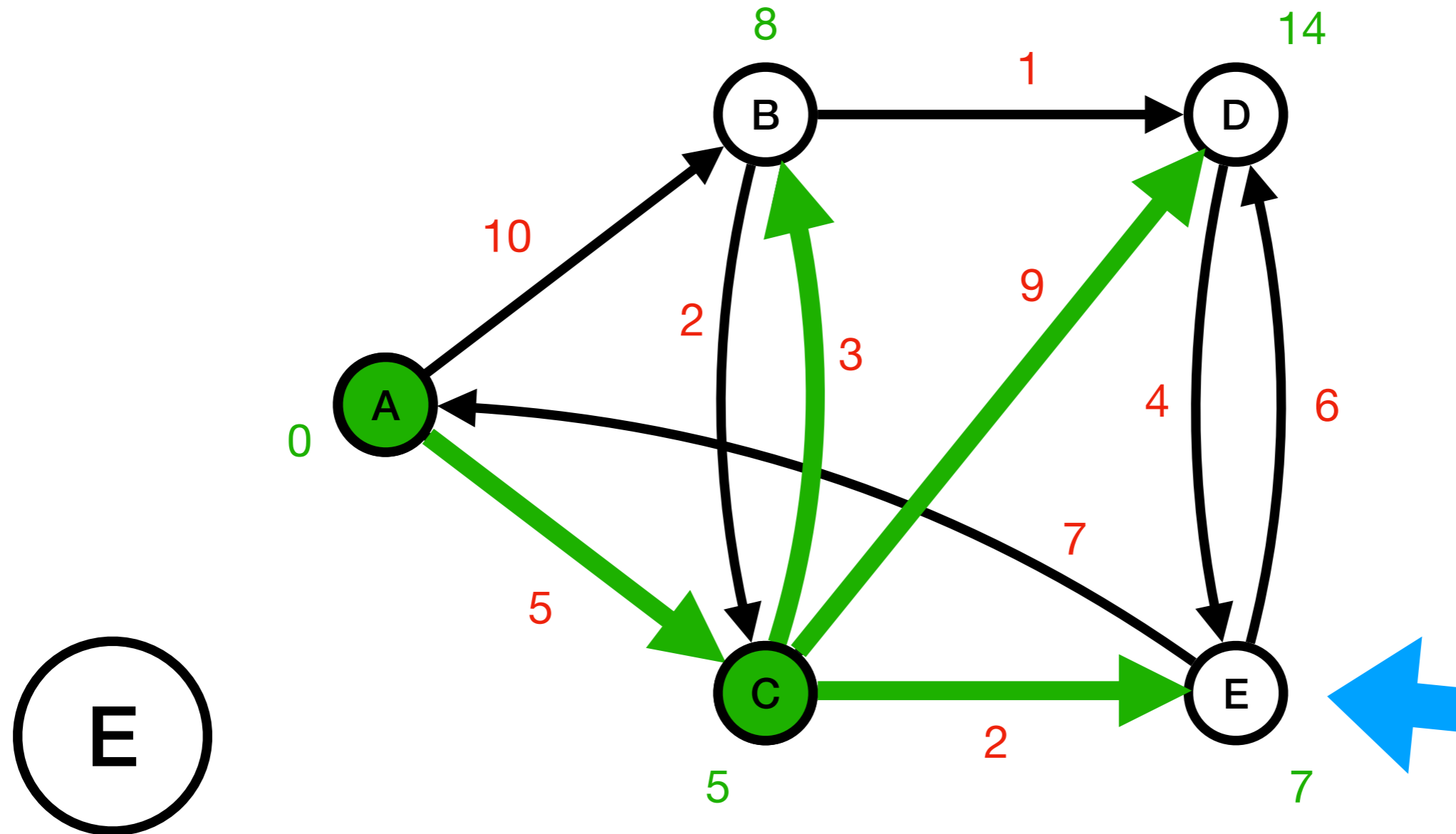
Algorithme de Dijkstra



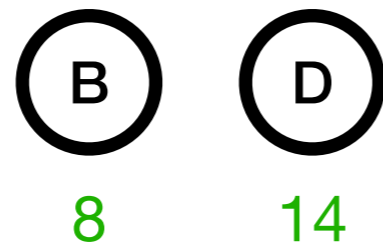
File de priorité →



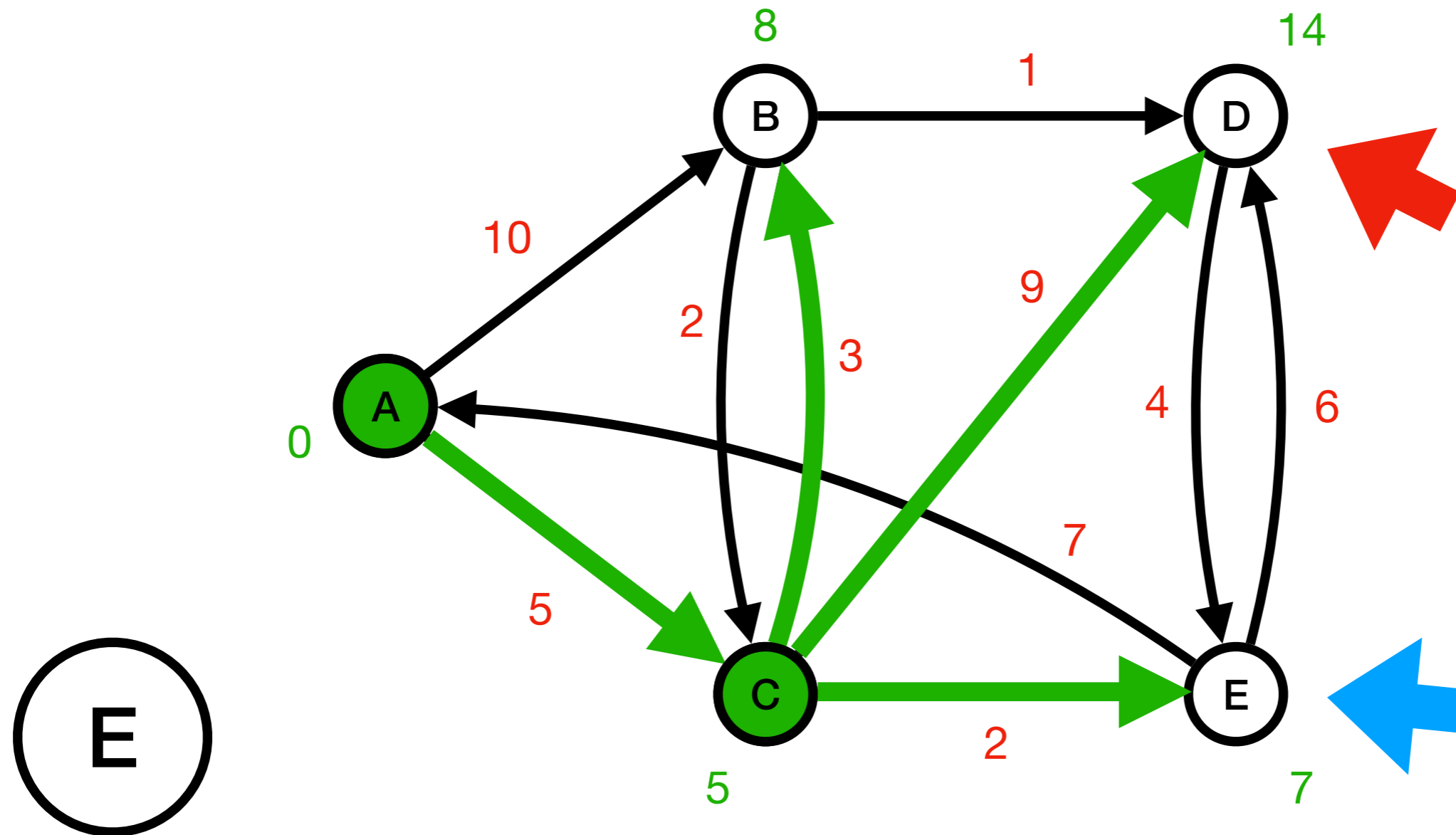
Algorithme de Dijkstra



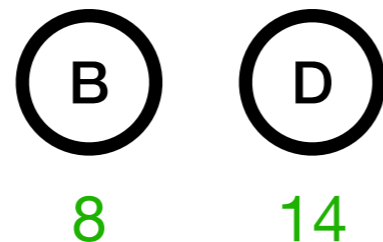
File de priorité →



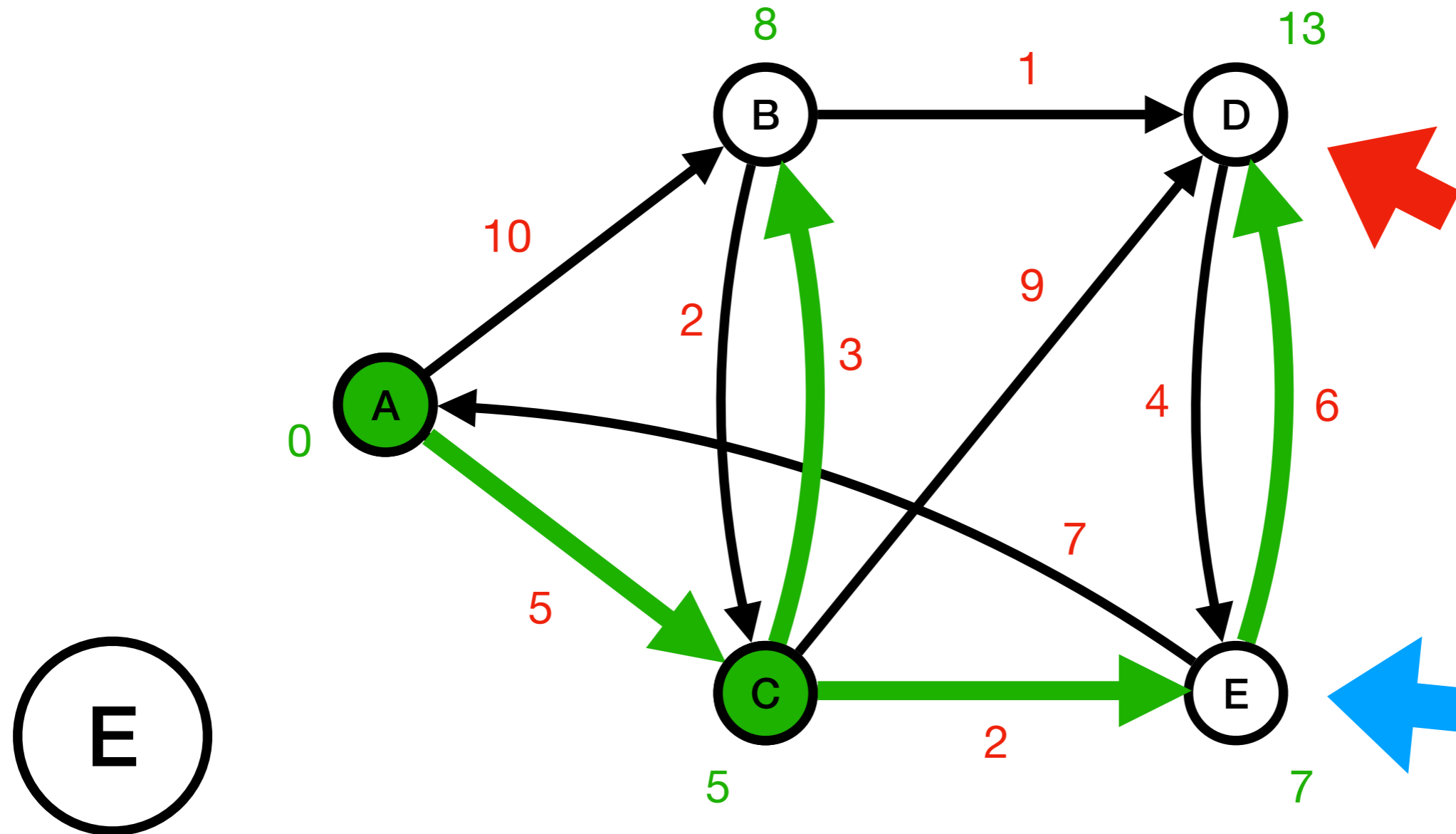
Algorithme de Dijkstra



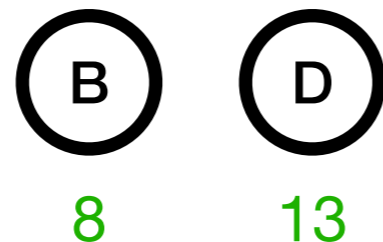
File de priorité →



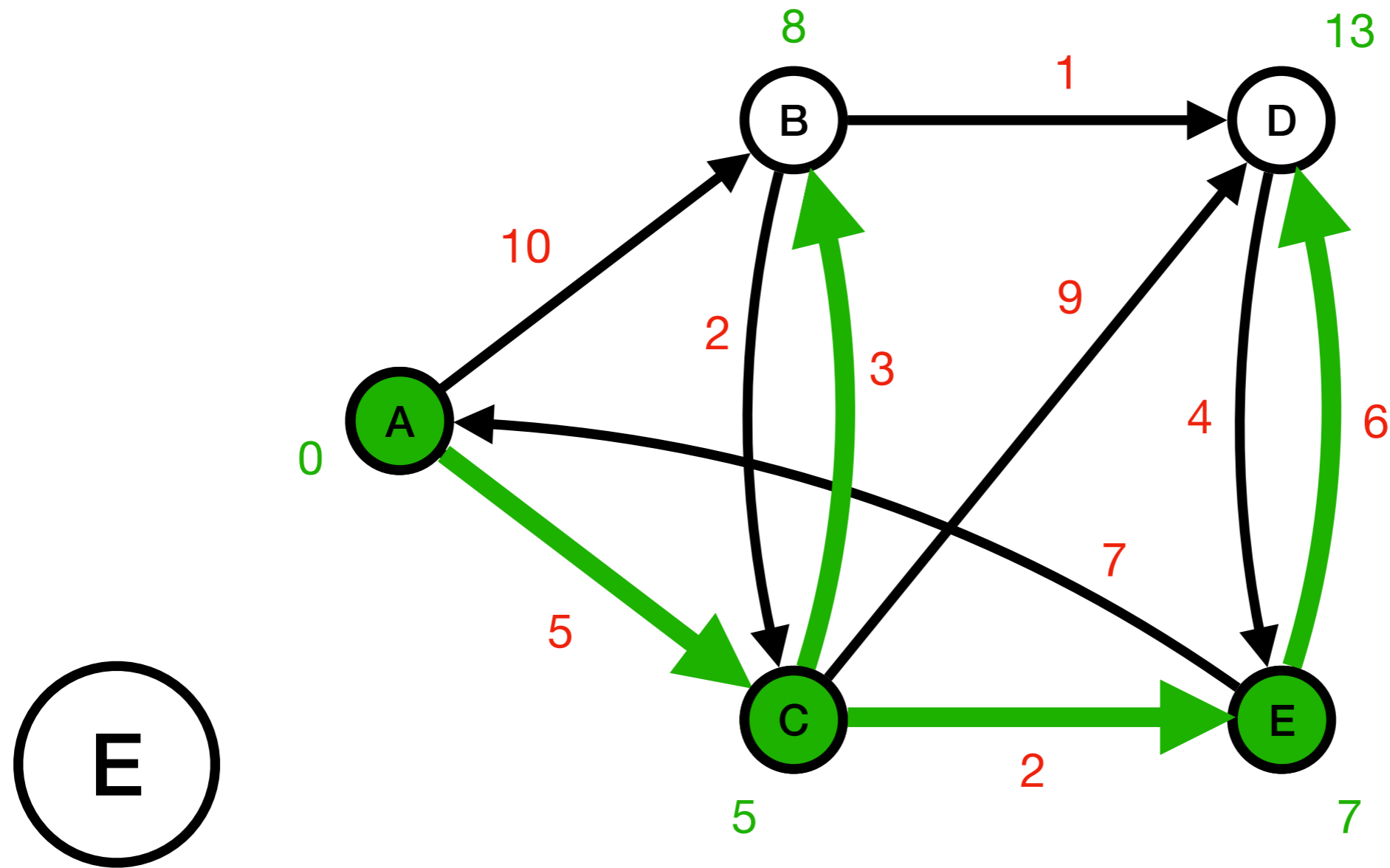
Algorithme de Dijkstra



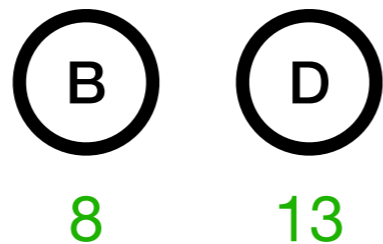
File de priorité →



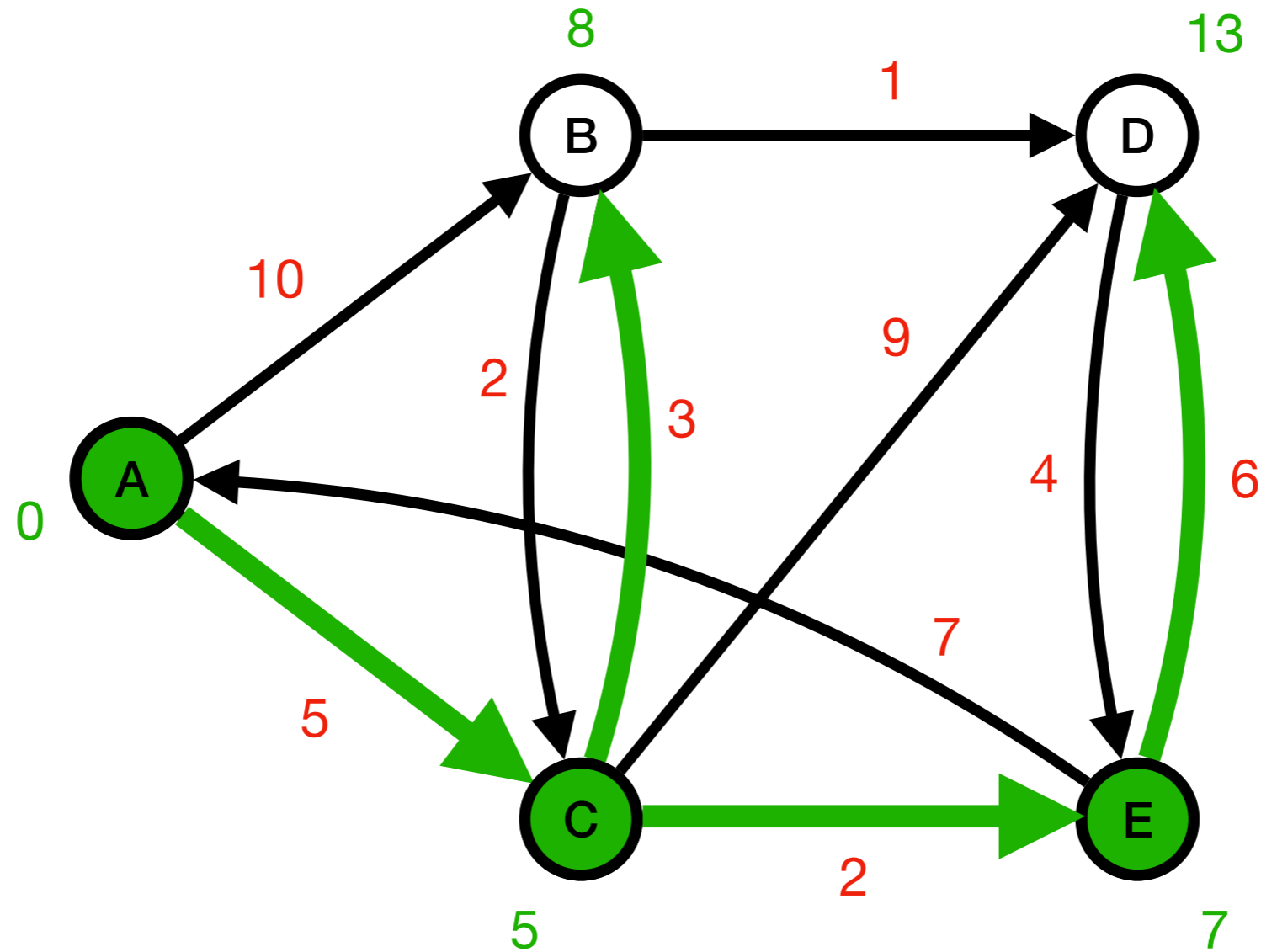
Algorithme de Dijkstra



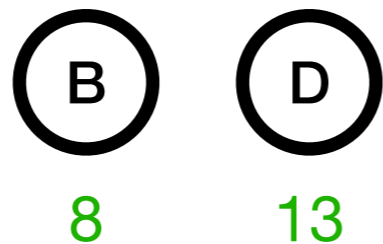
File de priorité →



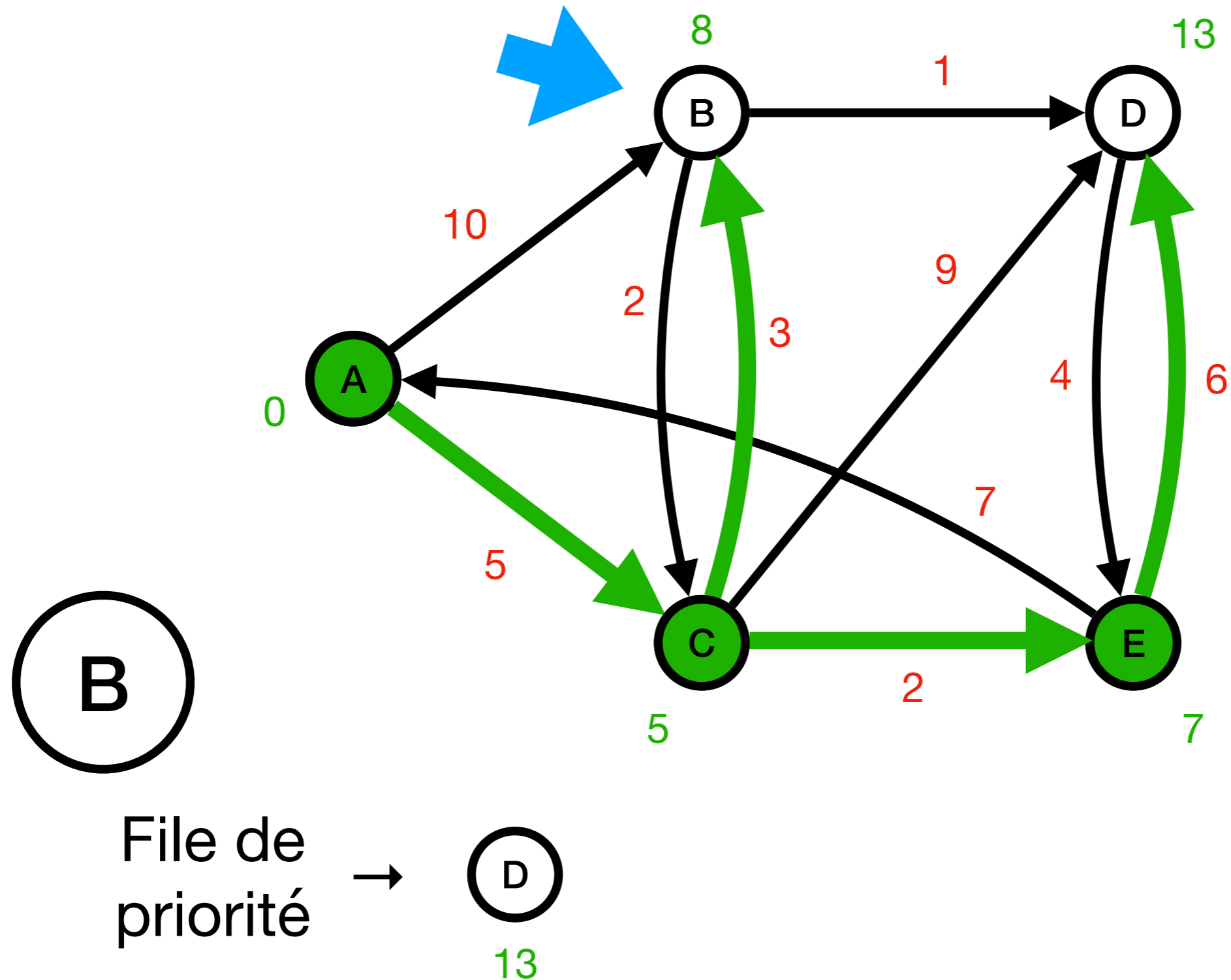
Algorithme de Dijkstra



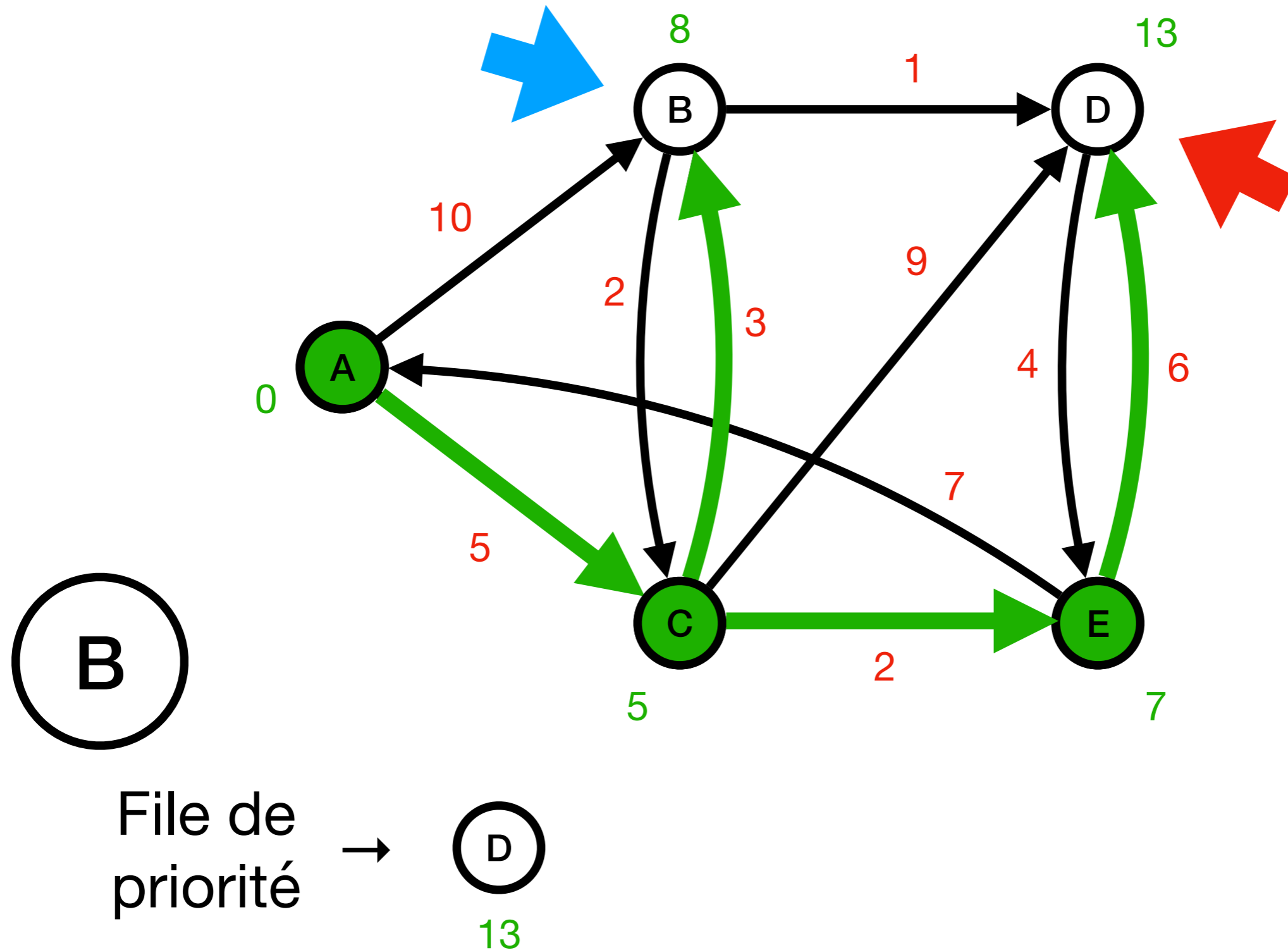
File de
priorité →



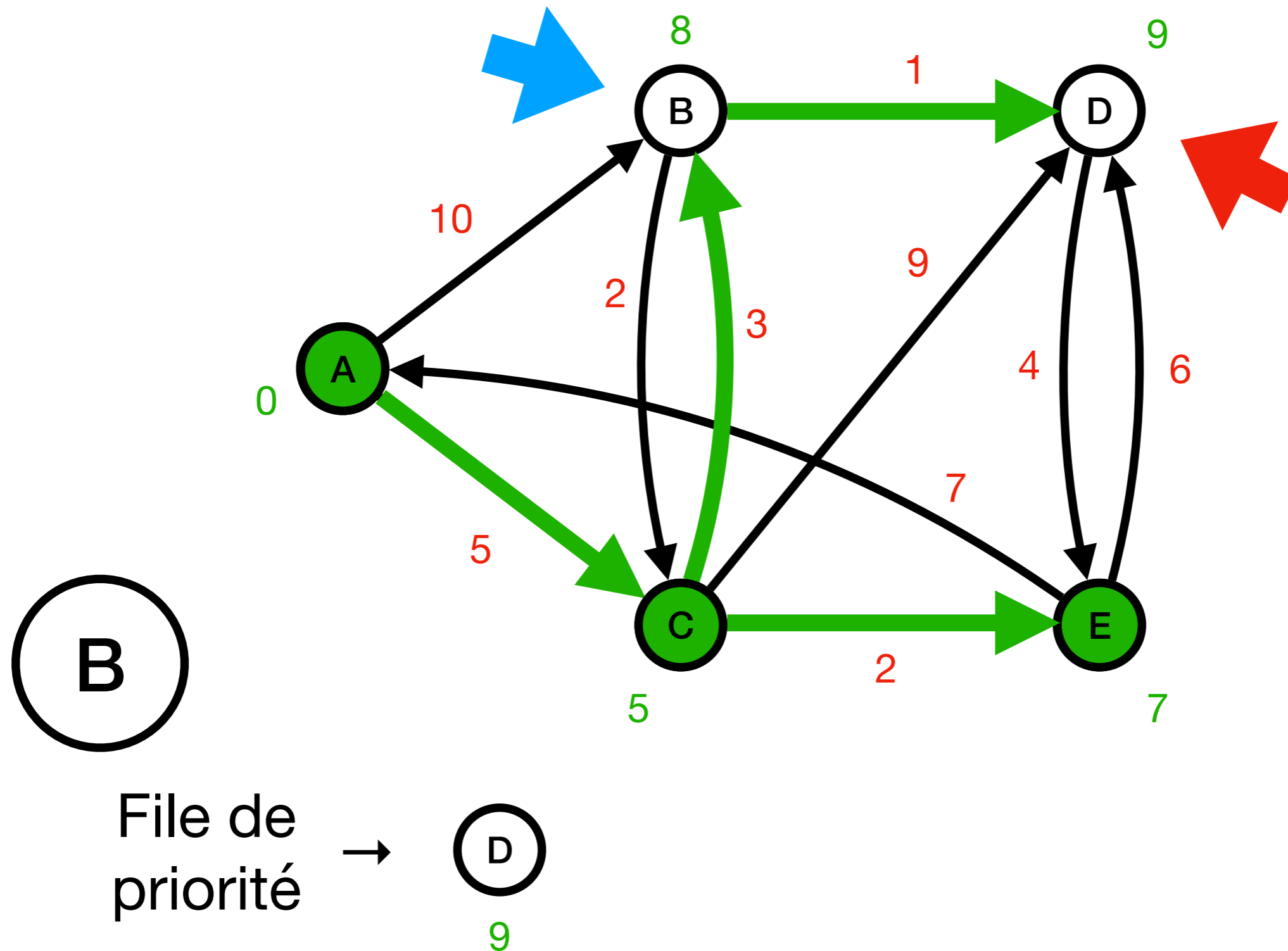
Algorithme de Dijkstra



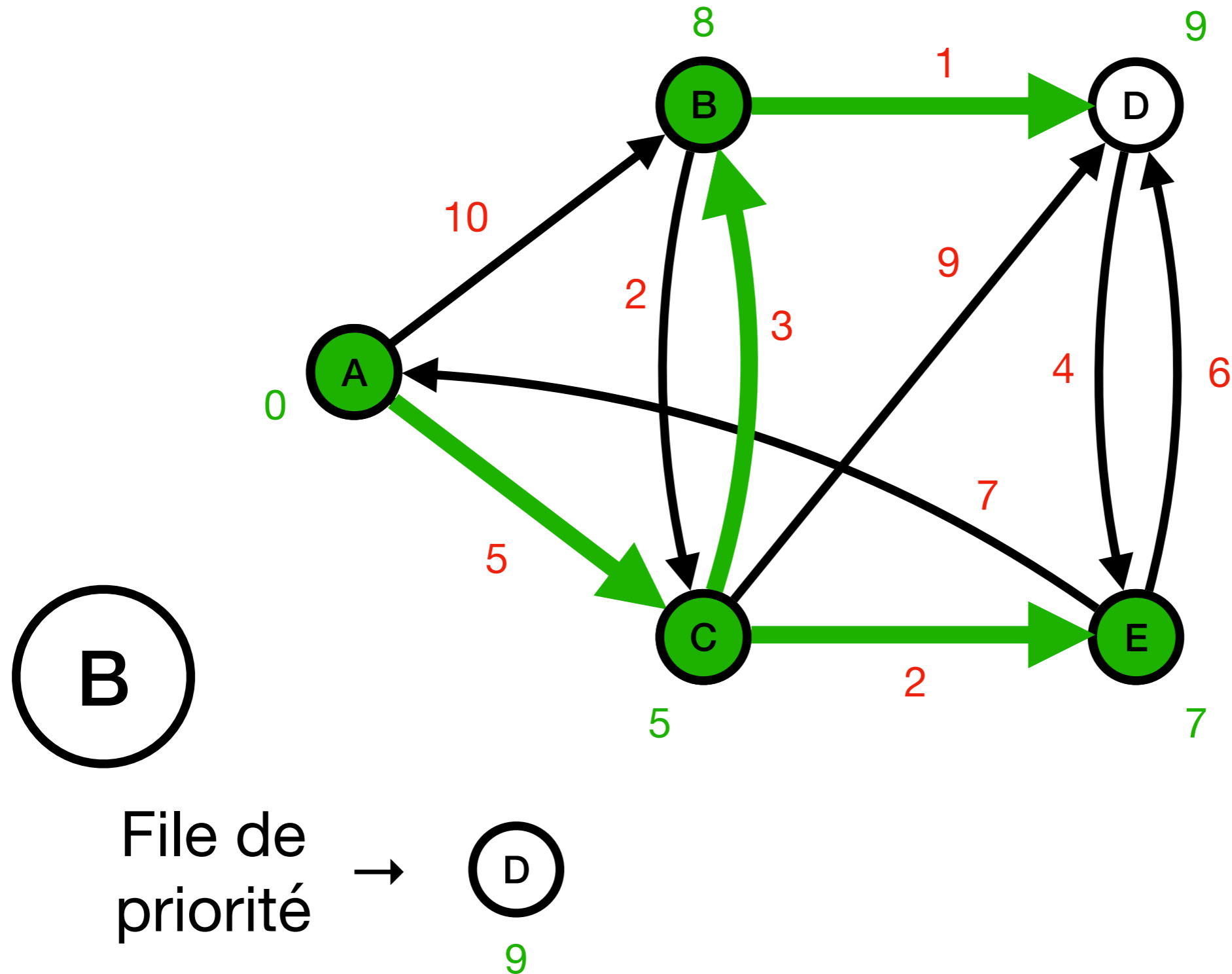
Algorithme de Dijkstra



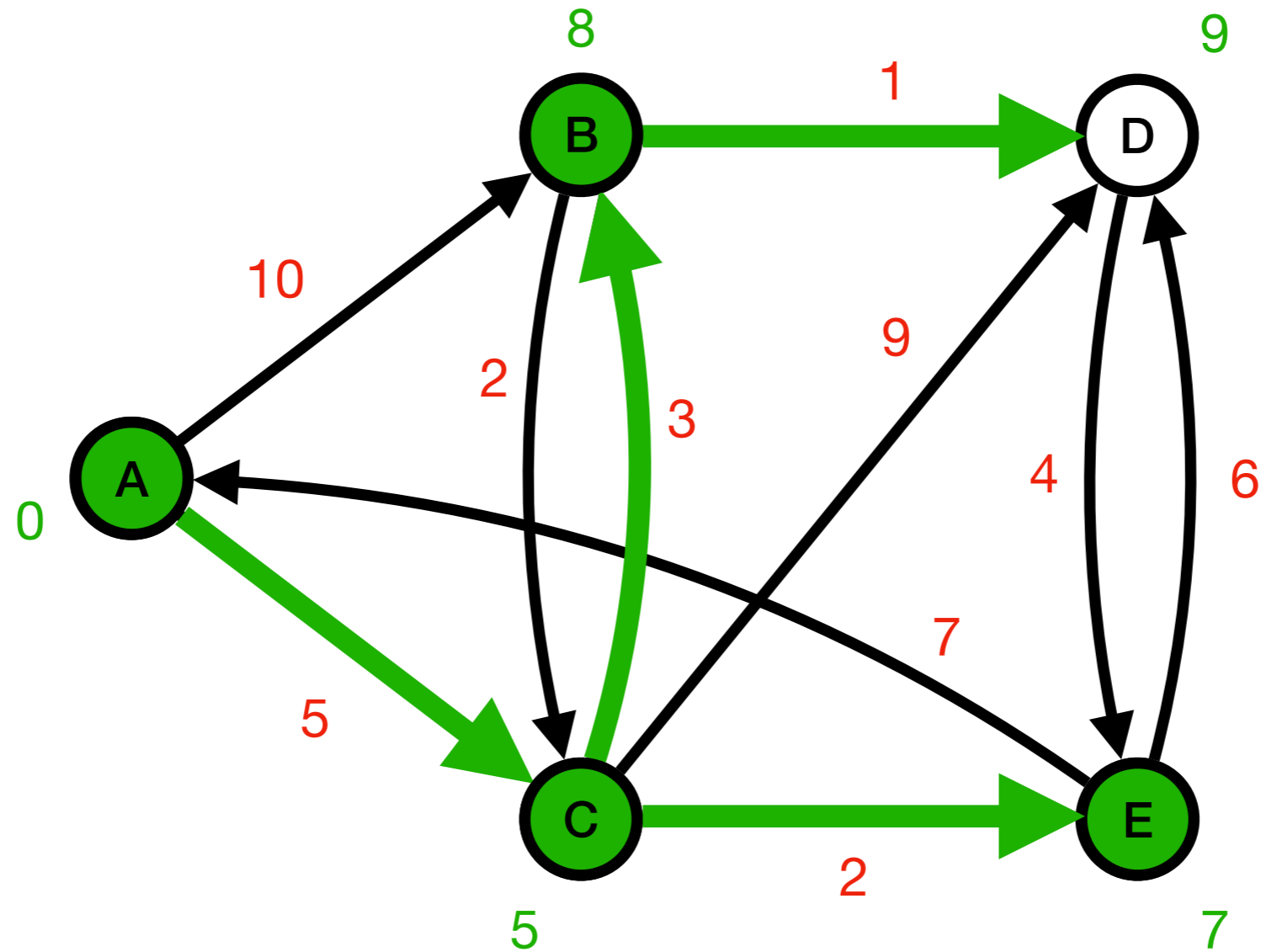
Algorithme de Dijkstra



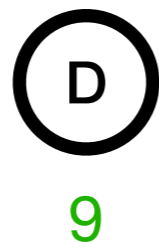
Algorithme de Dijkstra



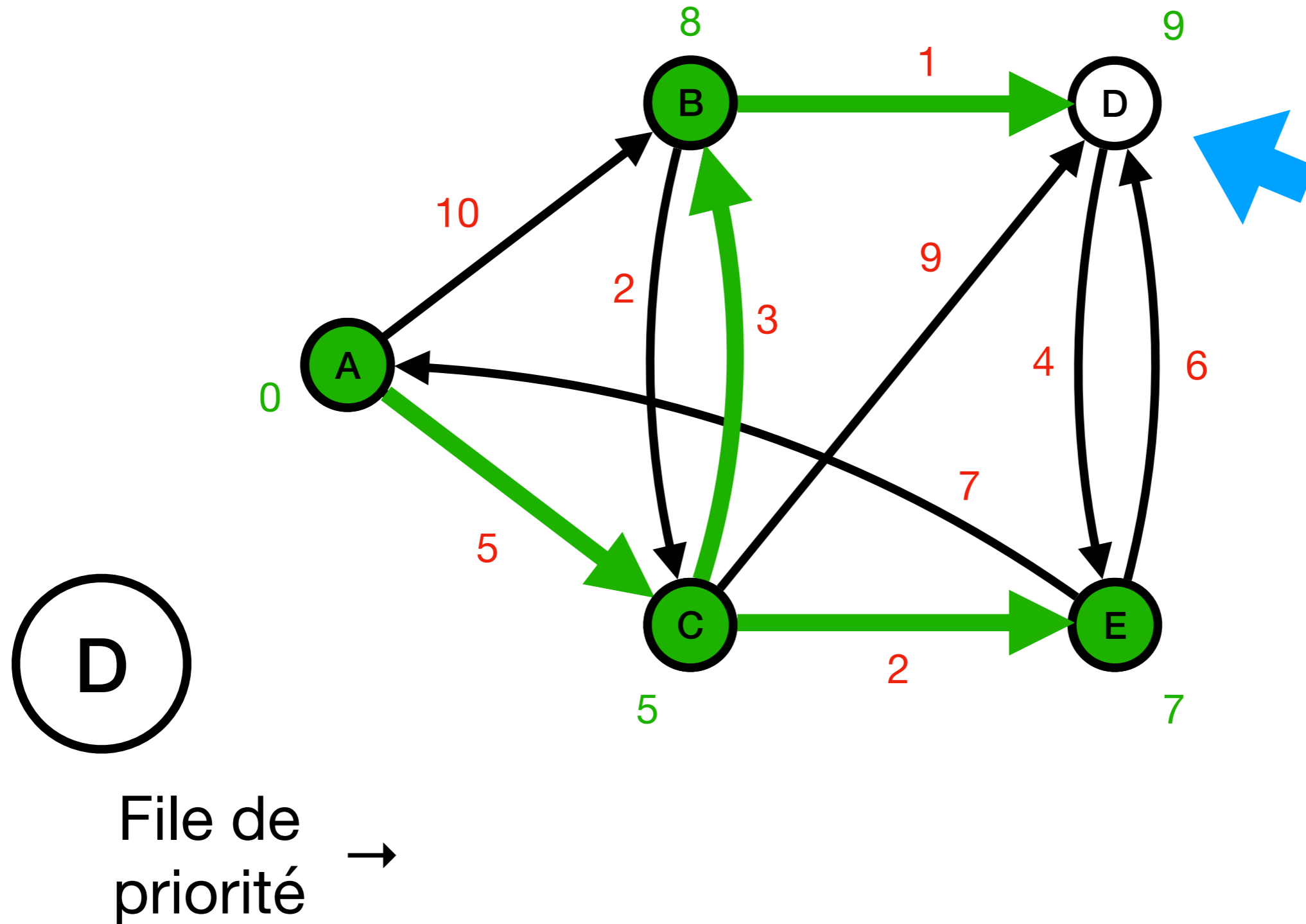
Algorithme de Dijkstra



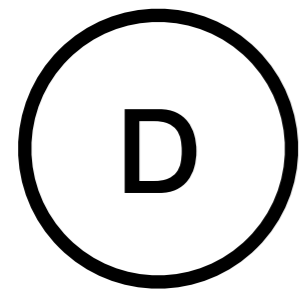
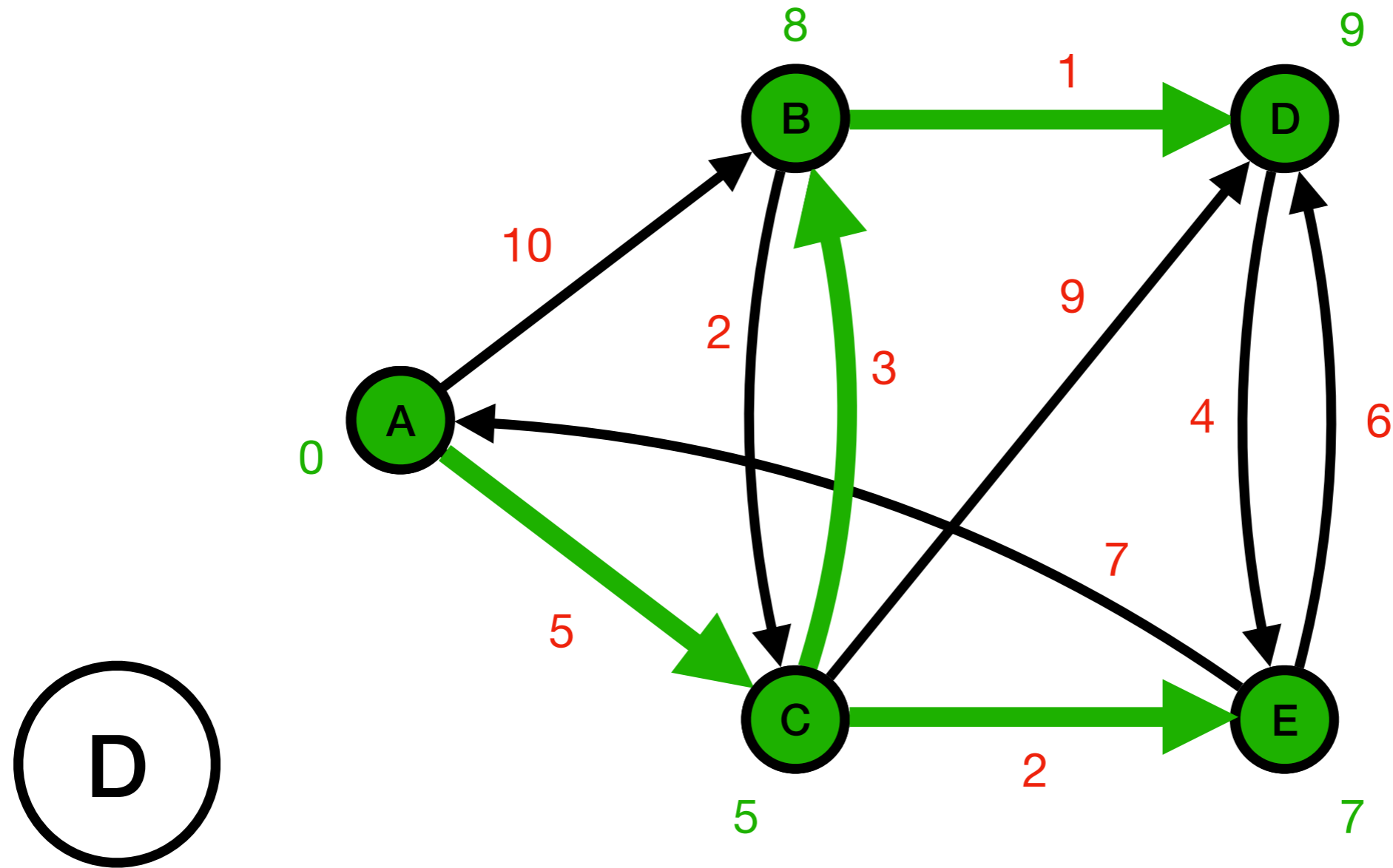
File de priorité →



Algorithme de Dijkstra

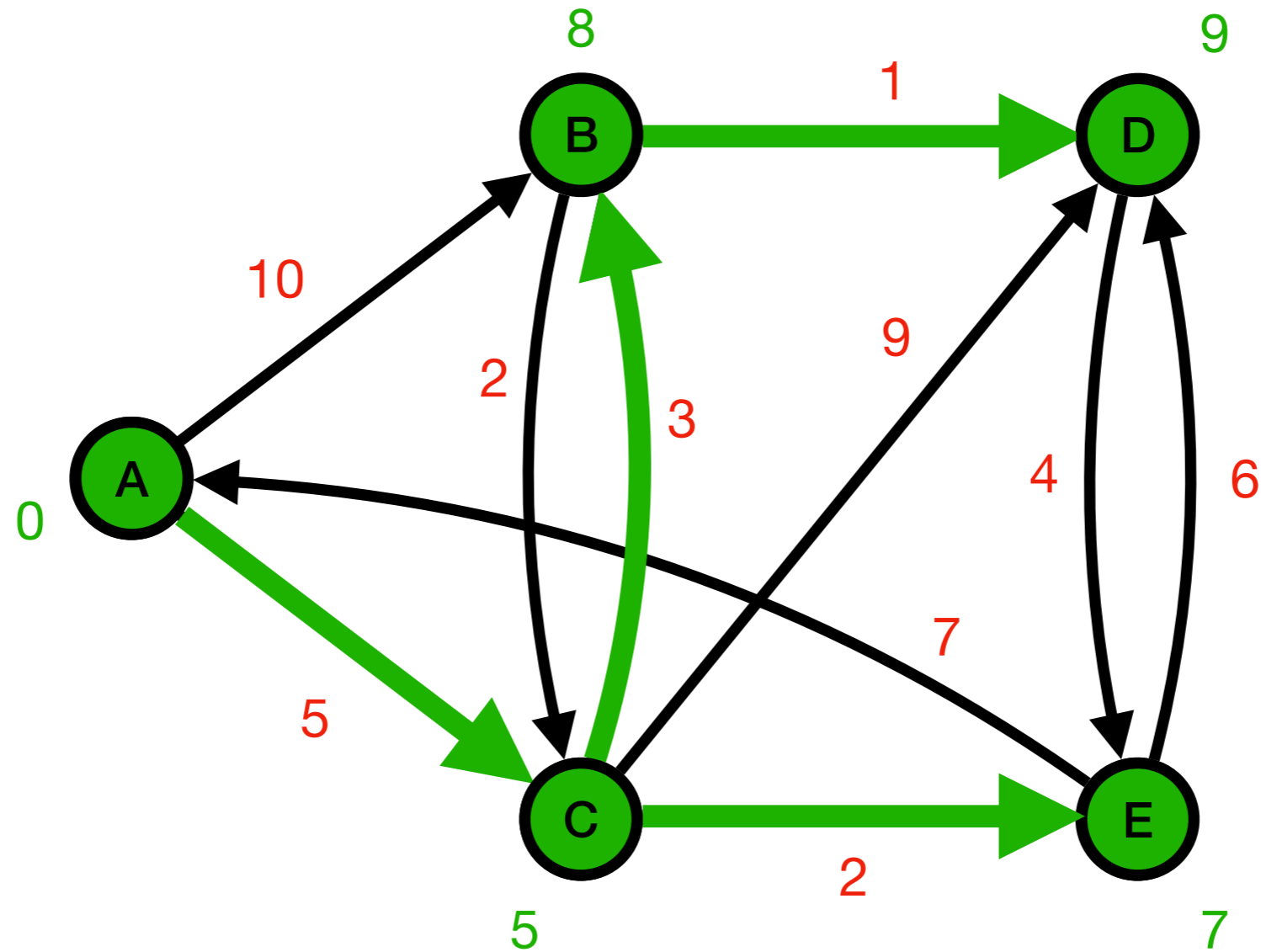


Algorithme de Dijkstra



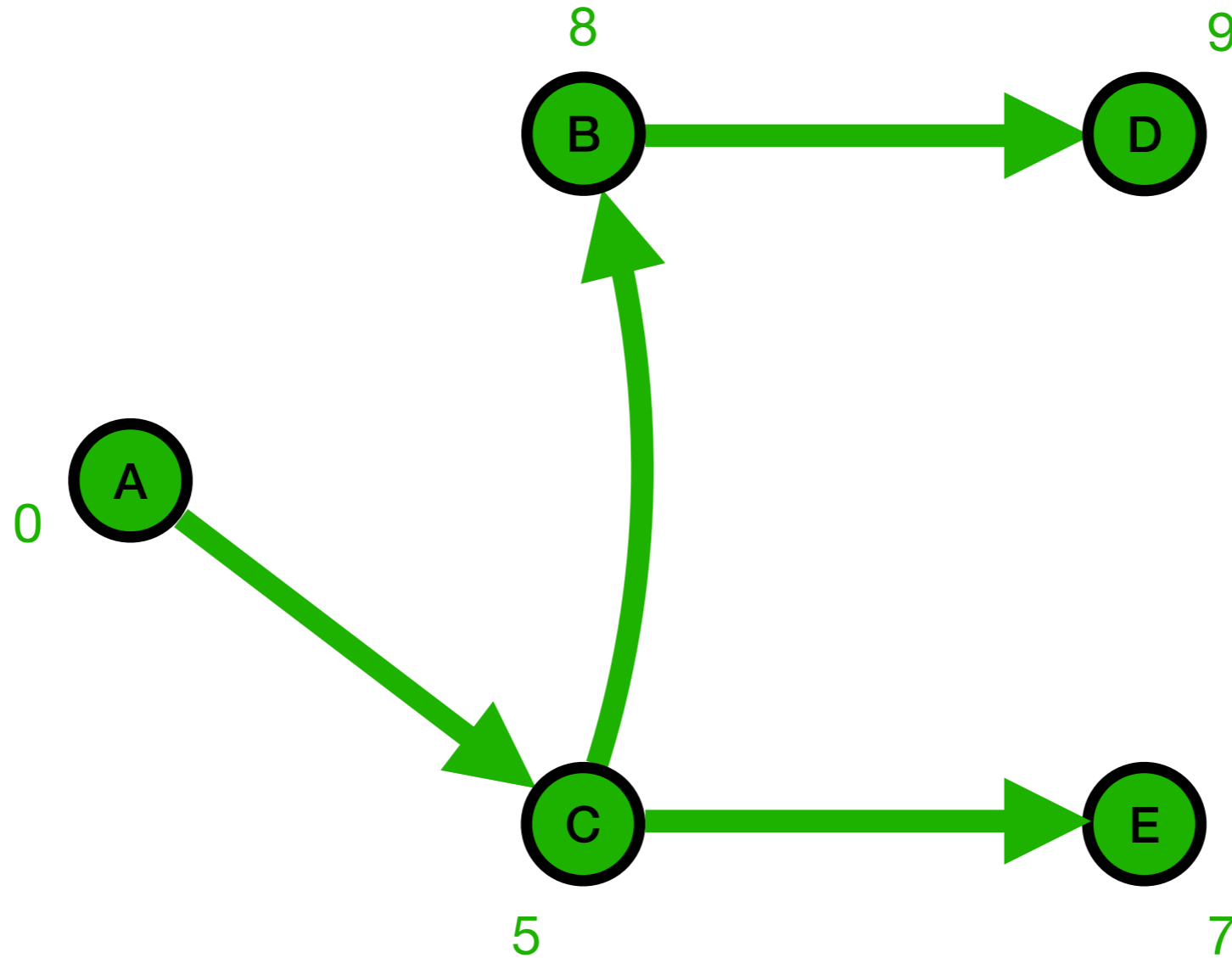
File de priorité →

Algorithme de Dijkstra



File de priorité →

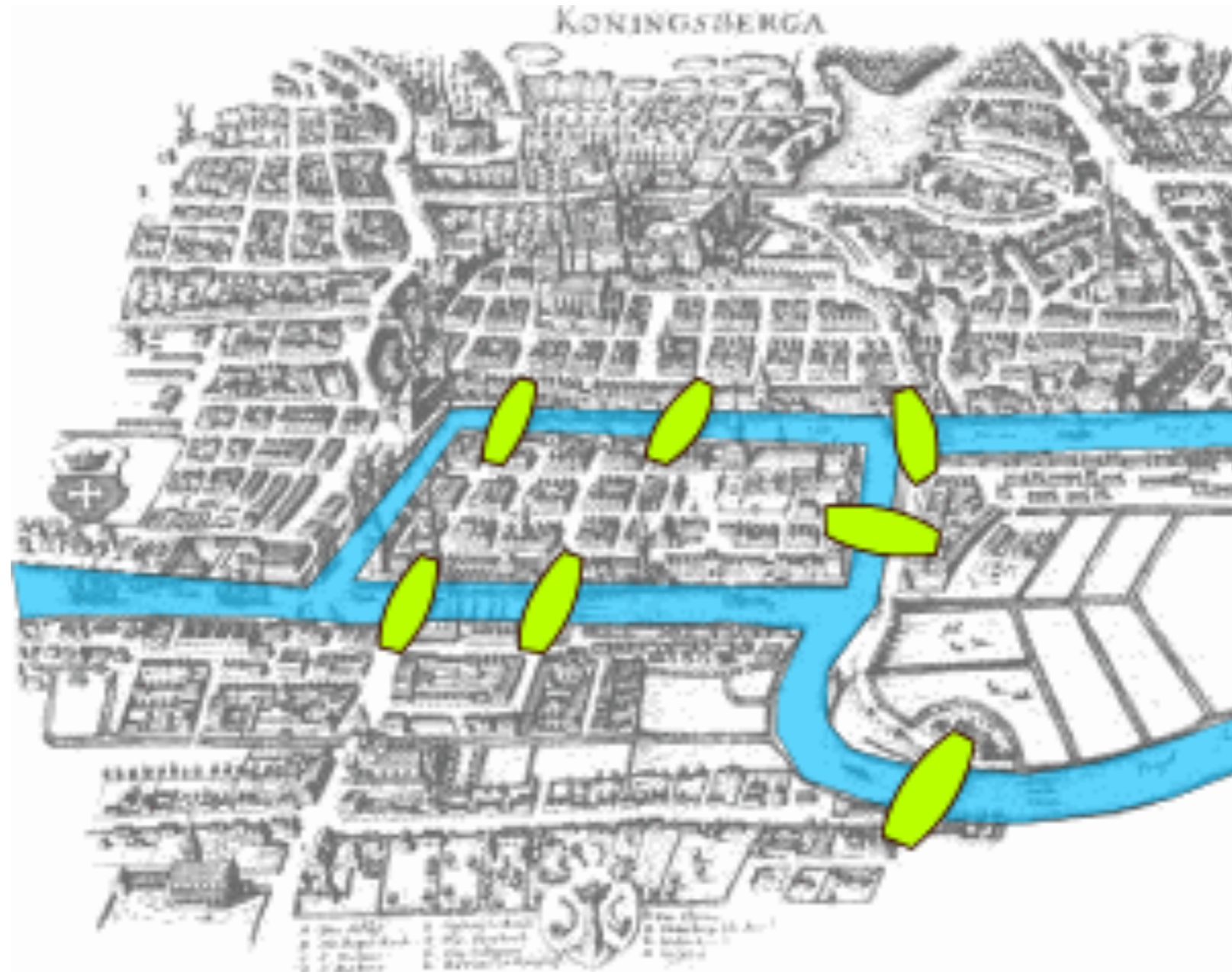
Algorithme de Dijkstra



Exercice 1 du TD4

Graphes non orientés eulériens

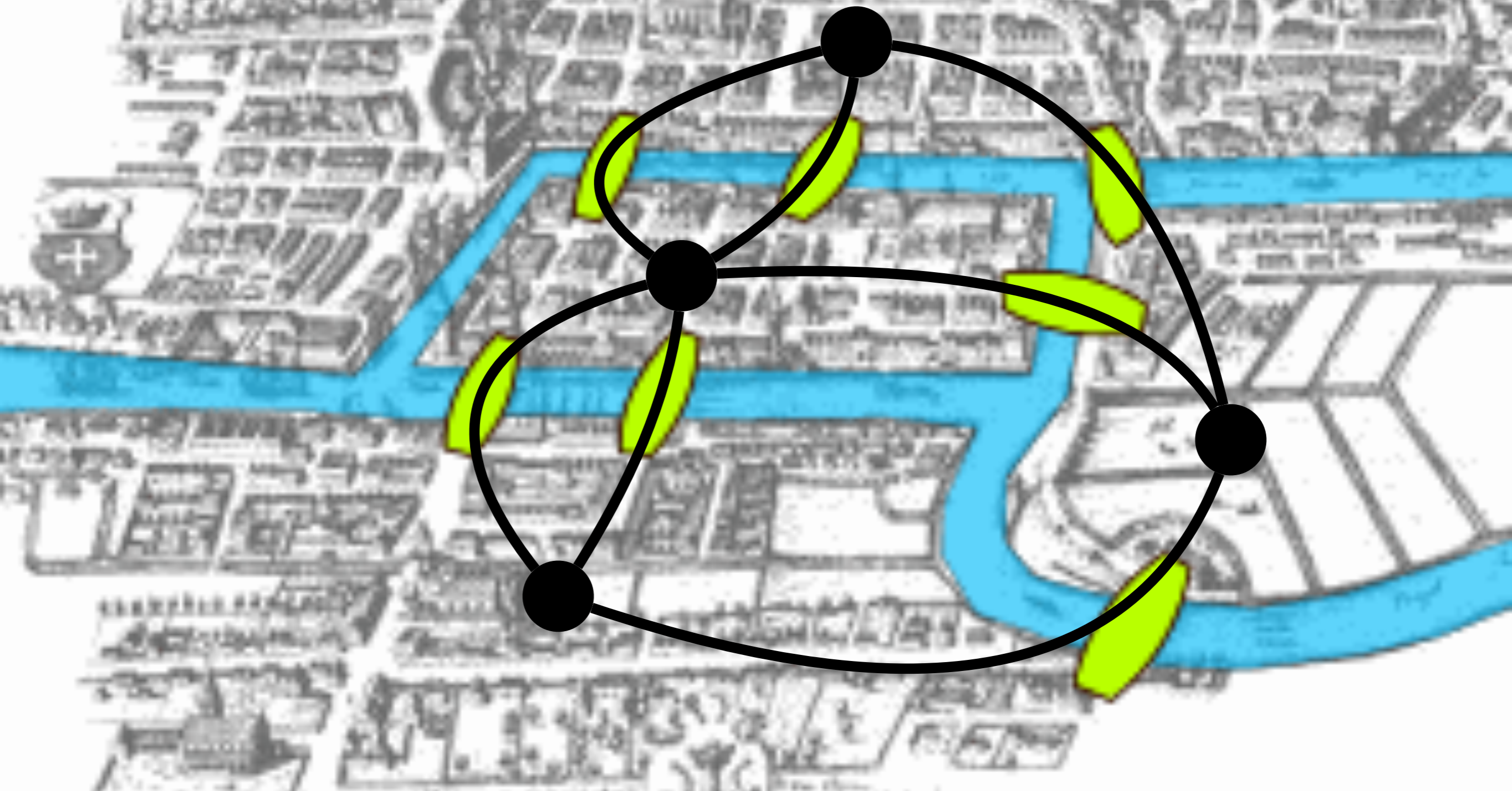
Problème des sept ponts de Königsberg



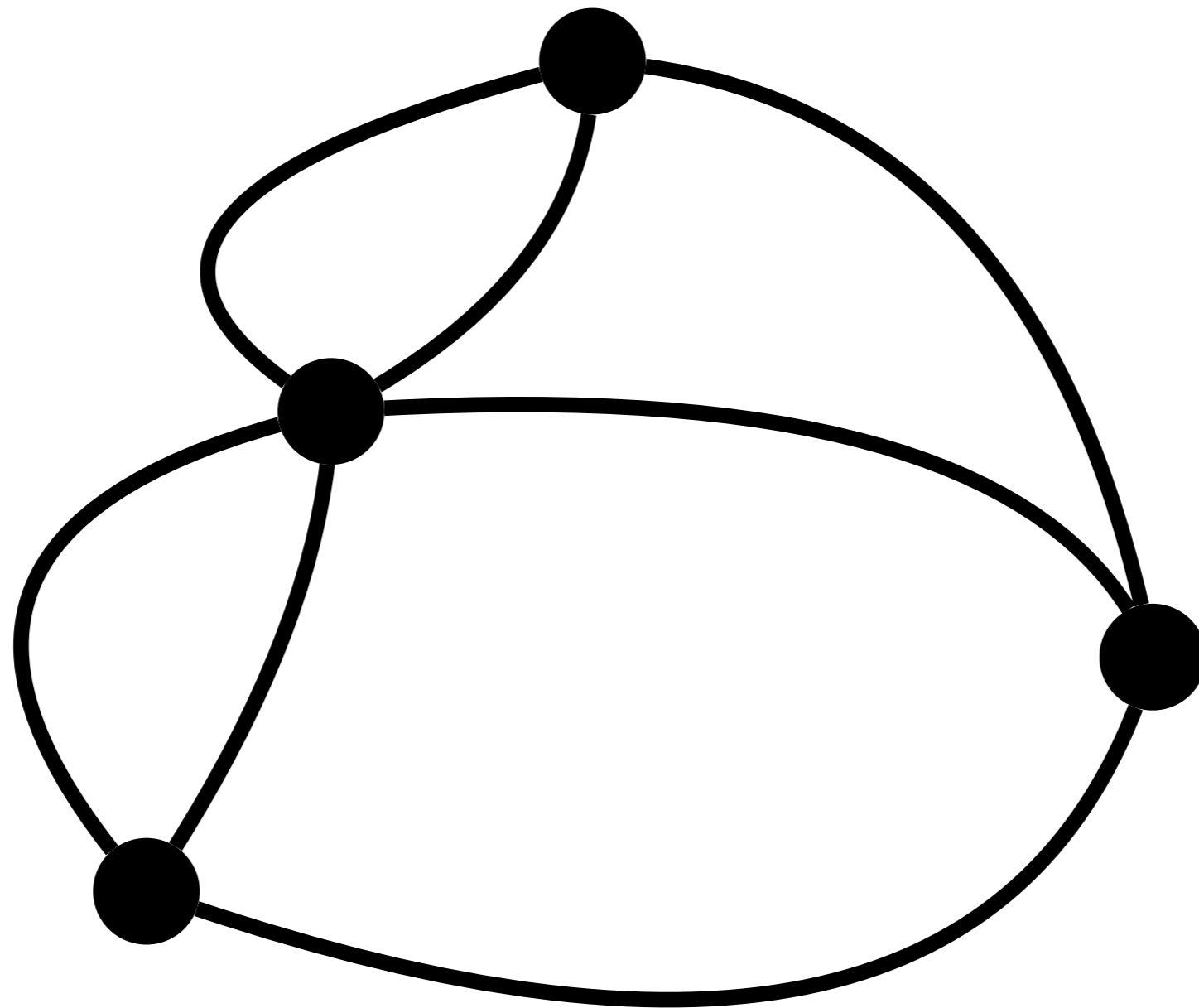
Problème des sept ponts de Königsberg



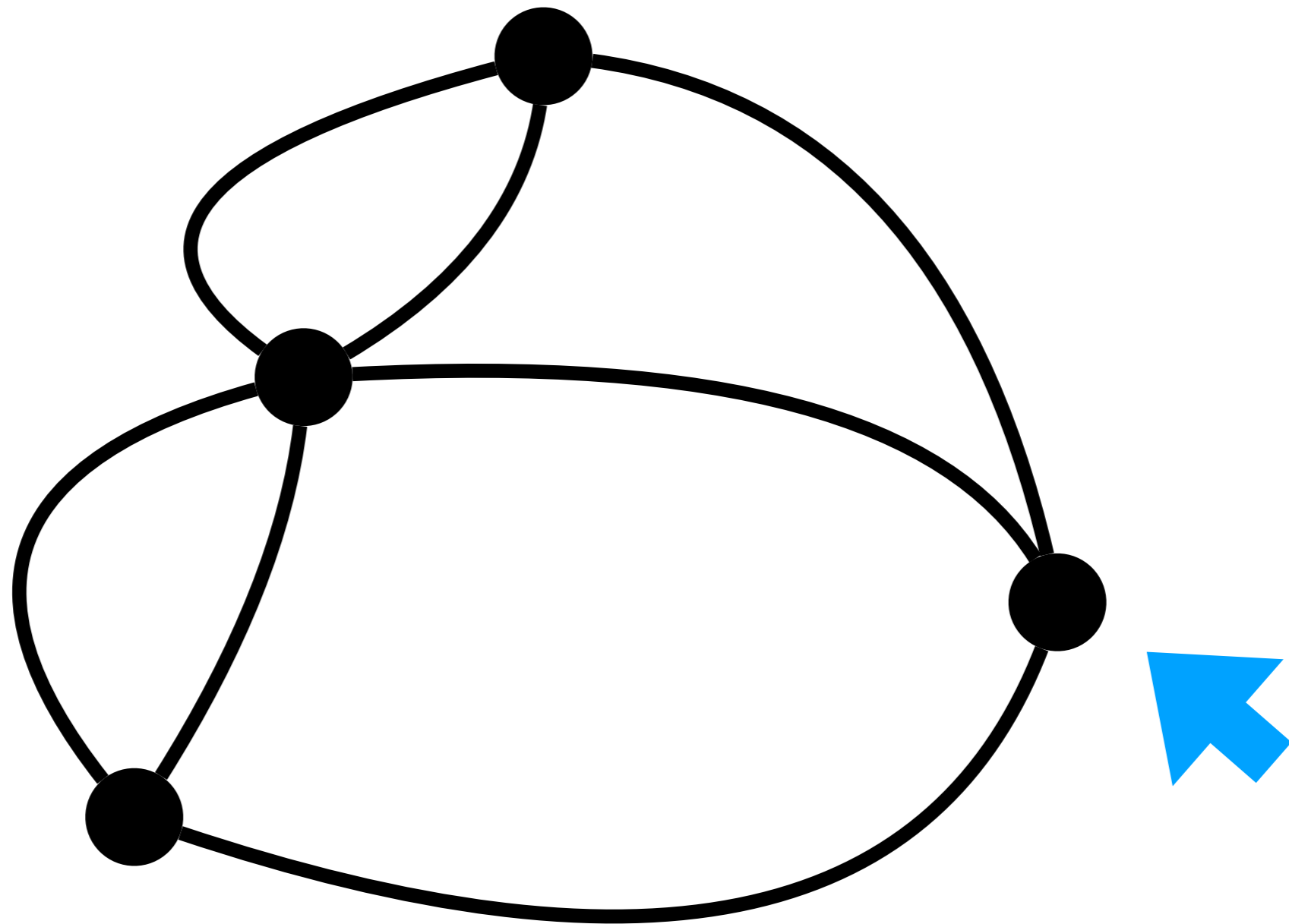
Problème des sept ponts de Königsberg



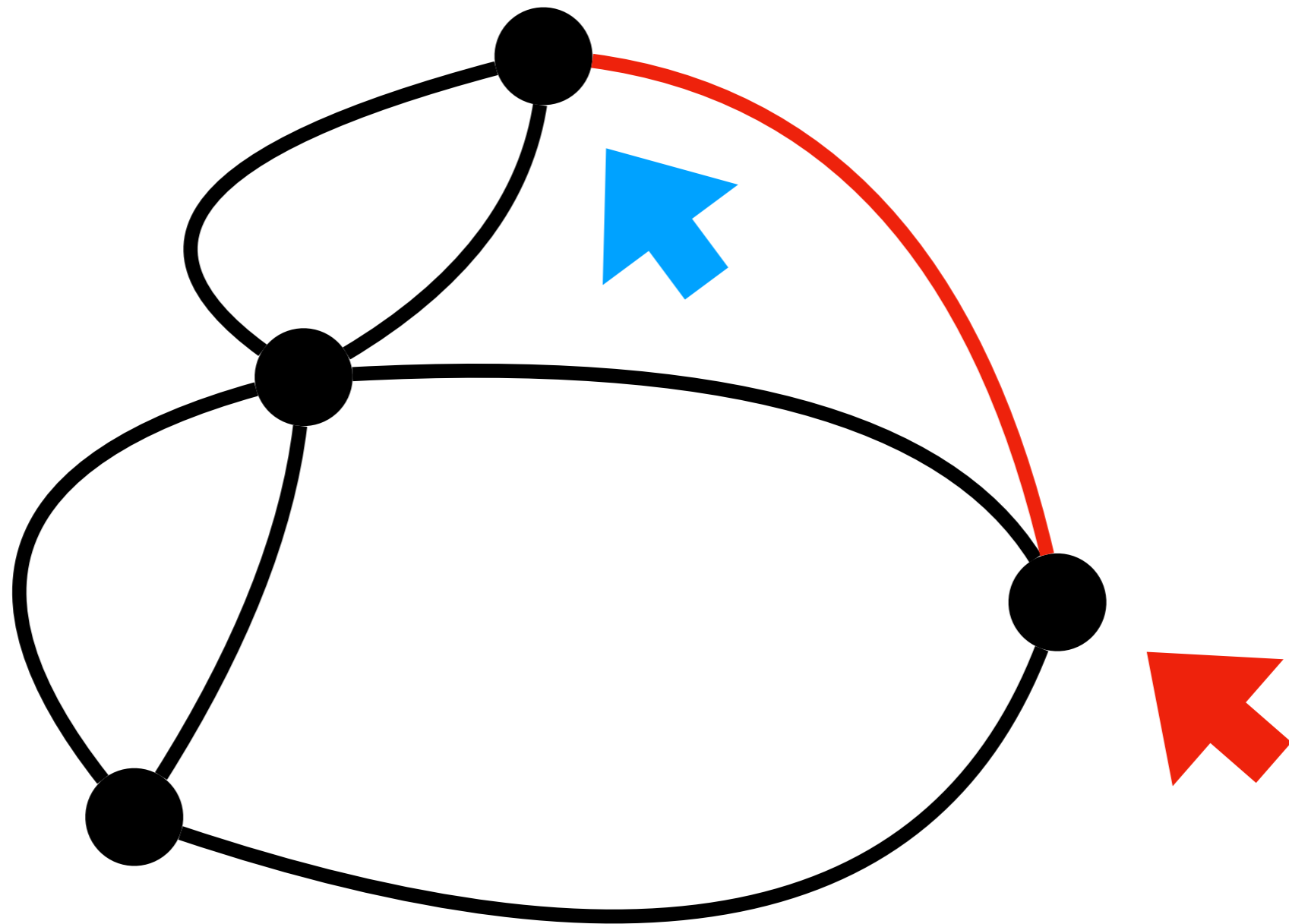
Problème des sept ponts de Königsberg



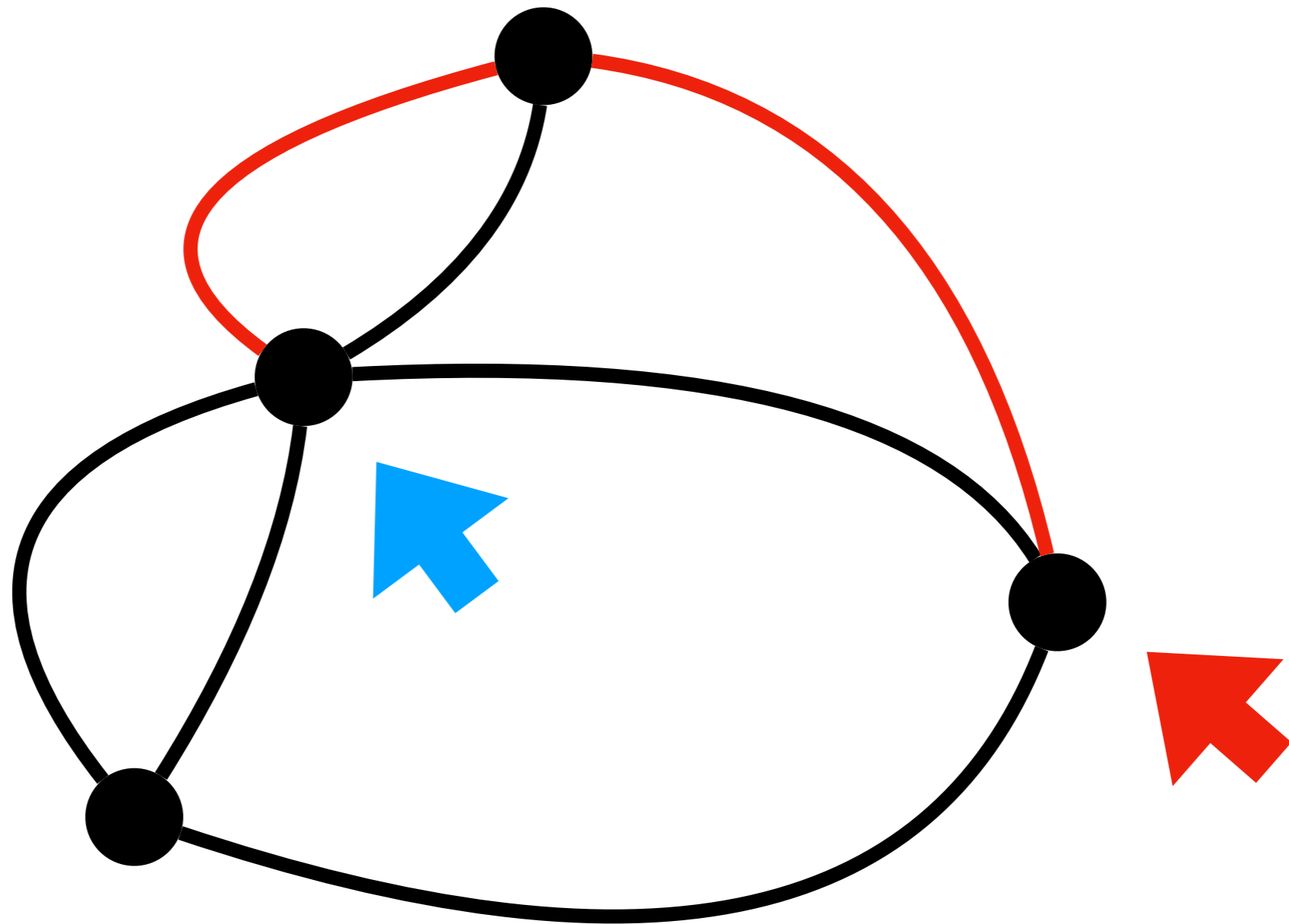
Problème des sept ponts de Königsberg



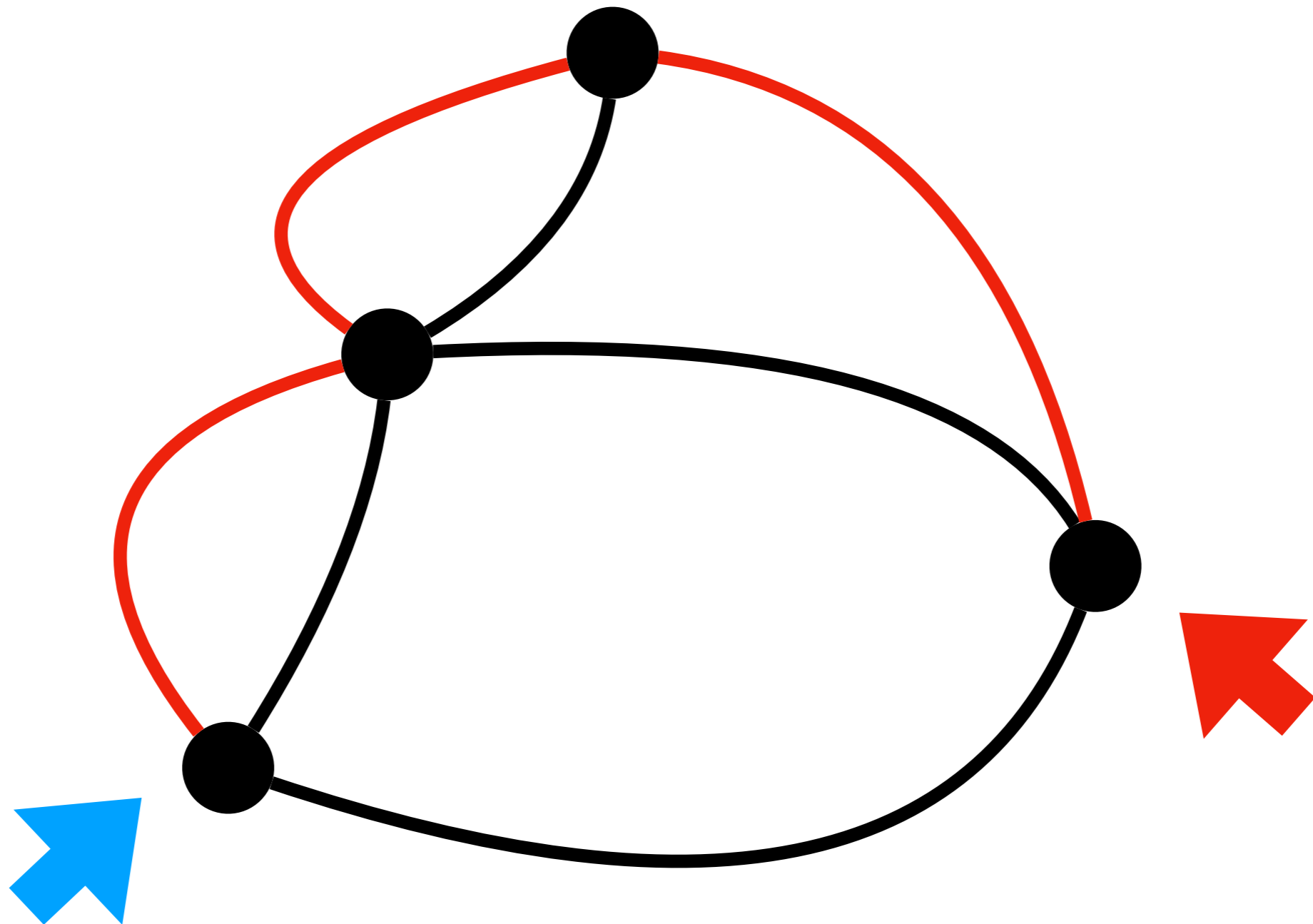
Problème des sept ponts de Königsberg



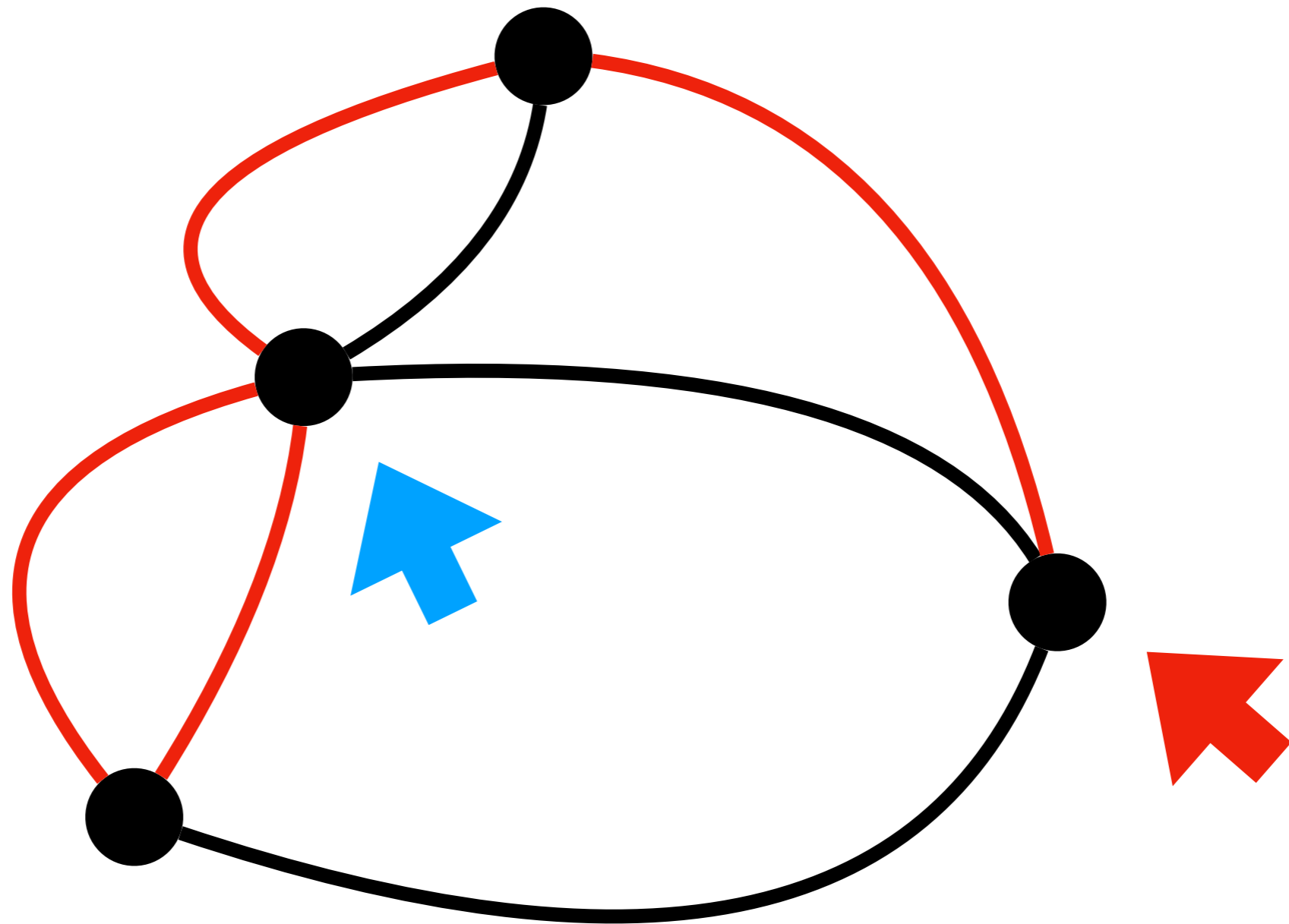
Problème des sept ponts de Königsberg



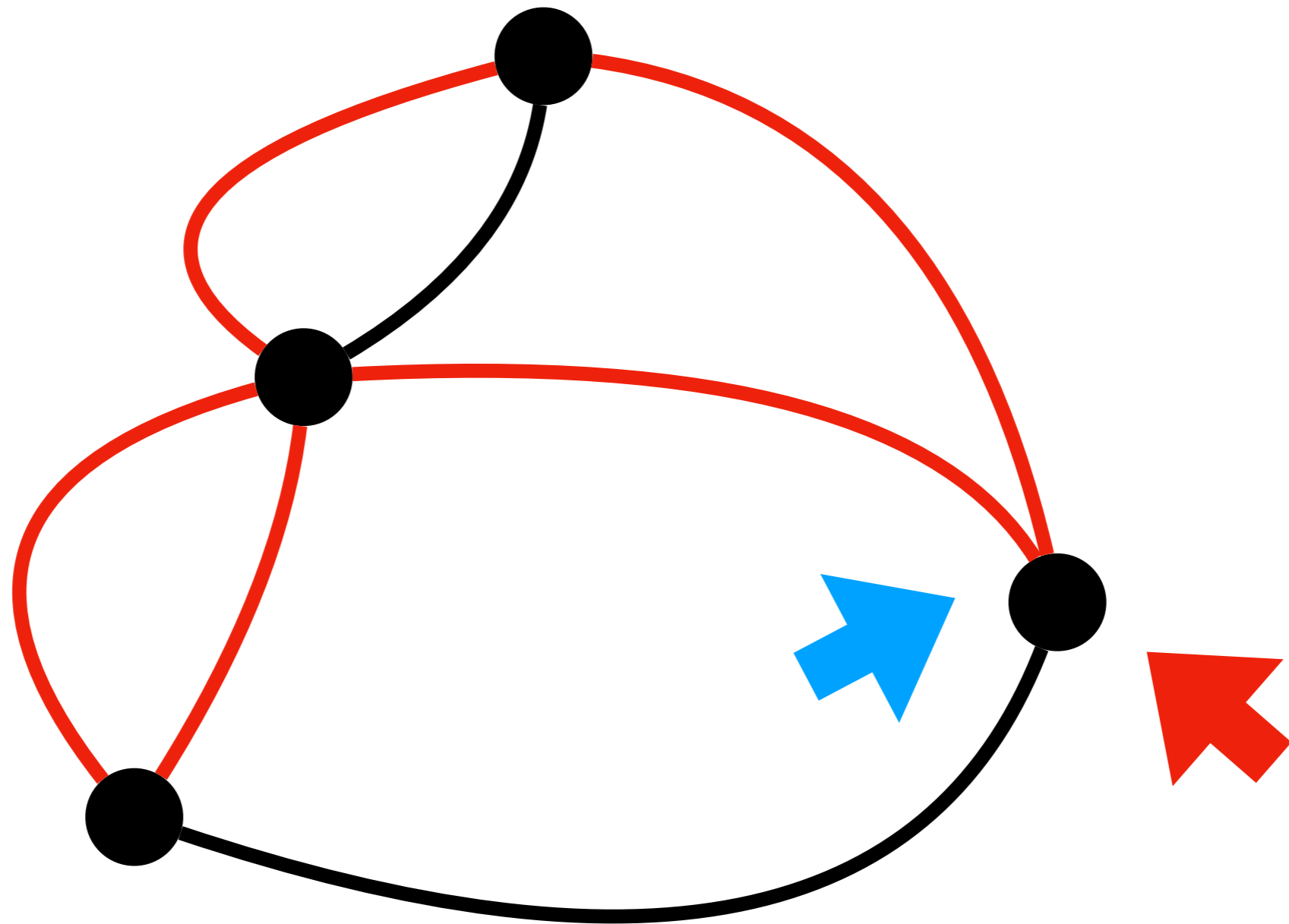
Problème des sept ponts de Königsberg



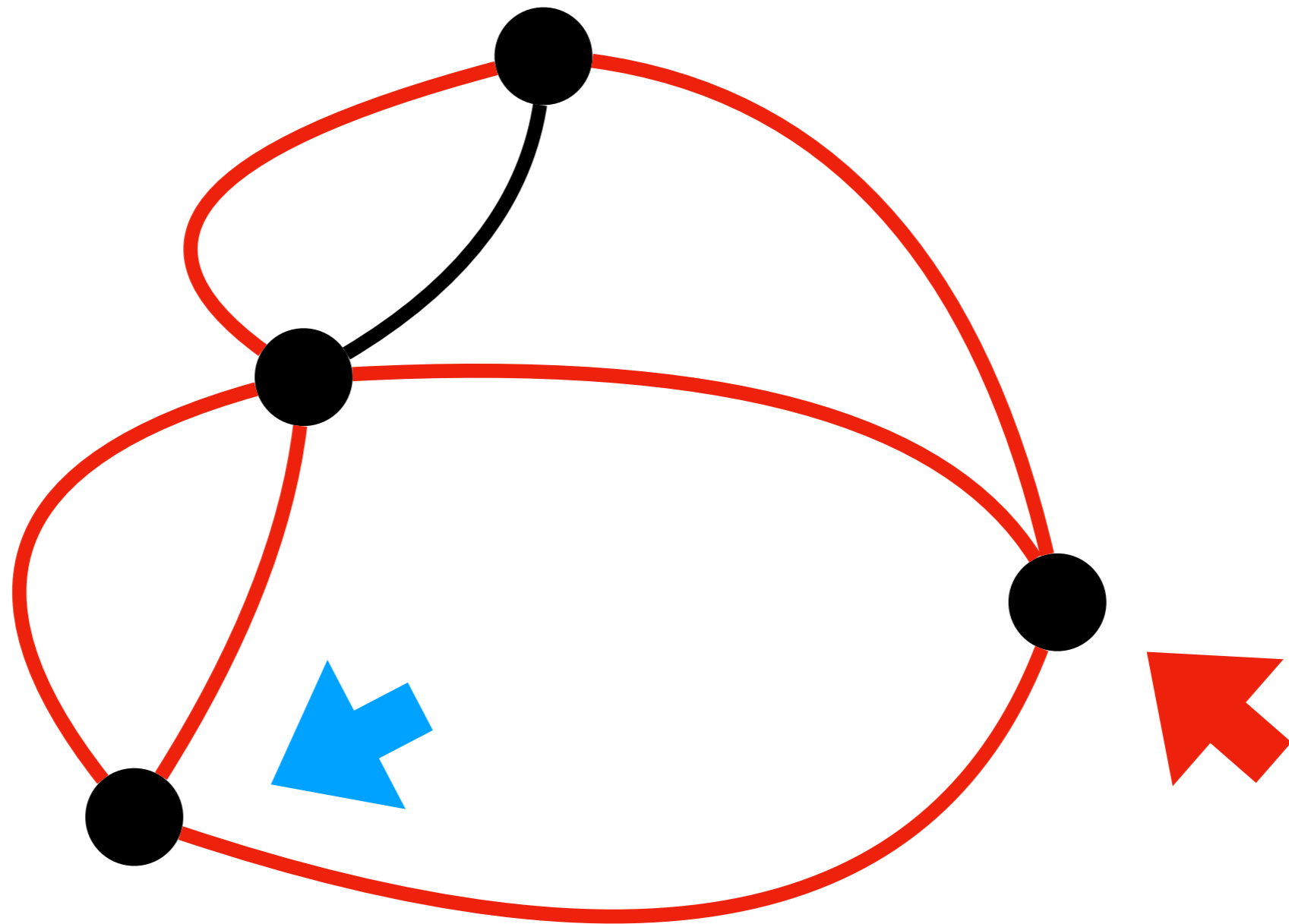
Problème des sept ponts de Königsberg



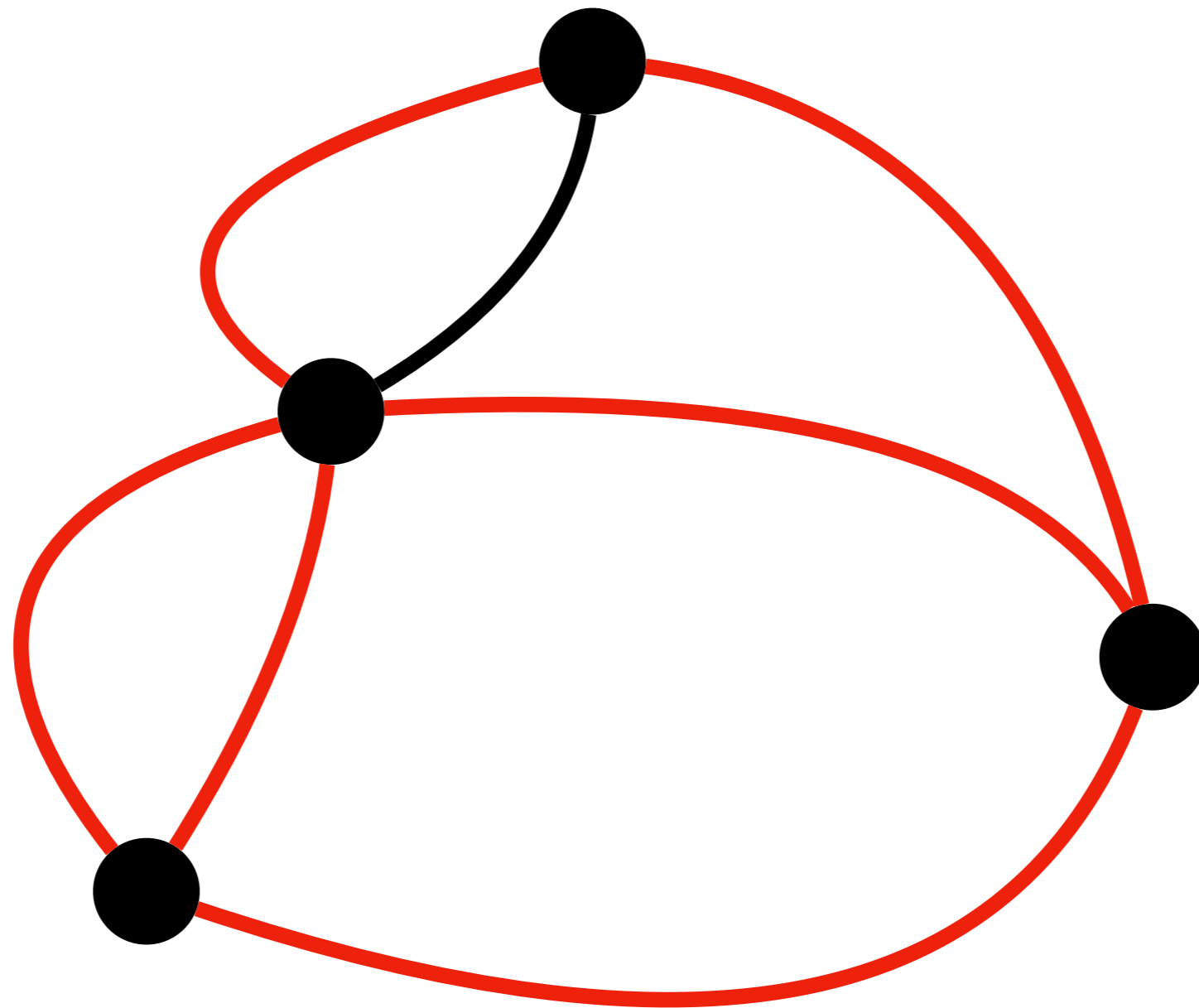
Problème des sept ponts de Königsberg



Problème des sept ponts de Königsberg



Problème des sept ponts de Königsberg



Problème des sept ponts de Königsberg

Il n'y a pas de
solution !



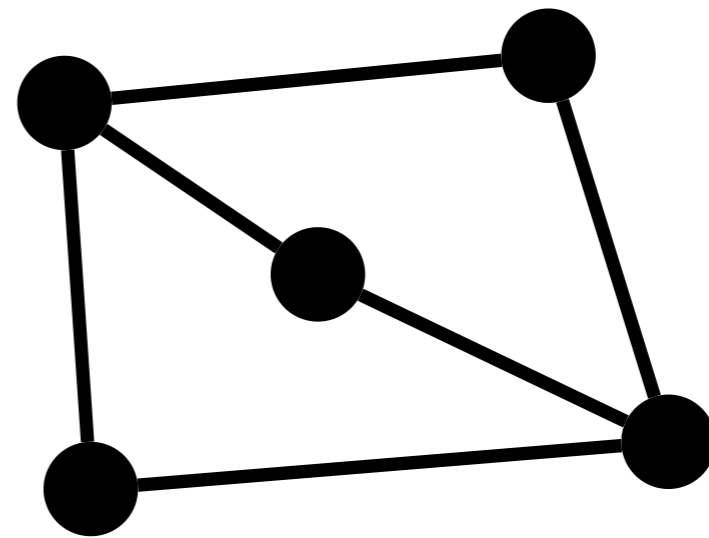
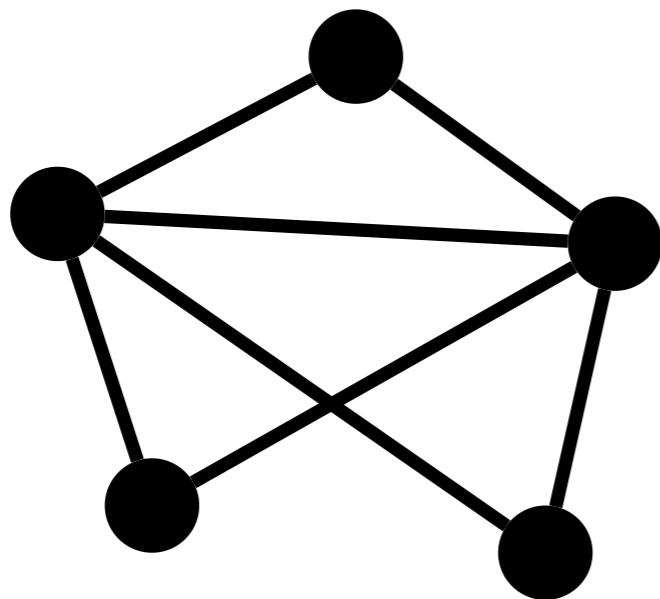
👉 Leonhard Euler

Théorème d'Euler (1736)

Un graphe non orienté connexe admet un cycle qui traverse chaque arête une et une seule fois (un « cycle eulérien ») si et seulement si chaque sommet a un nombre pair de voisins connectés par une arête.

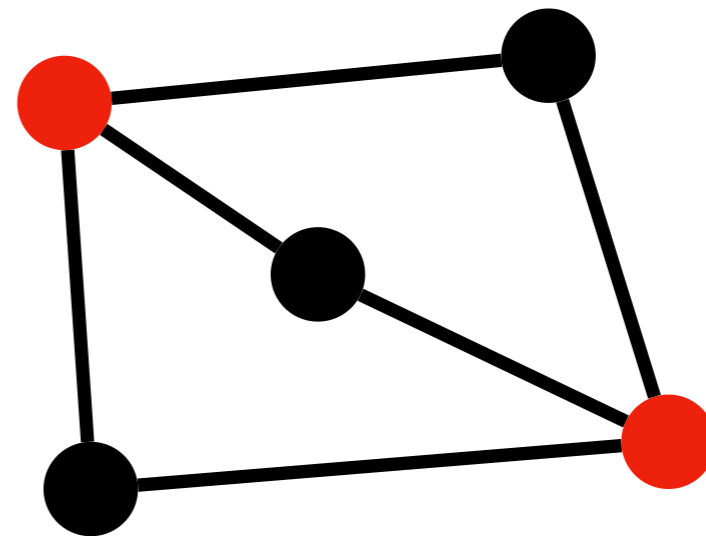
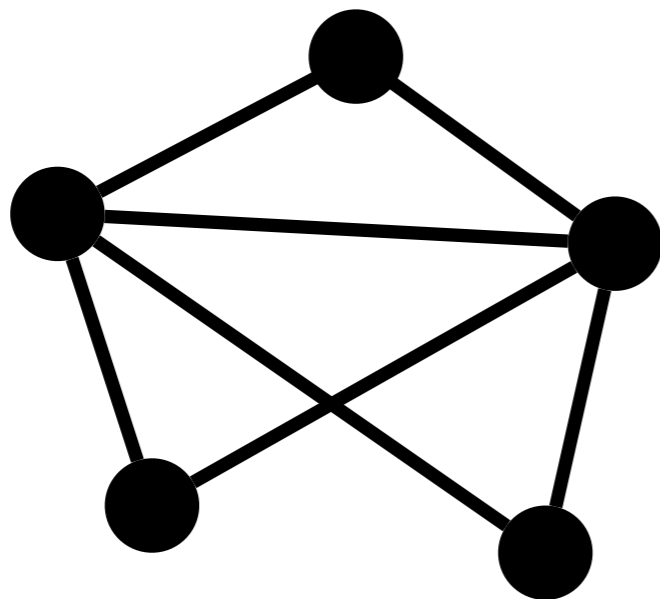
Théorème d'Euler (1736)

Un graphe non orienté connexe admet un cycle qui traverse chaque arête une et une seule fois (un « cycle eulérien ») si et seulement si chaque sommet a un nombre pair de voisins connectés par un arête.

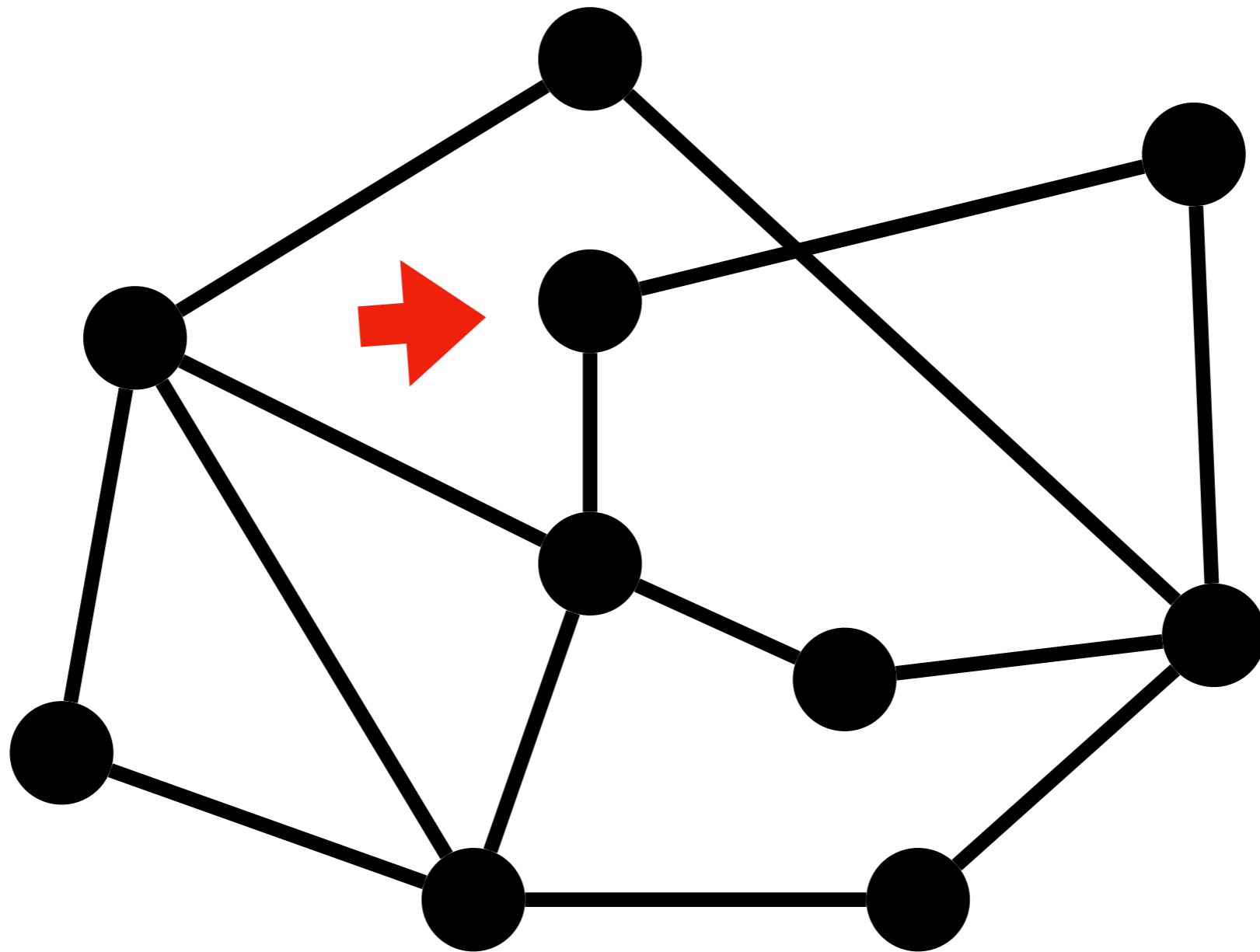


Théorème d'Euler (1736)

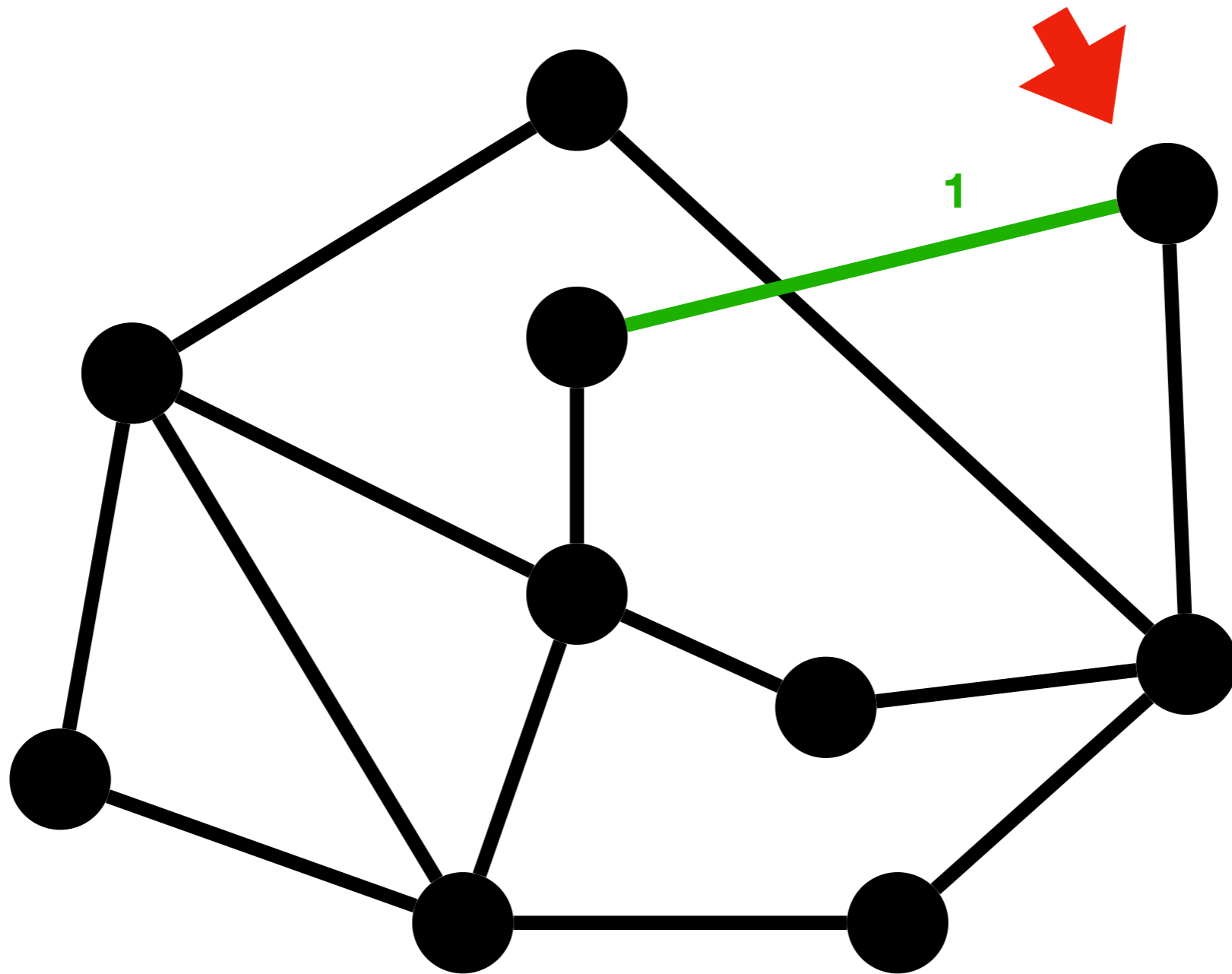
Un graphe non orienté connexe admet un cycle qui traverse chaque arête une et une seule fois (un « cycle eulérien ») si et seulement si chaque sommet a un nombre pair de voisins connectés par un arête.



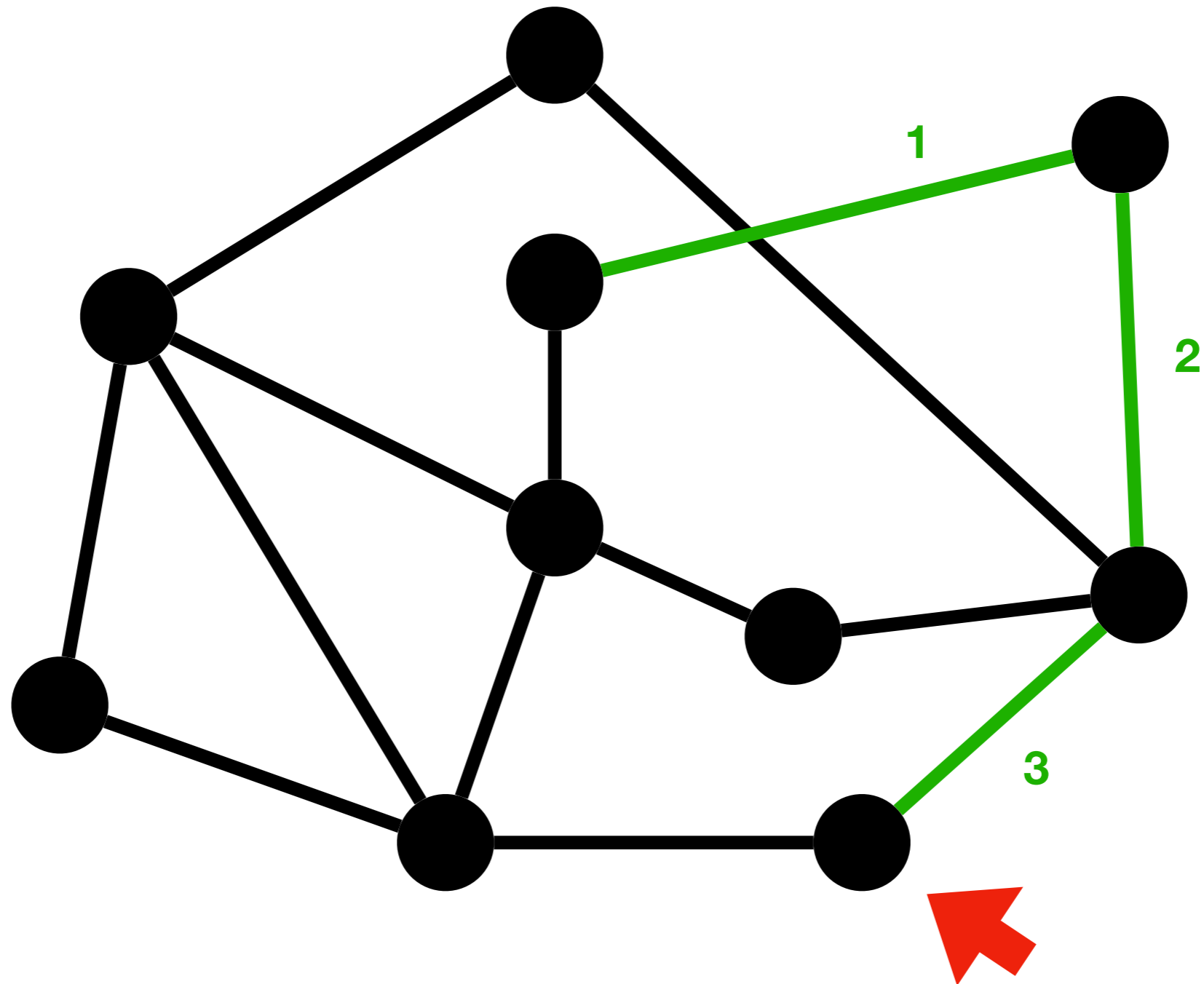
Algorithme de Hierholzer



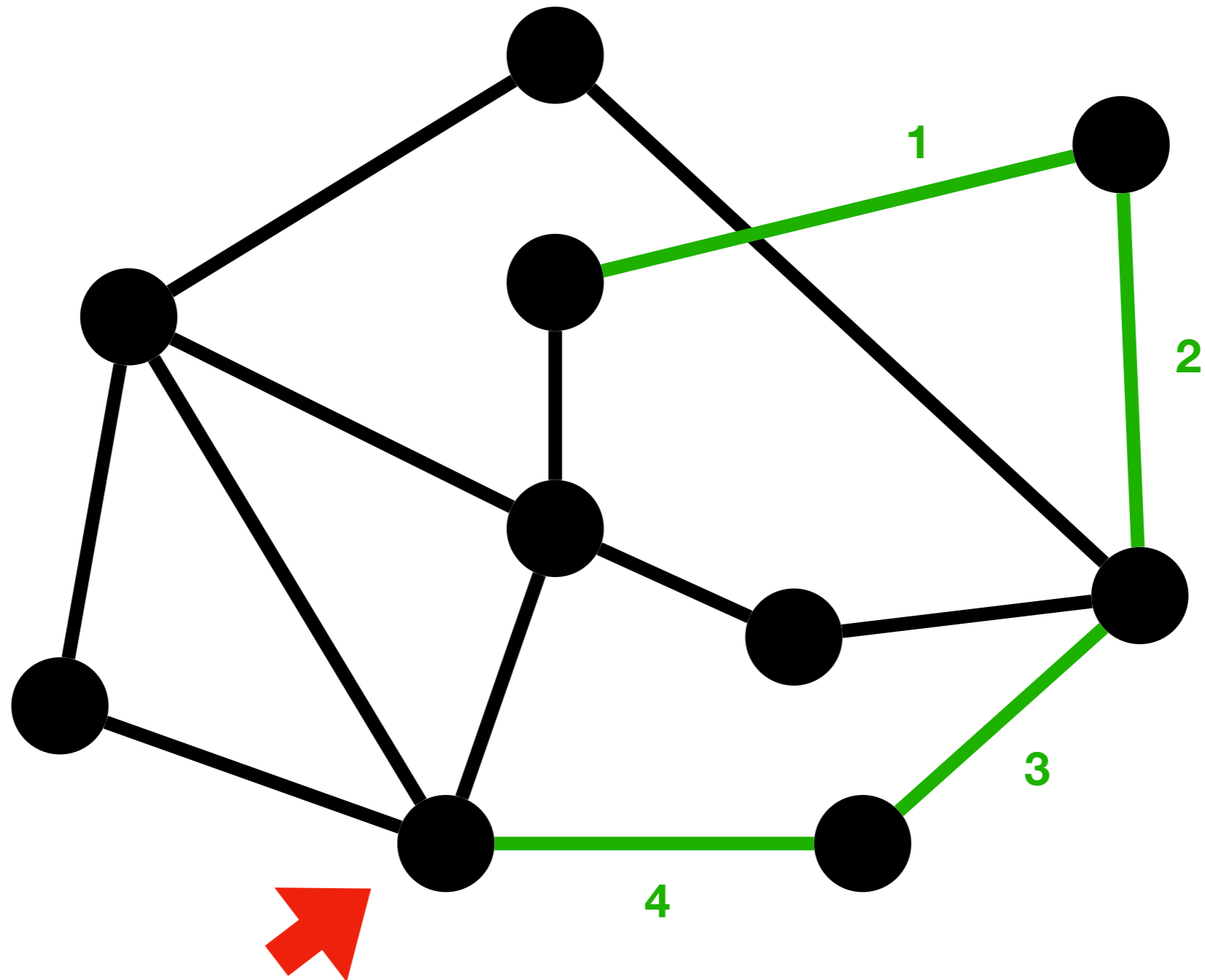
Algorithme de Hierholzer



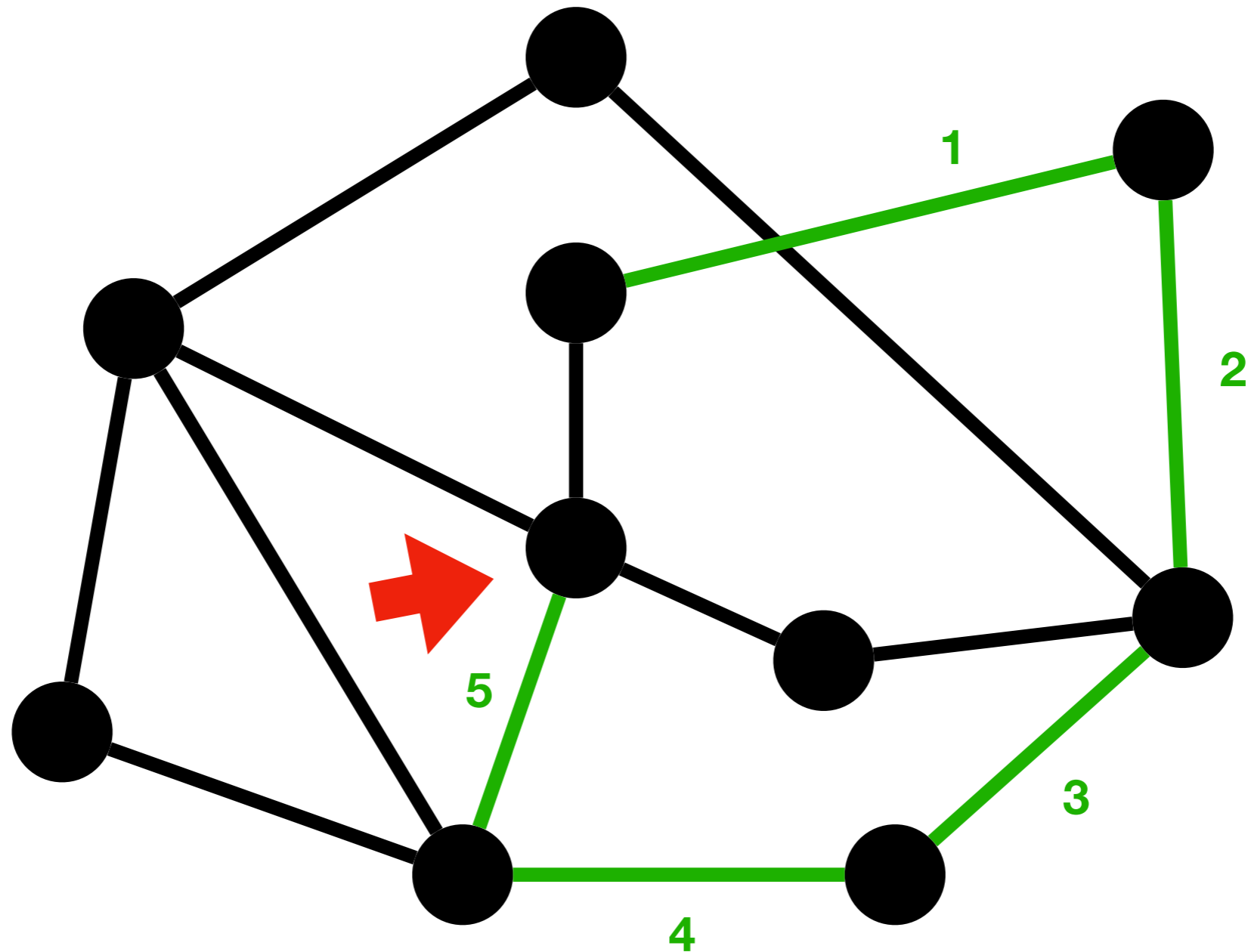
Algorithme de Hierholzer



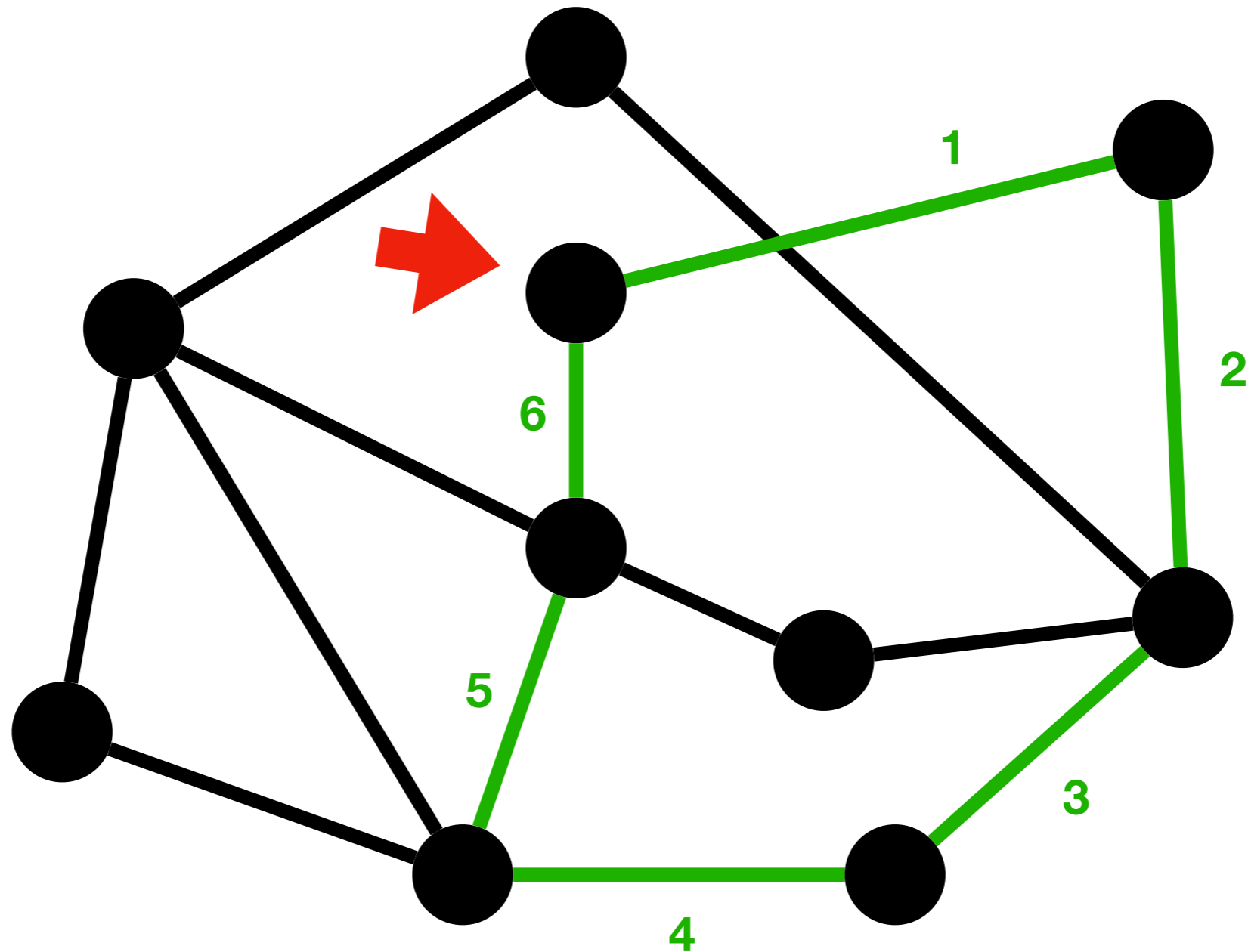
Algorithme de Hierholzer



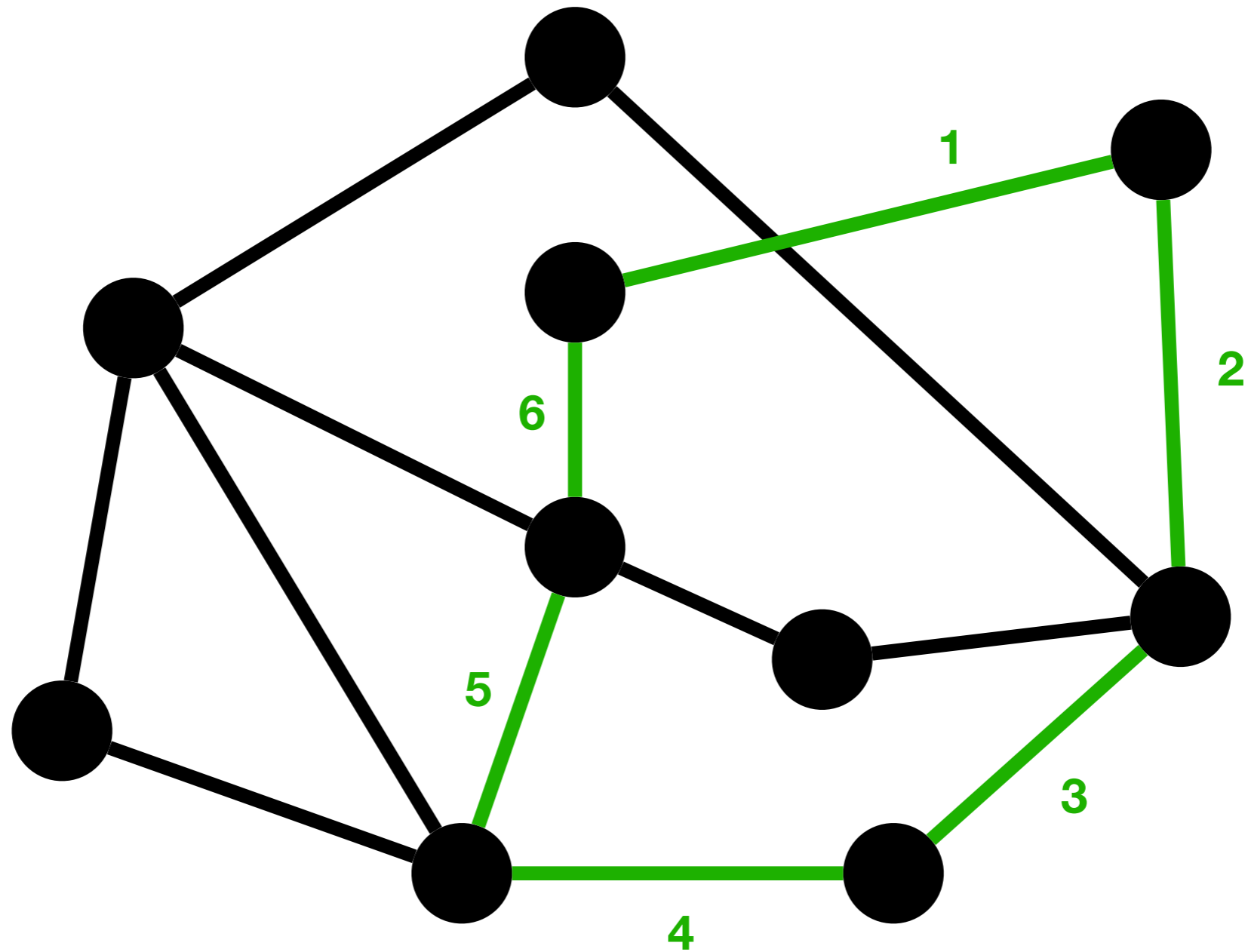
Algorithme de Hierholzer



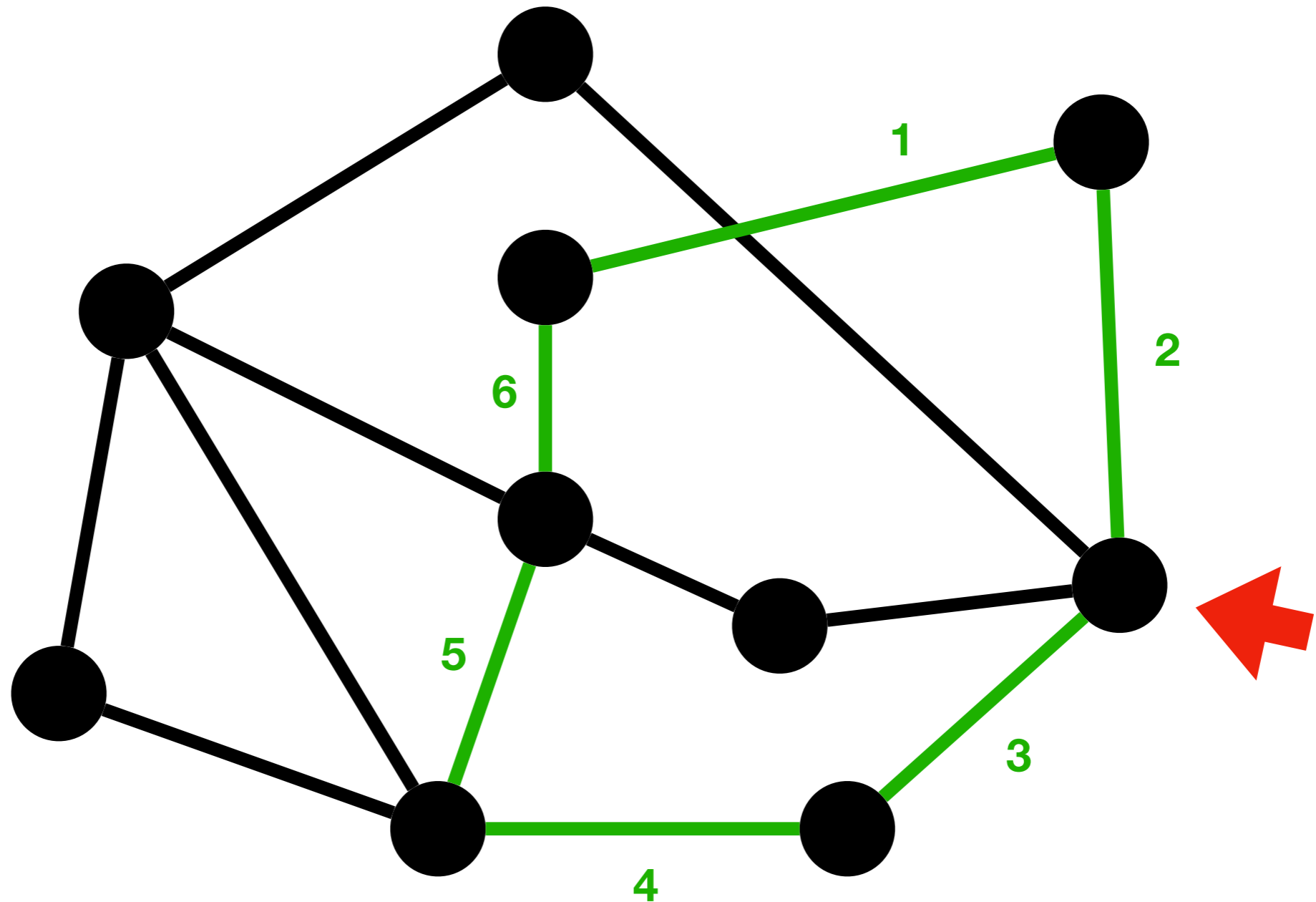
Algorithme de Hierholzer



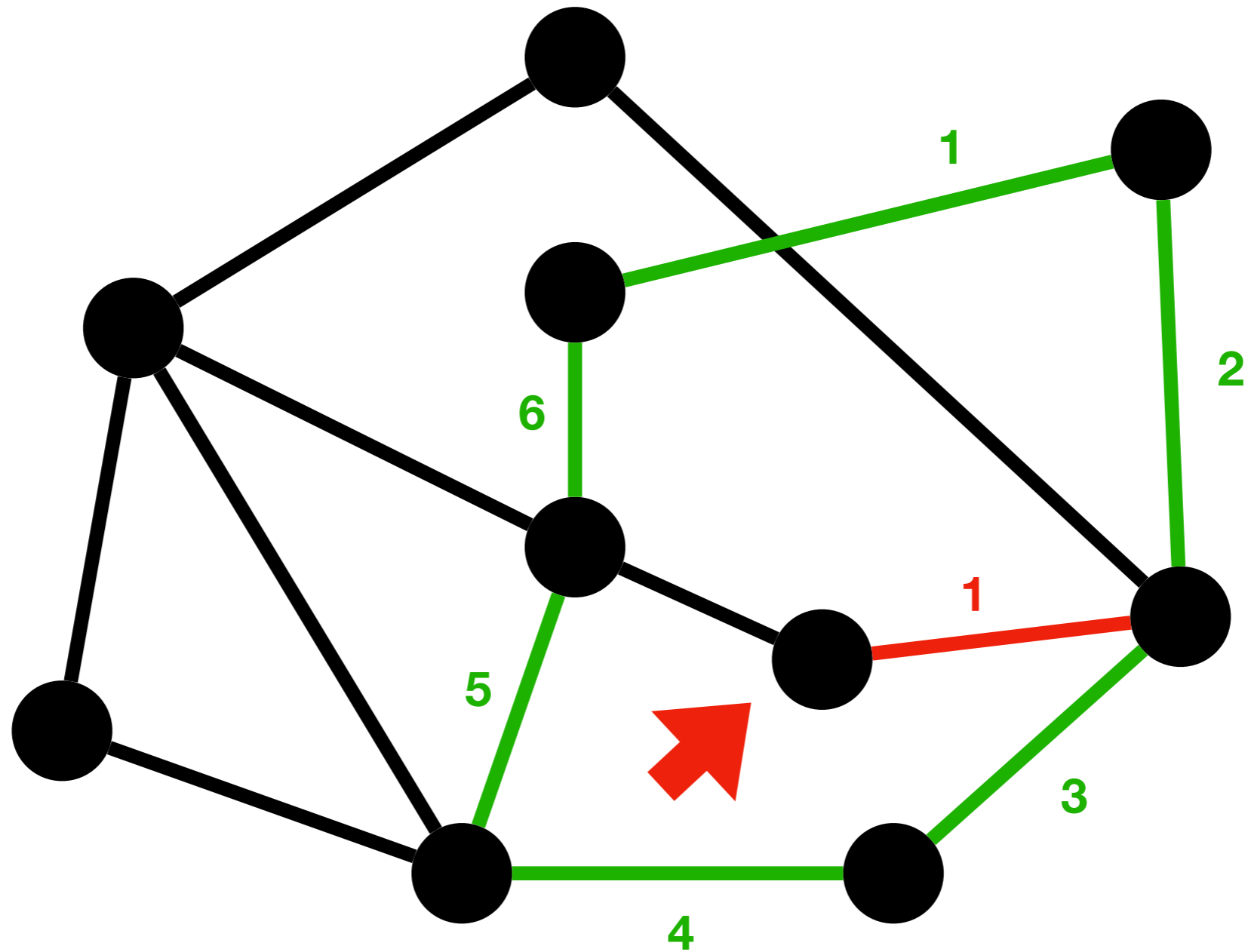
Algorithme de Hierholzer



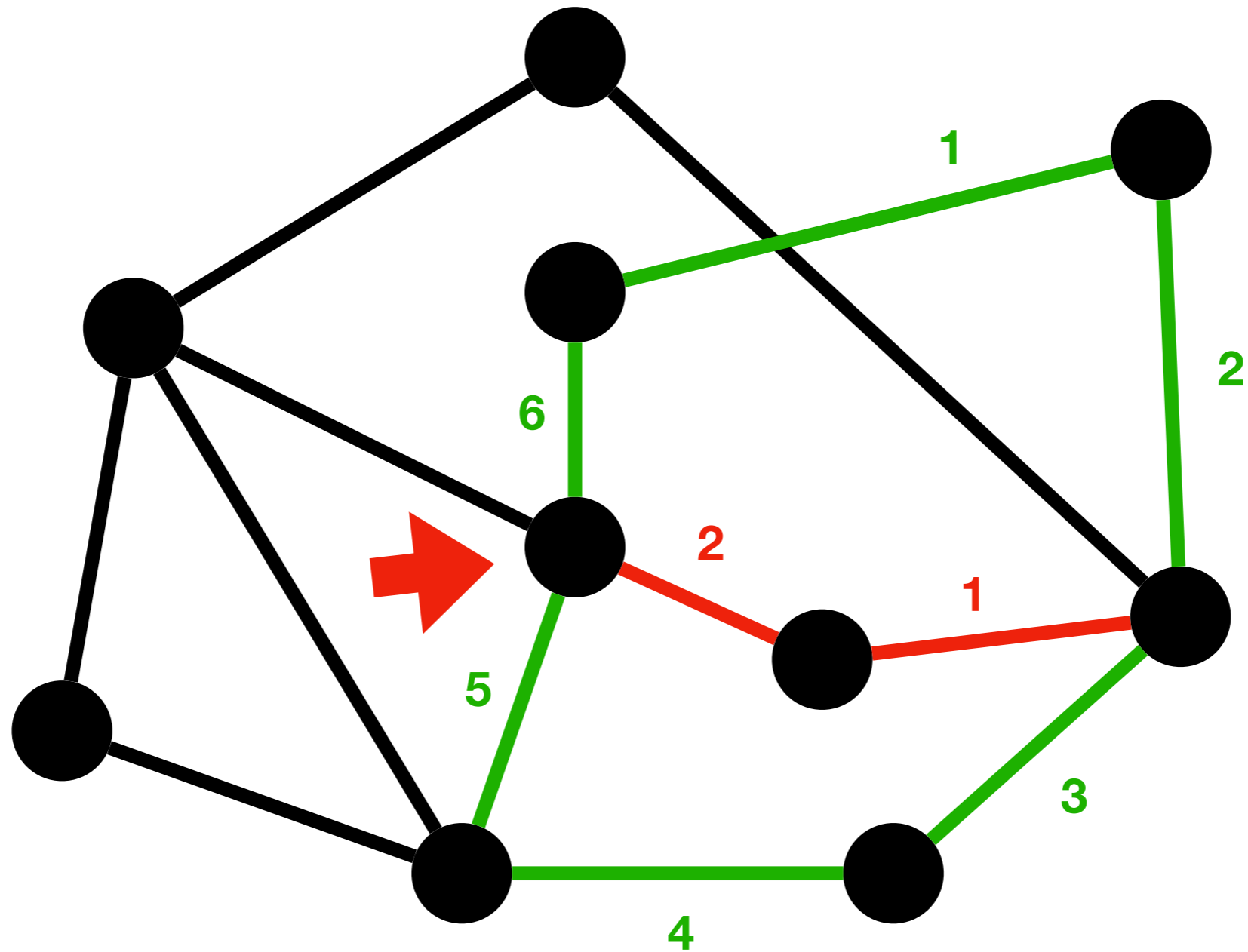
Algorithme de Hierholzer



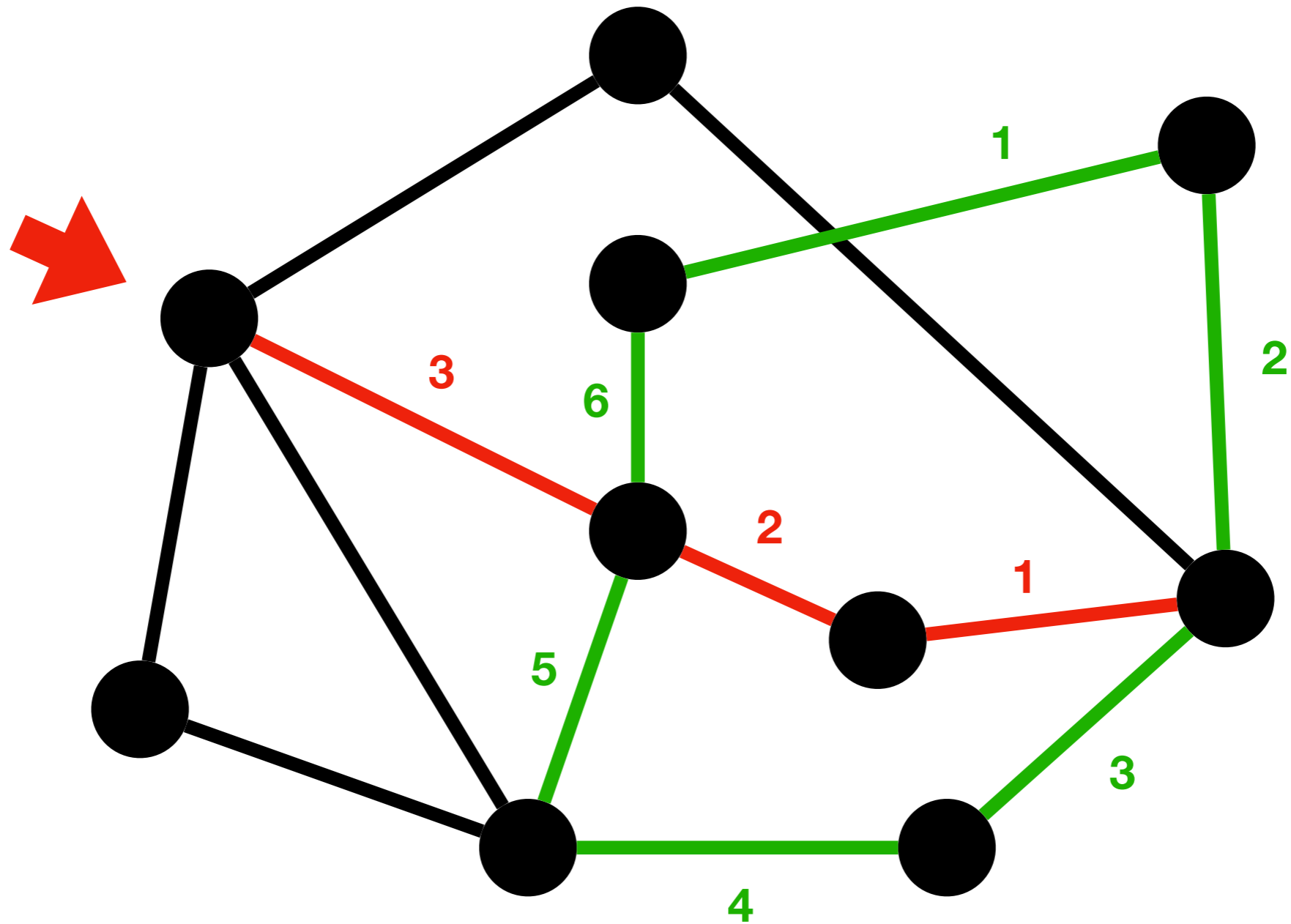
Algorithme de Hierholzer



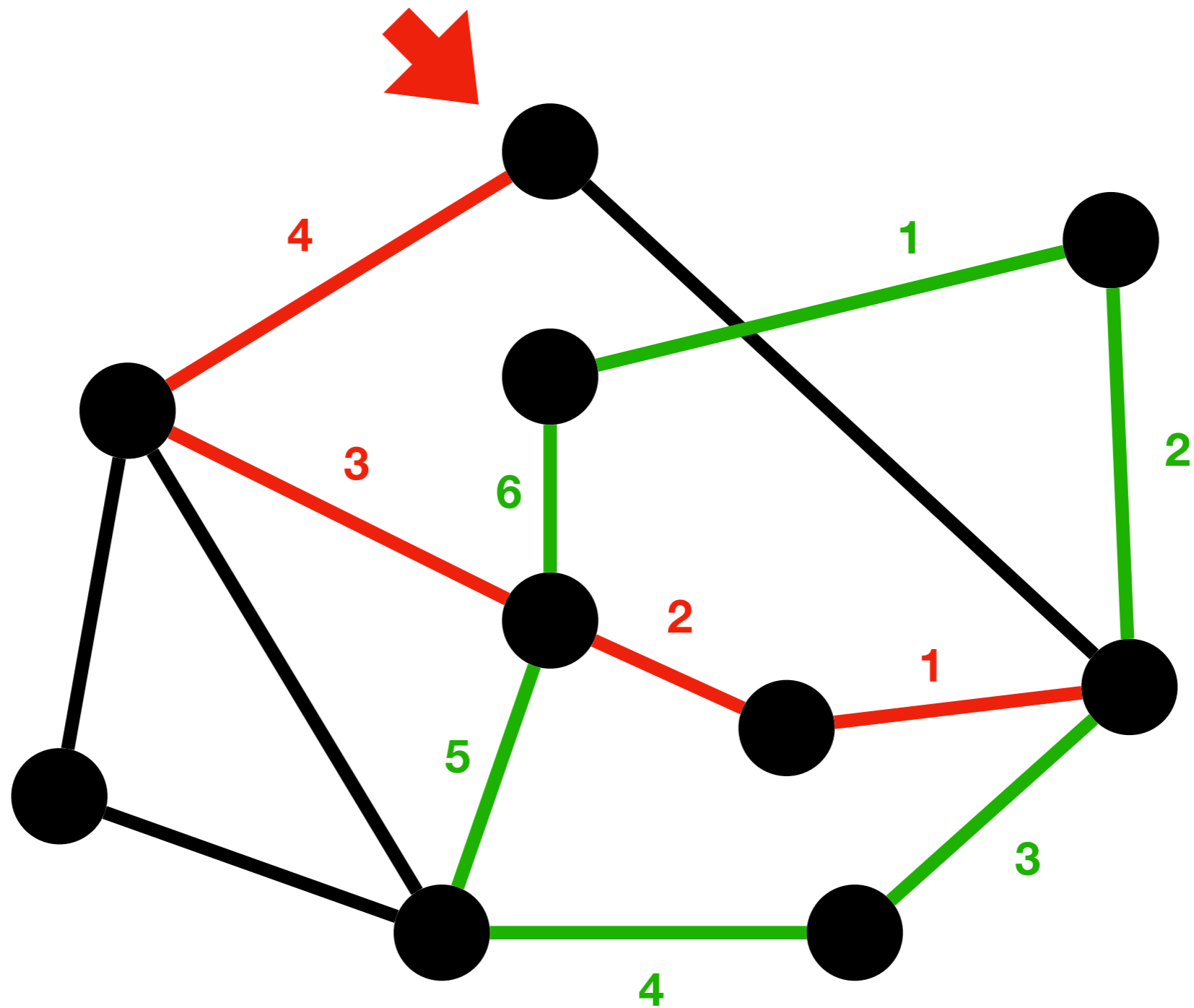
Algorithme de Hierholzer



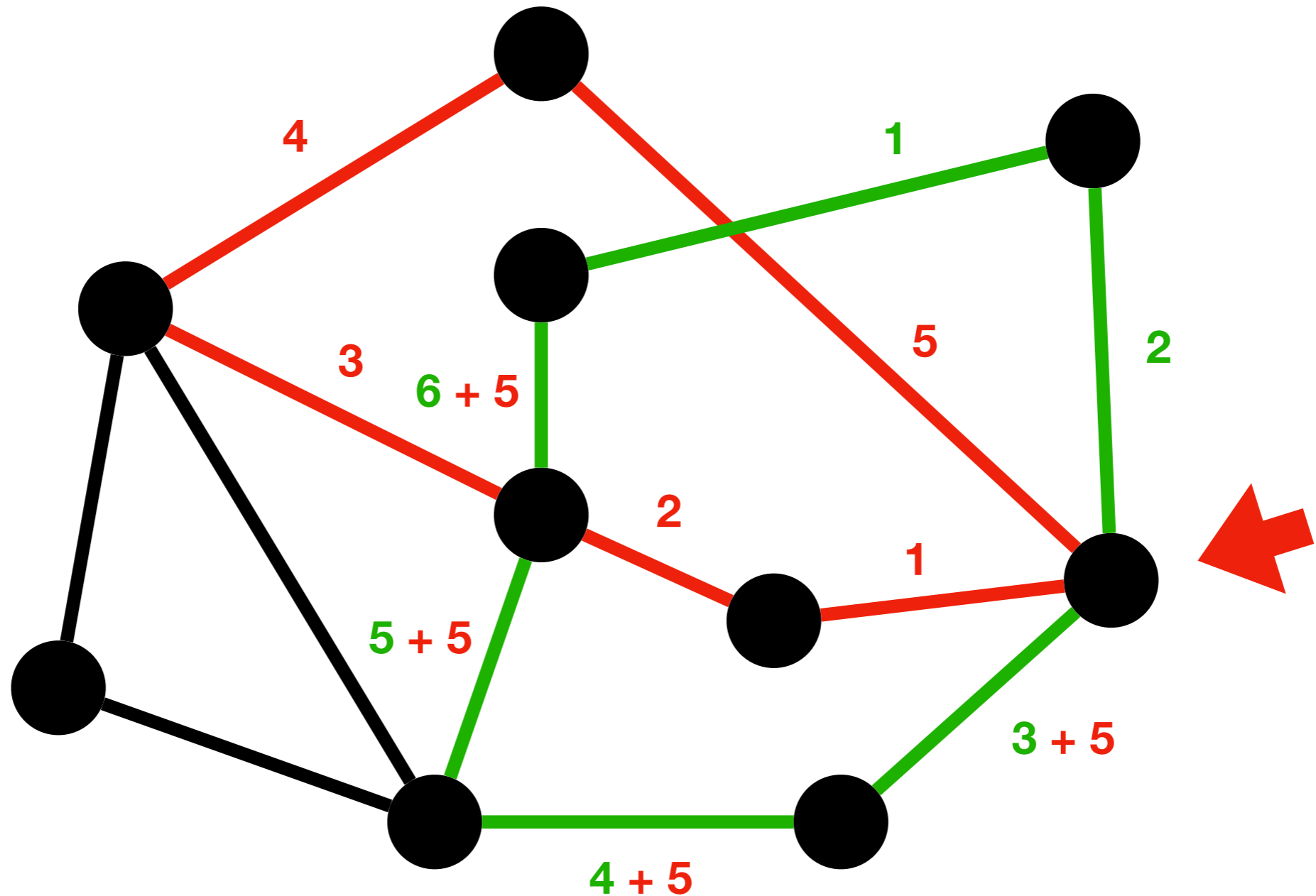
Algorithme de Hierholzer



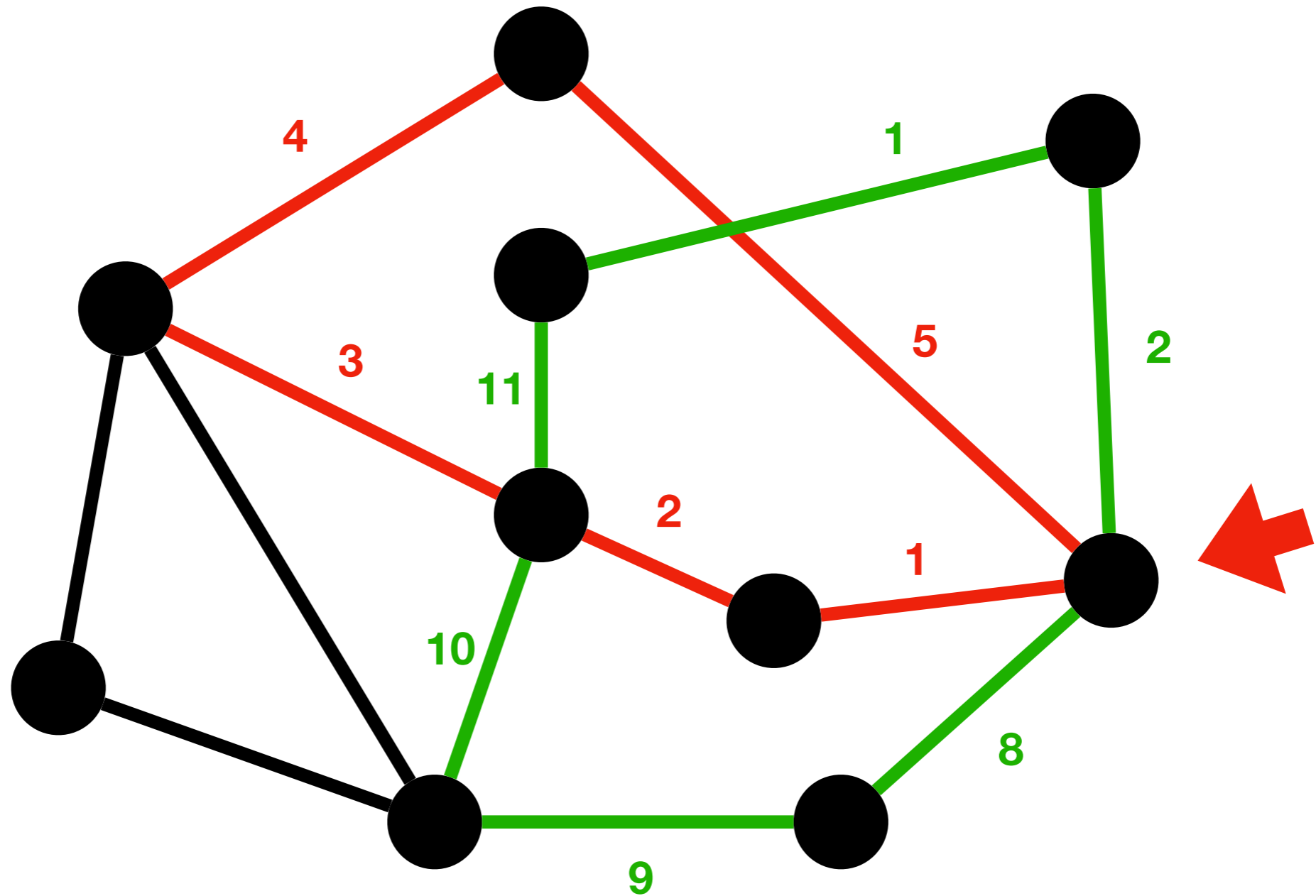
Algorithme de Hierholzer



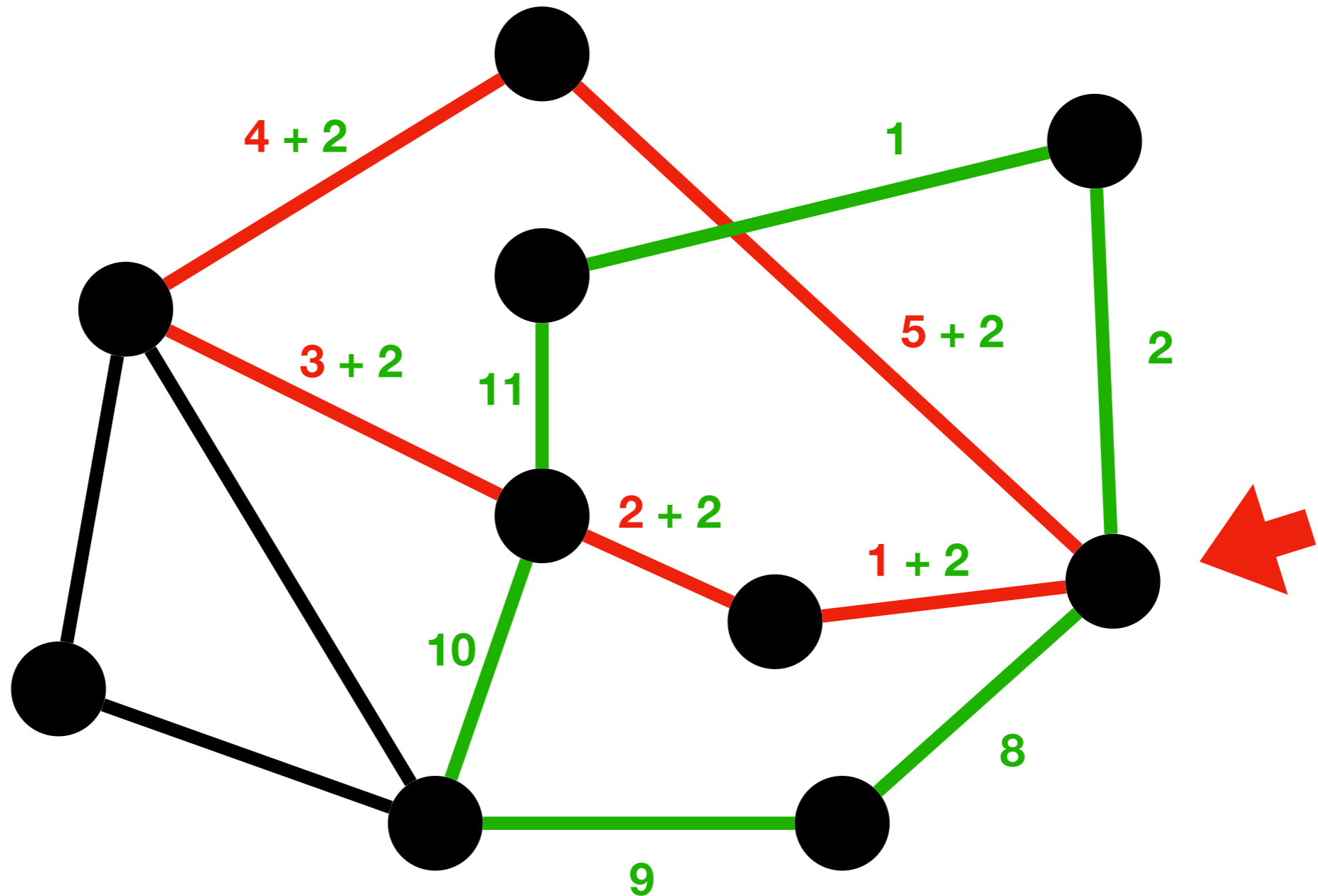
Algorithme de Hierholzer



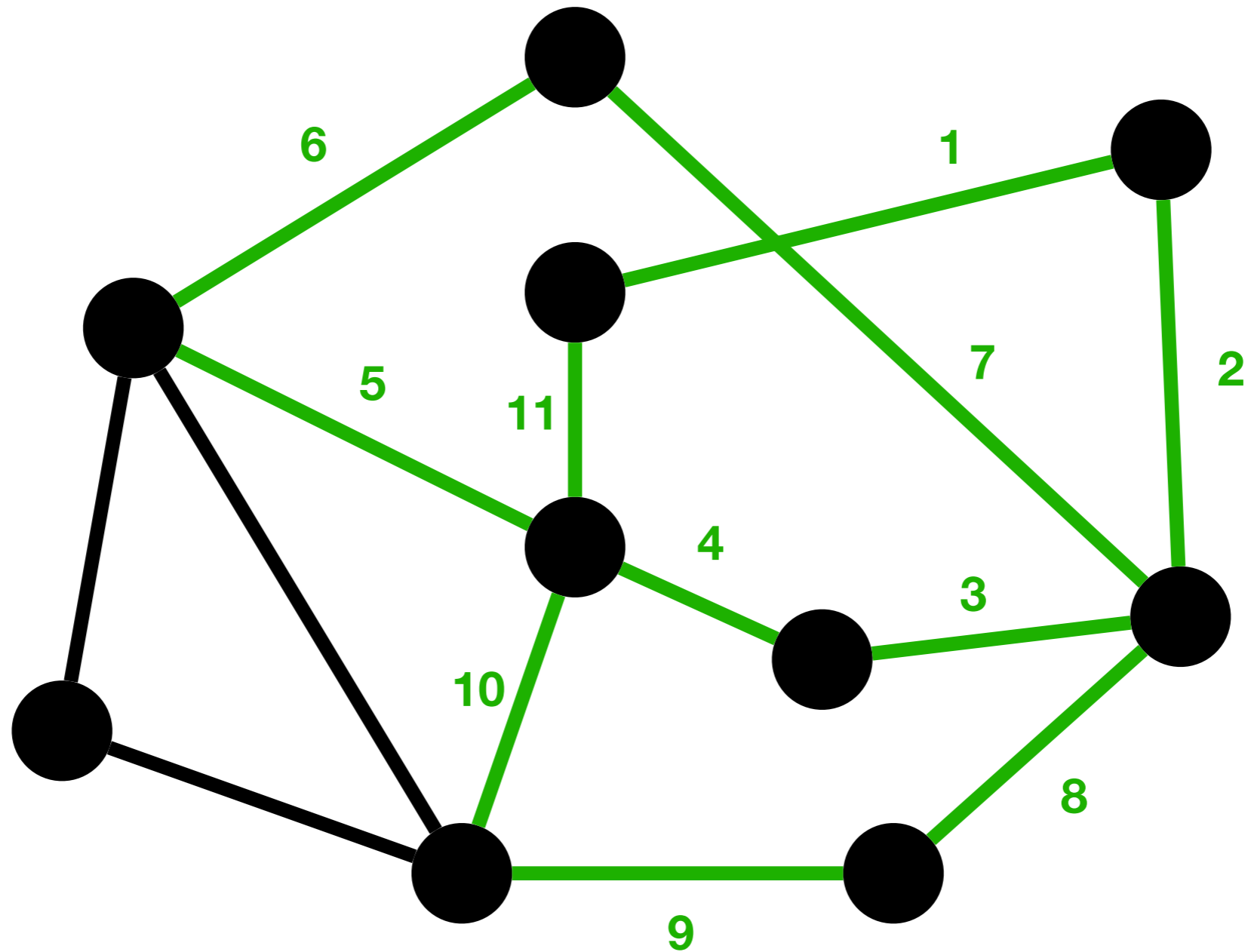
Algorithme de Hierholzer



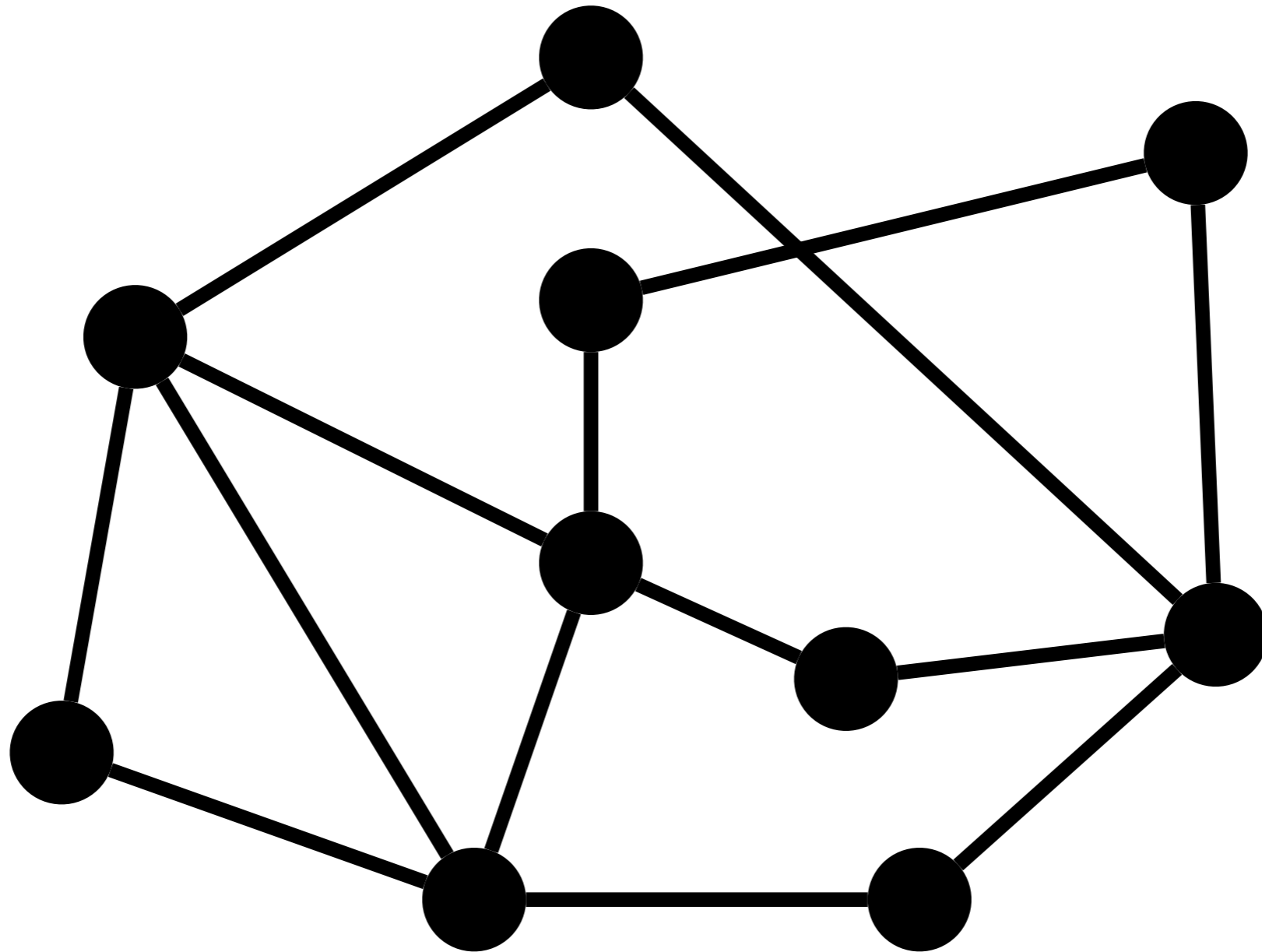
Algorithme de Hierholzer



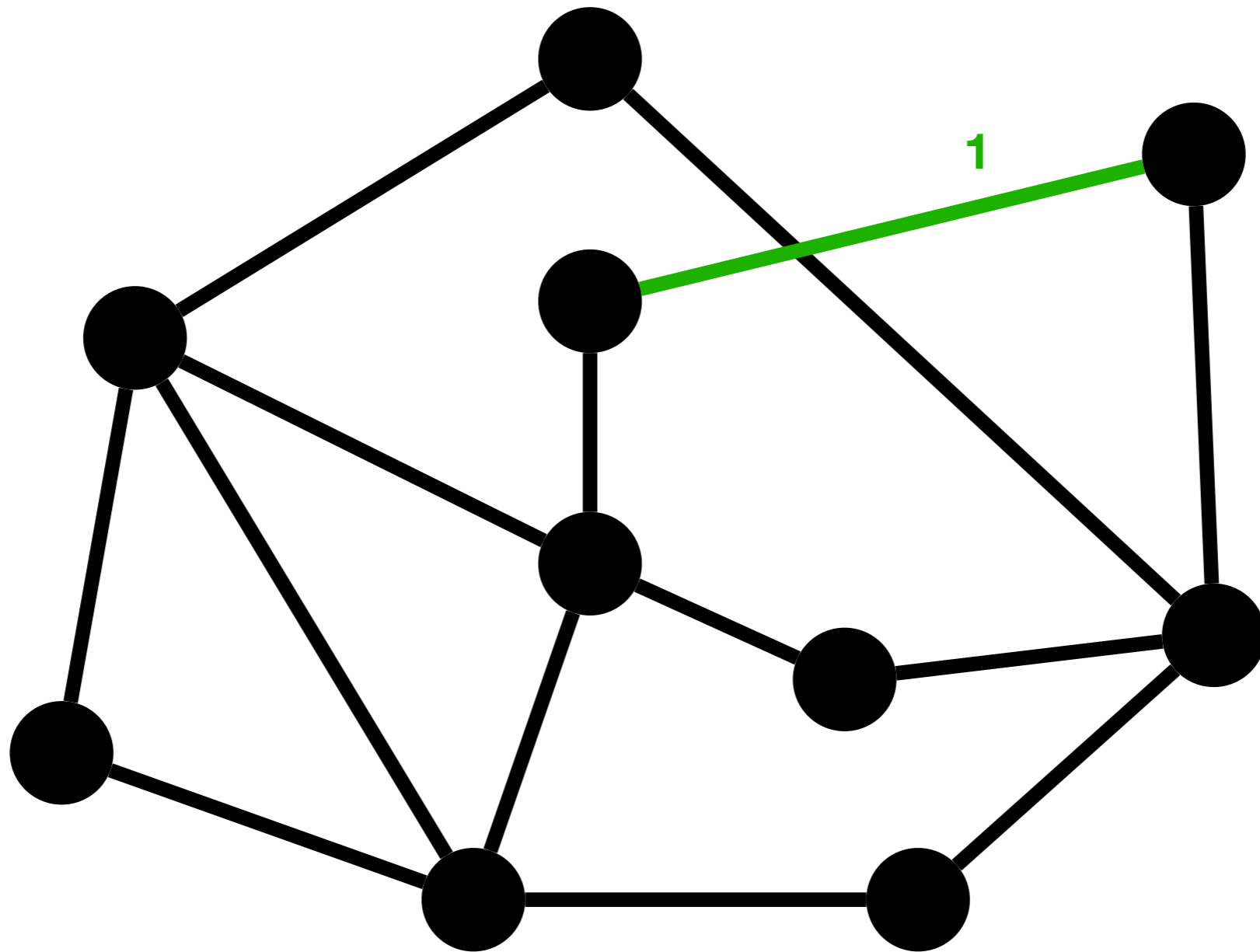
Algorithme de Hierholzer



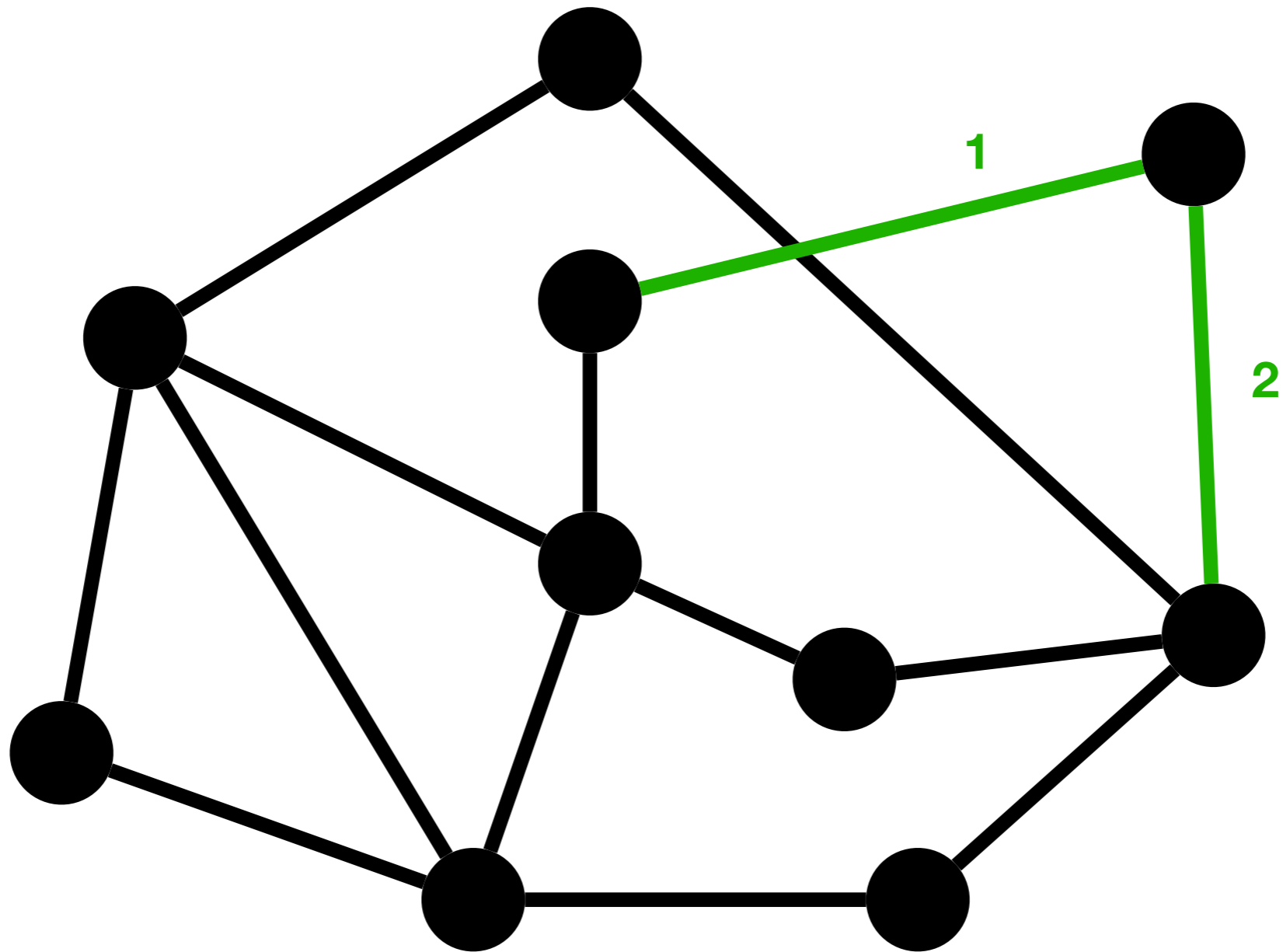
Algorithme de Hierholzer



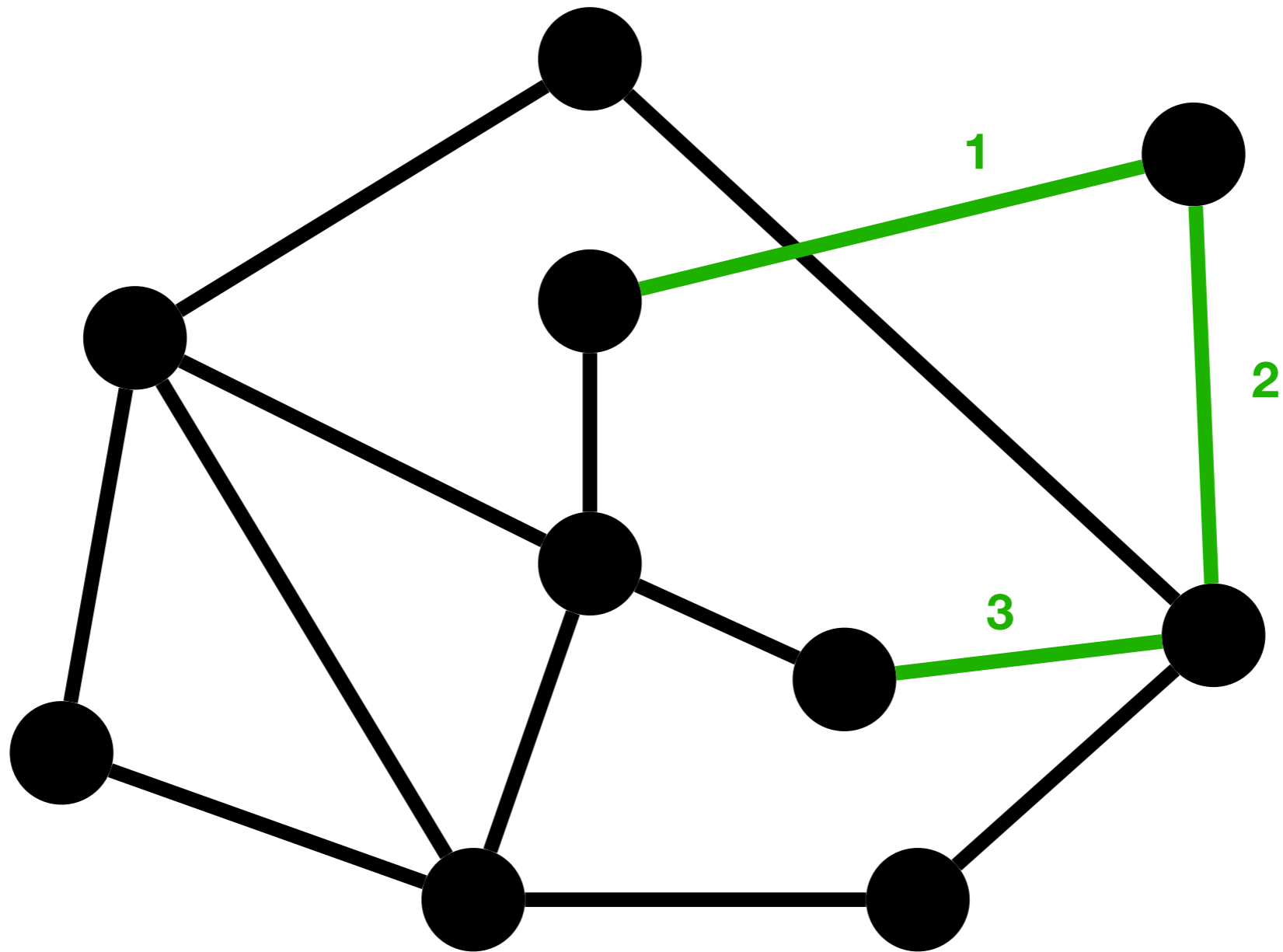
Algorithme de Hierholzer



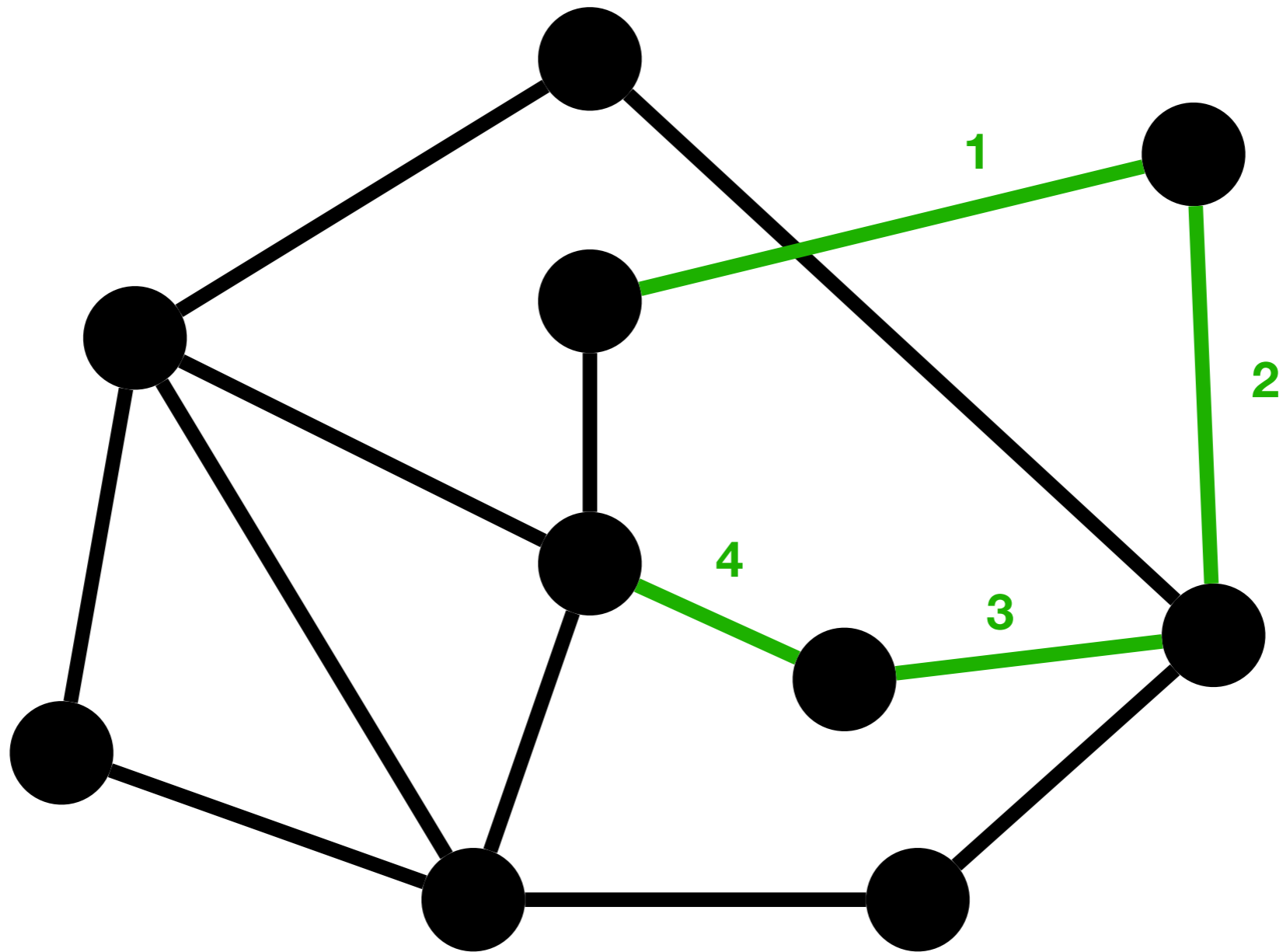
Algorithme de Hierholzer



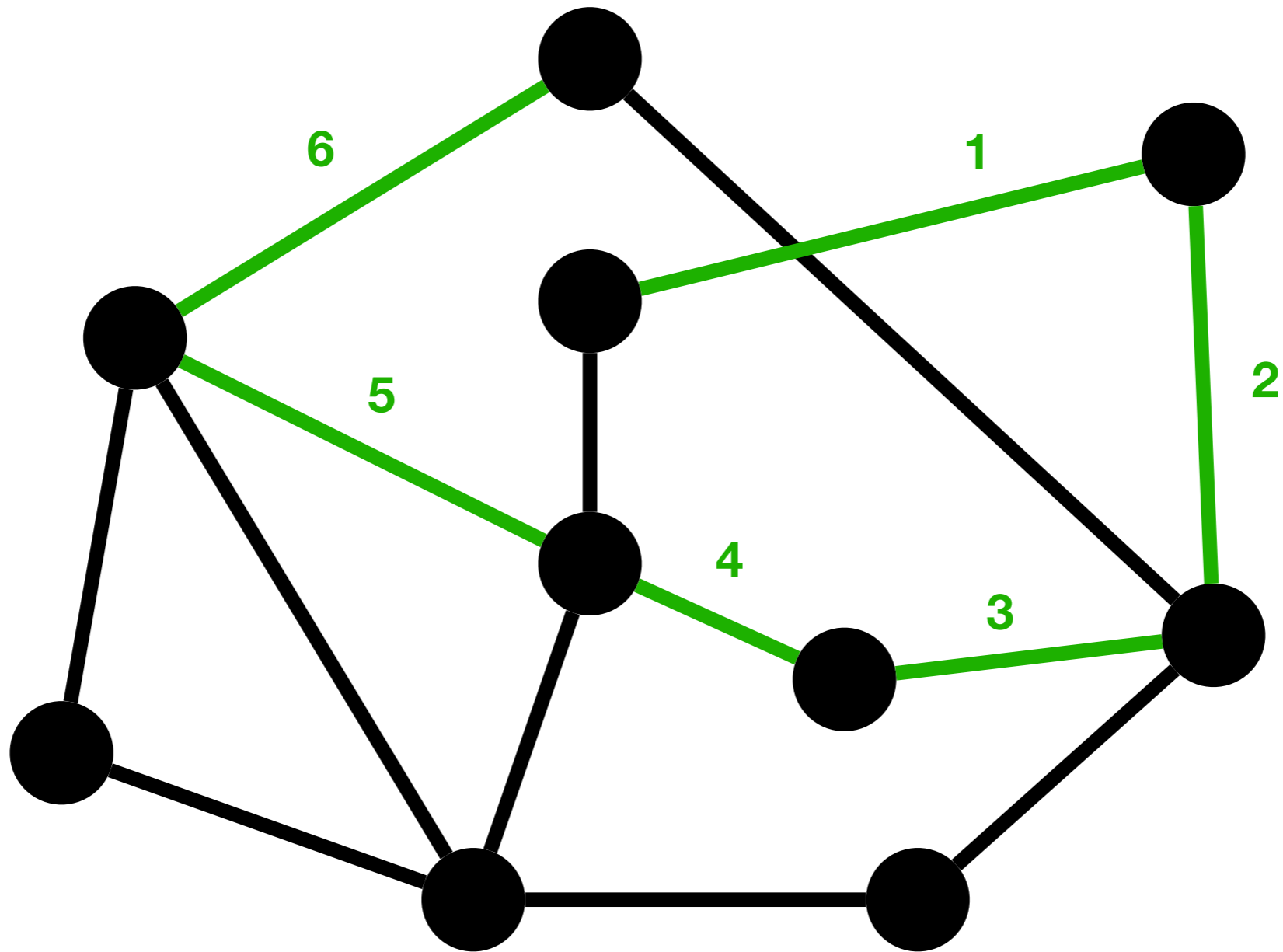
Algorithme de Hierholzer



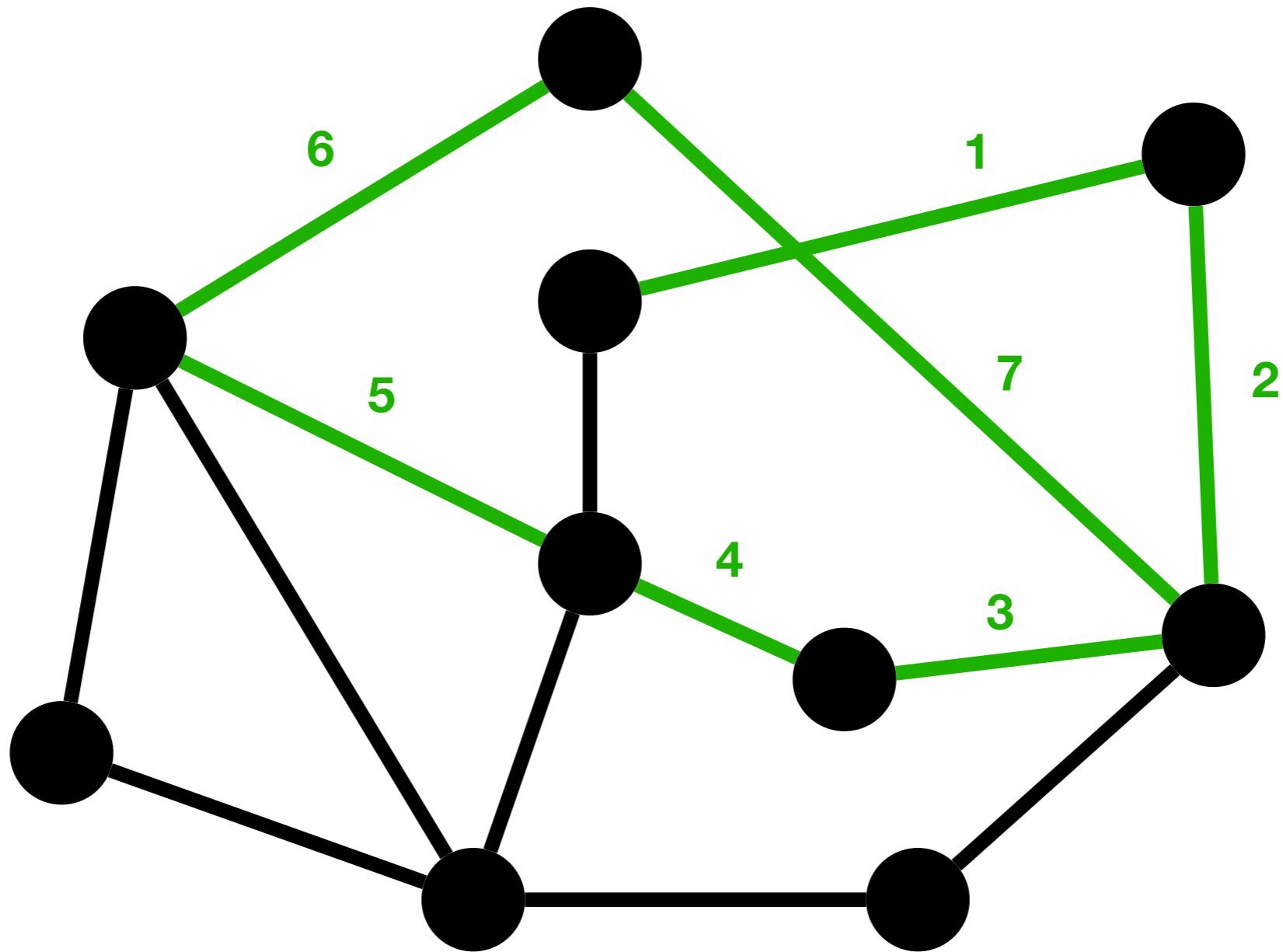
Algorithme de Hierholzer



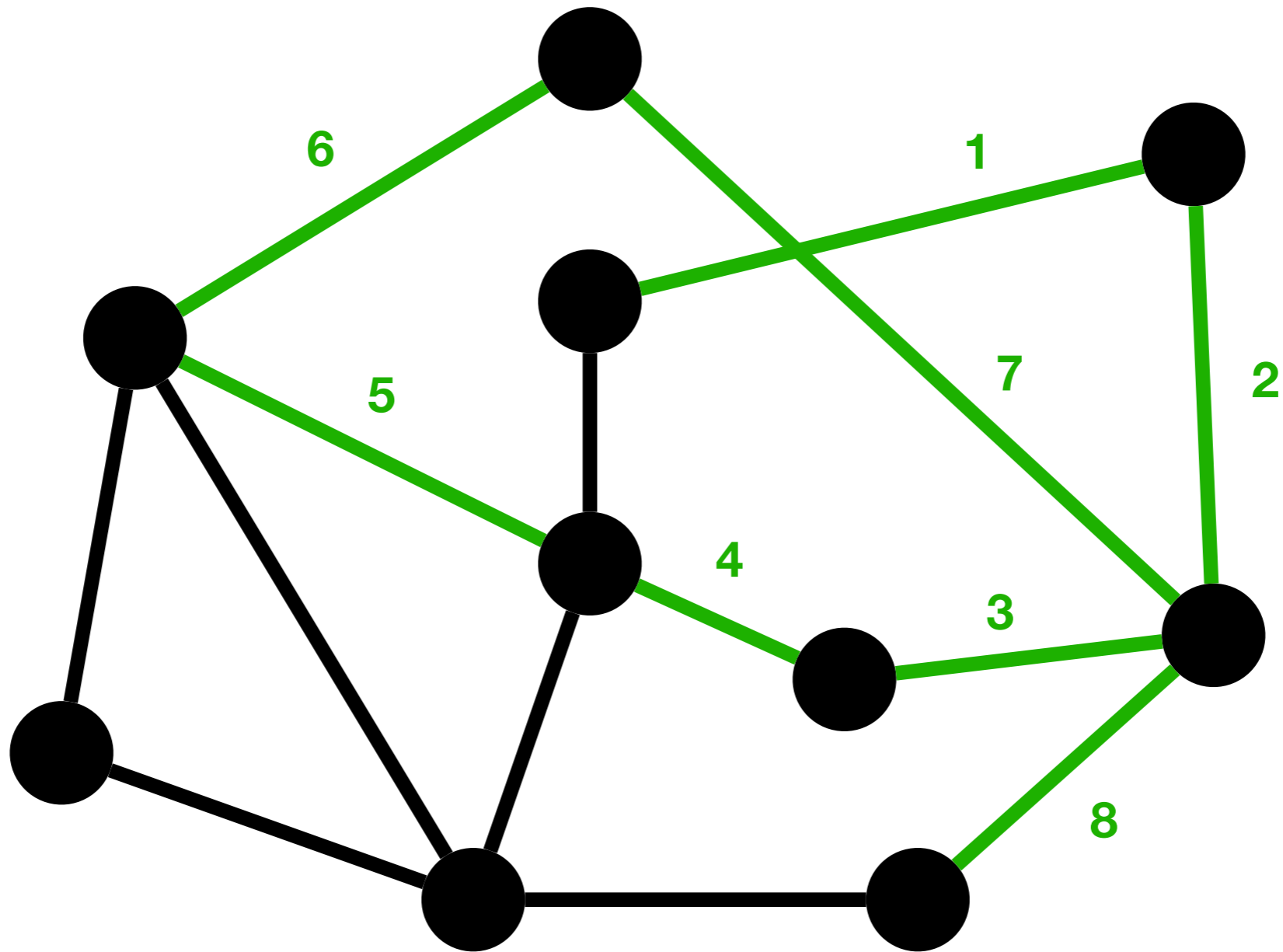
Algorithme de Hierholzer



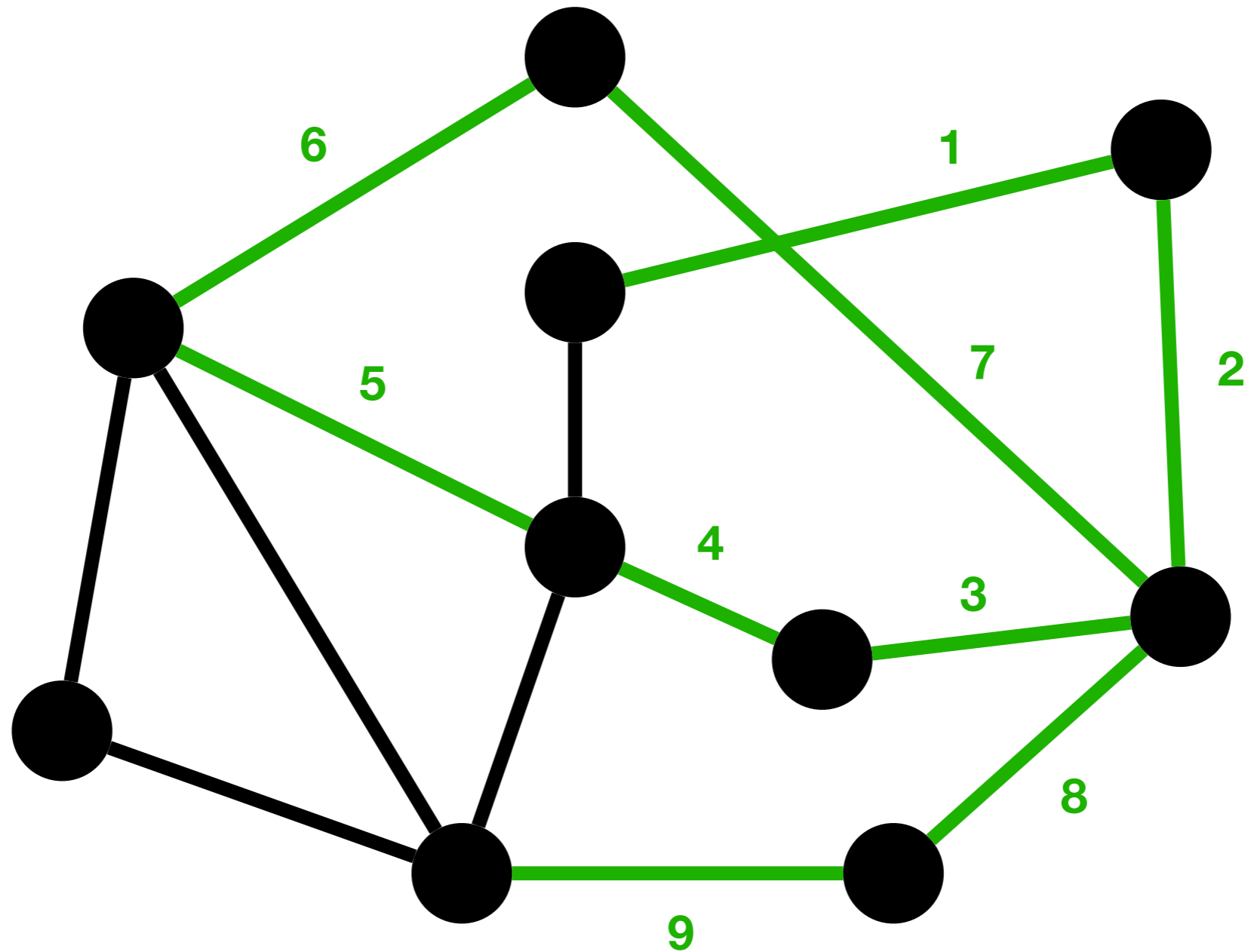
Algorithme de Hierholzer



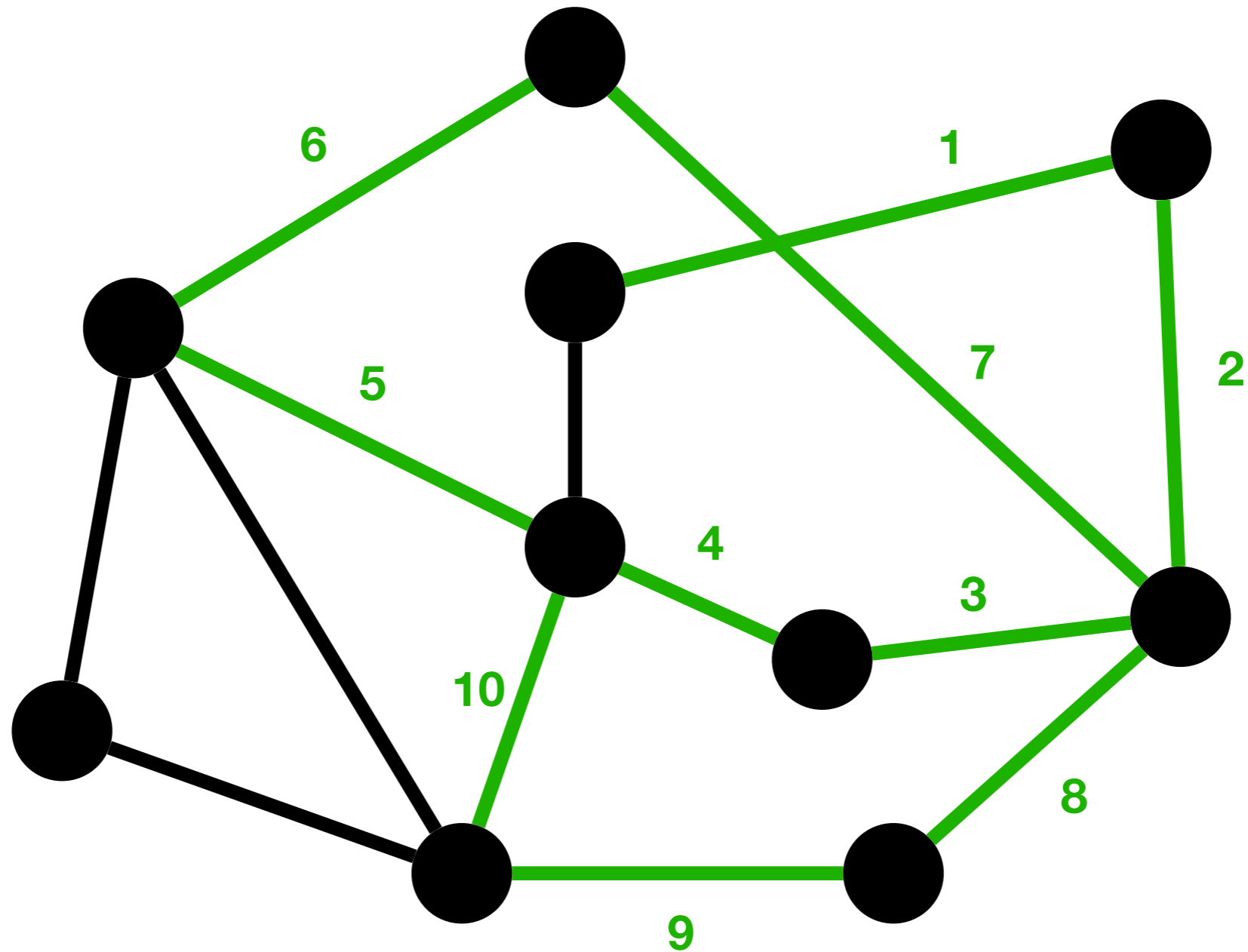
Algorithme de Hierholzer



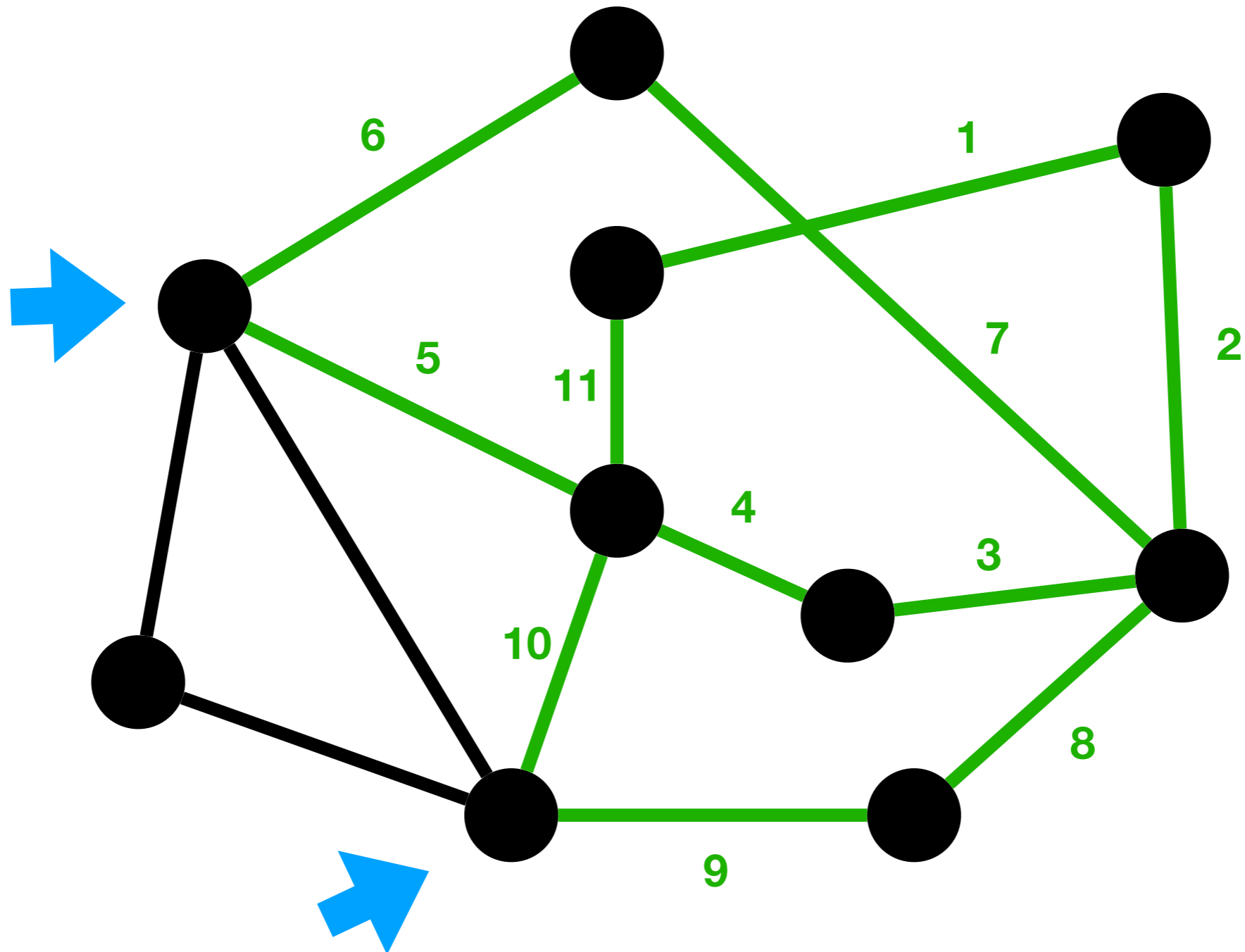
Algorithme de Hierholzer



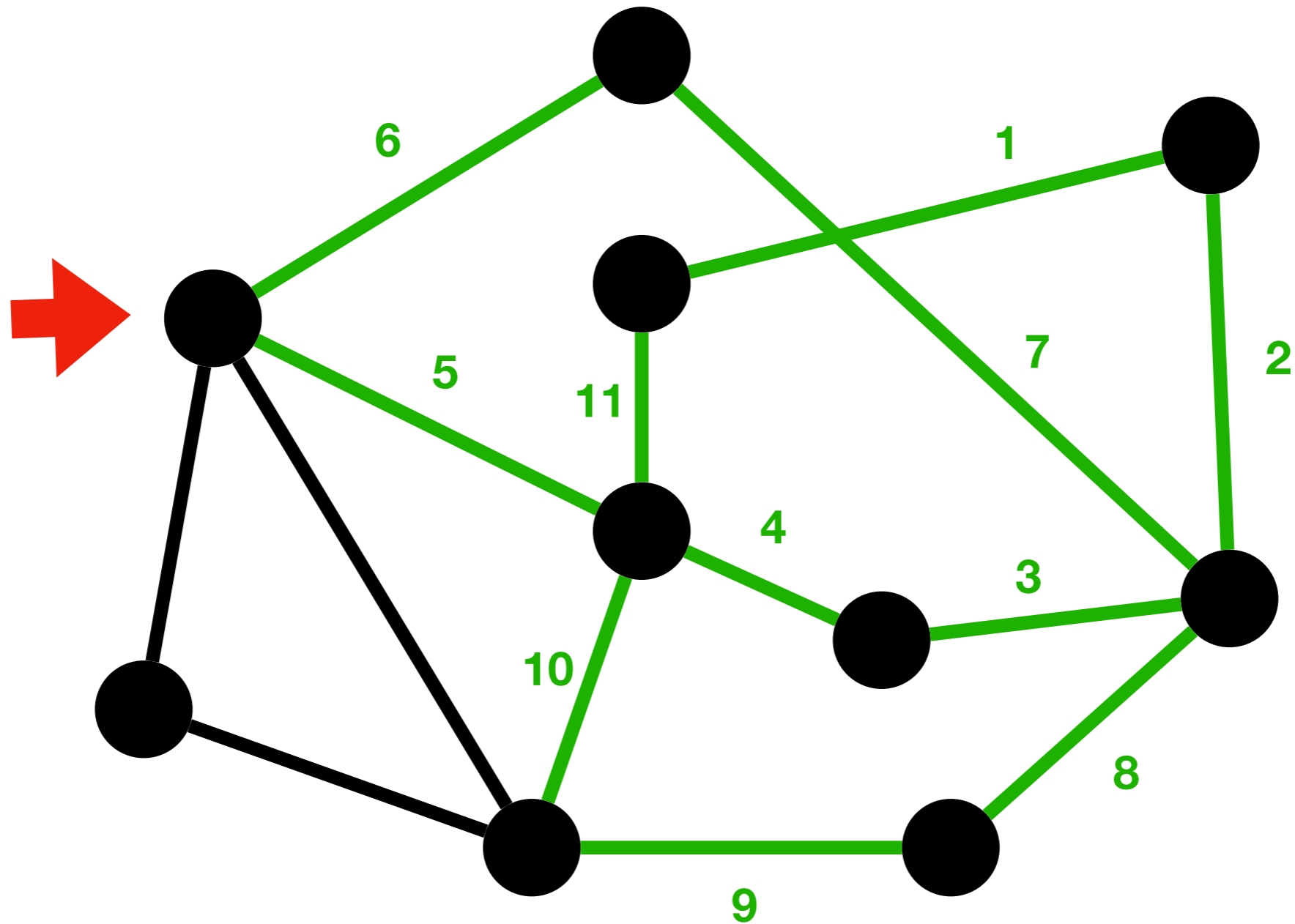
Algorithme de Hierholzer



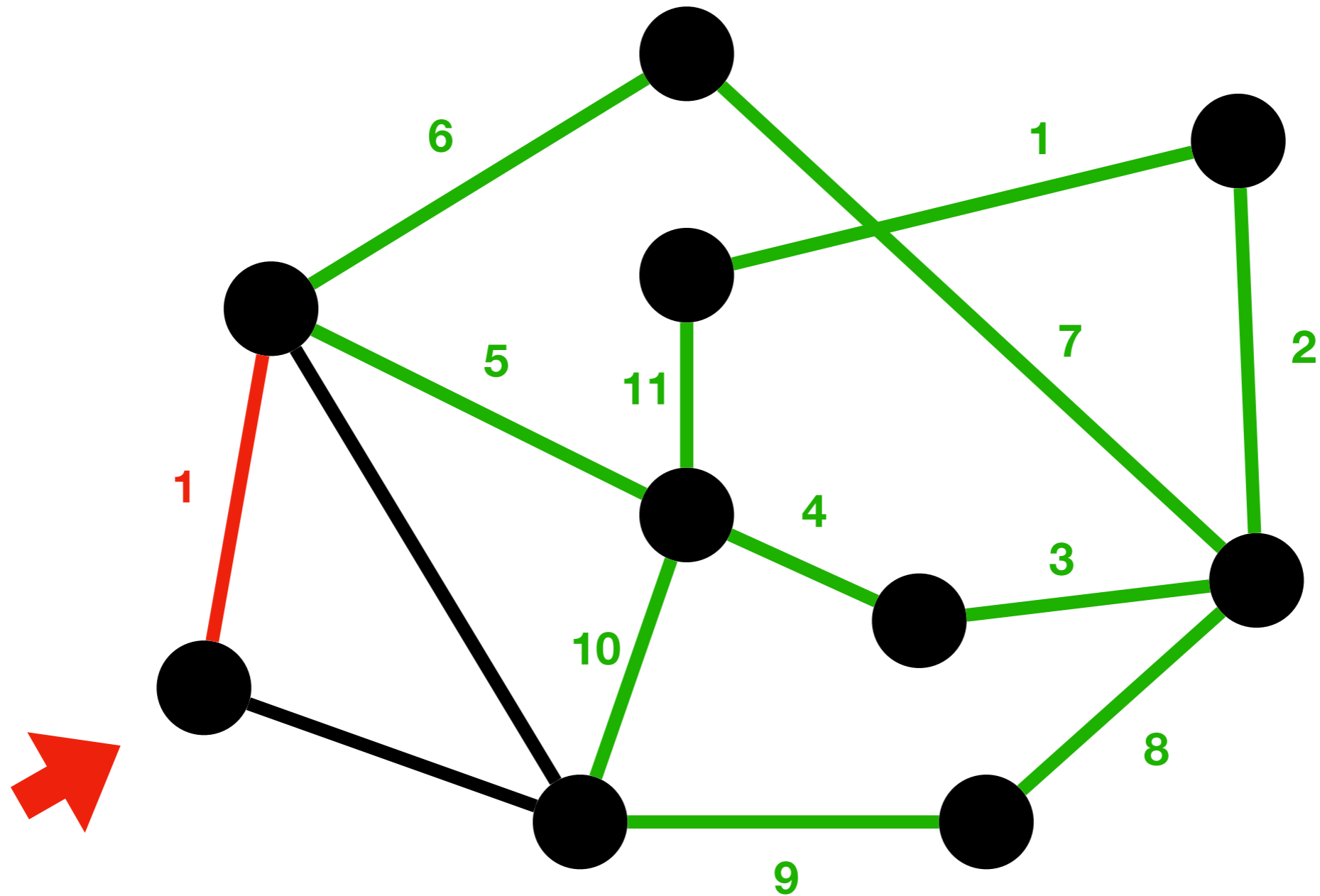
Algorithme de Hierholzer



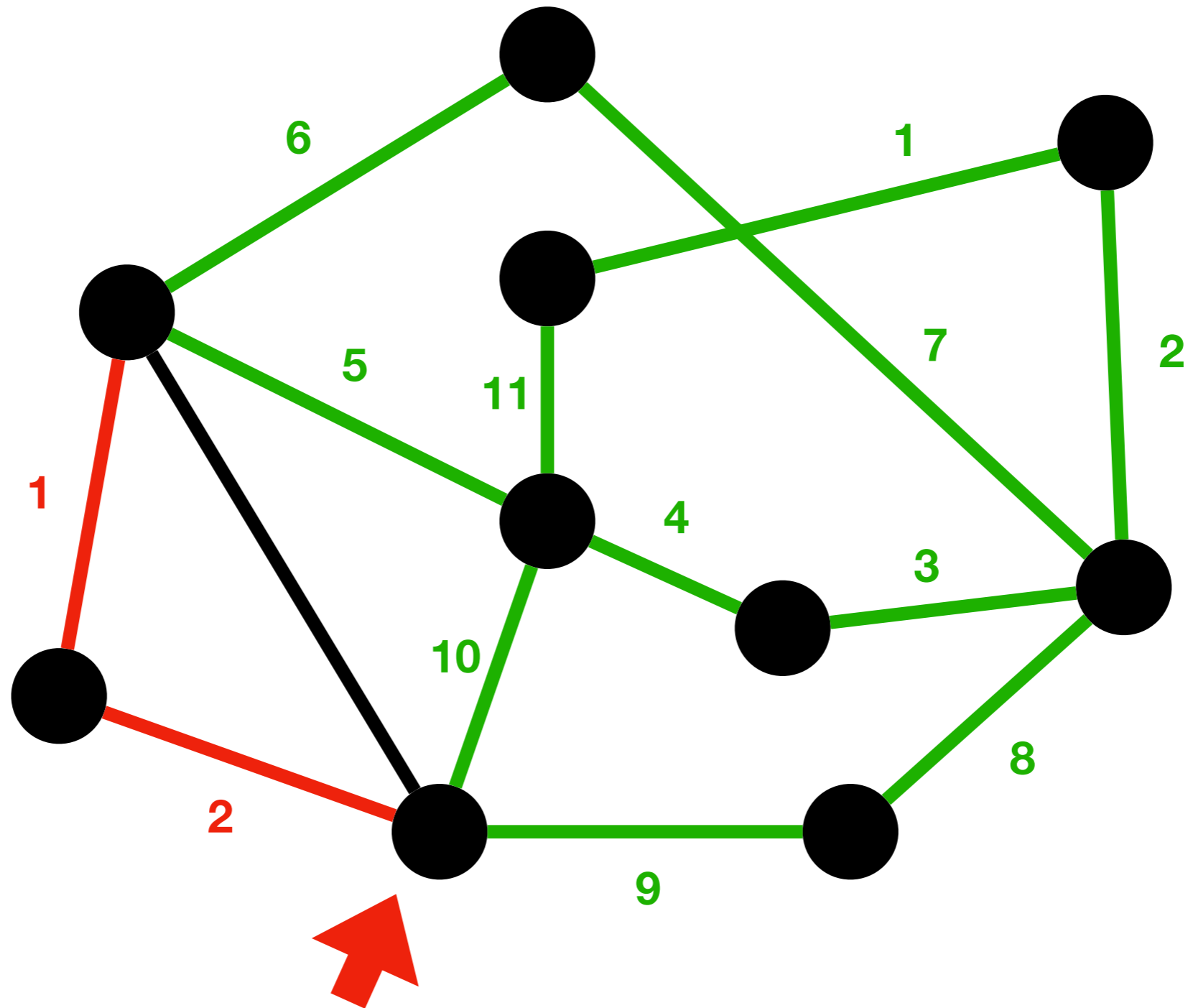
Algorithme de Hierholzer



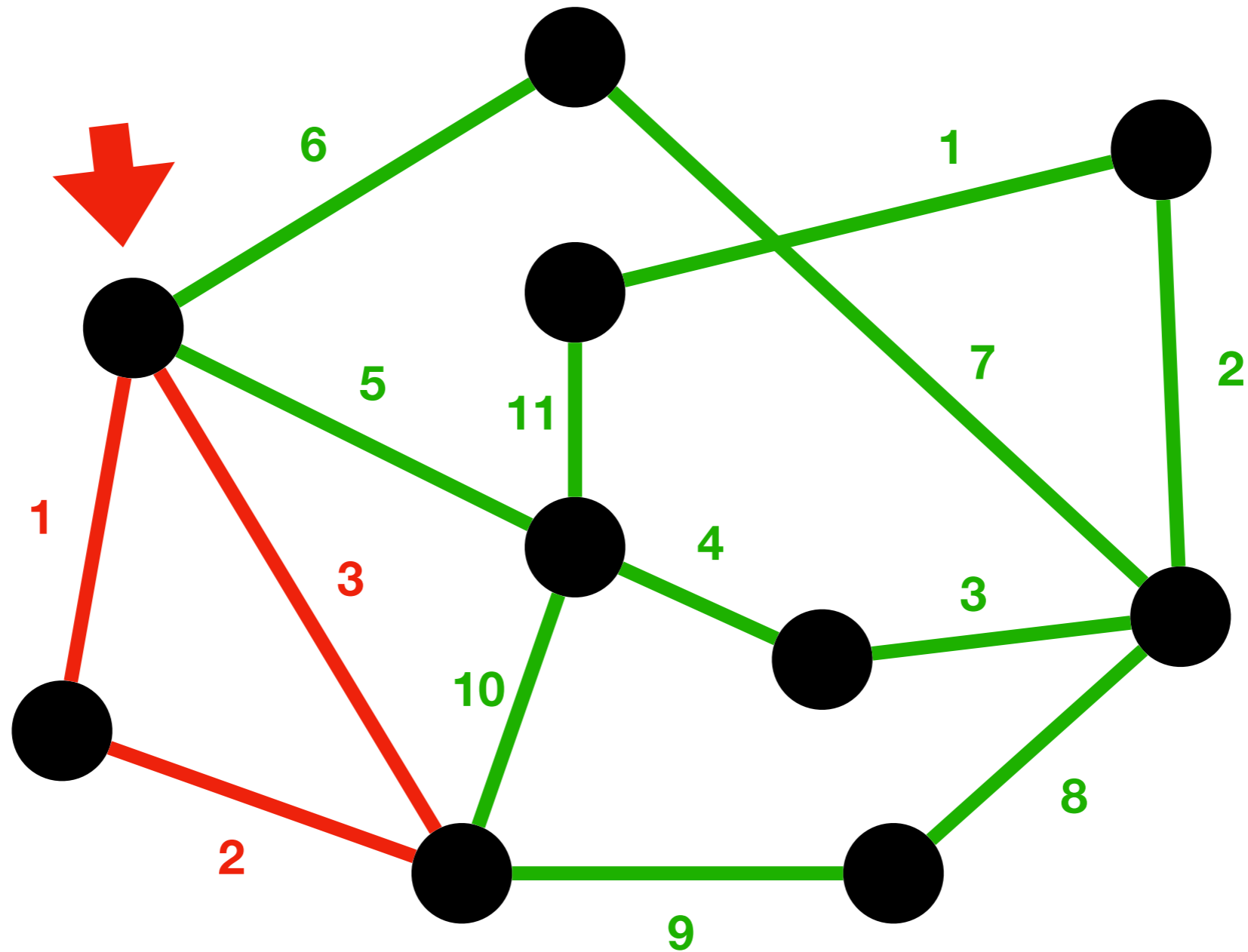
Algorithme de Hierholzer



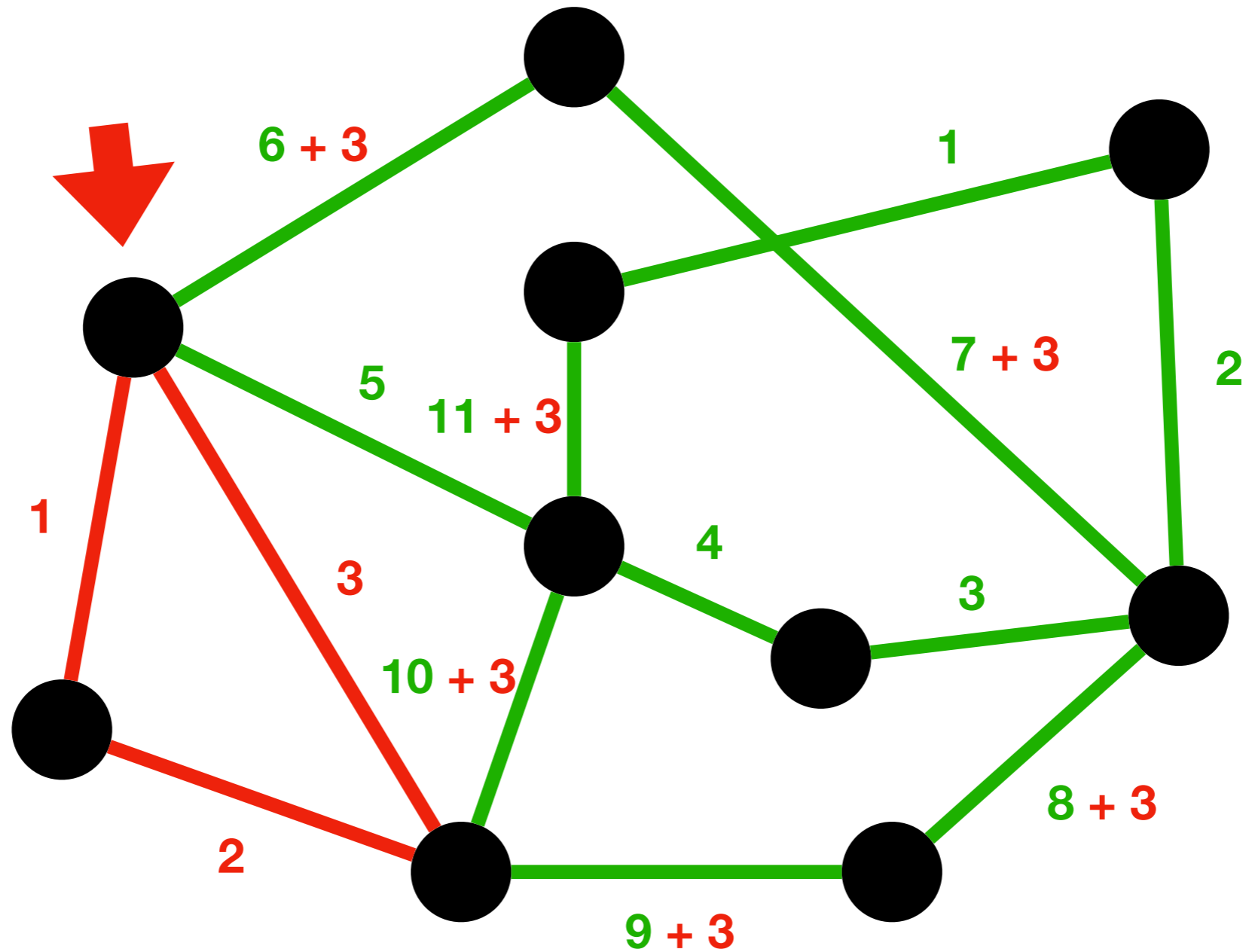
Algorithme de Hierholzer



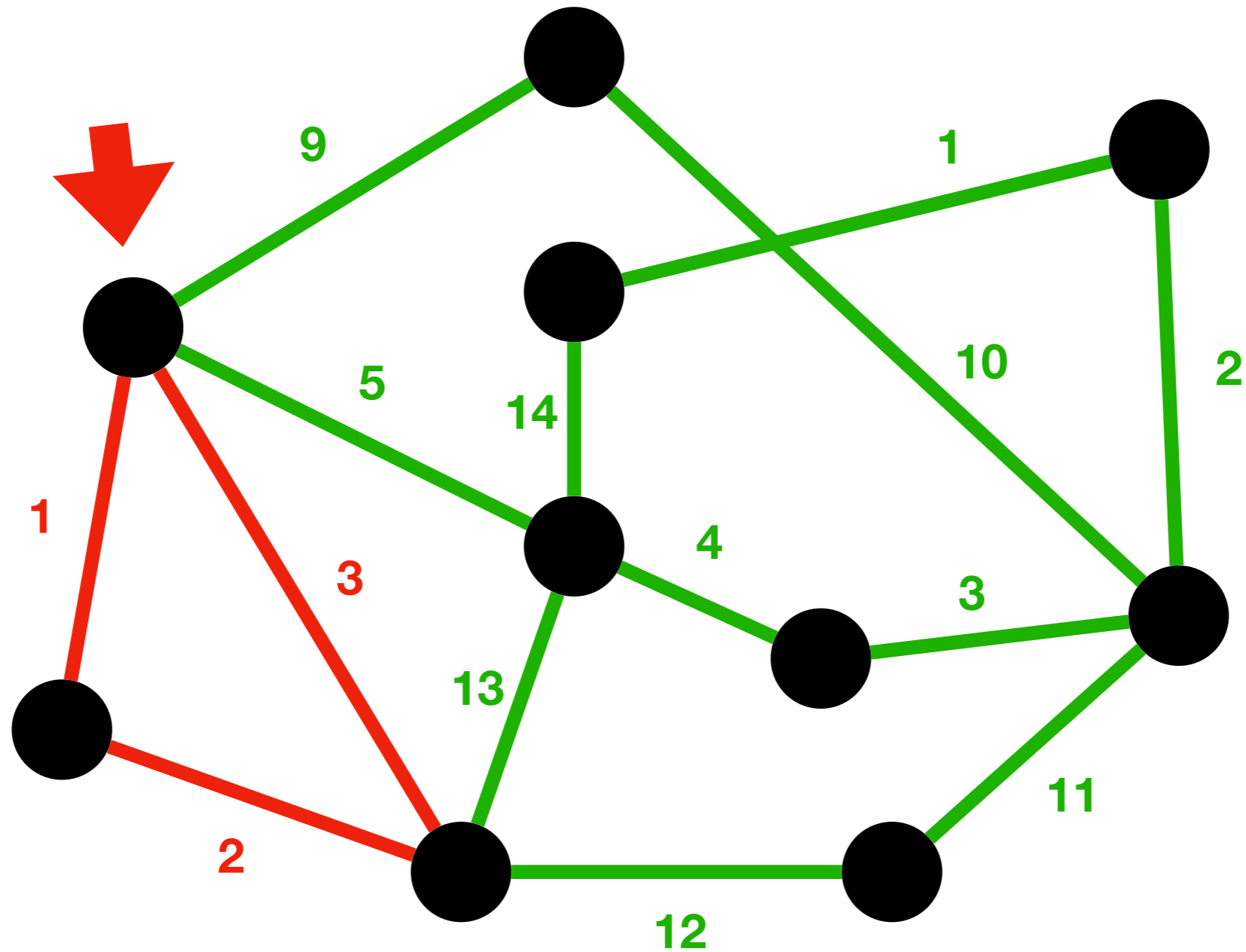
Algorithme de Hierholzer



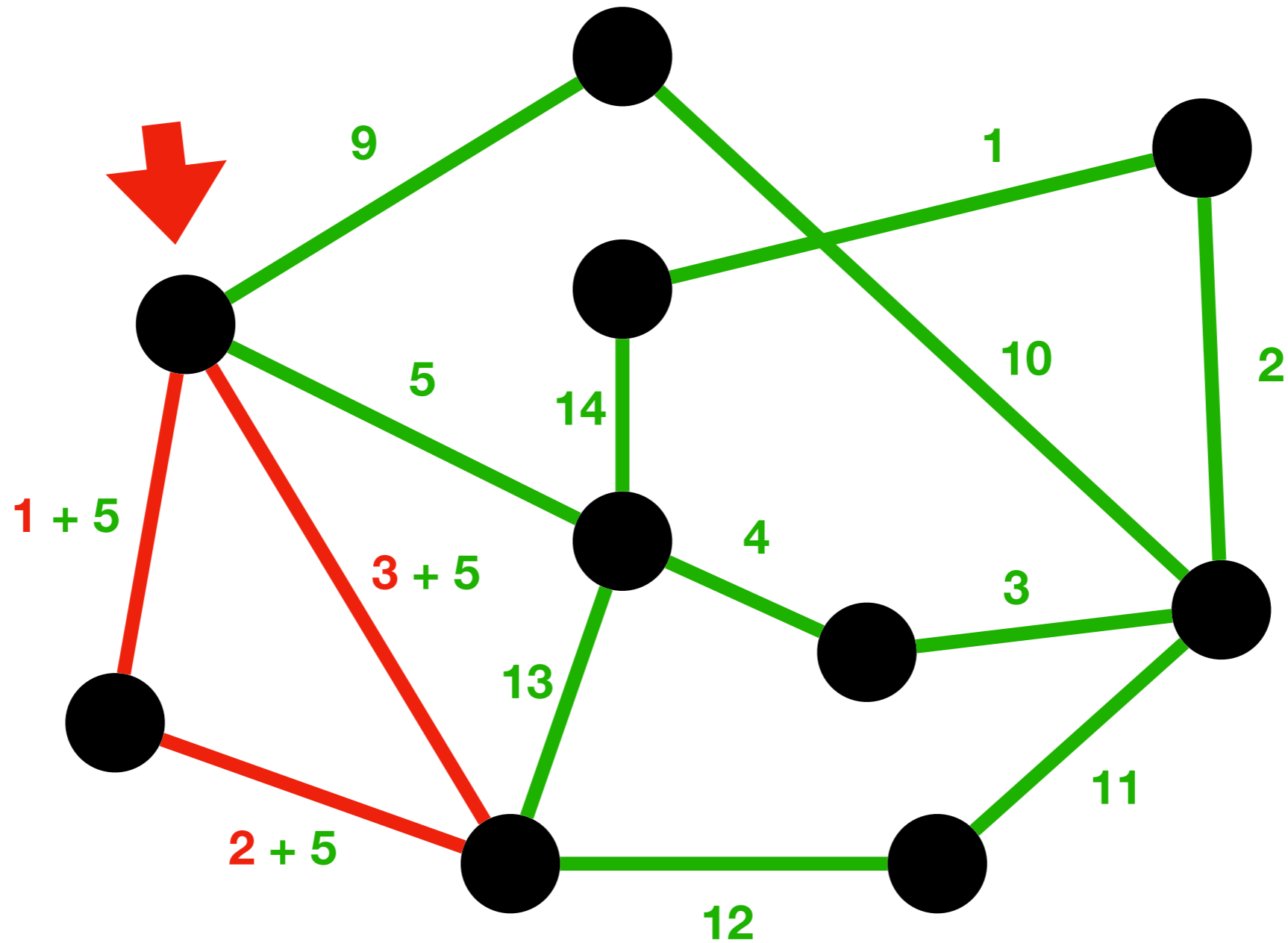
Algorithme de Hierholzer



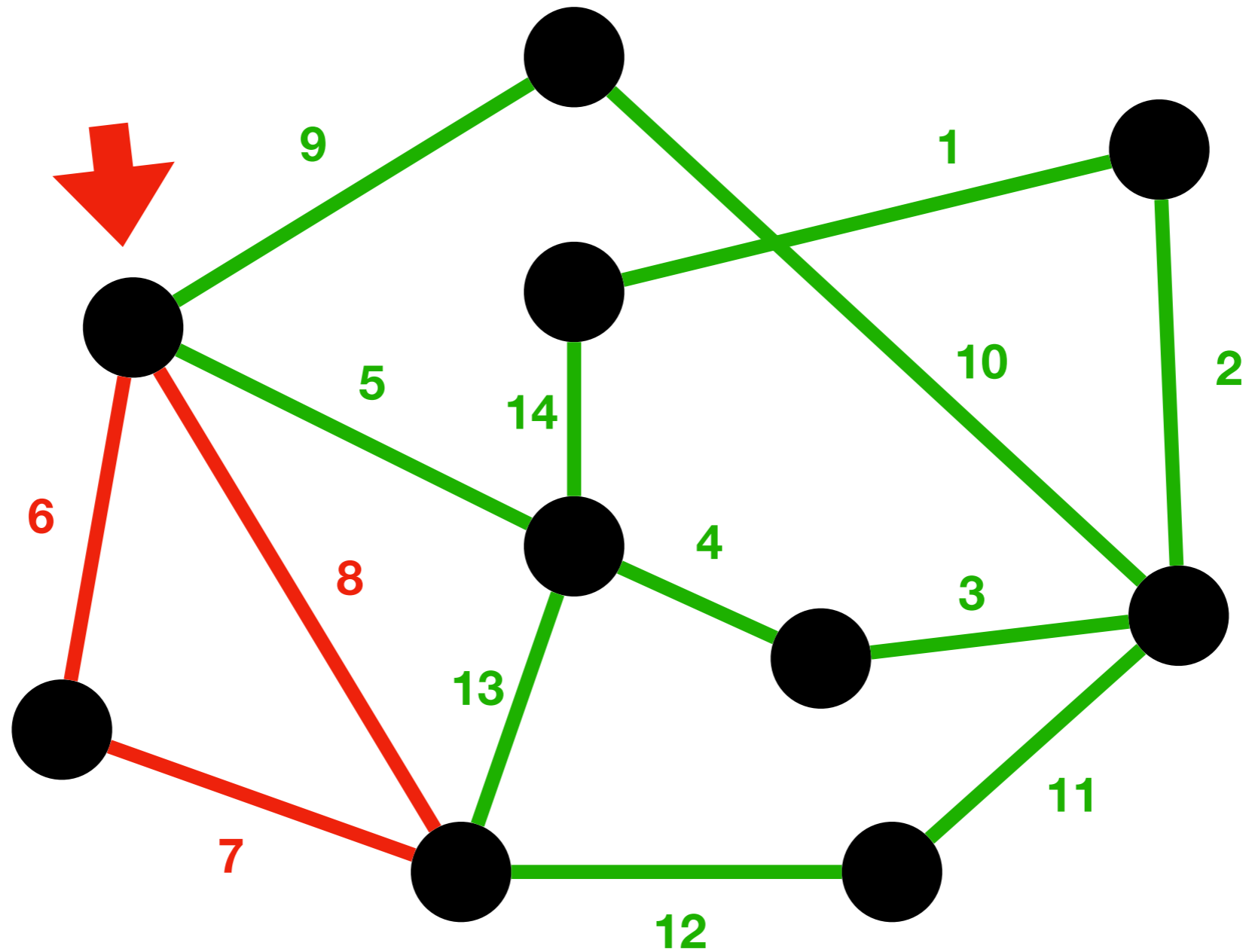
Algorithme de Hierholzer



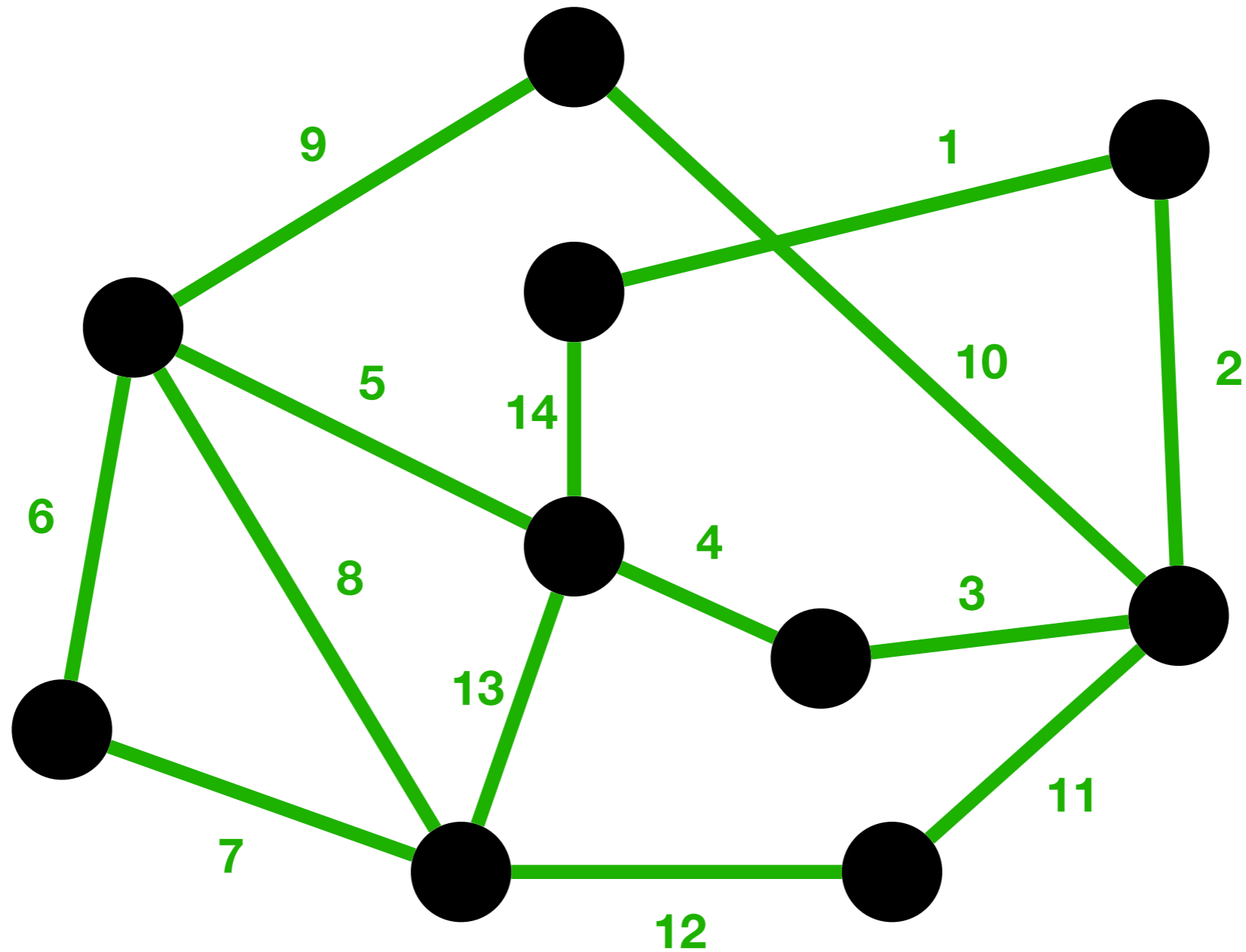
Algorithme de Hierholzer



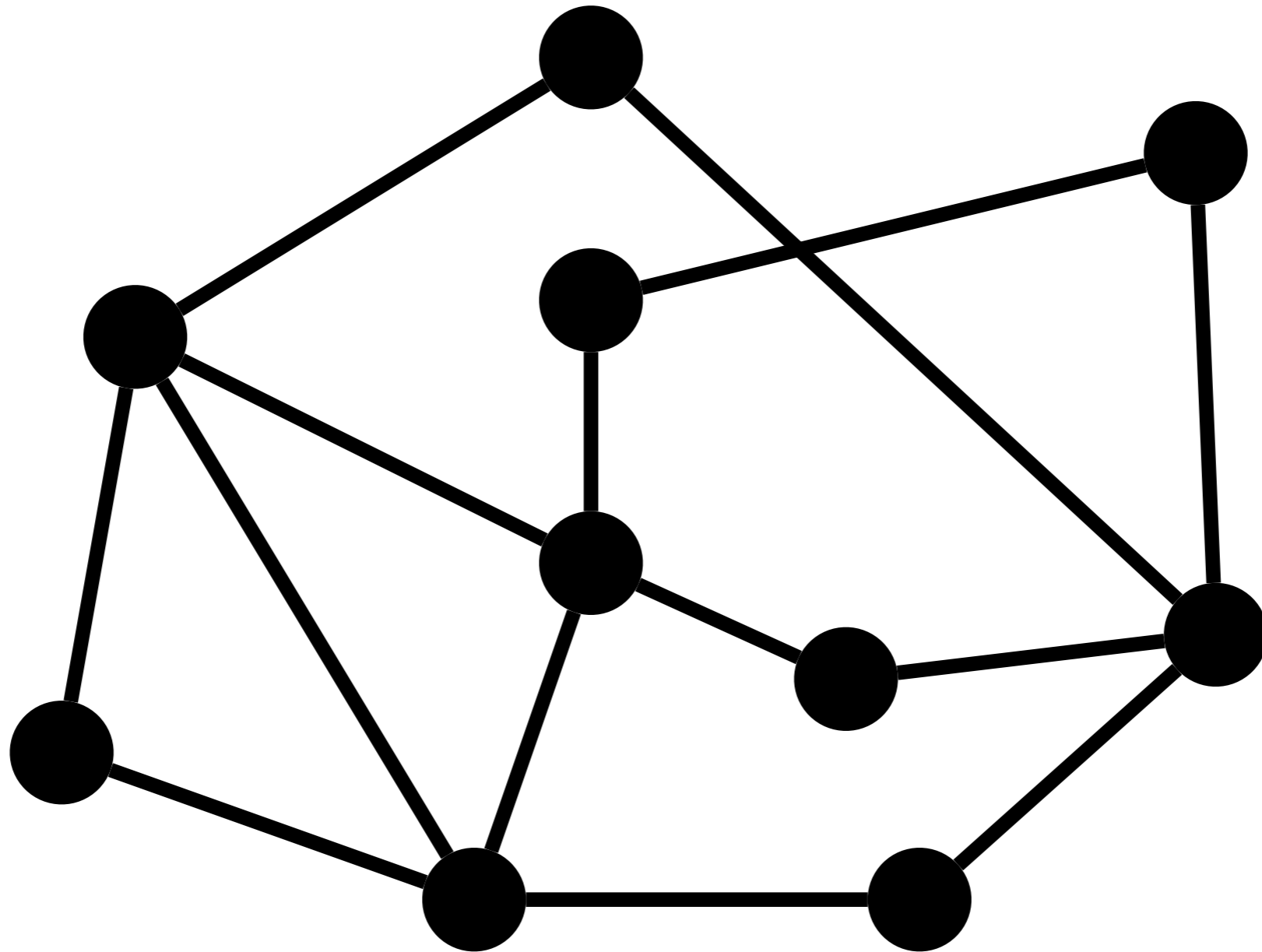
Algorithme de Hierholzer



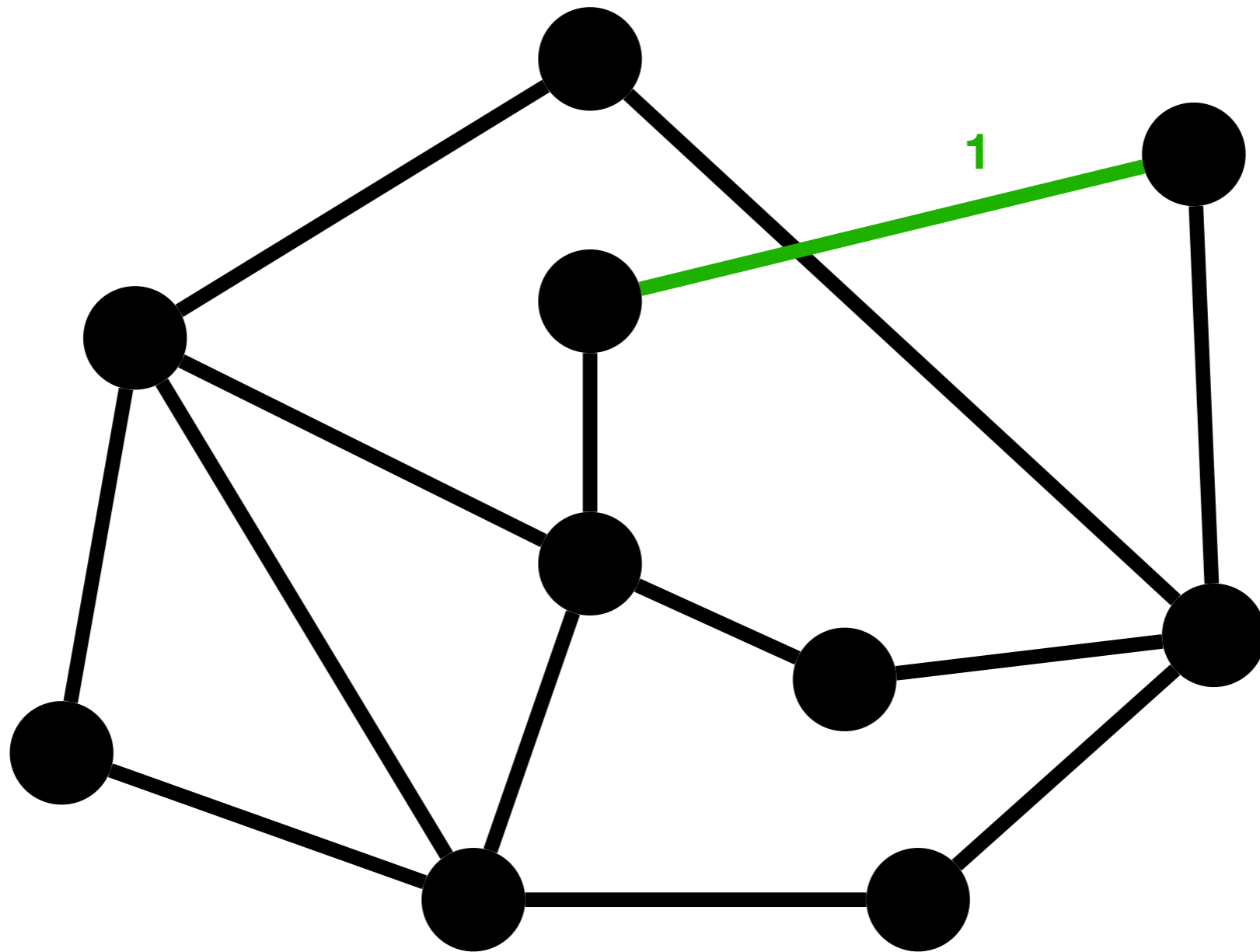
Algorithme de Hierholzer



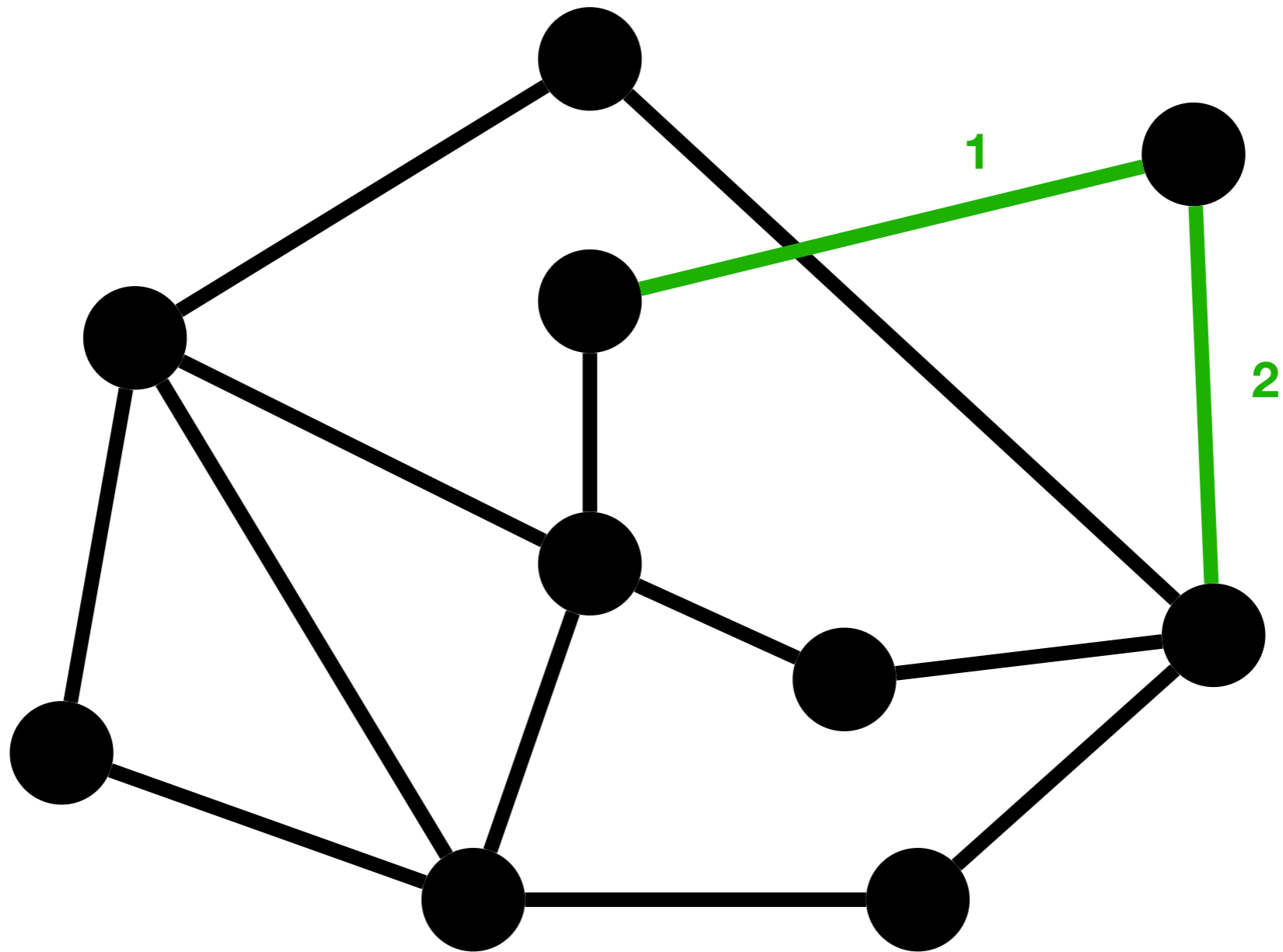
Algorithme de Hierholzer



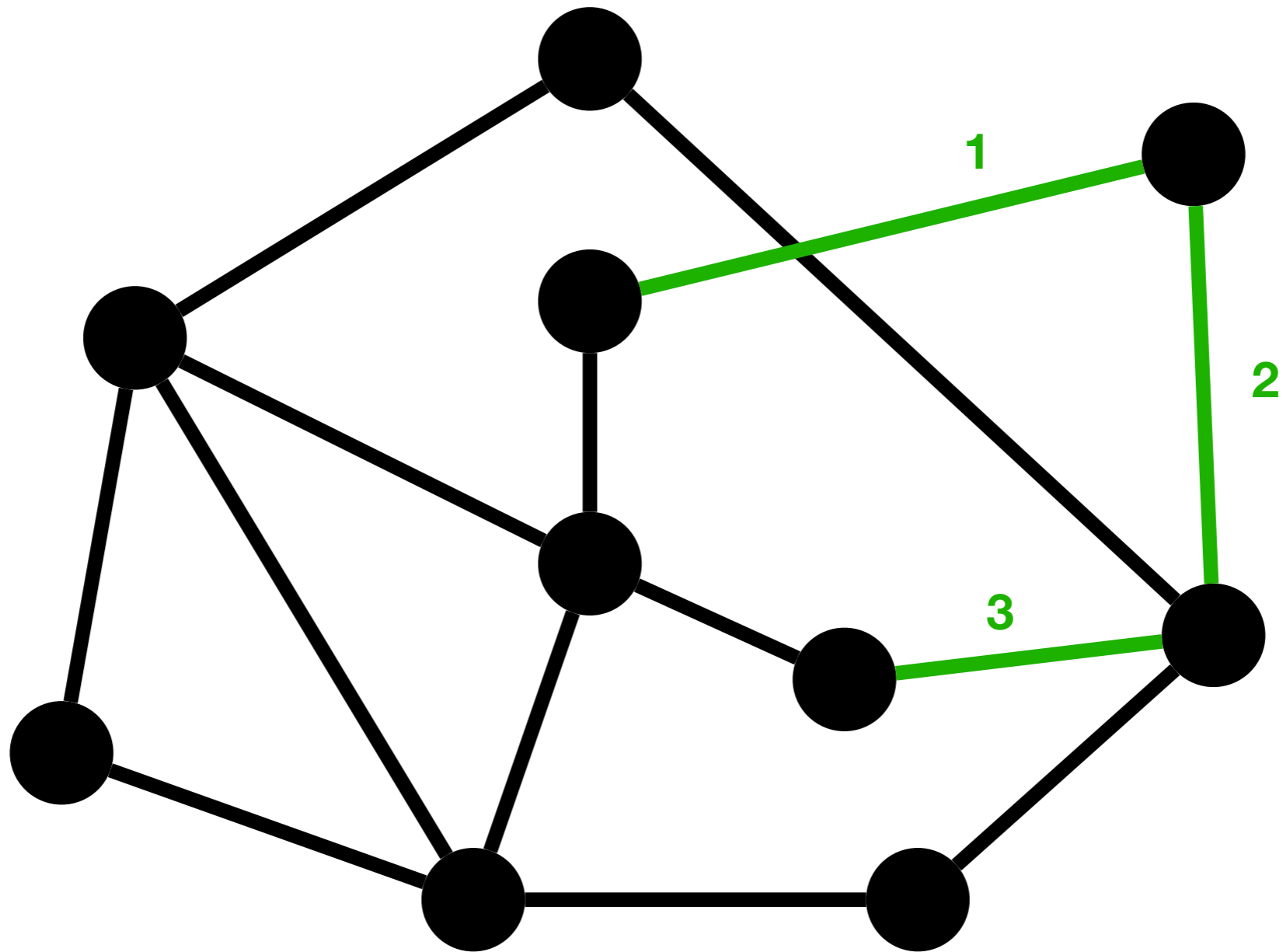
Algorithme de Hierholzer



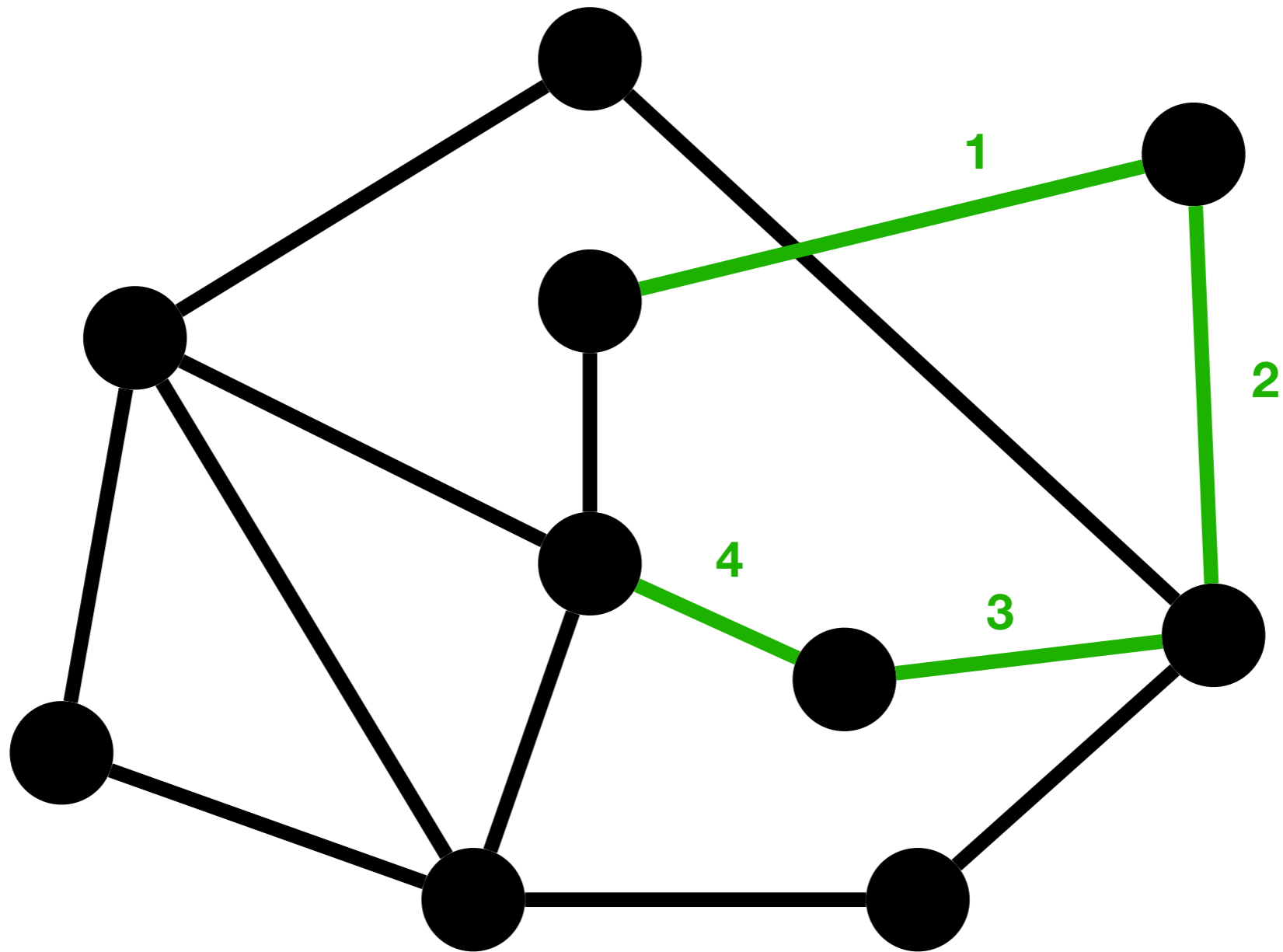
Algorithme de Hierholzer



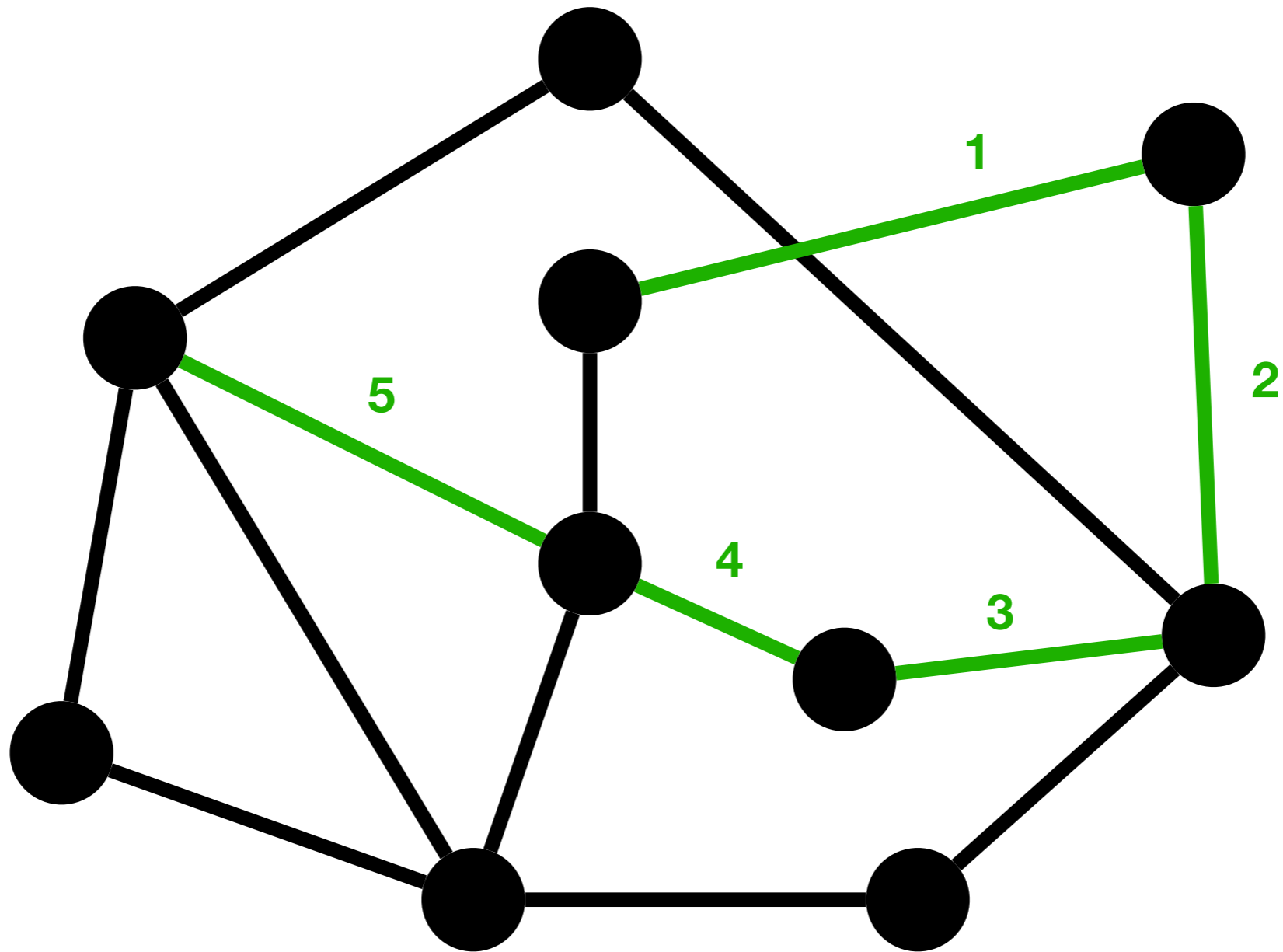
Algorithme de Hierholzer



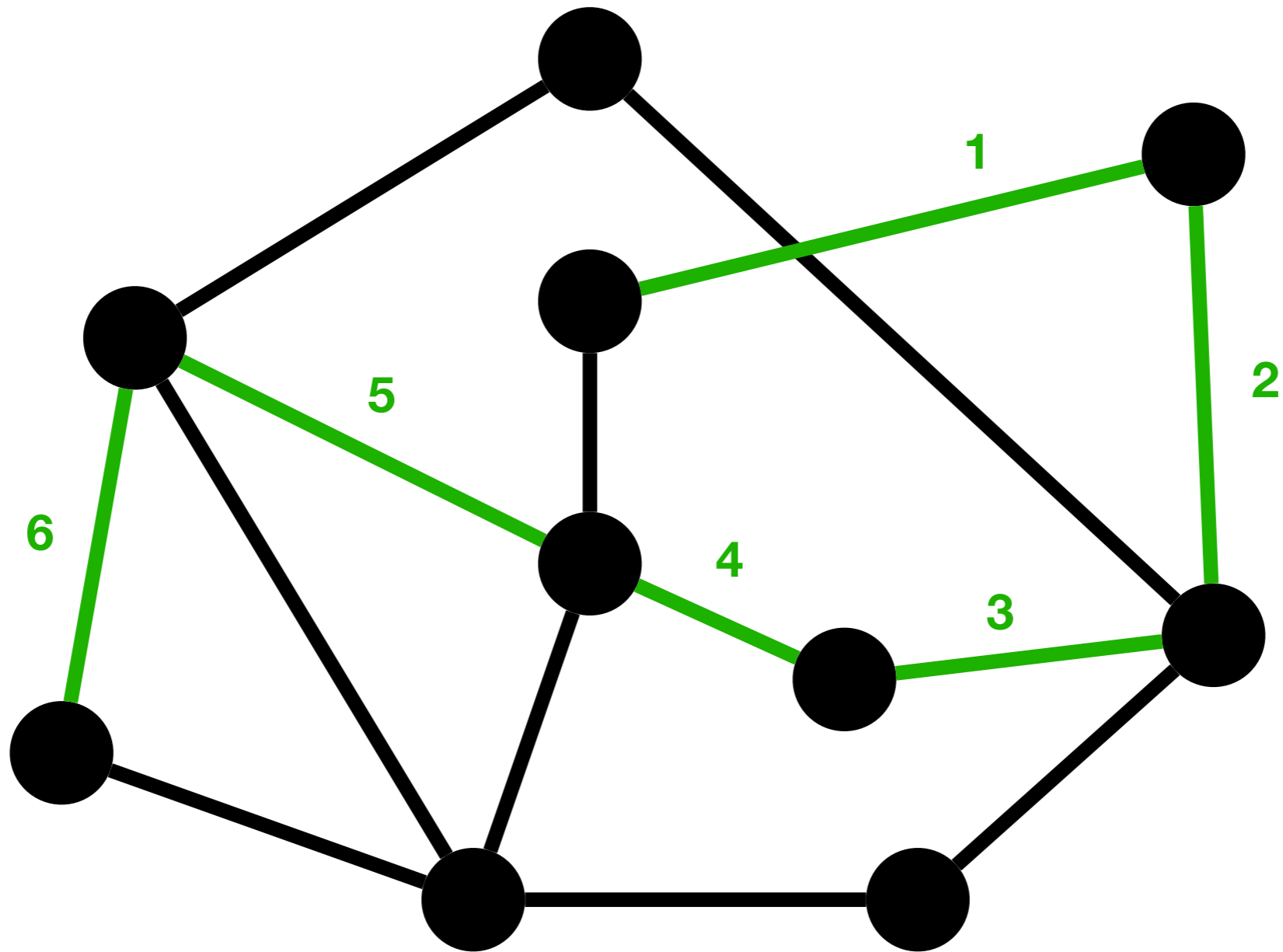
Algorithme de Hierholzer



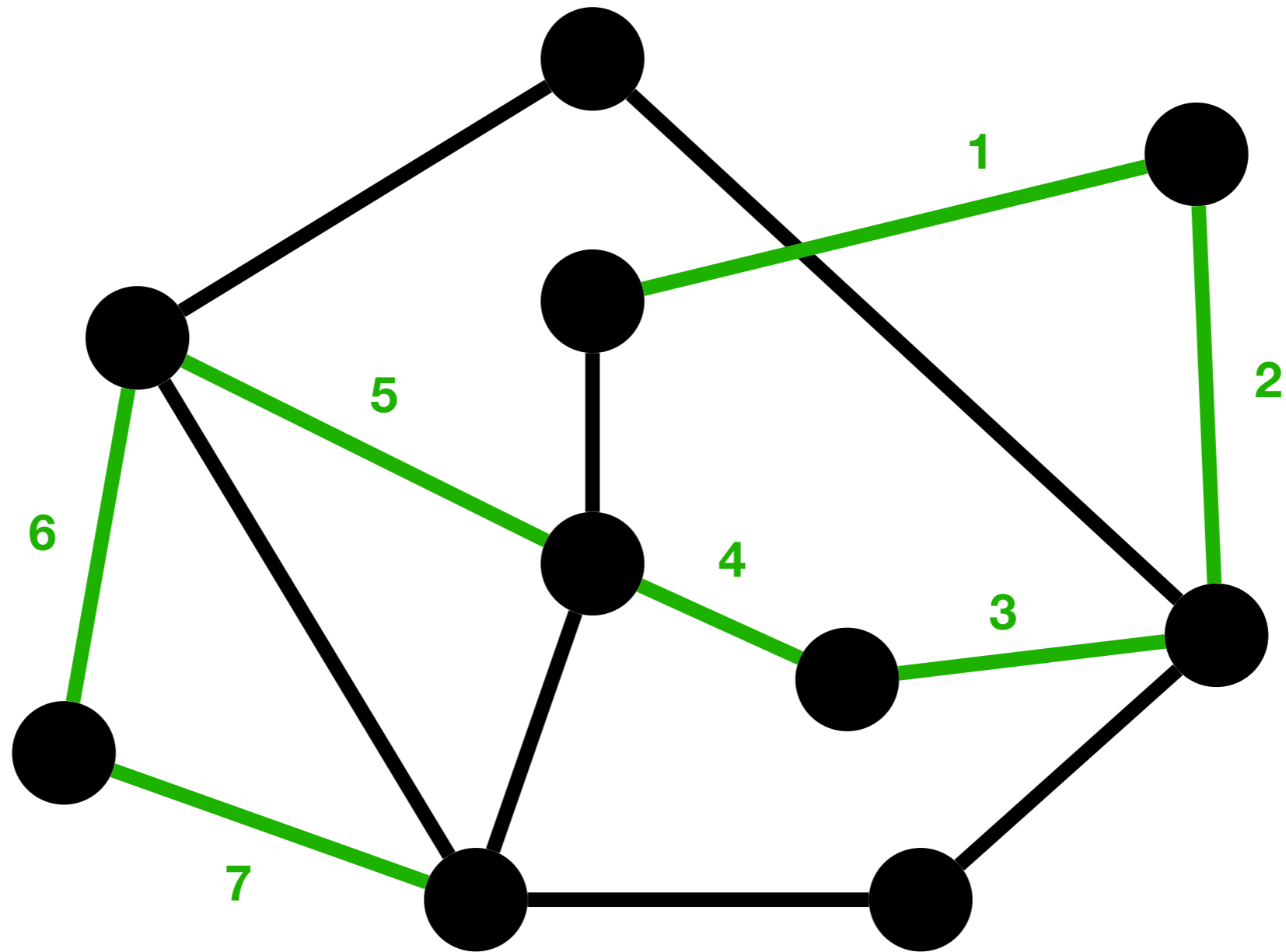
Algorithme de Hierholzer



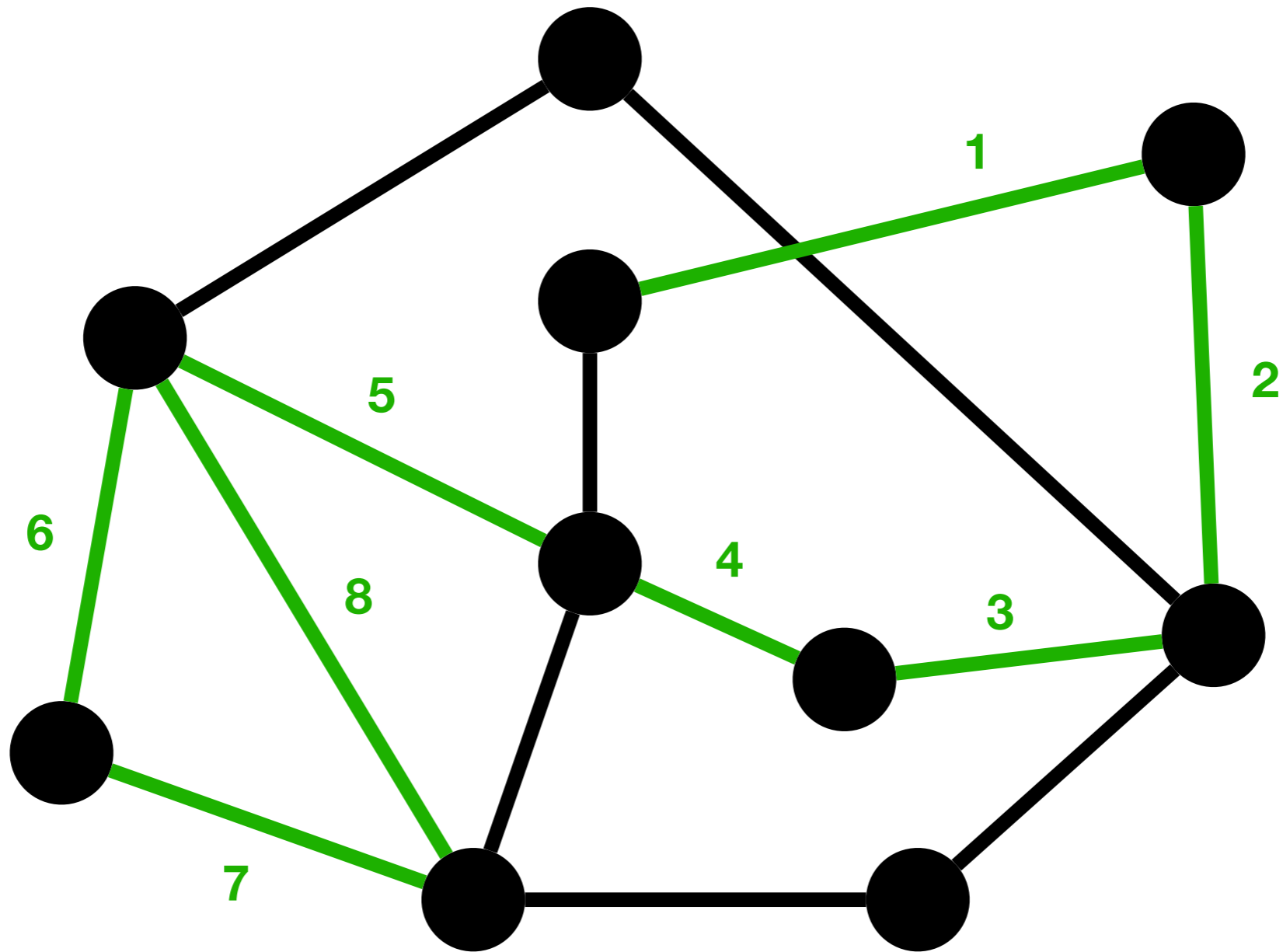
Algorithme de Hierholzer



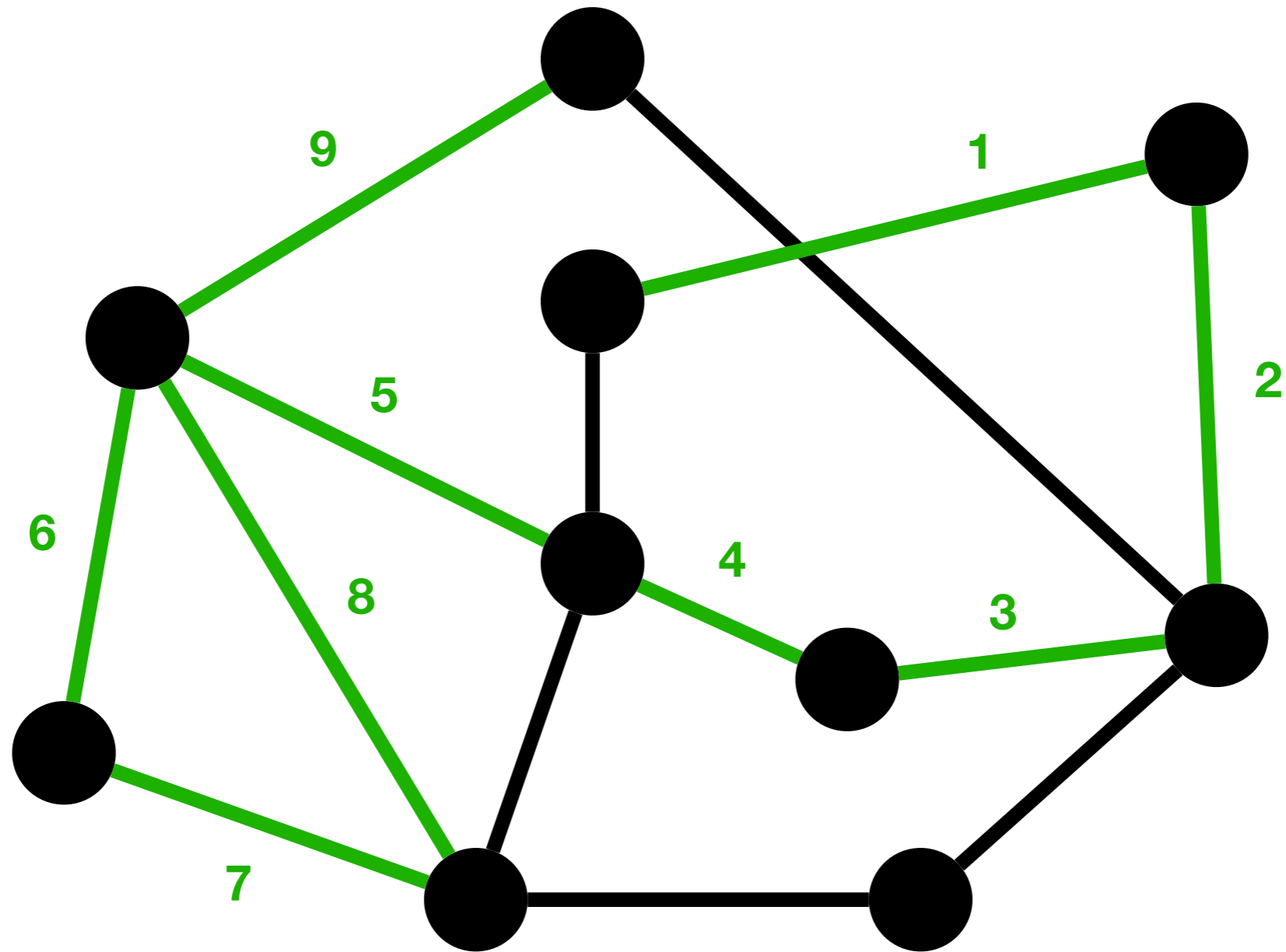
Algorithme de Hierholzer



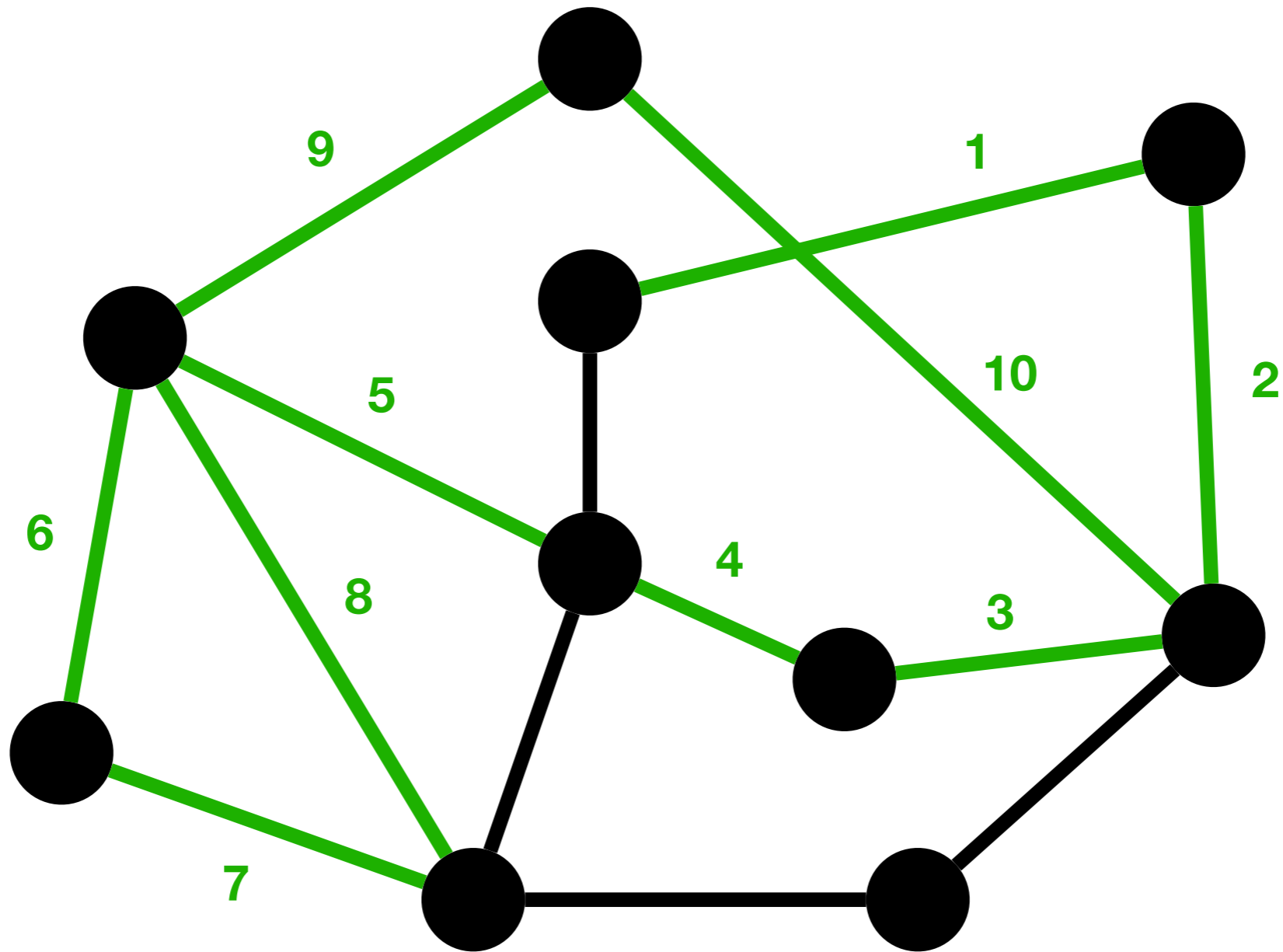
Algorithme de Hierholzer



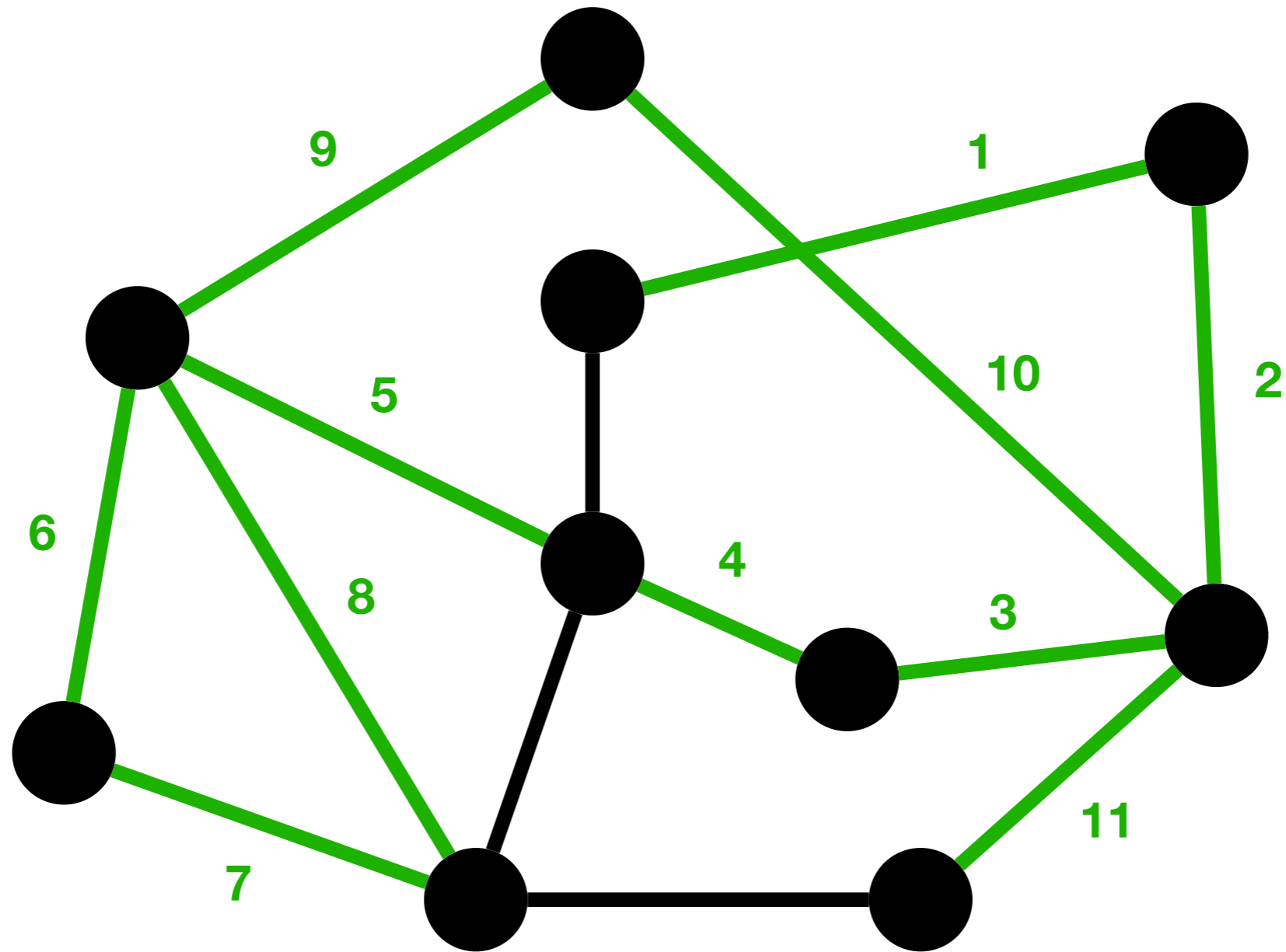
Algorithme de Hierholzer



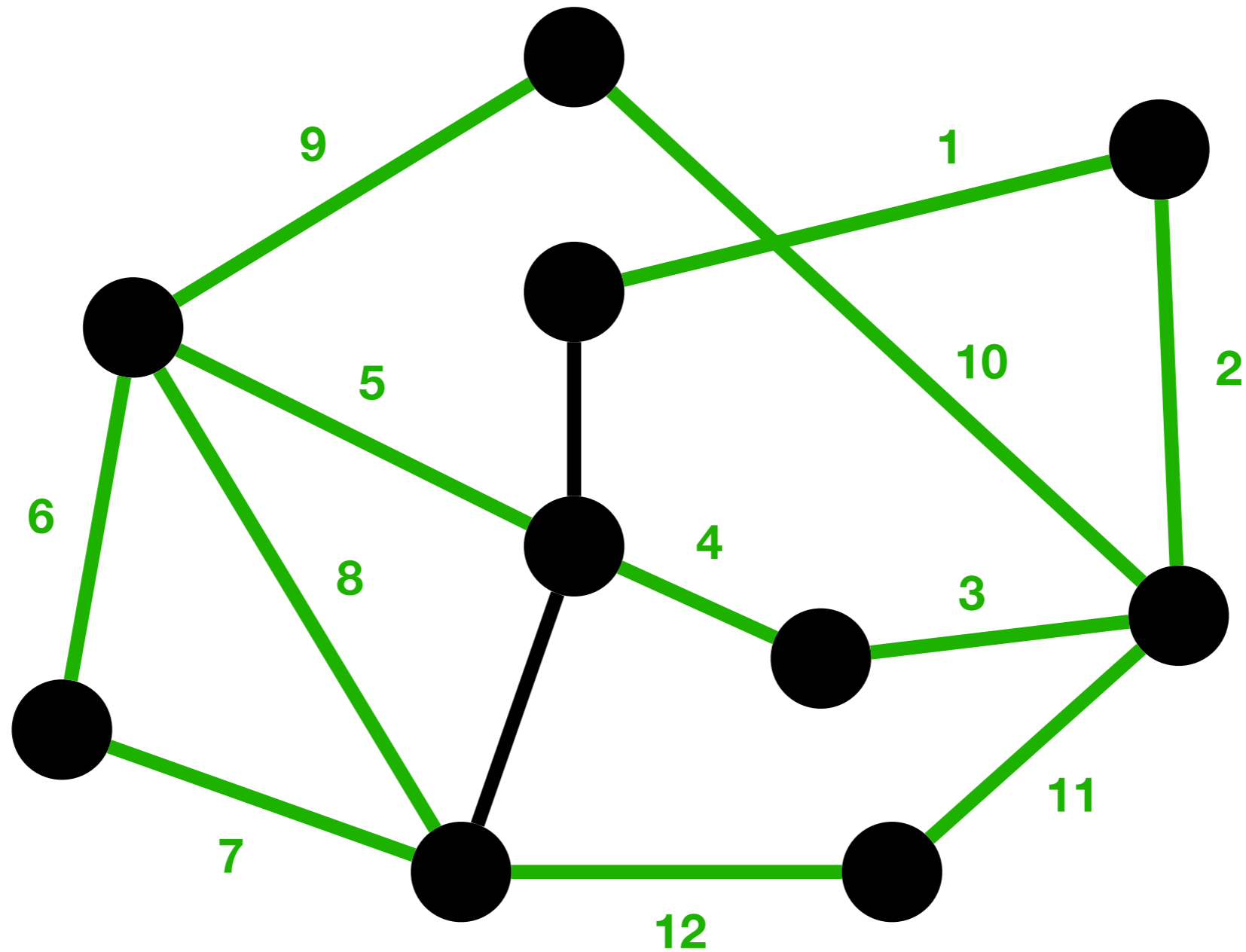
Algorithme de Hierholzer



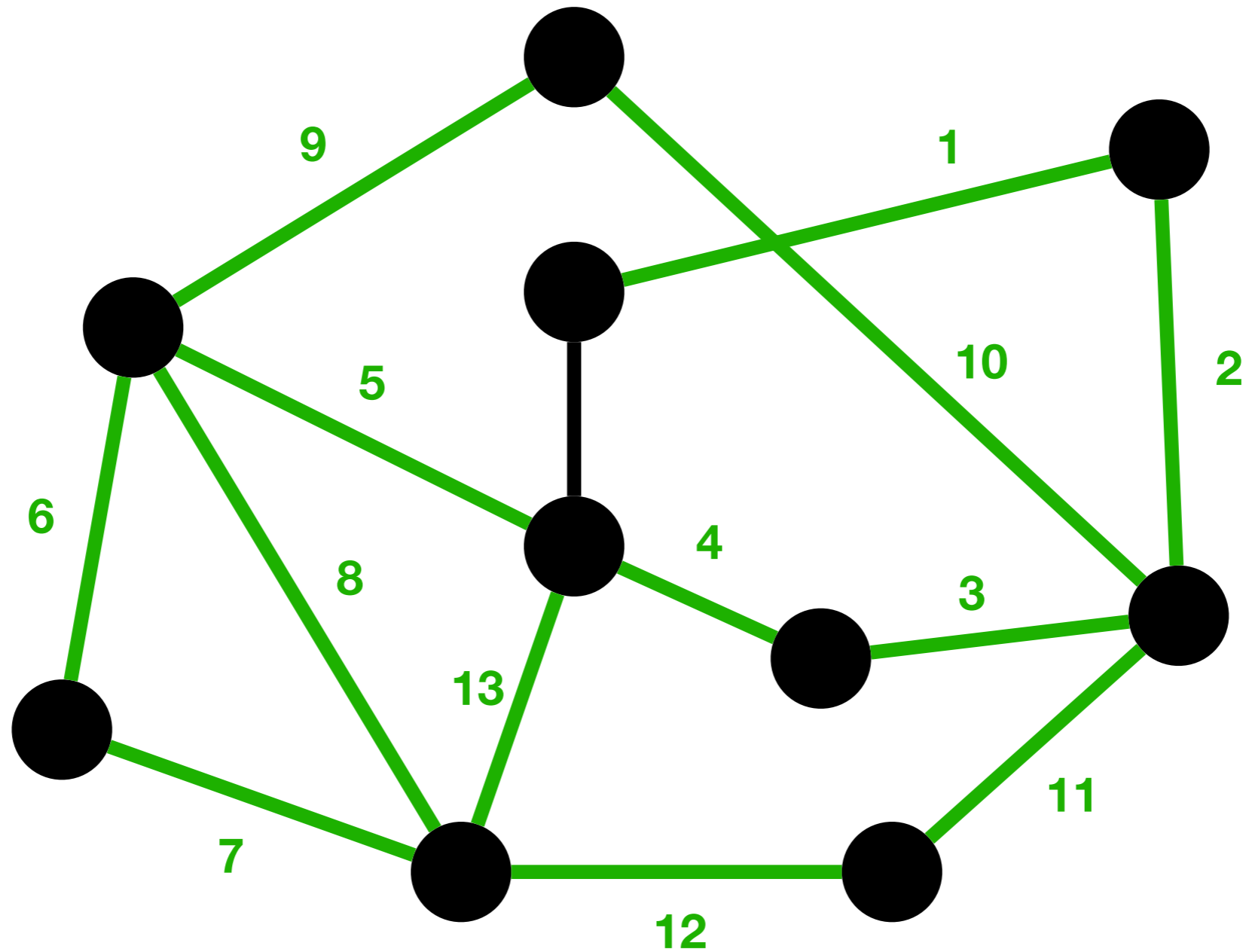
Algorithme de Hierholzer



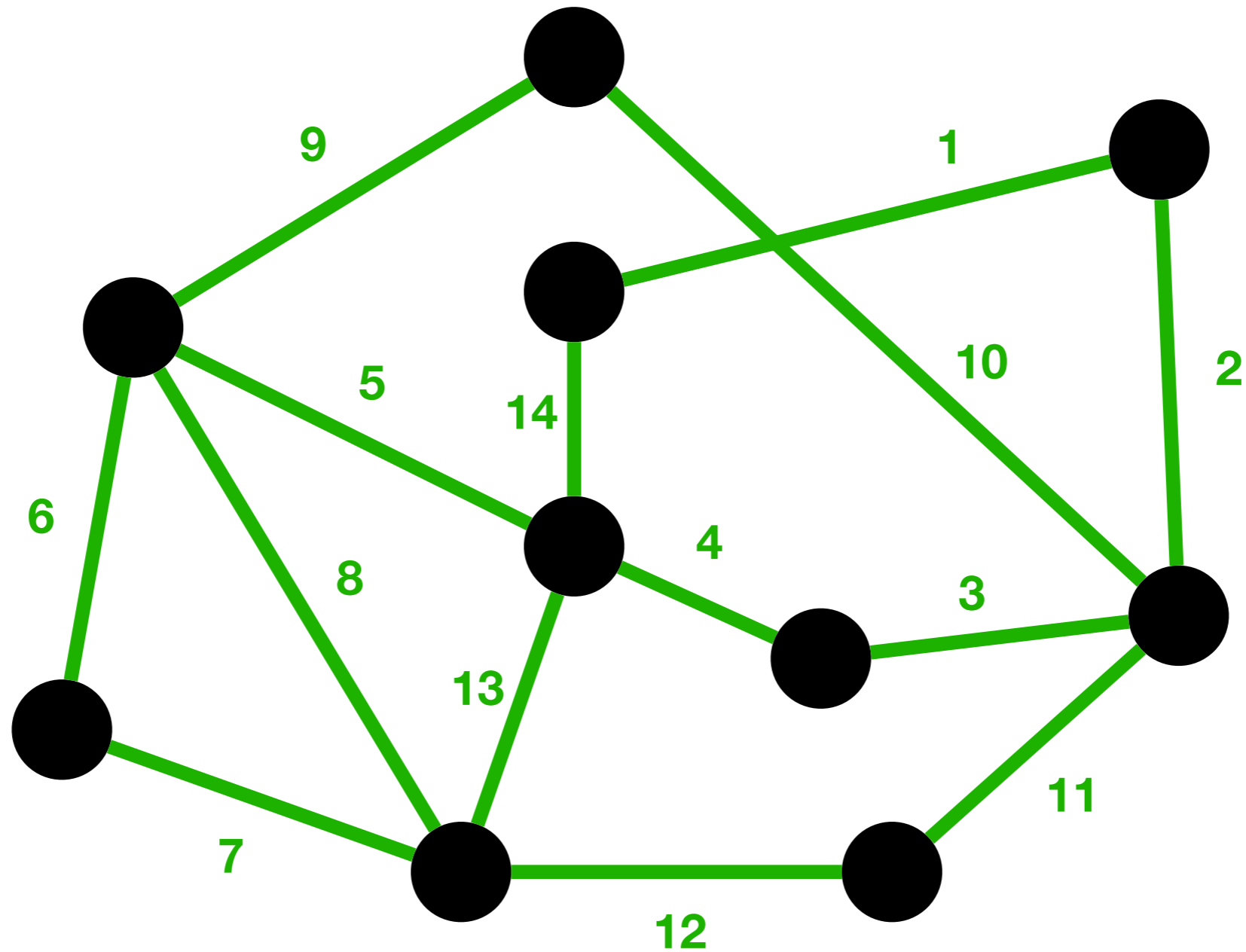
Algorithme de Hierholzer



Algorithme de Hierholzer



Algorithme de Hierholzer



Algorithme de Hierholzer

- Choisir n'importe quel sommet initial v
- Suivre un chemin arbitraire d'arêtes jusqu'à retourner à v , en traversant chaque arête une fois au plus, obtenant ainsi un cycle partiel c
- **Tant qu'il y a des sommets u dans le cycle c avec des arêtes qu'on n'a pas encore choisi **faire****
 - Suivre un chemin de sommets à partir de u jusqu'à retourner à u , obtenant un cycle c'
 - Prolonger le cycle c par c'

Terminaison et correction de l'algorithme de Hierholzer

Terminaison et correction de l'algorithme de Hierholzer

- « Suivre un chemin arbitraire d'arêtes jusqu'à retourner à v , en traversant chaque arête une fois au plus, obtenant ainsi un cycle partiel c »

Terminaison et correction de l'algorithme de Hierholzer

- « Suivre un chemin arbitraire d'arêtes jusqu'à retourner à v , en traversant chaque arête une fois au plus, obtenant ainsi un cycle partiel c »
- On ne peut pas rester bloqué dans aucun sommet différent de v , puisque ils ont toujours un nombre pair d'arêtes pas encore explorées (« si on entre dans un sommet, on peut toujours en sortir »)

Terminaison et correction de l'algorithme de Hierholzer

- « Suivre un chemin arbitraire d'arêtes jusqu'à retourner à v , en traversant chaque arête une fois au plus, obtenant ainsi un cycle partiel c »
- On ne peut pas rester bloqué dans aucun sommet différent de v , puisque ils ont toujours un nombre pair d'arêtes pas encore explorées (« si on entre dans un sommet, on peut toujours en sortir »)
- Et, tôt ou tard, on retourne à v , puisque dans le pire des cas on épuise l'ensemble des autres arêtes, et v a un nombre **impair** d'arêtes inexplorées (donc il reste une arête pour y entrer)

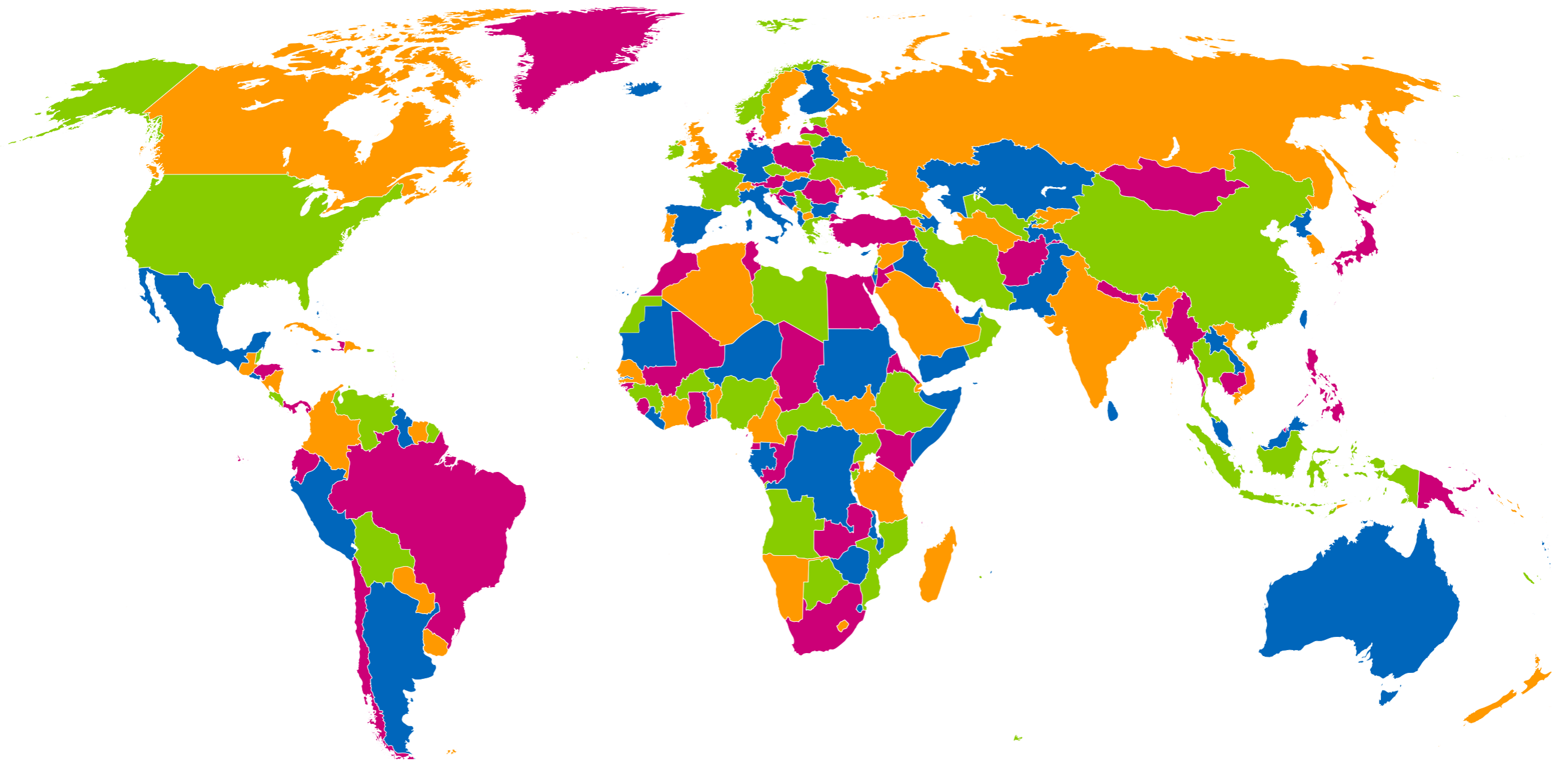
Terminaison et correction de l'algorithme de Hierholzer

- « Suivre un chemin arbitraire d'arêtes jusqu'à retourner à v , en traversant chaque arête une fois au plus, obtenant ainsi un cycle partiel c »
- On ne peut pas rester bloqué dans aucun sommet différent de v , puisque ils ont toujours un nombre pair d'arêtes pas encore explorées (« si on entre dans un sommet, on peut toujours en sortir »)
- Et, tôt ou tard, on retourne à v , puisque dans le pire des cas on épuise l'ensemble des autres arêtes, et v a un nombre **impair** d'arêtes inexplorées (donc il reste une arête pour y entrer)
- Ça est également vrai pour chaque sommet u choisi après v

Exercice 2 du TD4

Coloration de graphes non orientés

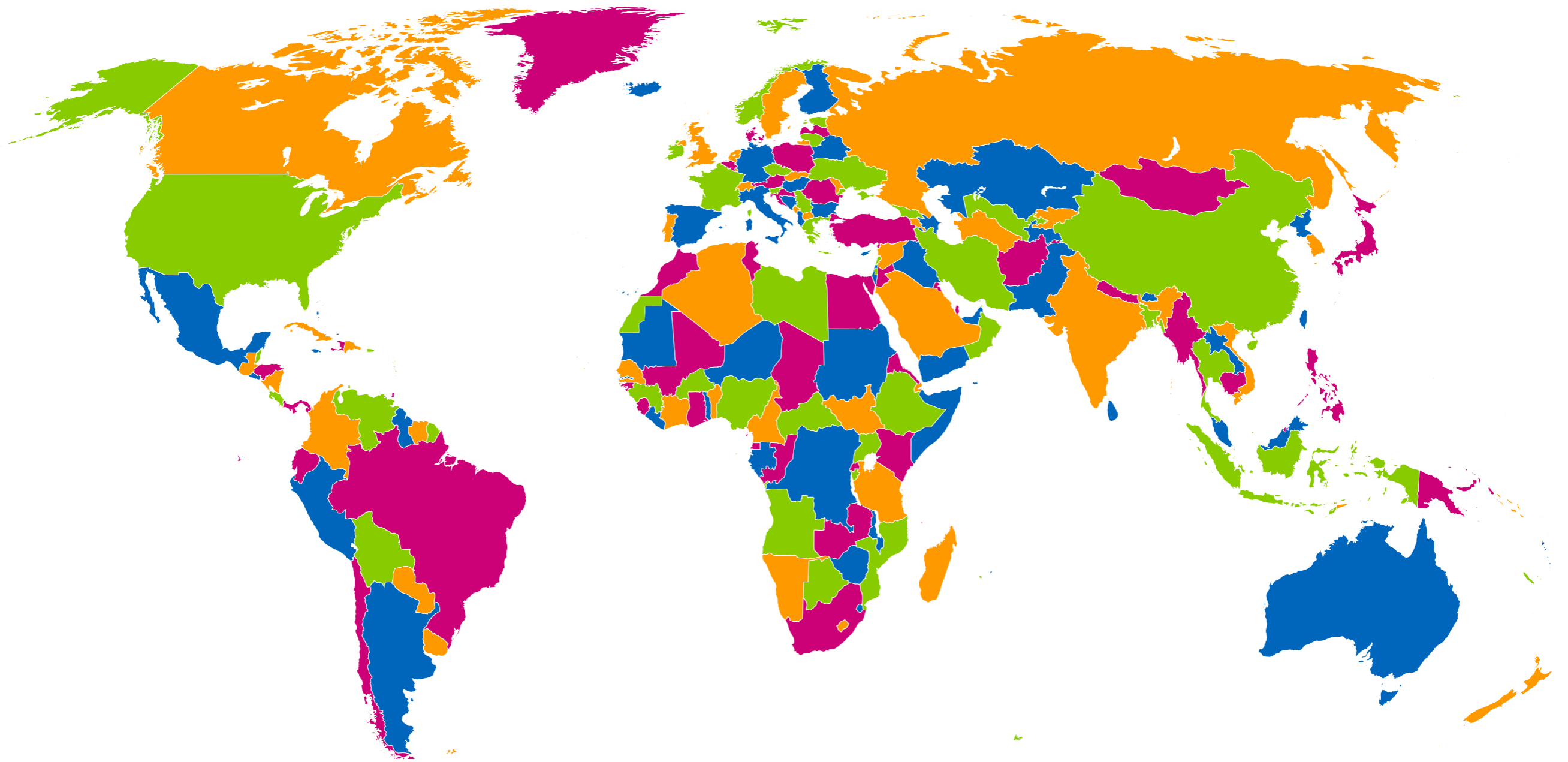
Coloration de cartes



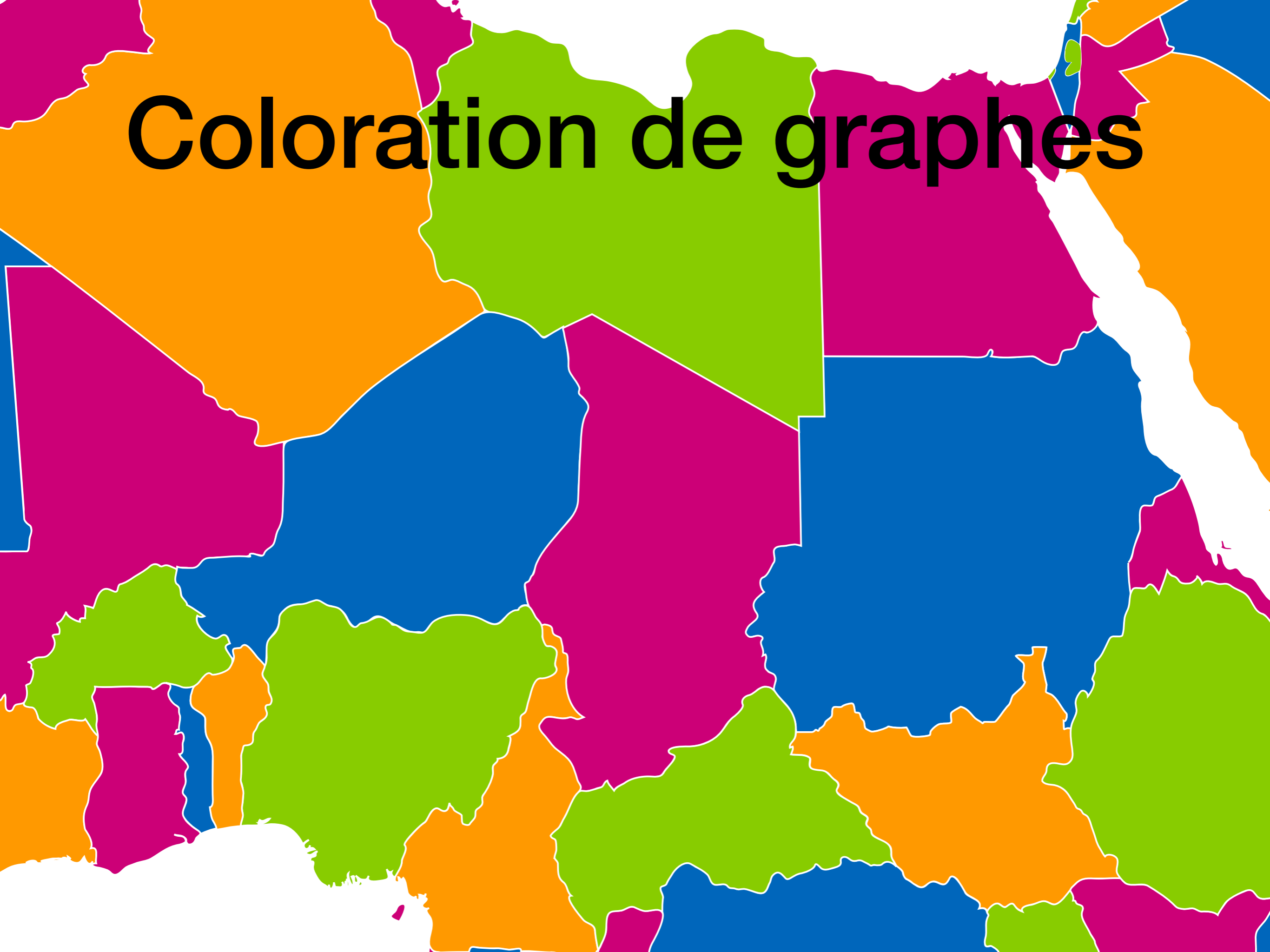
Théorème des quatre couleurs

On peut colorier n'importe quelle carte
avec un maximum de 4 couleurs
de sorte que les pays (régions, etc.) adjacents
reçoivent toujours deux couleurs distinctes

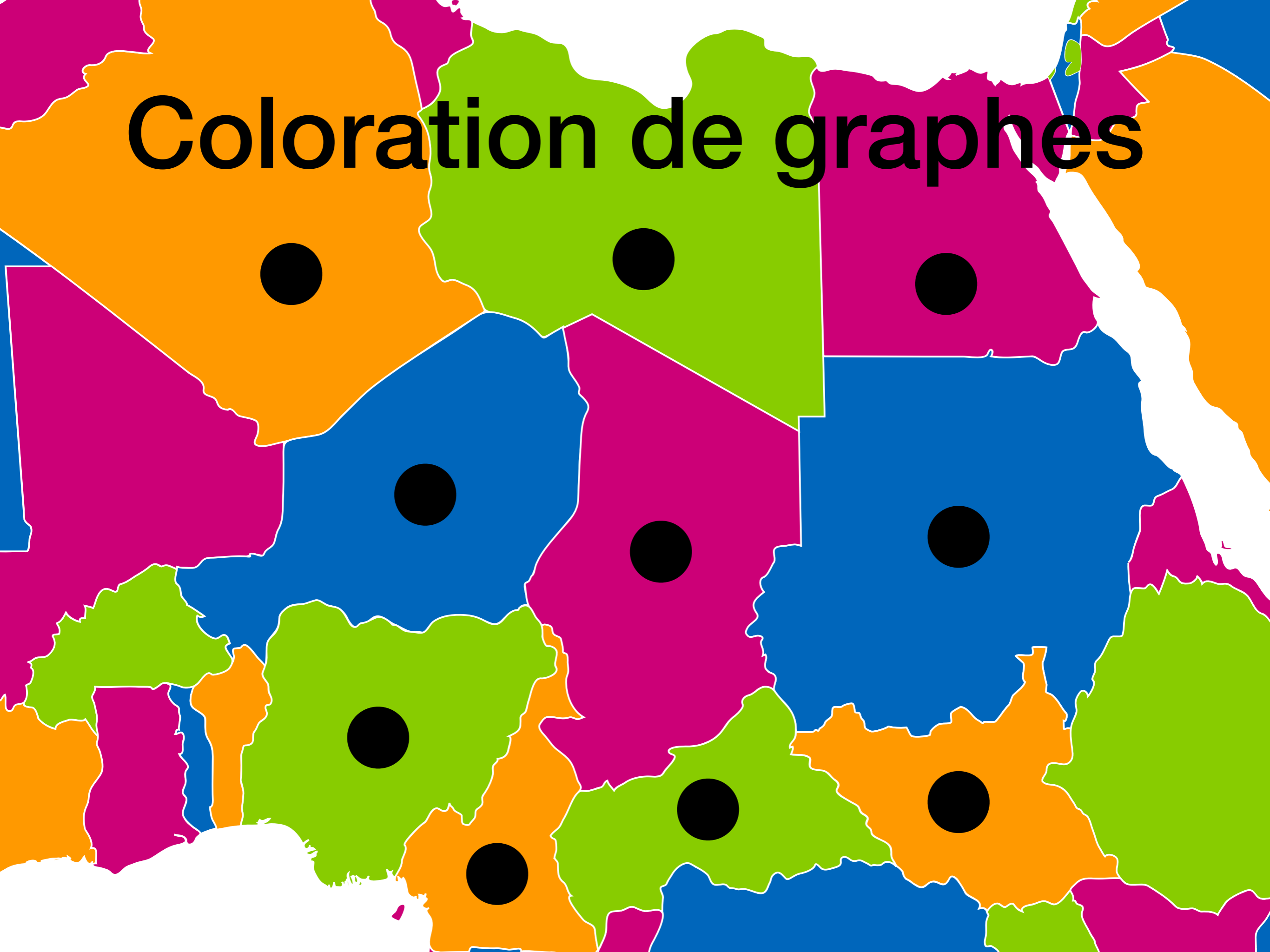
Coloration de graphes



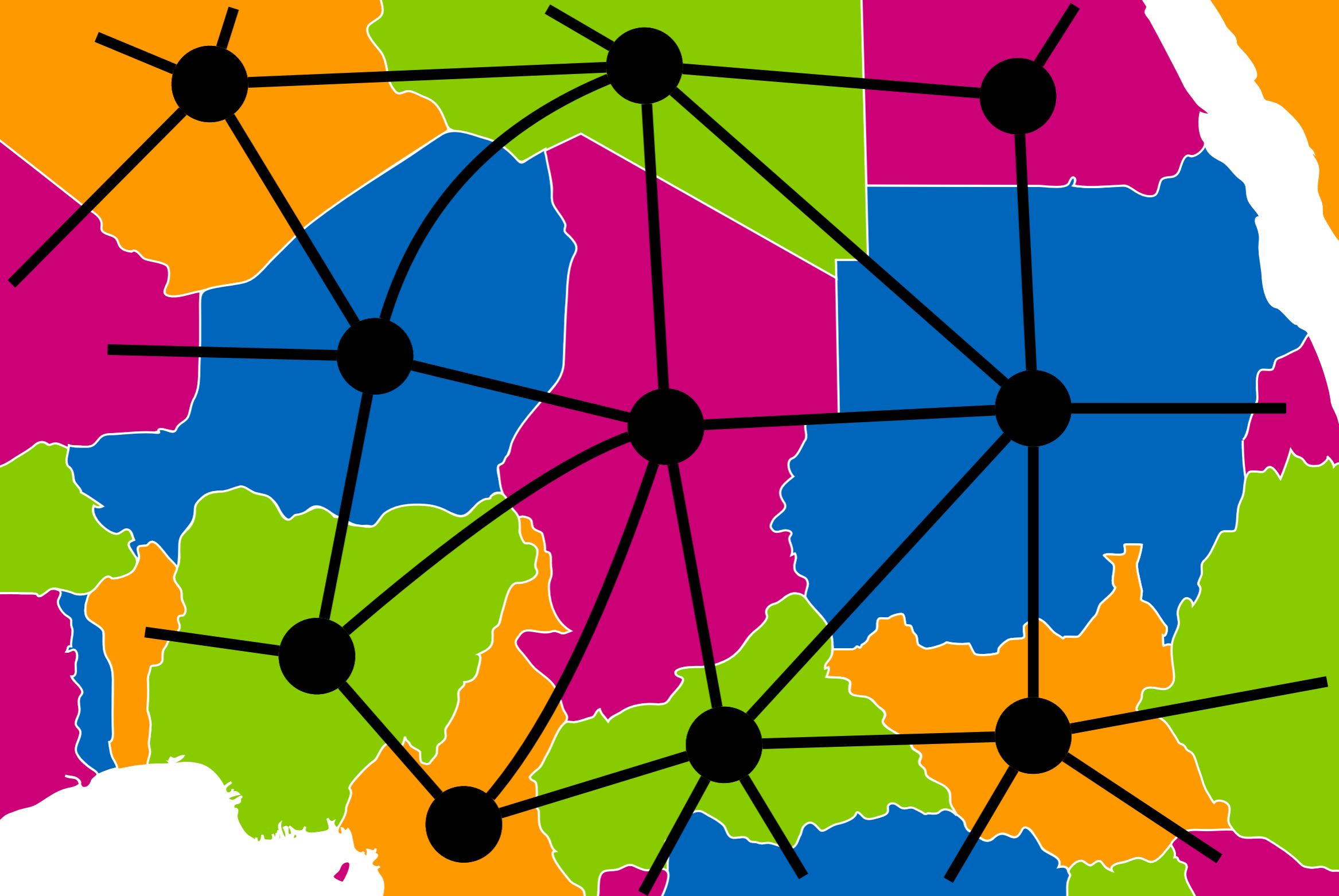
Coloration de graphes



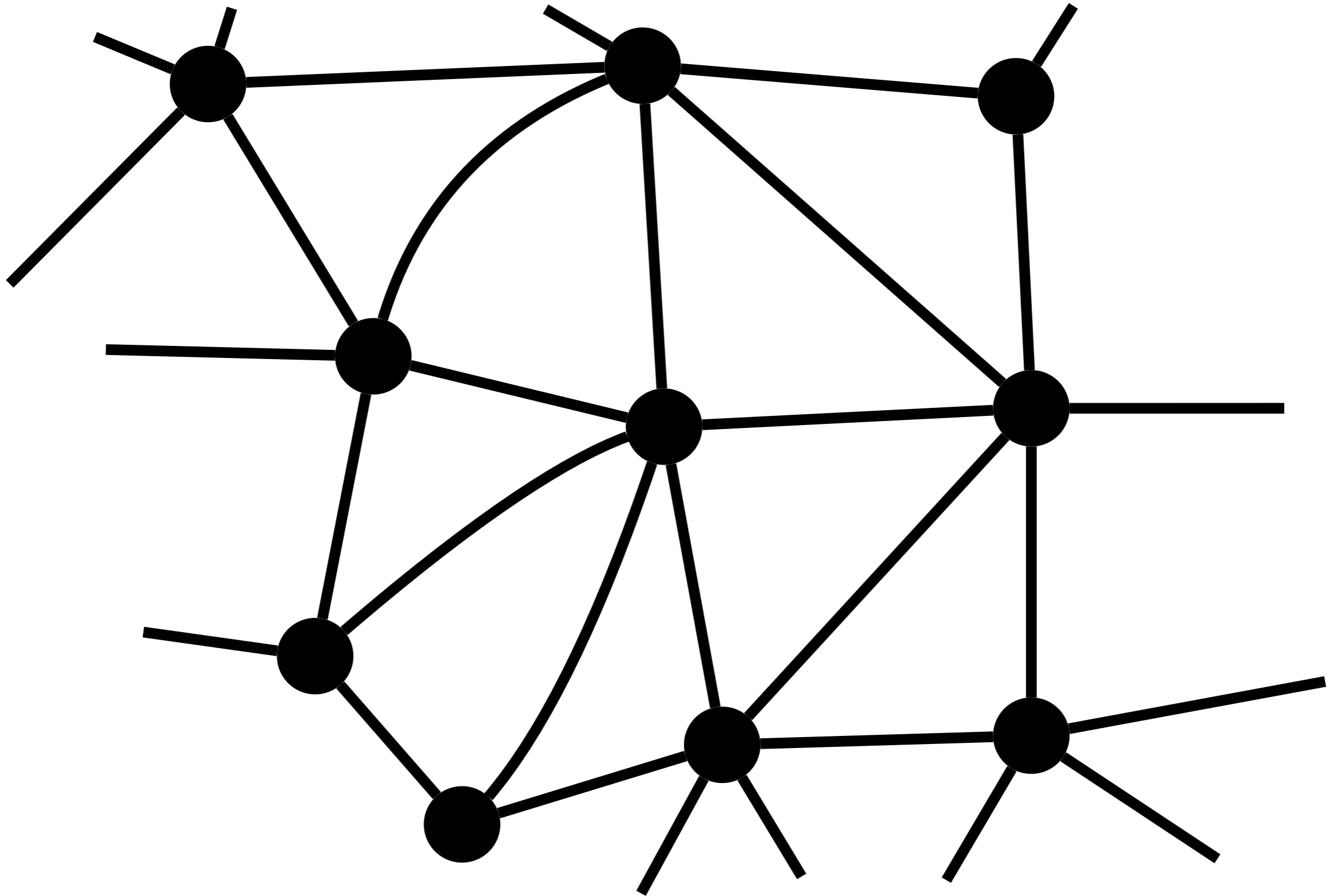
Coloration de graphes



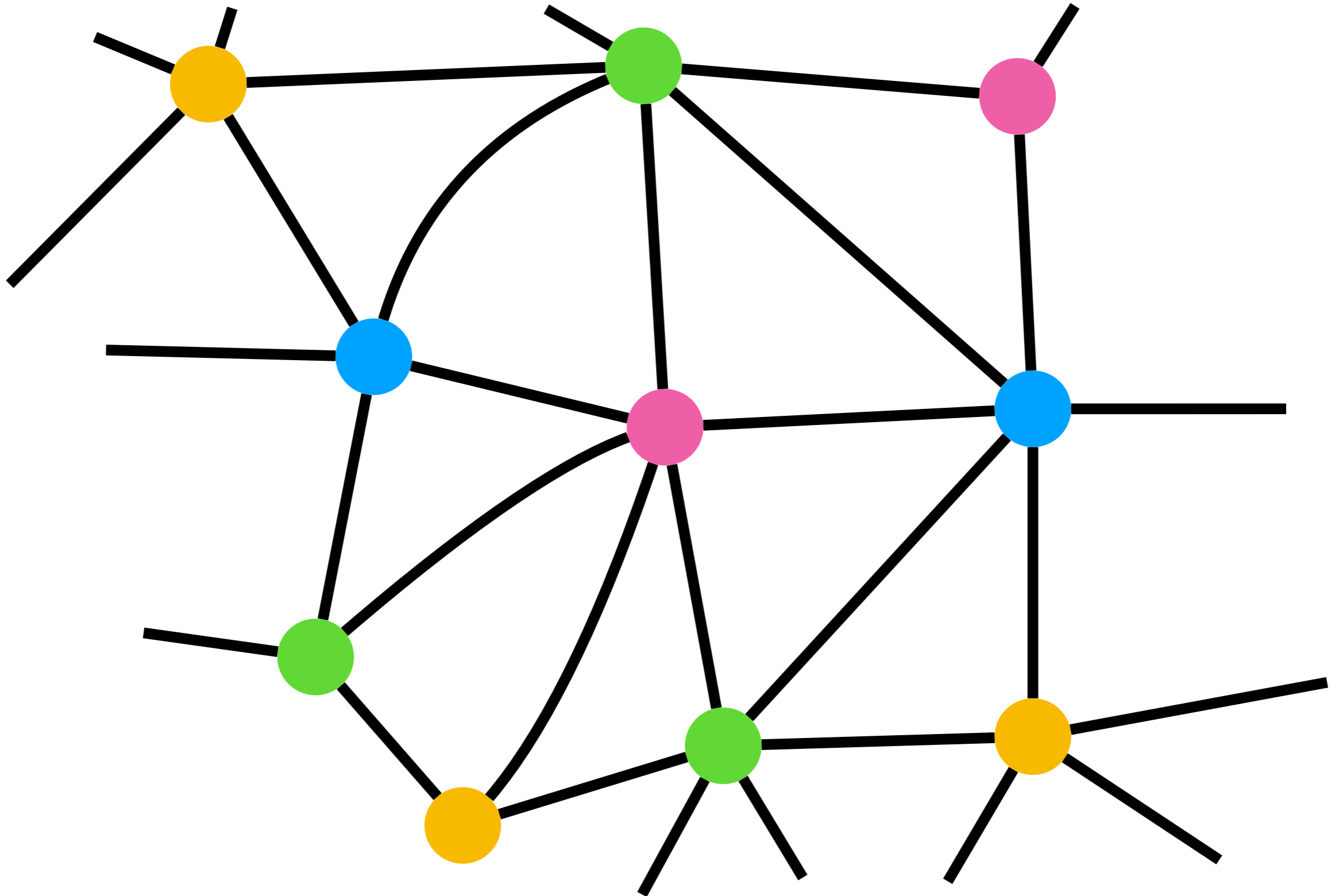
Coloration de graphes



Coloration de graphes



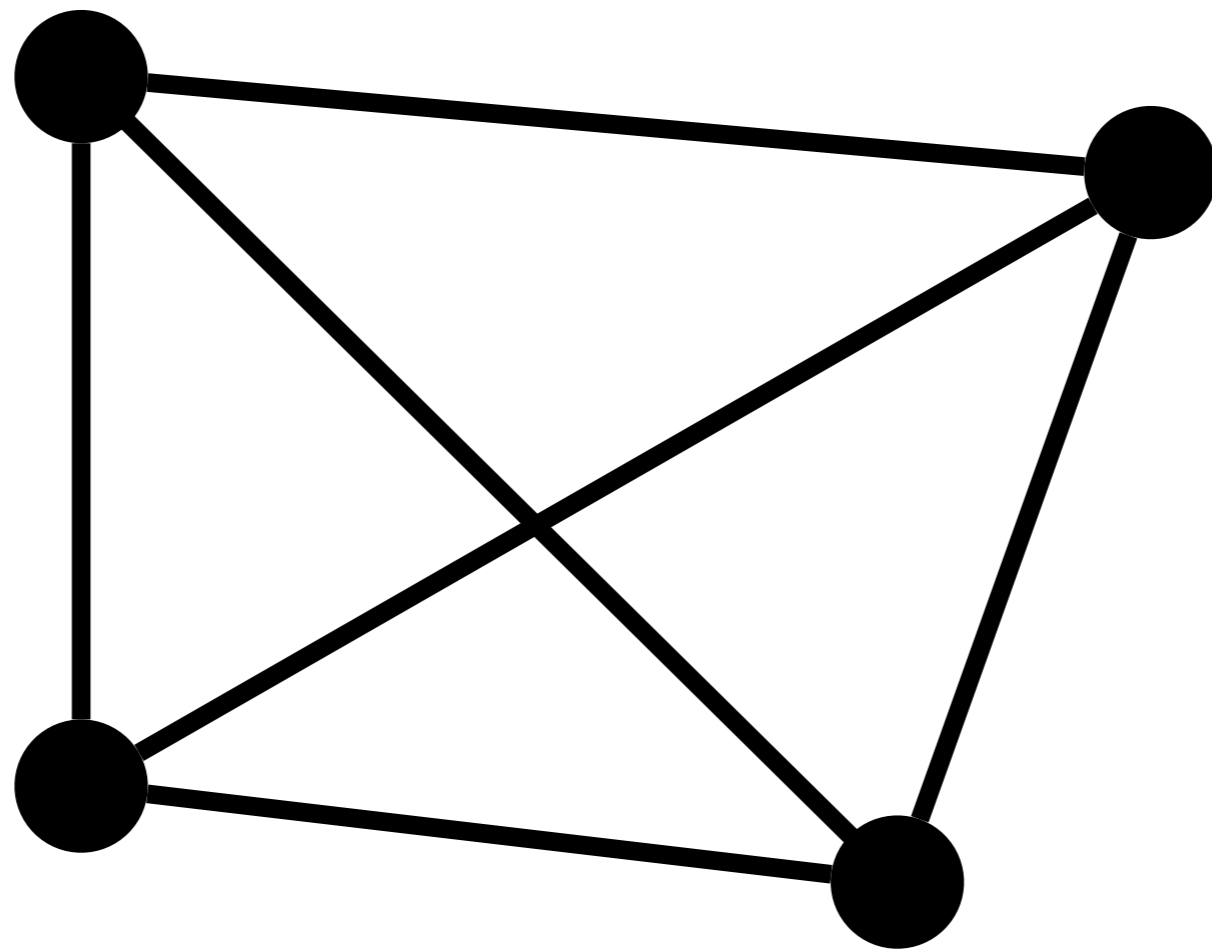
Coloration de graphes



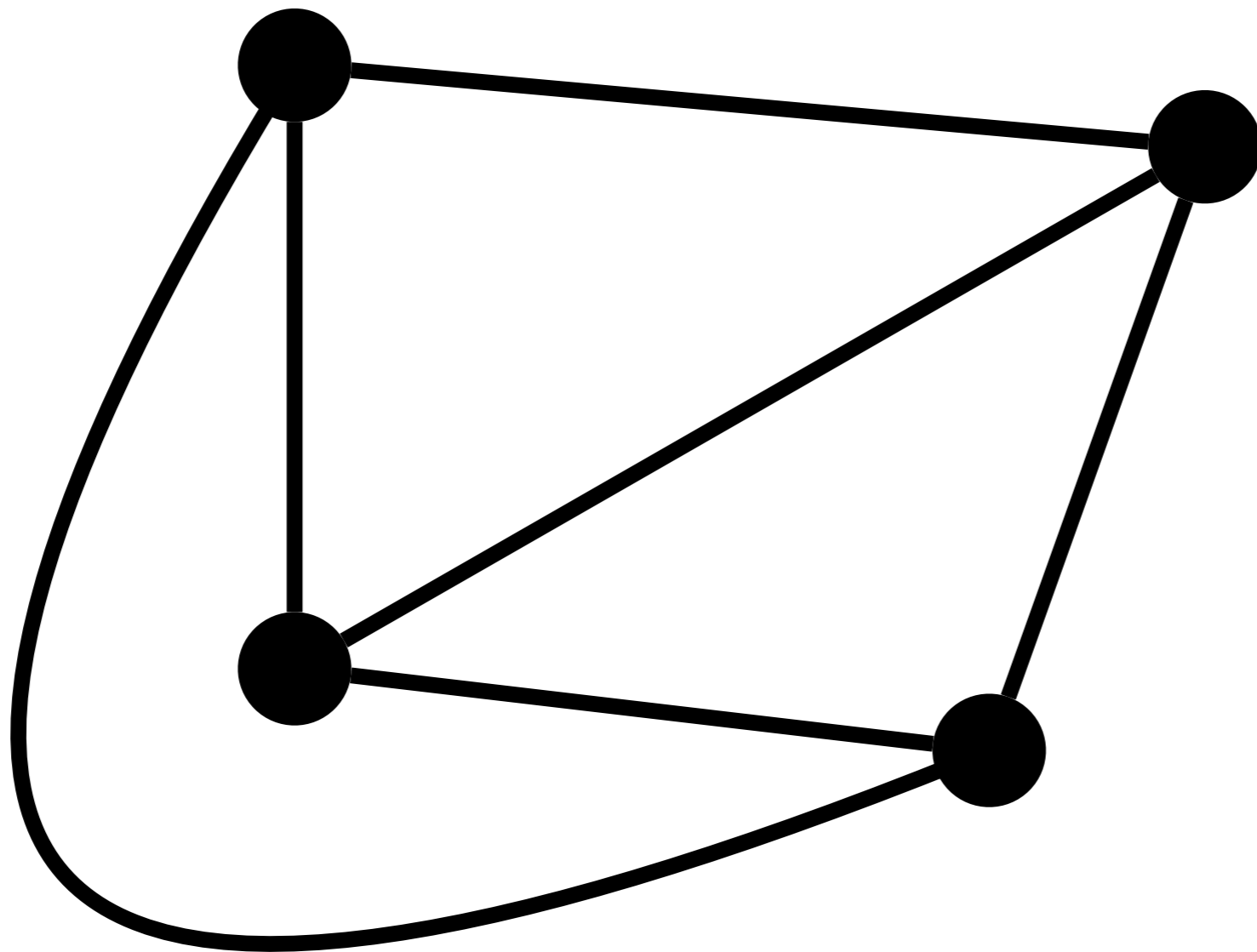
Graphes planaires

- Un **graphe** (non orienté) **planaire** est un graphe qu'on peut dessiner sur un plan (une feuille de papier) sans qu'aucune arête n'en croise une autre
- Un graphe non orienté G est planaire si et seulement si il existe une carte ayant G comme graphe associé

Graphes qui ne semblent pas planaires



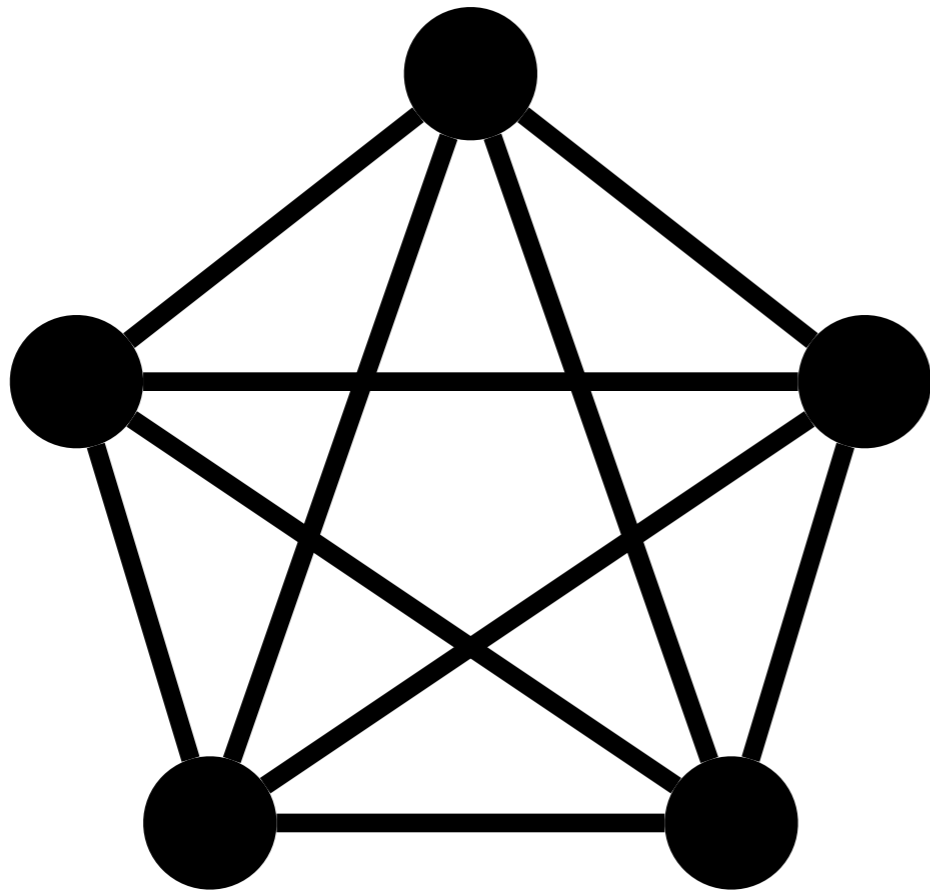
Graphes qui ne semblent pas planaires



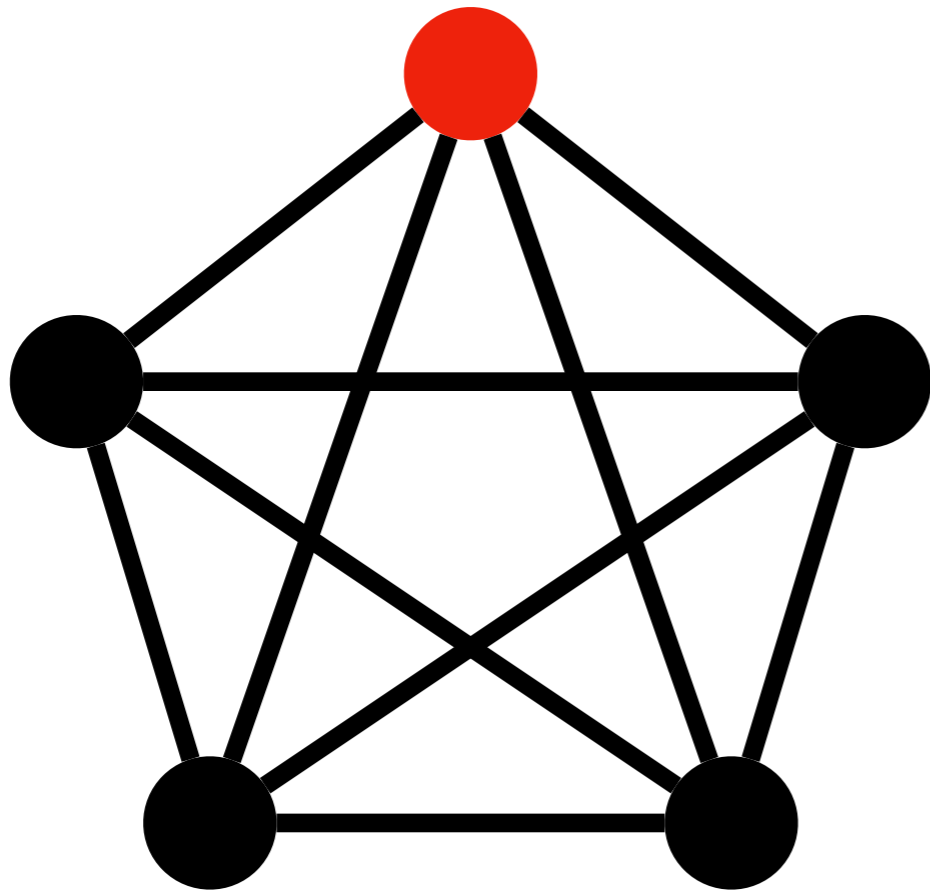
Théorème des quatre couleurs

On peut colorier les sommets de n'importe quel graphe non orienté planaire avec **un maximum de 4 couleurs** de sorte que les sommets adjacents reçoivent toujours deux couleurs distinctes

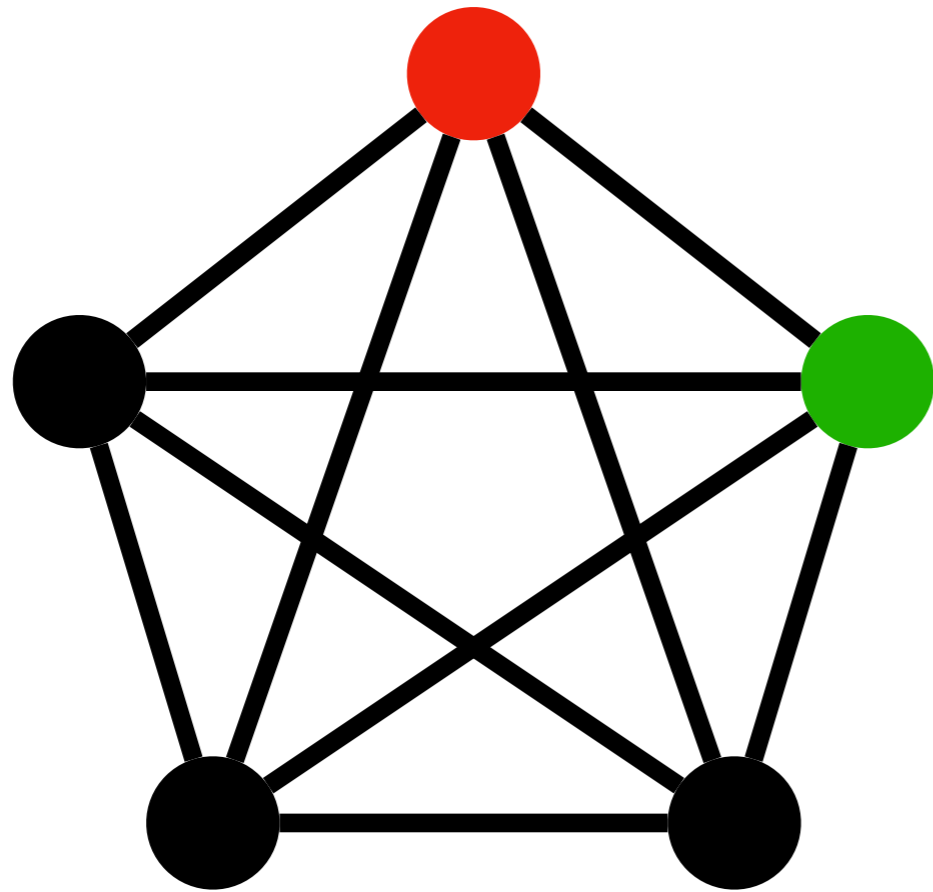
Graphes non planaires



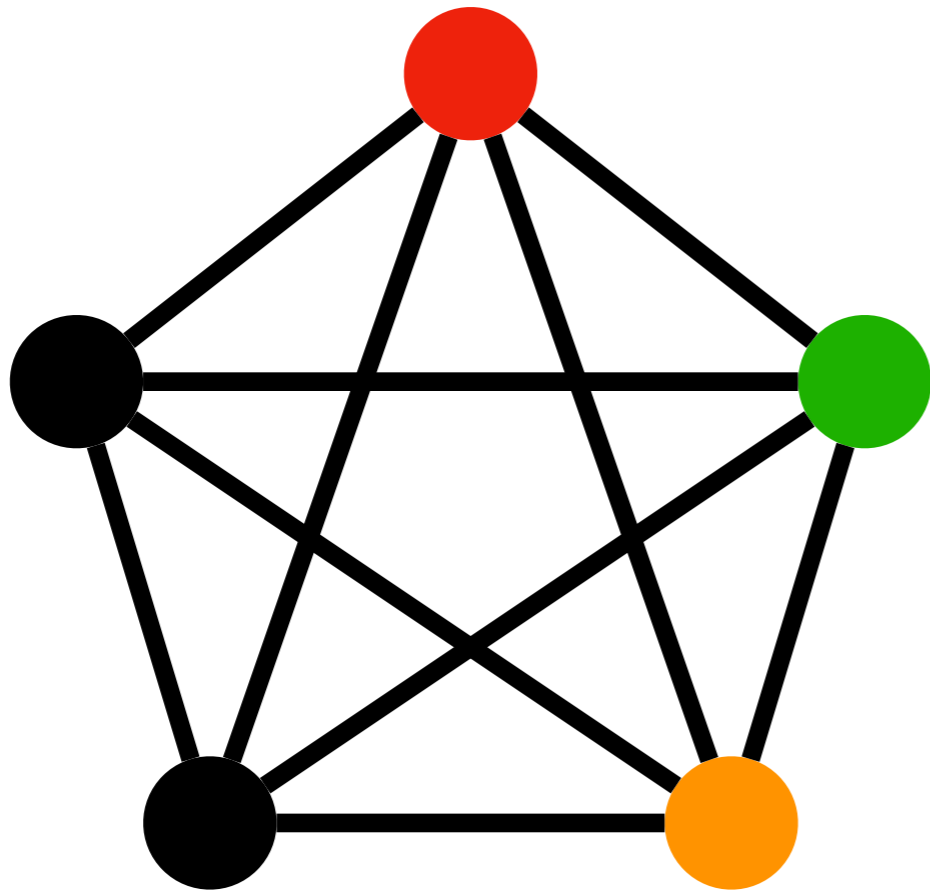
Graphes non planaires



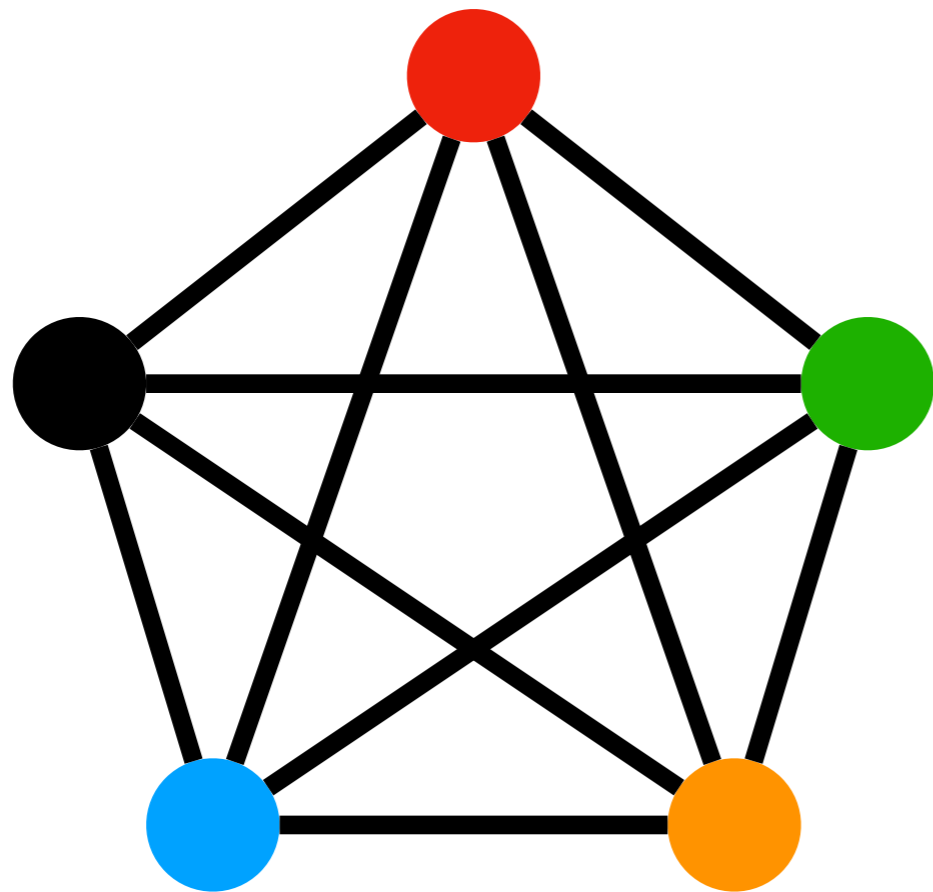
Graphes non planaires



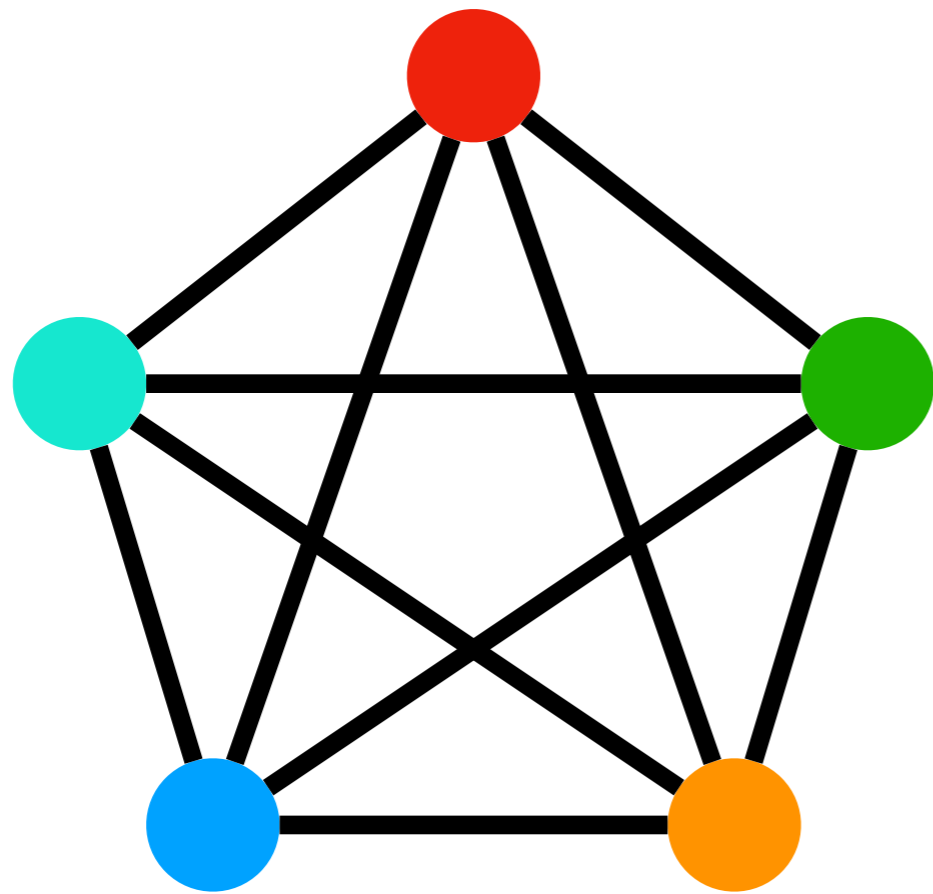
Graphes non planaires



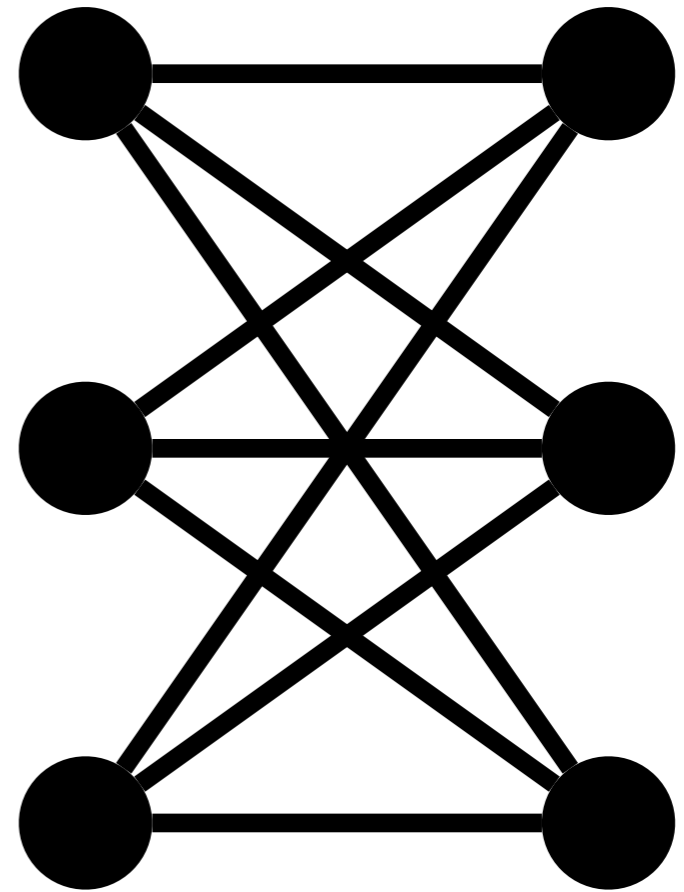
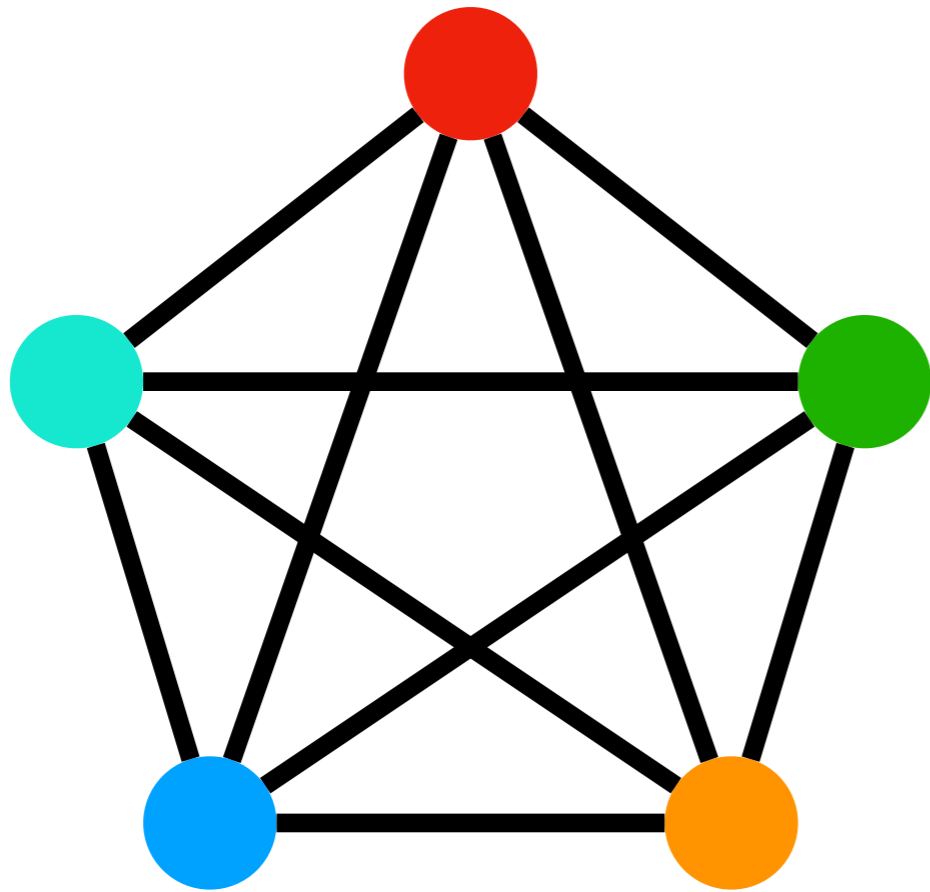
Graphes non planaires



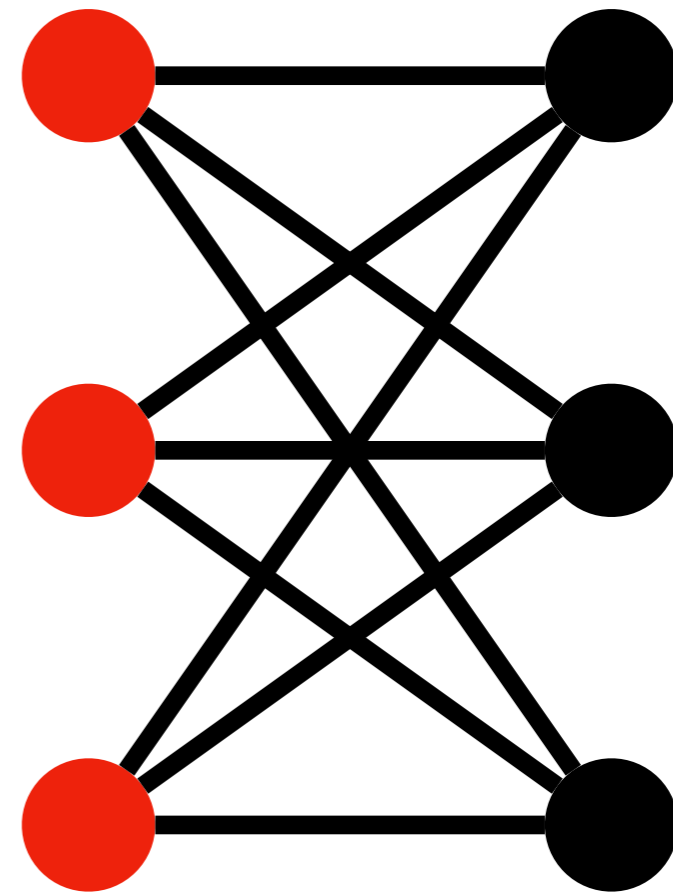
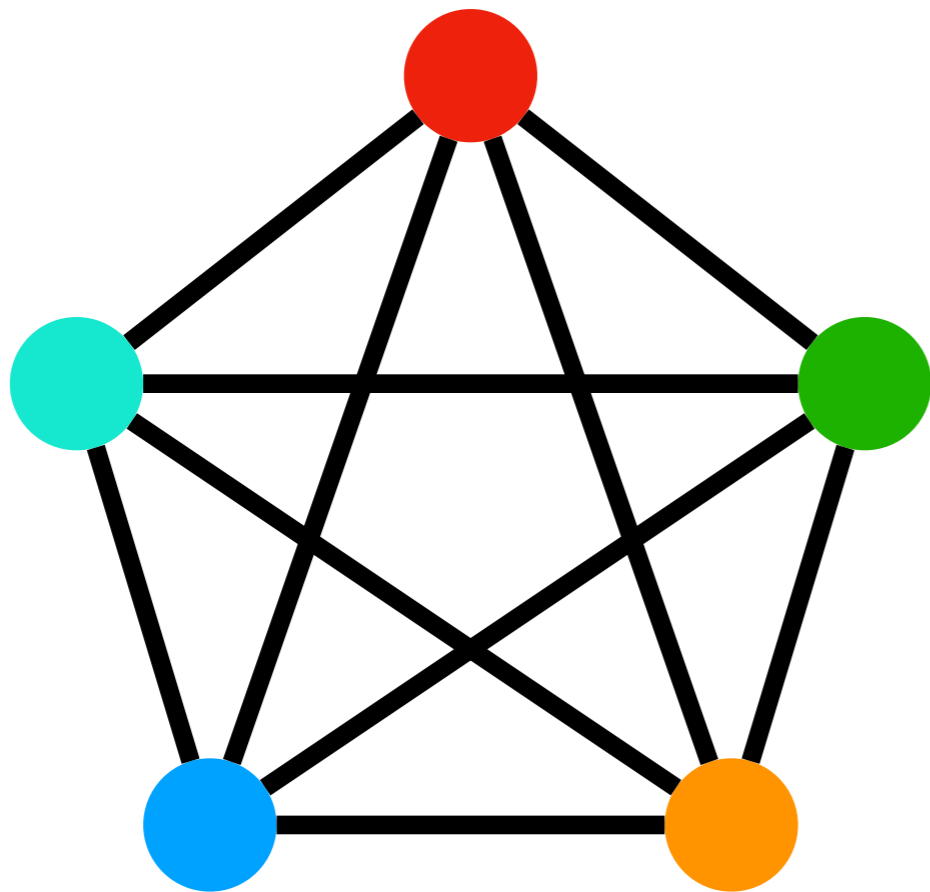
Graphes non planaires



Graphes non planaires



Graphes non planaires



Graphes non planaires

