

Chercher et Décider

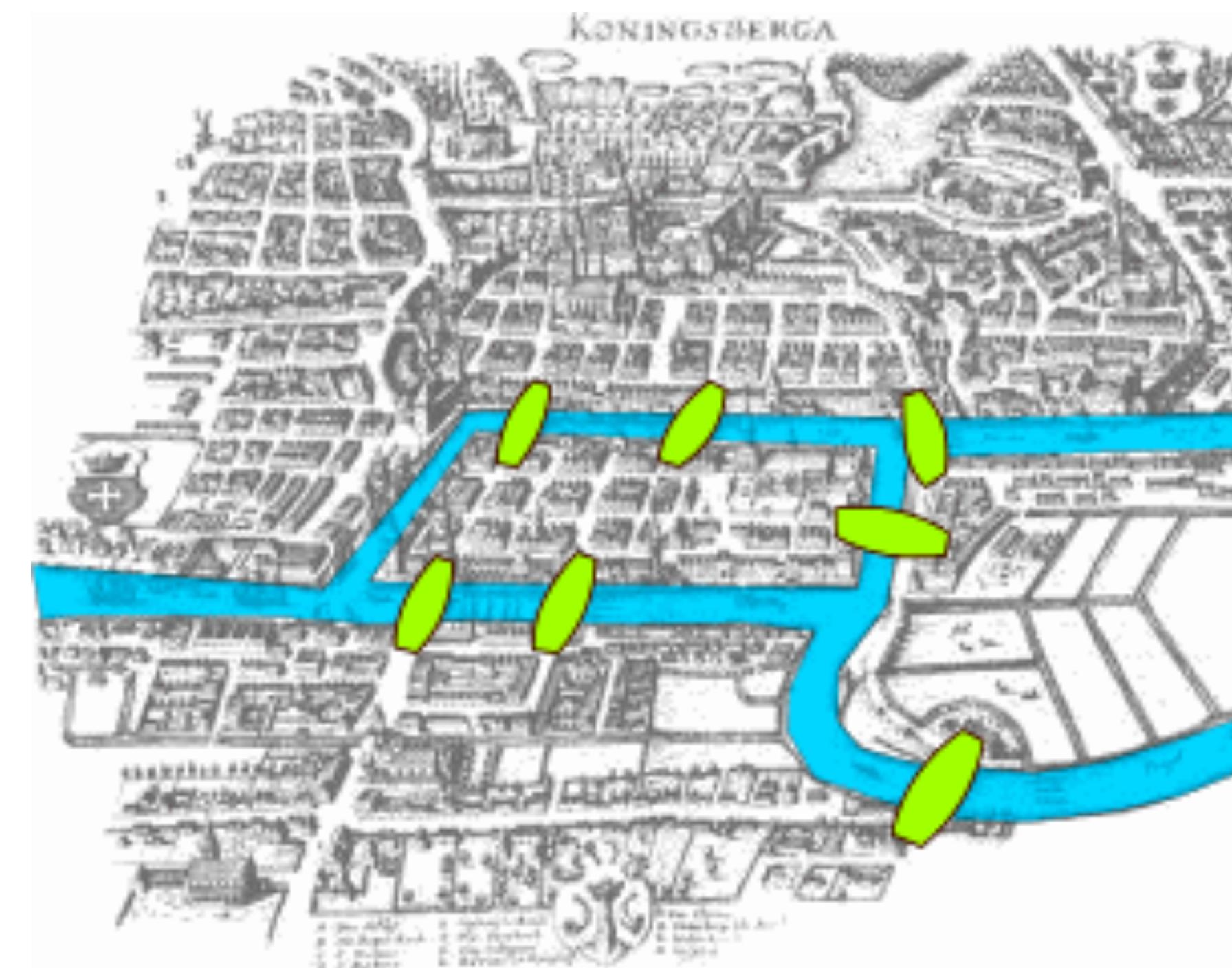
La théorie des graphes

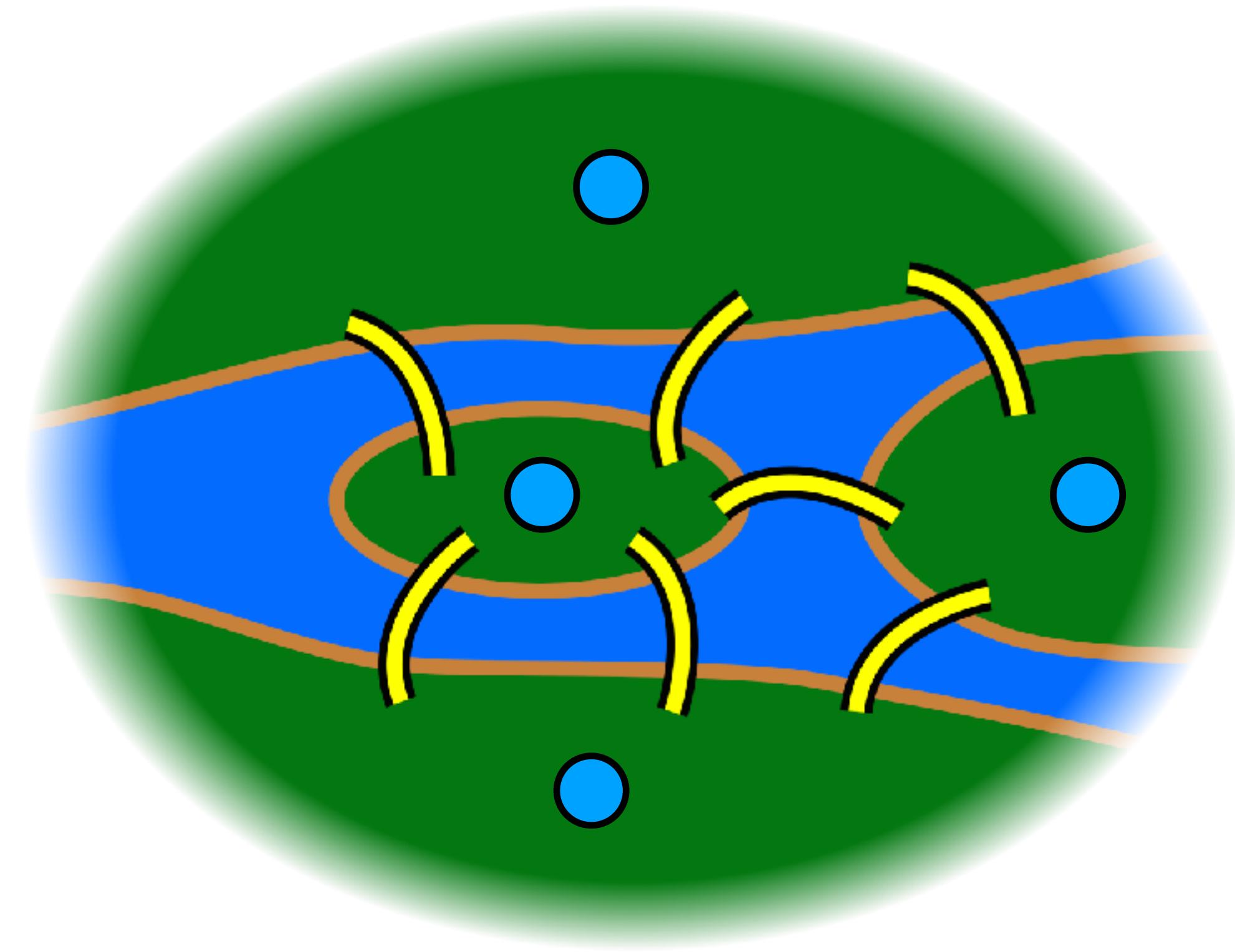
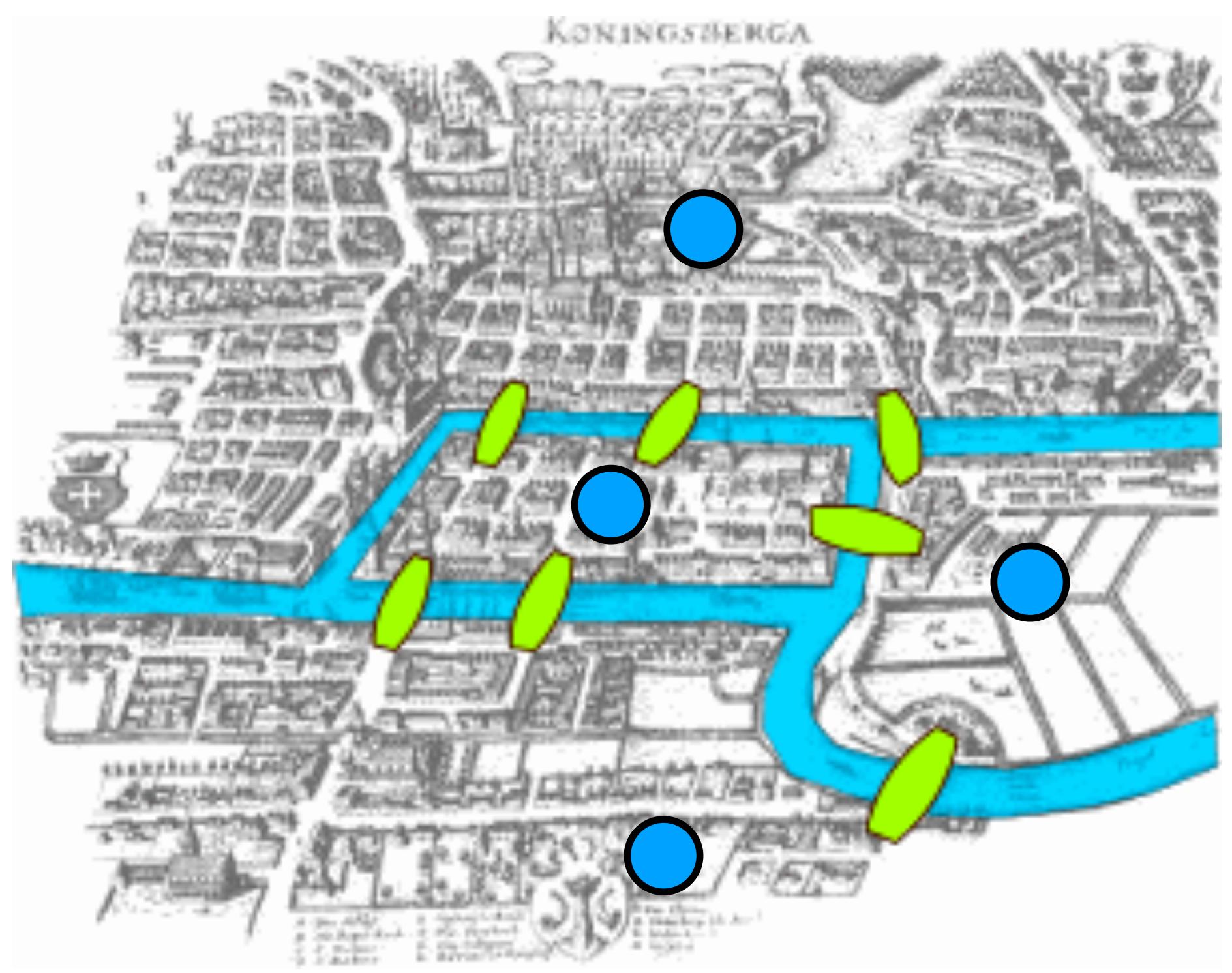
Martin Vielvoye 2021

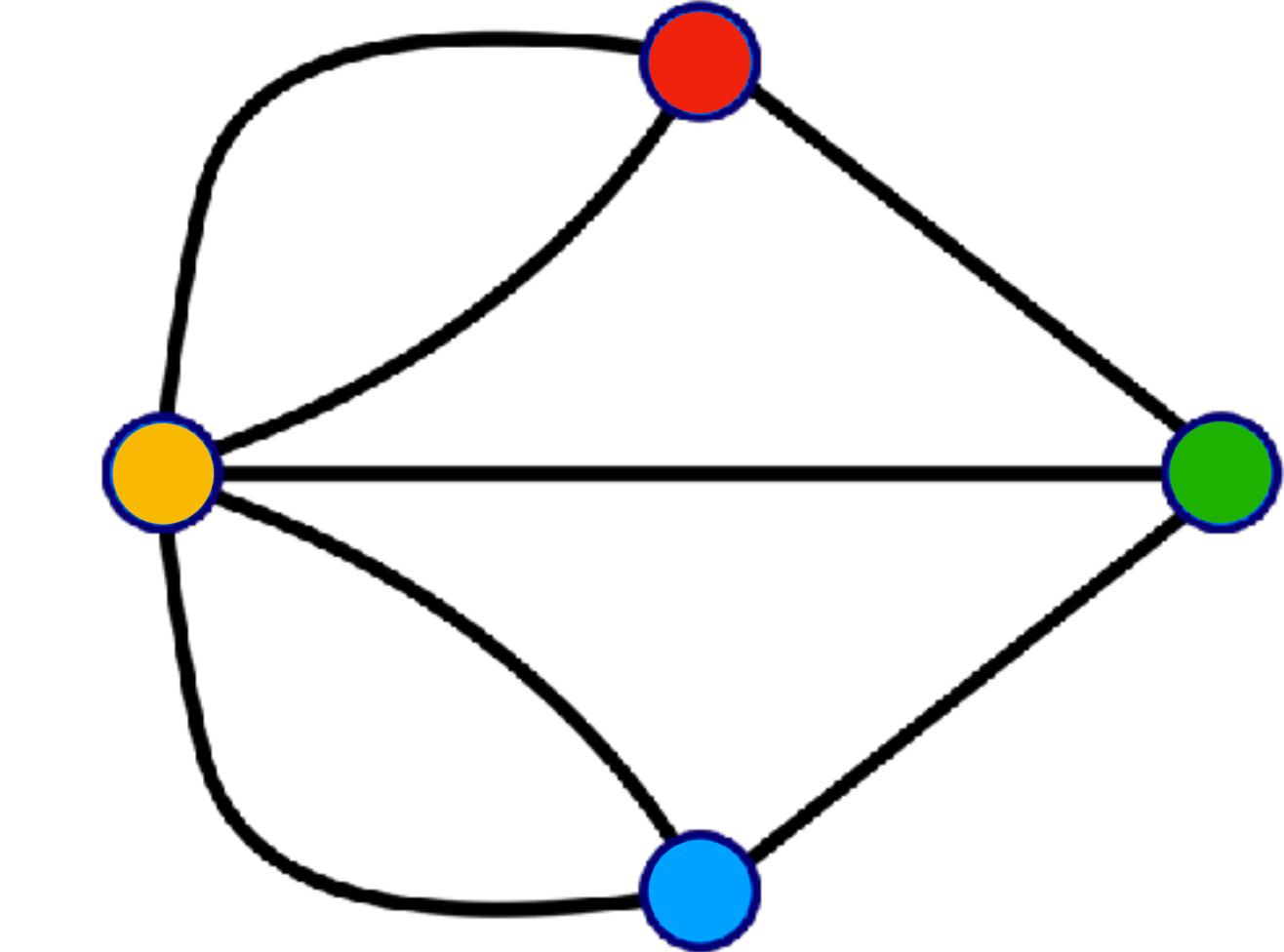
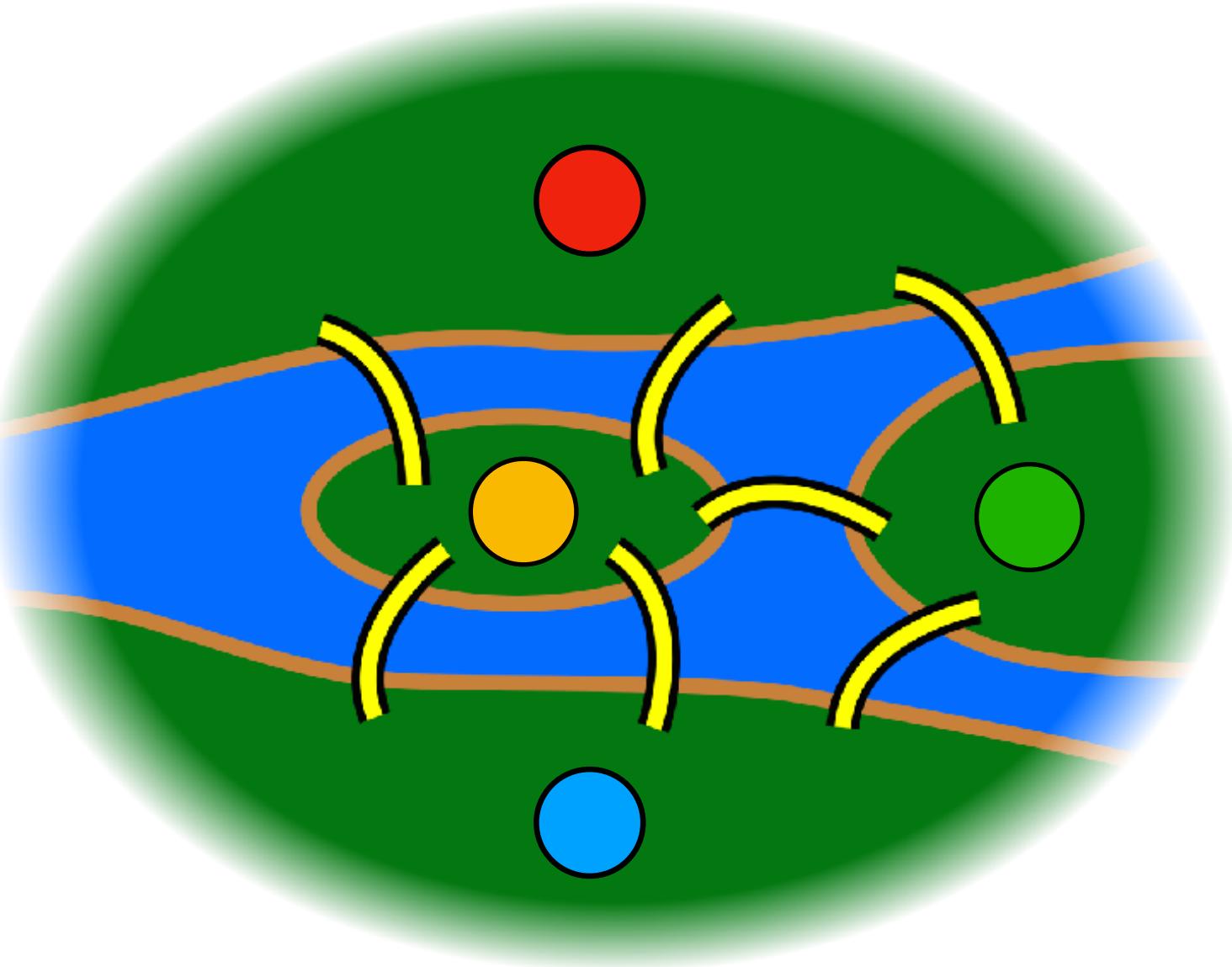
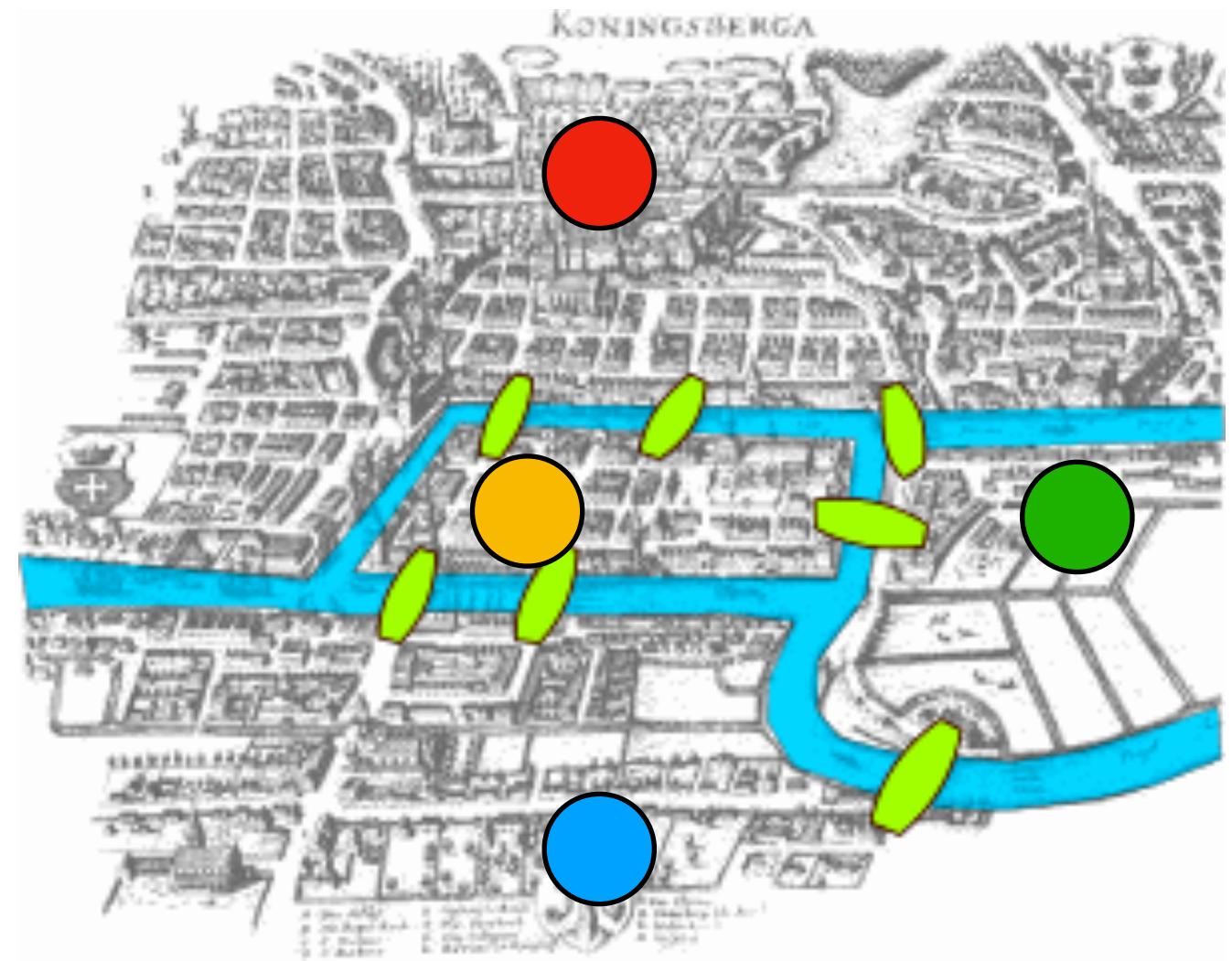
La théorie des graphes

Ou comment se représenter une balade sur des ponts.

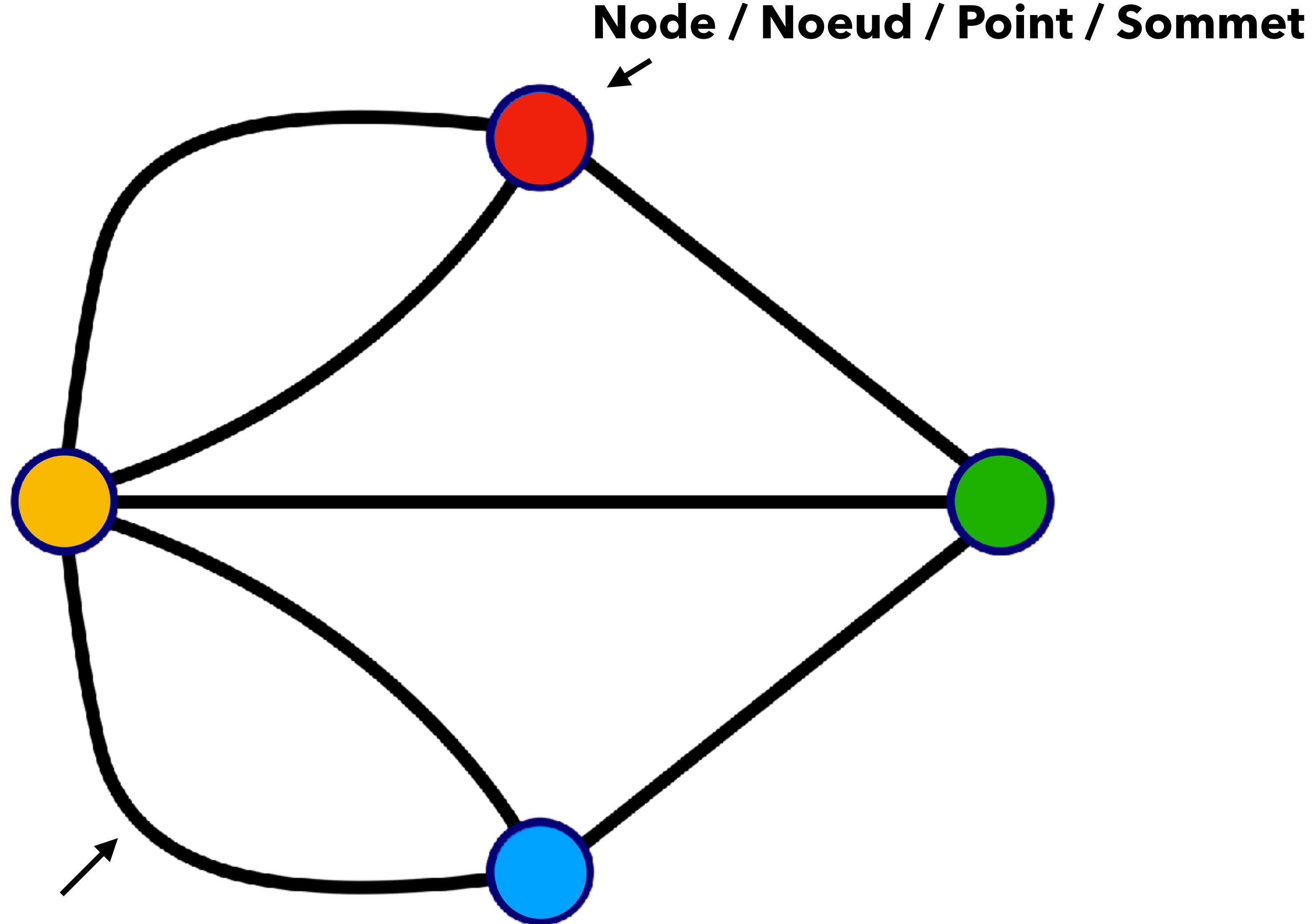
**"Existe-t-il un chemin par lequel je ne passe
par chaque ponts, qu'une seule fois?"**





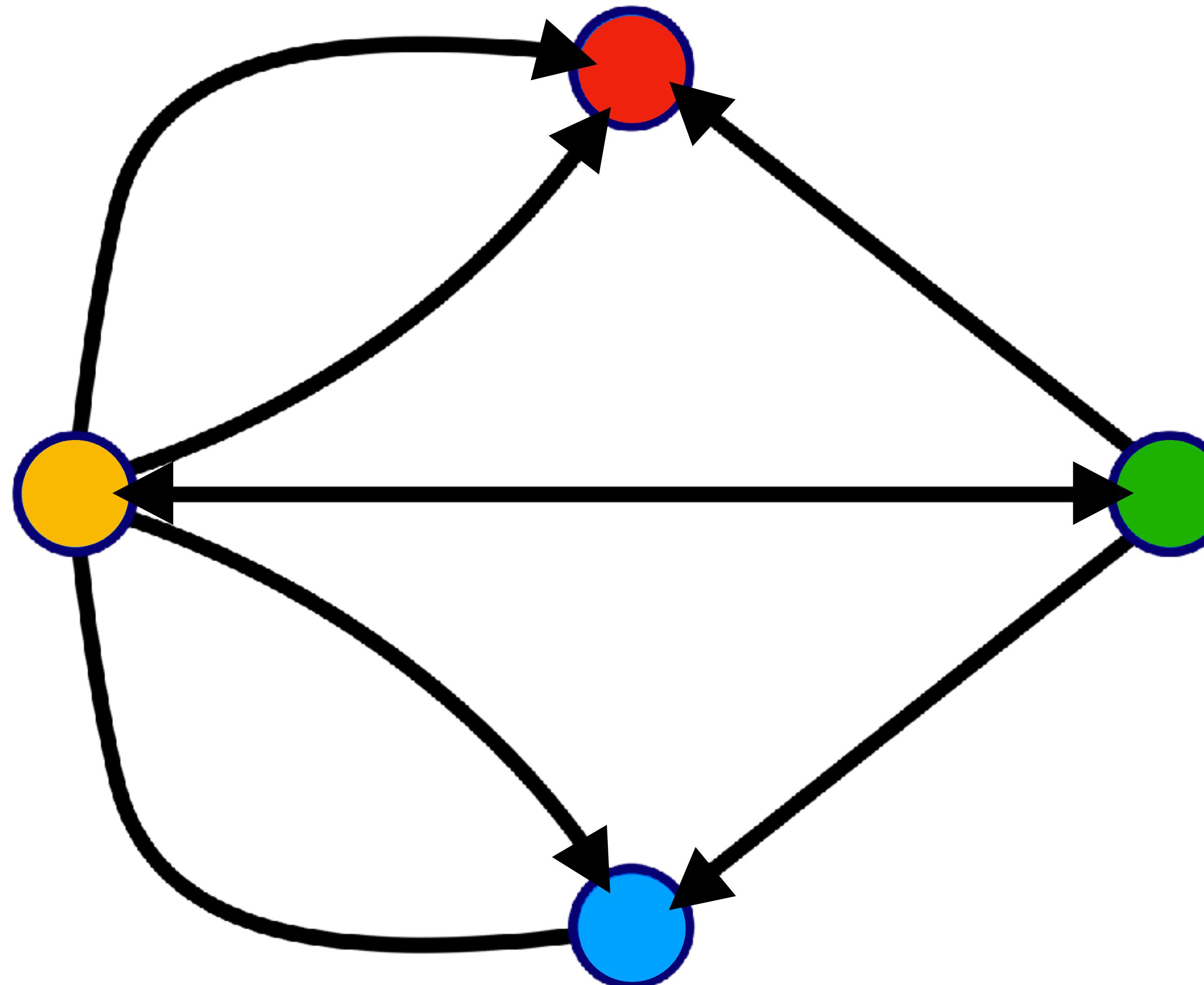


Graphe

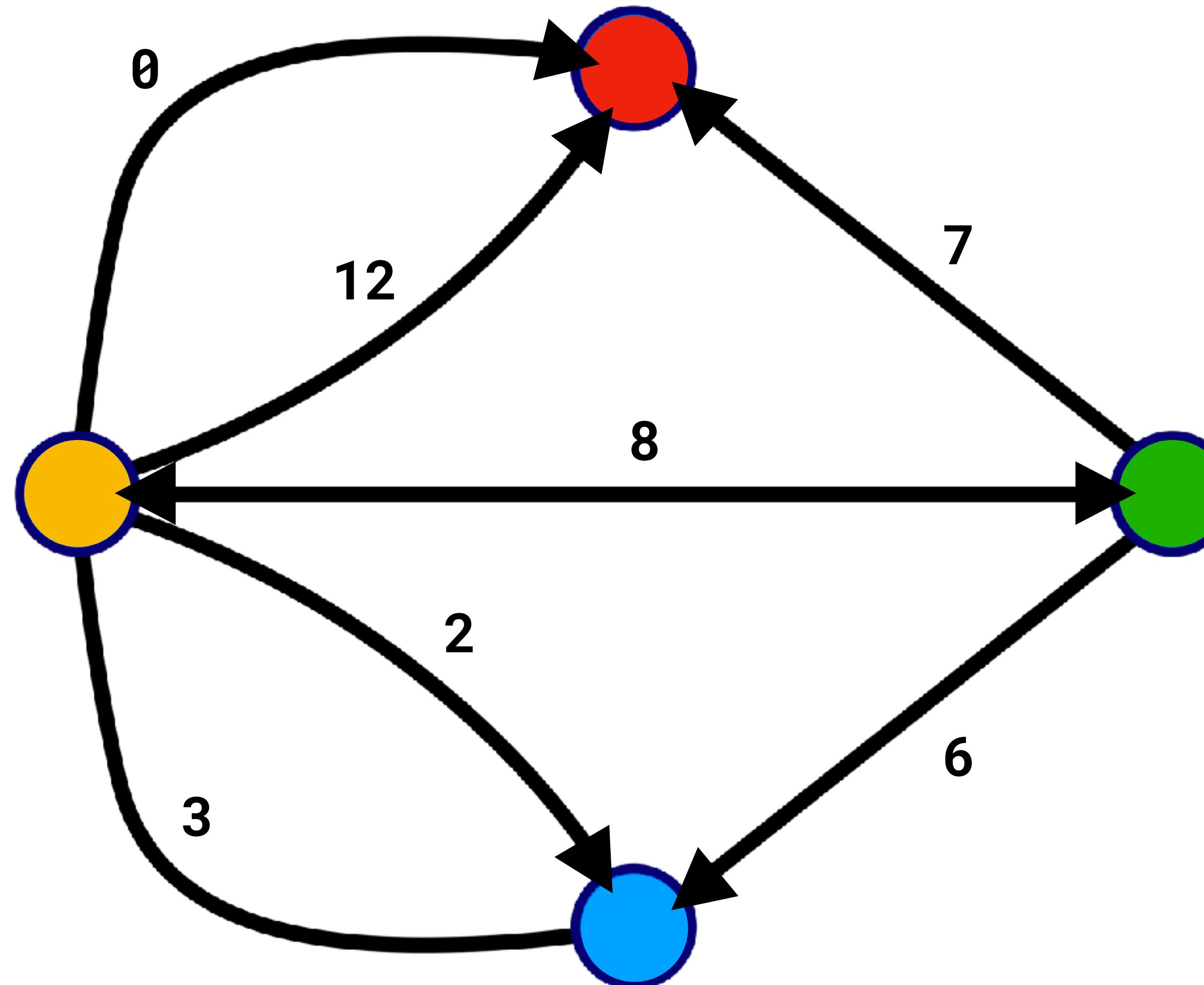


Vertice / Arête / Lien / Ligne

Graphe orienté



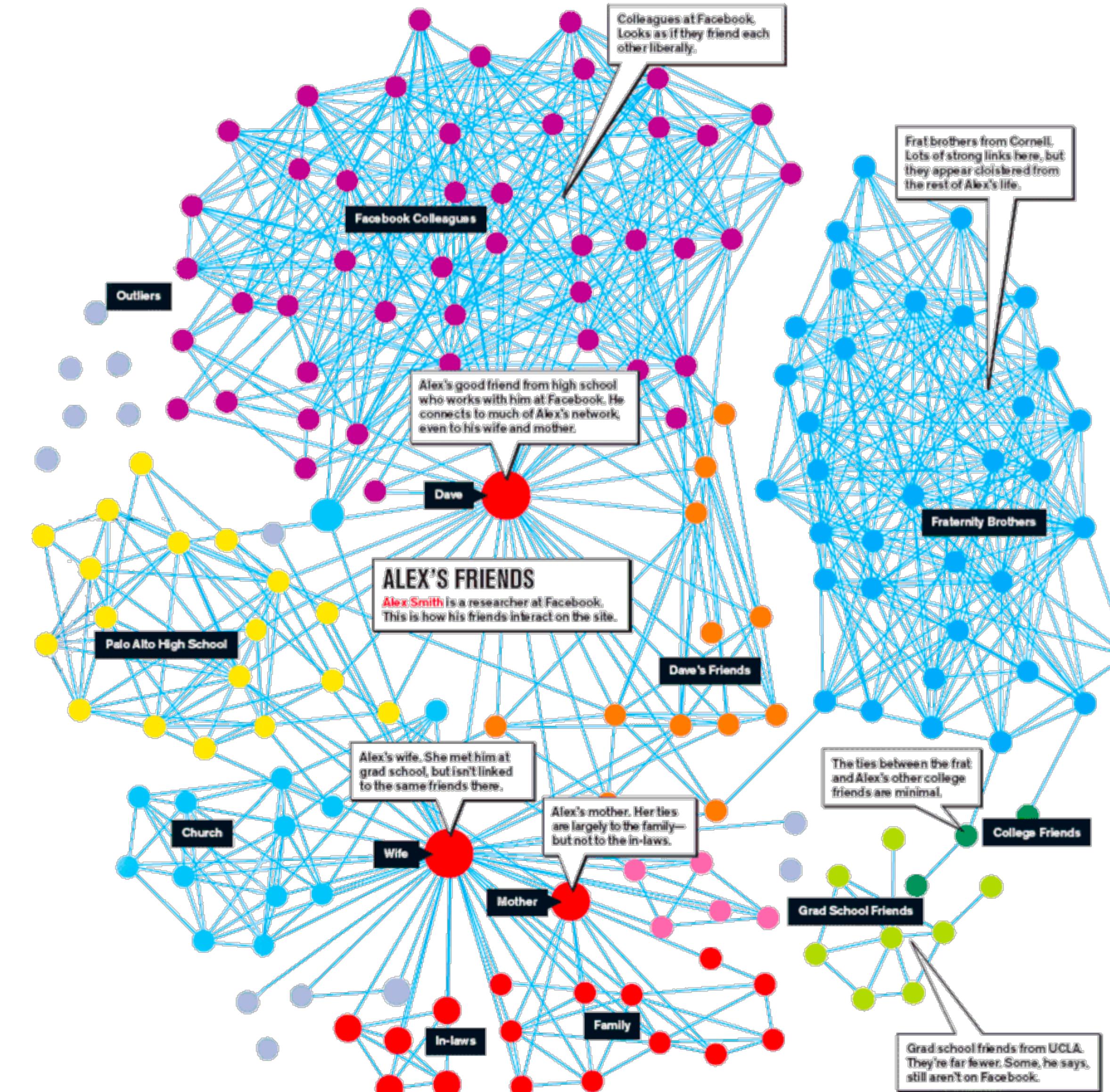
Graphe pondéré



Représenter des relations entre les données

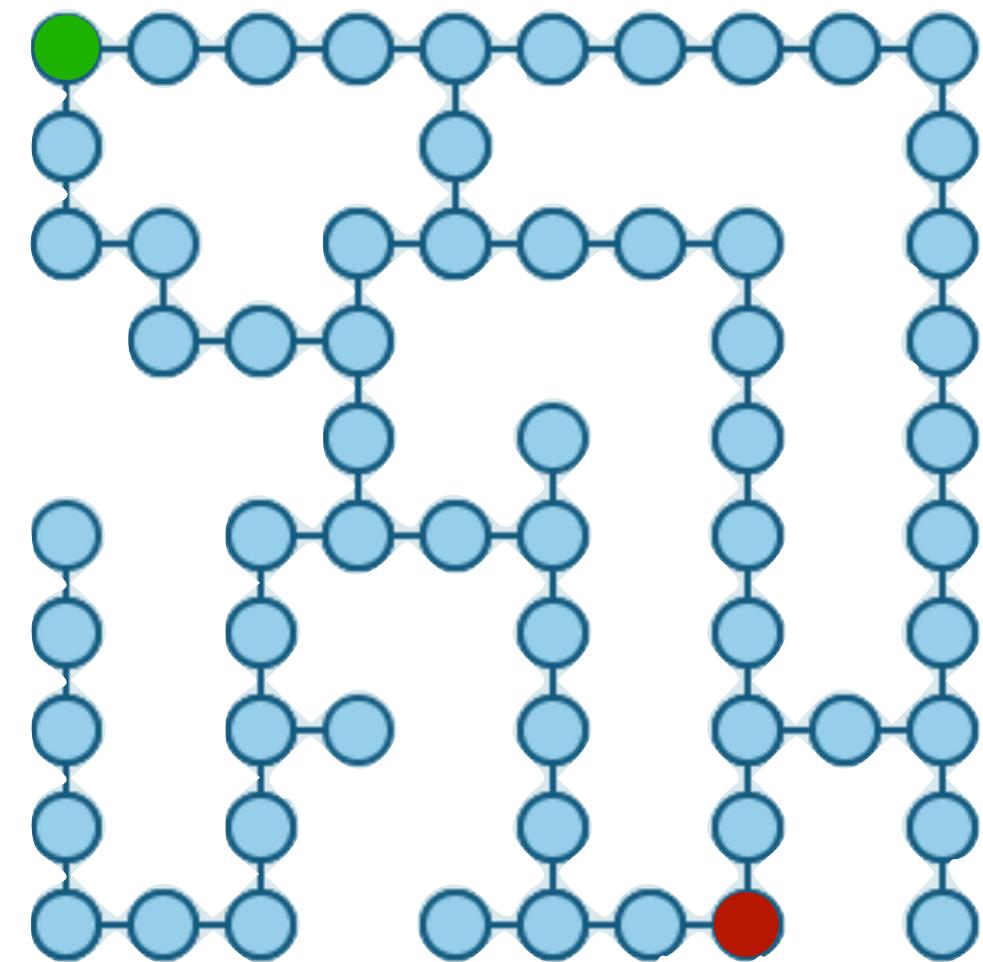
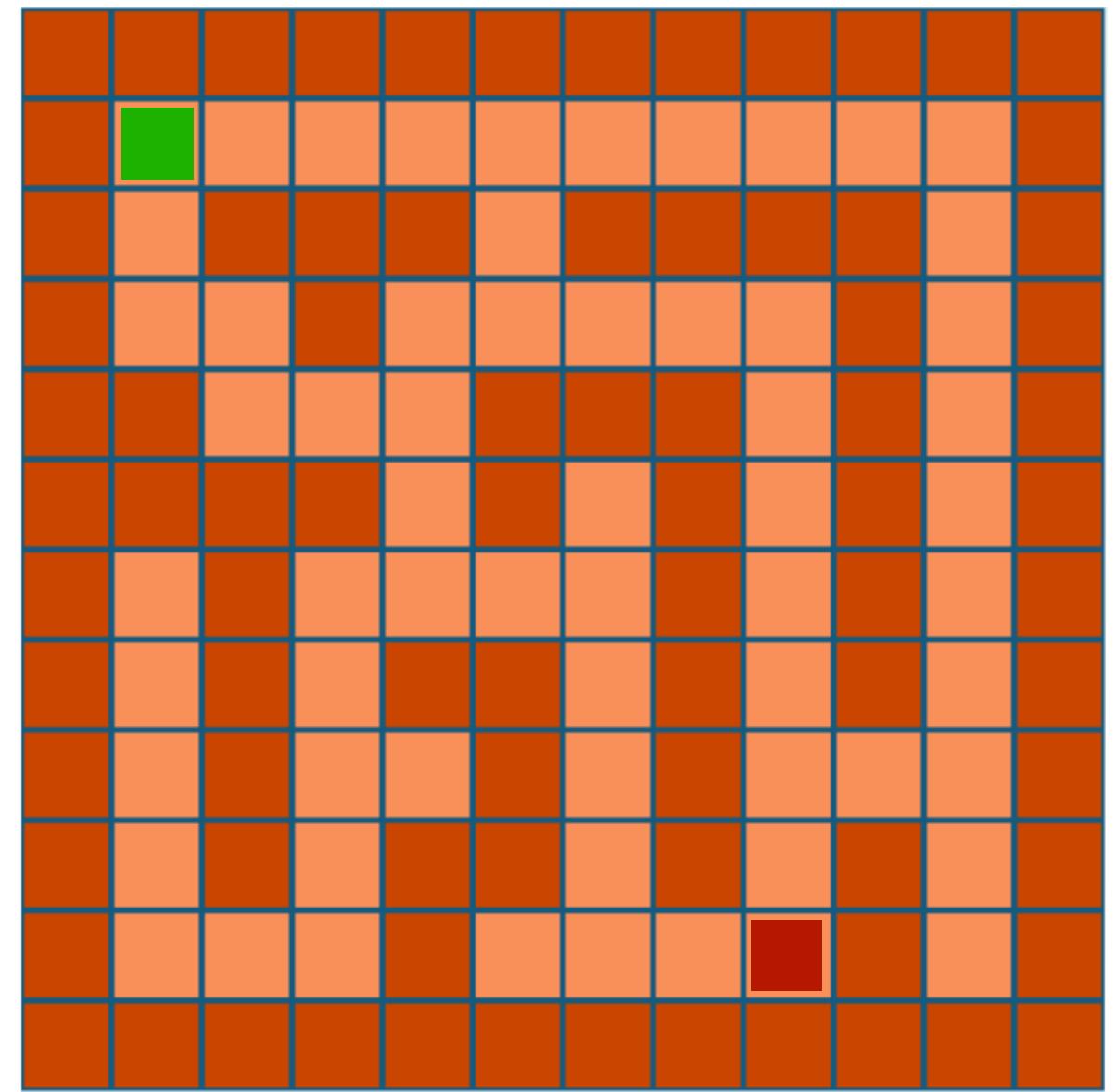
Graph Social

- Chaque ligne indique que deux personnes se connaissent.
 - 'Alex' connaît plusieurs personnes, qui peuvent se connaître entre elles.
 - Chaque personne connaît d'autres personnes aussi.
 - Exemples :
 - *Le nombre de Dunbar*
 - *La théorie des six poignées de main*



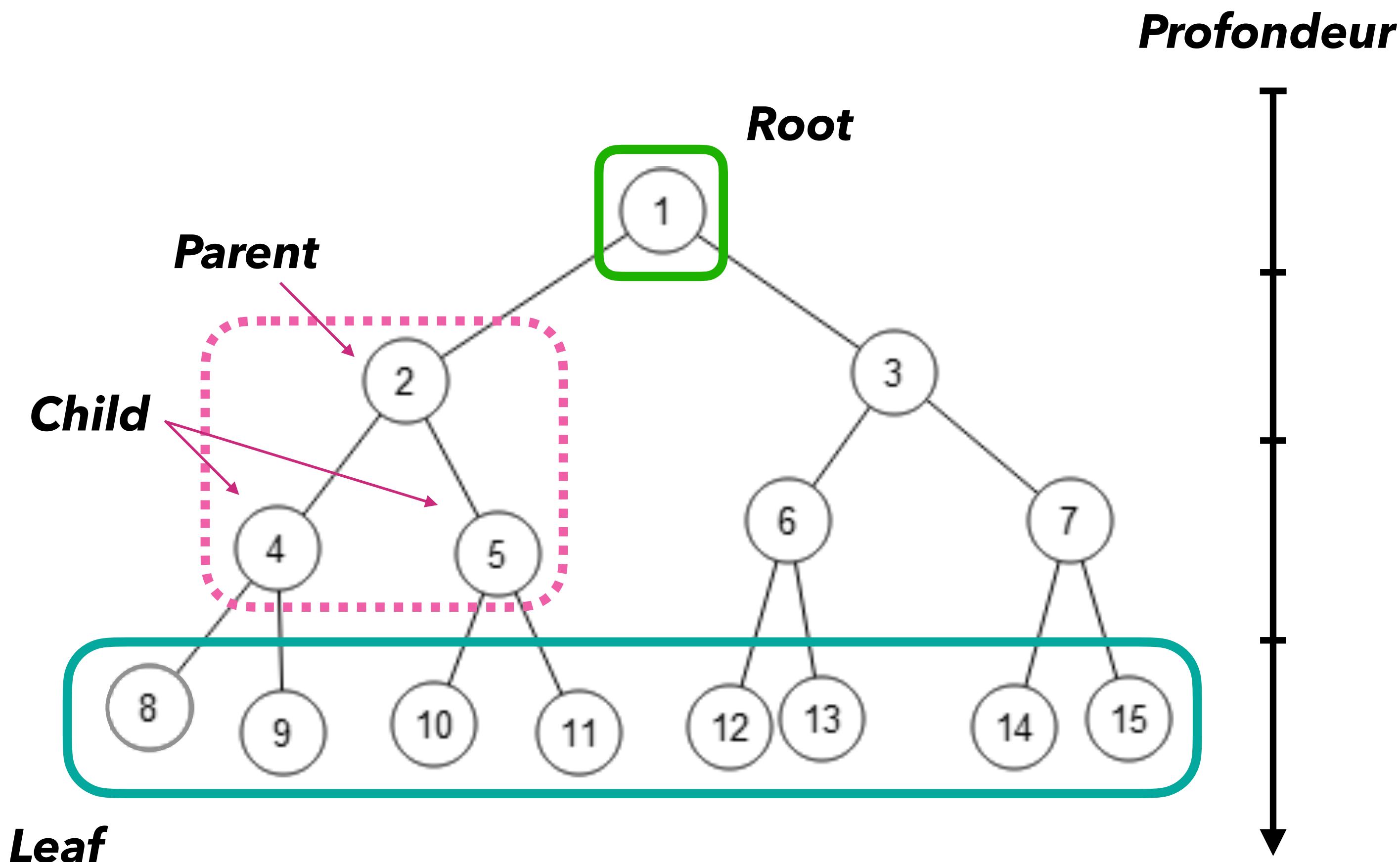
Path-finding

- Relier et créer des chemins dans un environnement.
- Le nombre d'incidences permet de savoir quand le chemin se sépare.
- Pouvoir trouver les chemins les plus courts ou les plus longs.



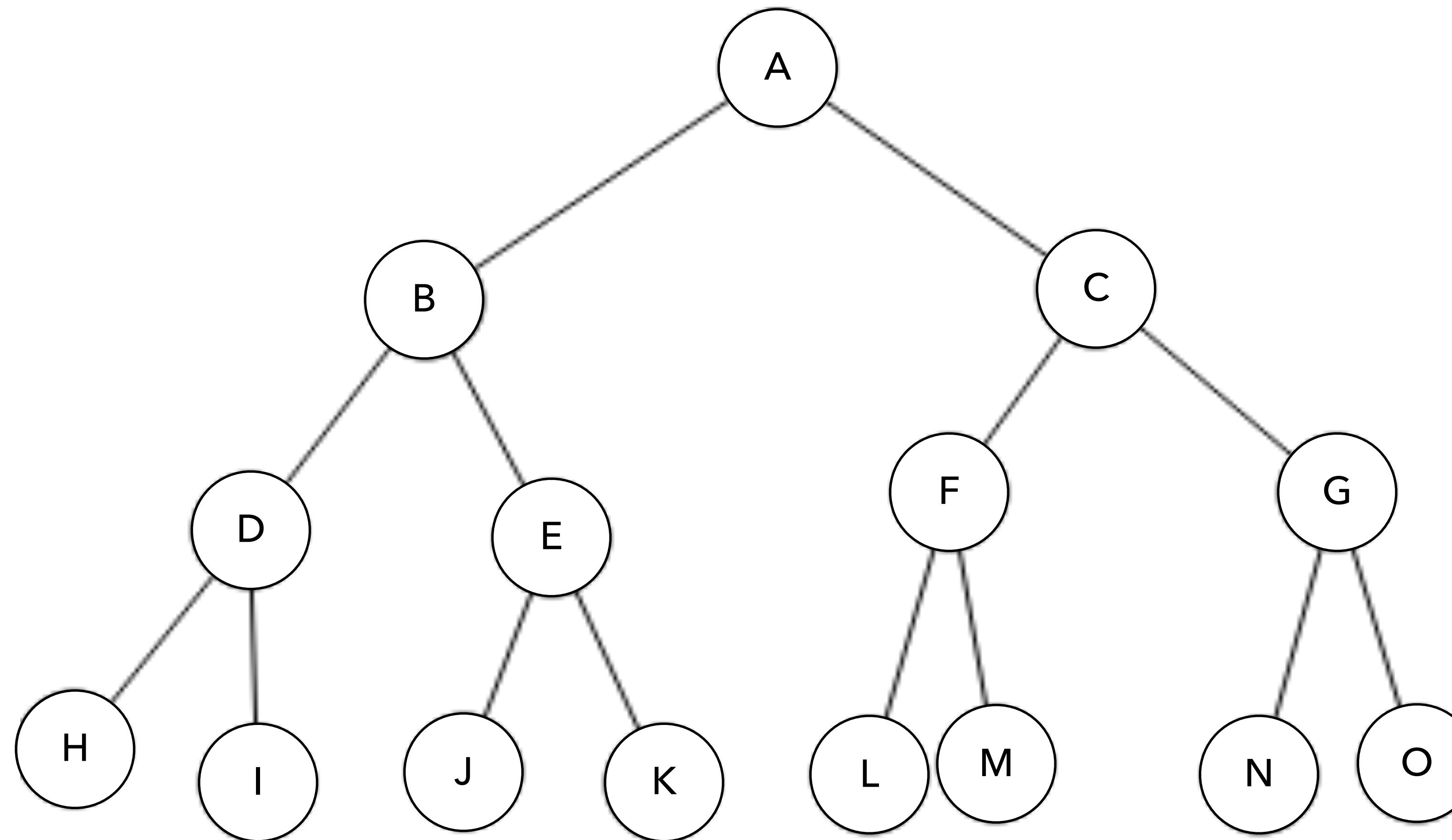
Arbre

- Termes importants :
- Structures fréquemment utilisées :
 - Arbre Général
 - Arbre Binaire
 - Arbre Binaire de Recherche
 - Arbre Rouge-Noir
 - Arbre Splay



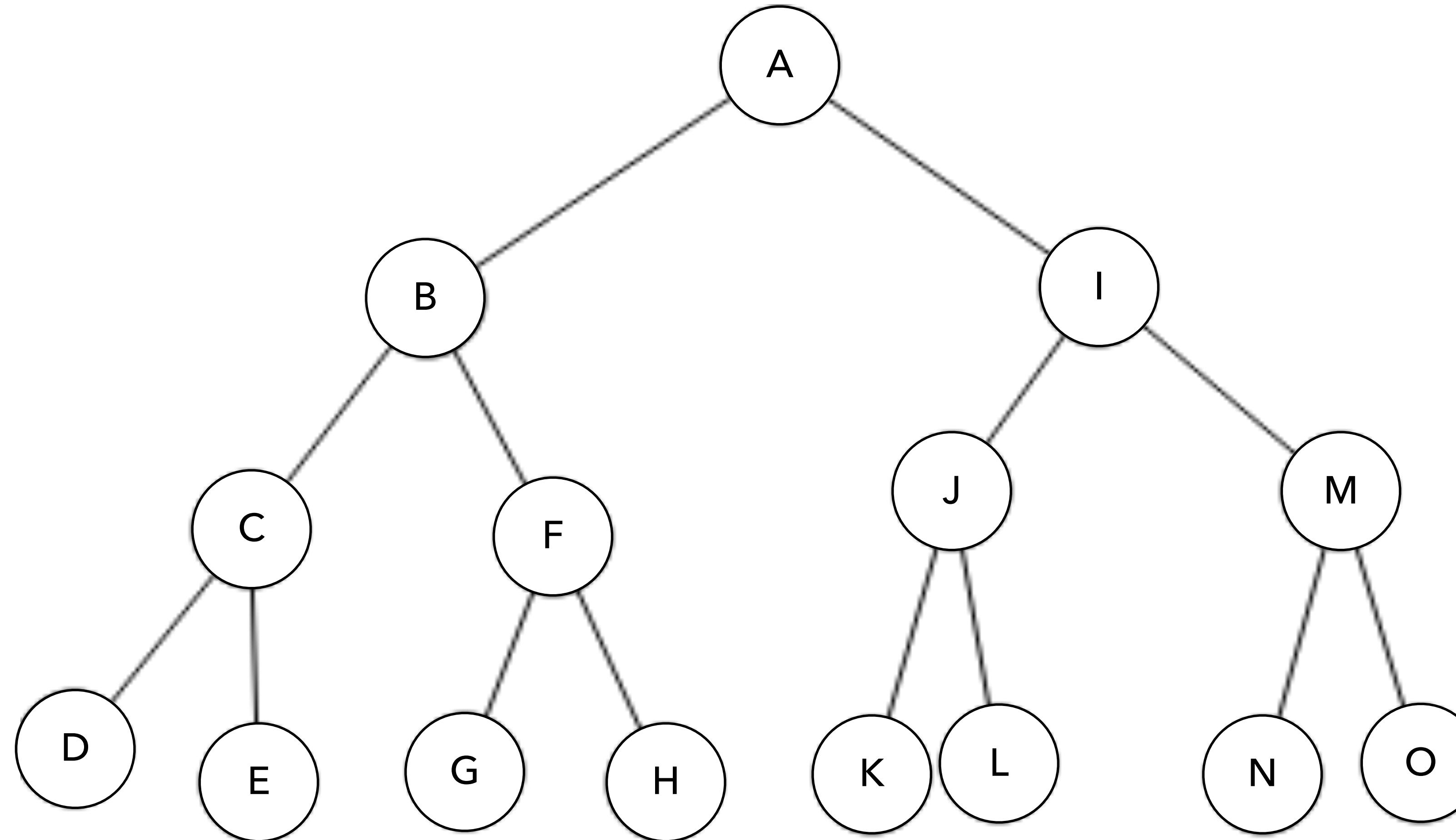
Parcourir un arbre

Breadth-First Search (BFS)



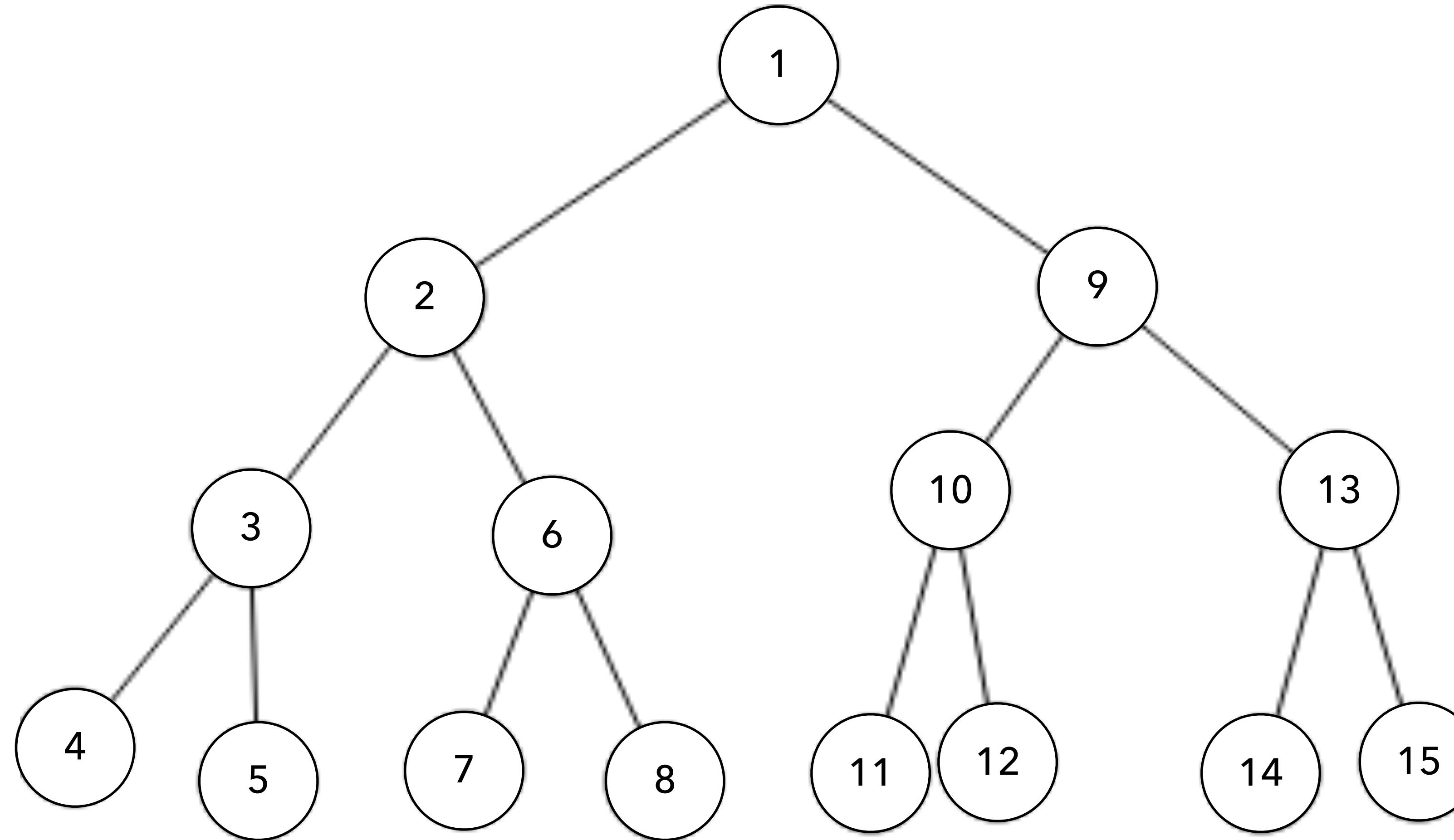
Parcourir un arbre

Depth-First Search (DFS)

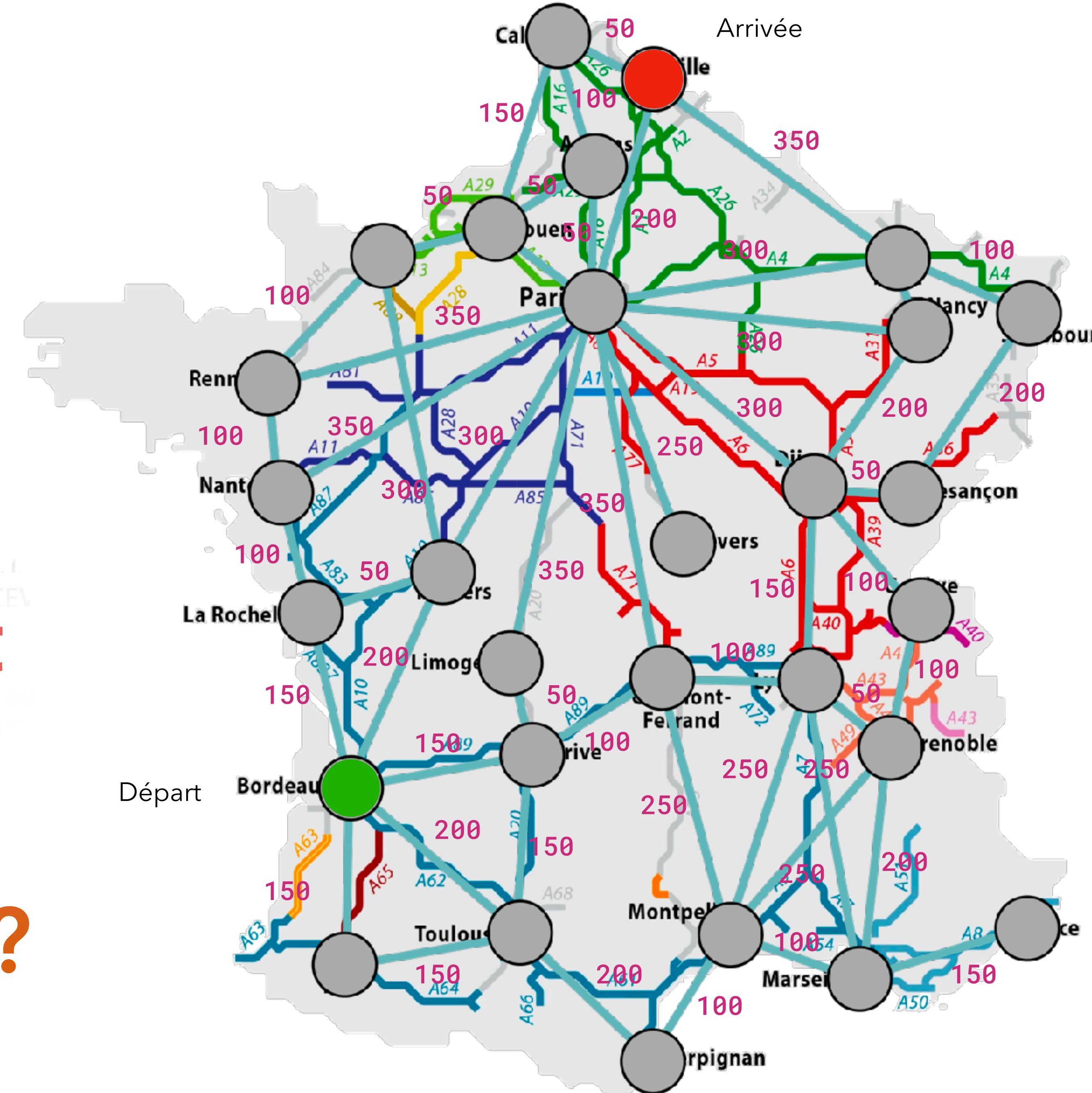


Parcourir un arbre

Uniform Cost Search (UCS)

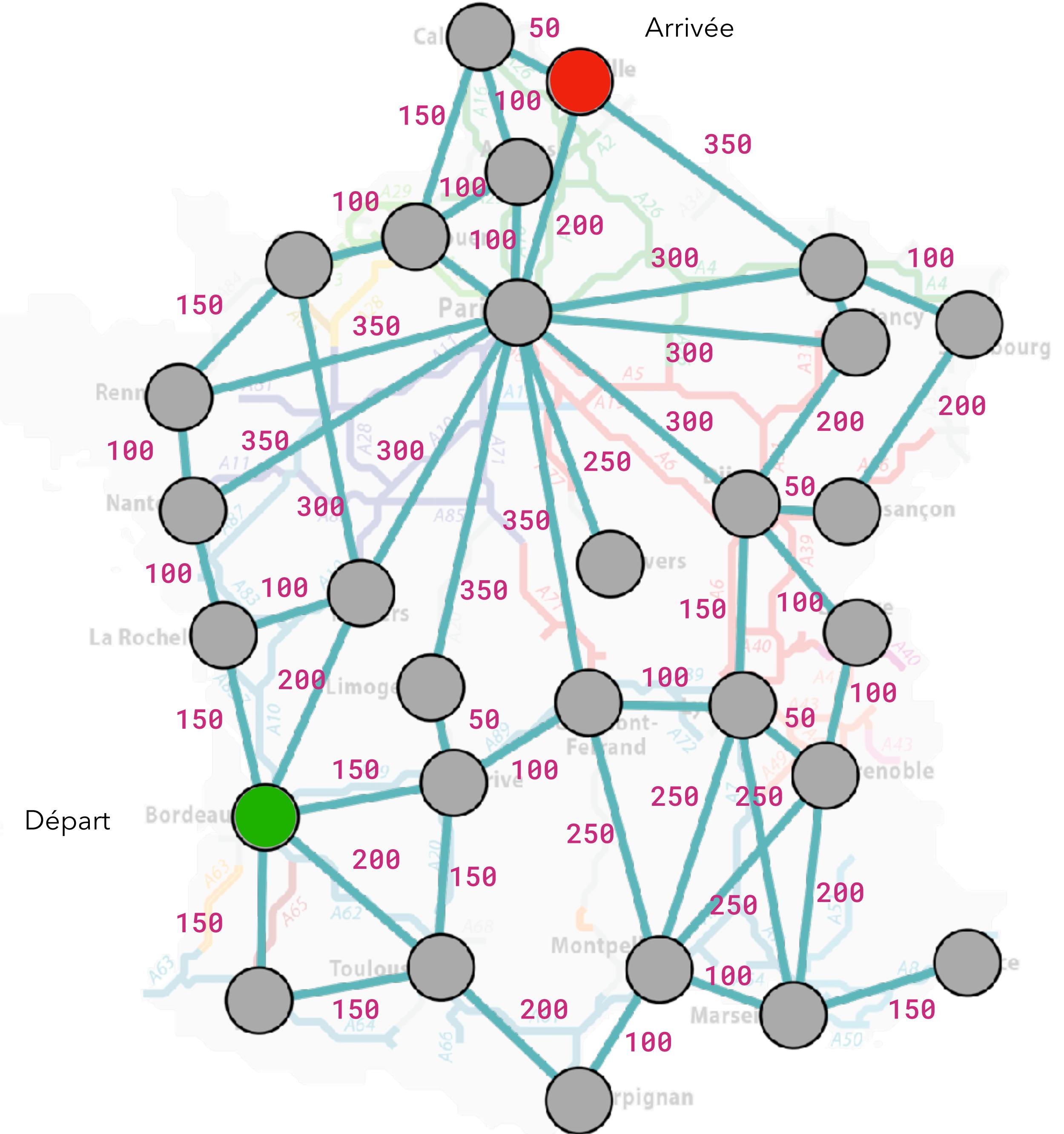


Comment trouver le chemin le plus court ?



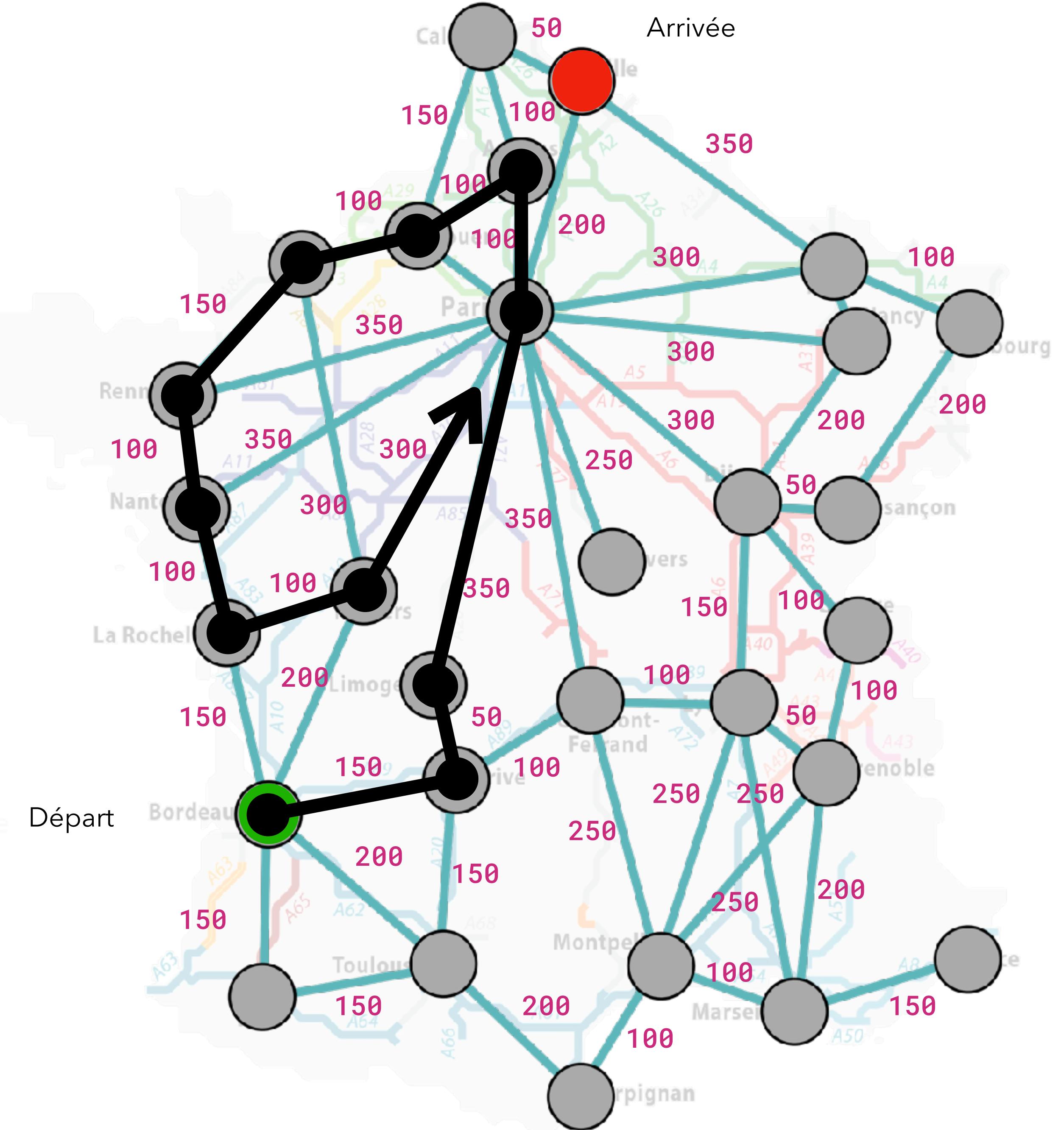
Analyser tous les chemins possibles ?

- Une fois tous les chemins trouvés, on trie simplement par ordre de grandeur et on trouve le plus court.
 - Nombre de possibilités factorielles pour chaque nouveau noeud.
- Solution optimale ? True
 - Temps ? $O(n!)$



Prendre l'arête la plus courte (UCS)

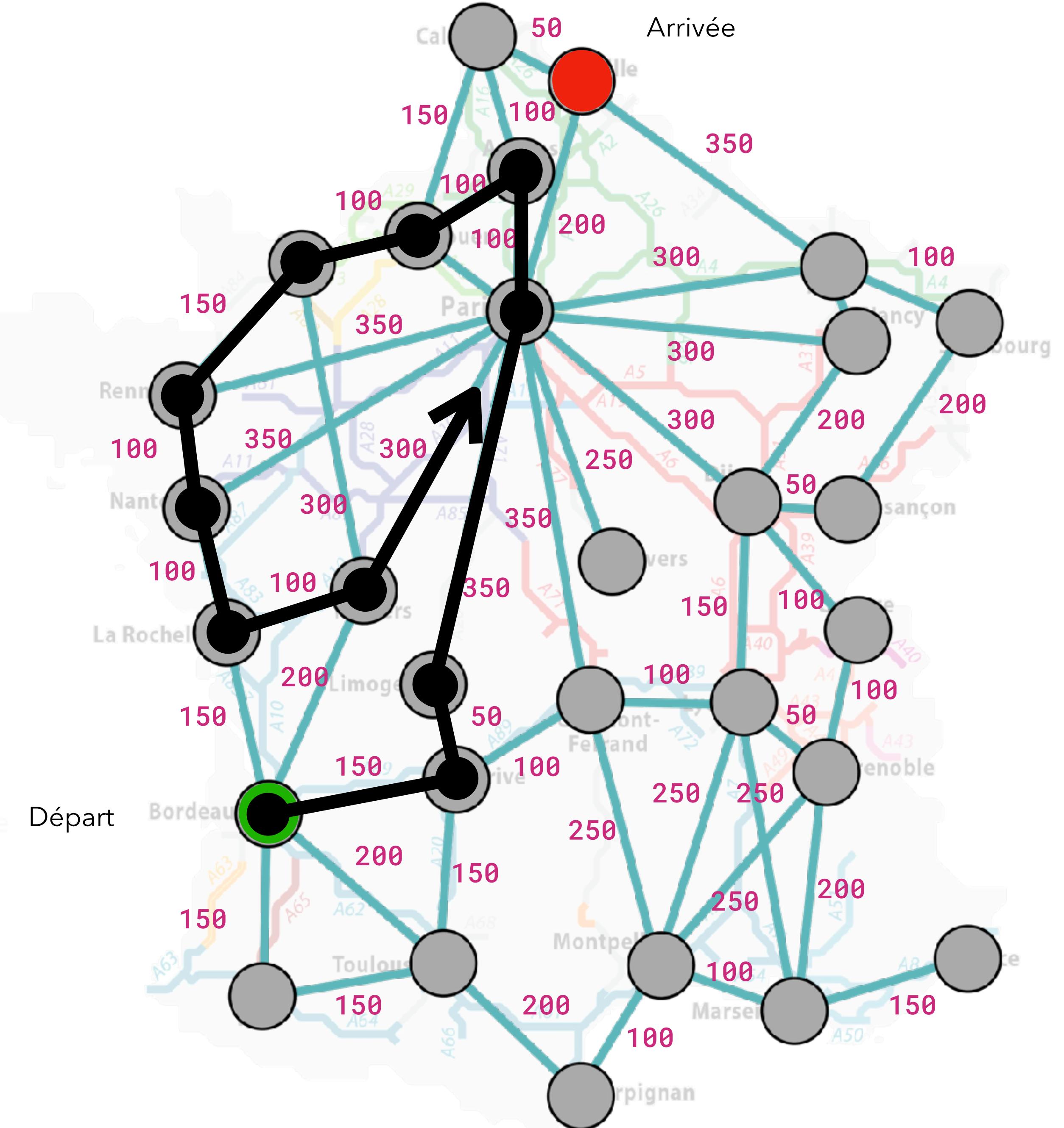
- On commence au départ, et on prend immédiatement l'arête la plus courte.
 - S'il y a égalité, on prend celui qui a l'angle le plus proche de la destination.
 - On est pas assuré de trouver la meilleure solution.
- Solution optimale ? *False*
- Temps ? $O(d)$



Trouver la solution optimale avec simplement les kilomètres ?

- Prendre un chemin qui est à contre-sens semble contre intuitif...
- On peut rajouter l'angle dans nos variables de recherche.

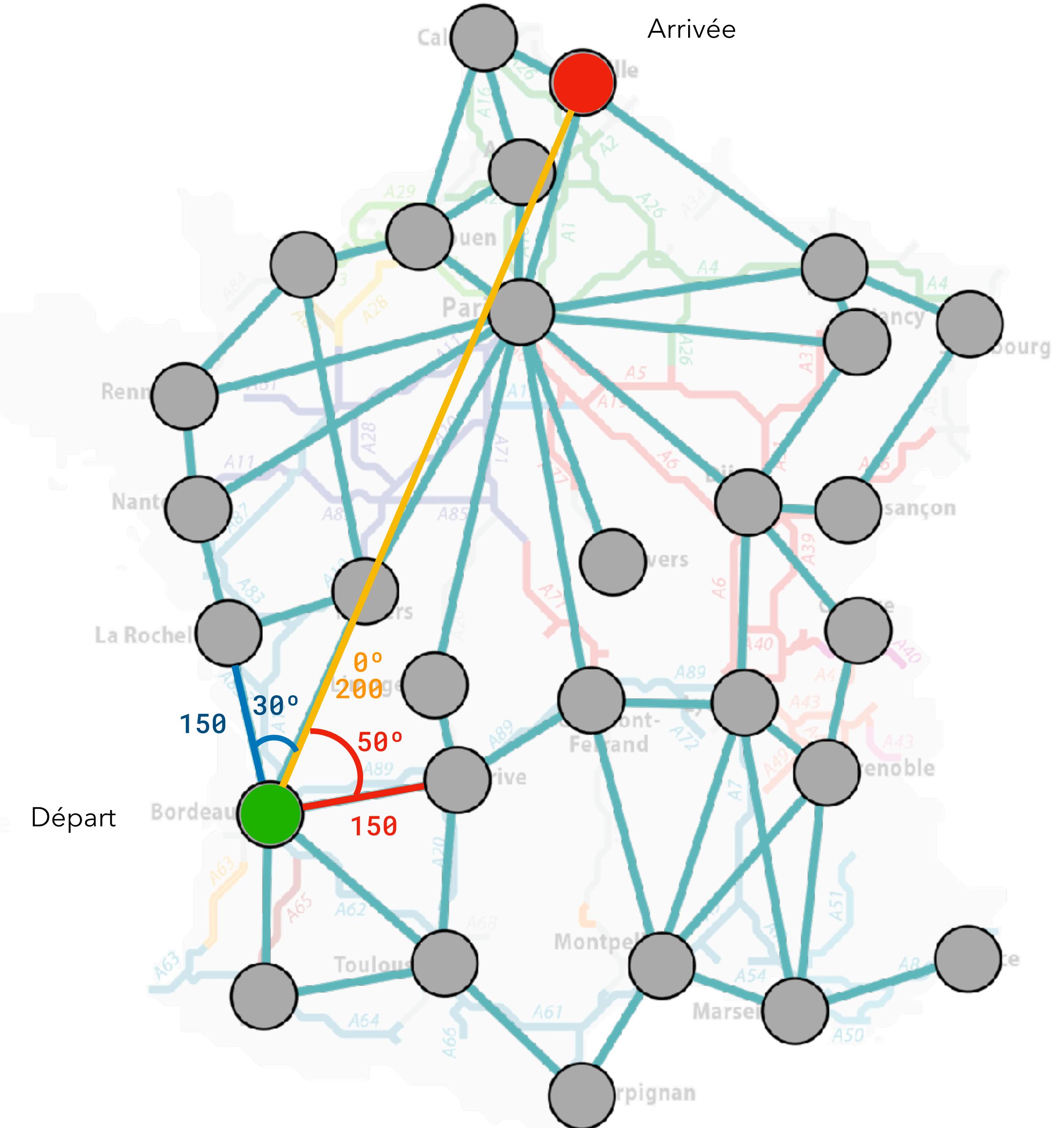
$$Km * (1 + (angle/180))$$



Trouver la solution optimale avec simplement les kilomètres ?

$$Km * (1 + (angle/180))$$

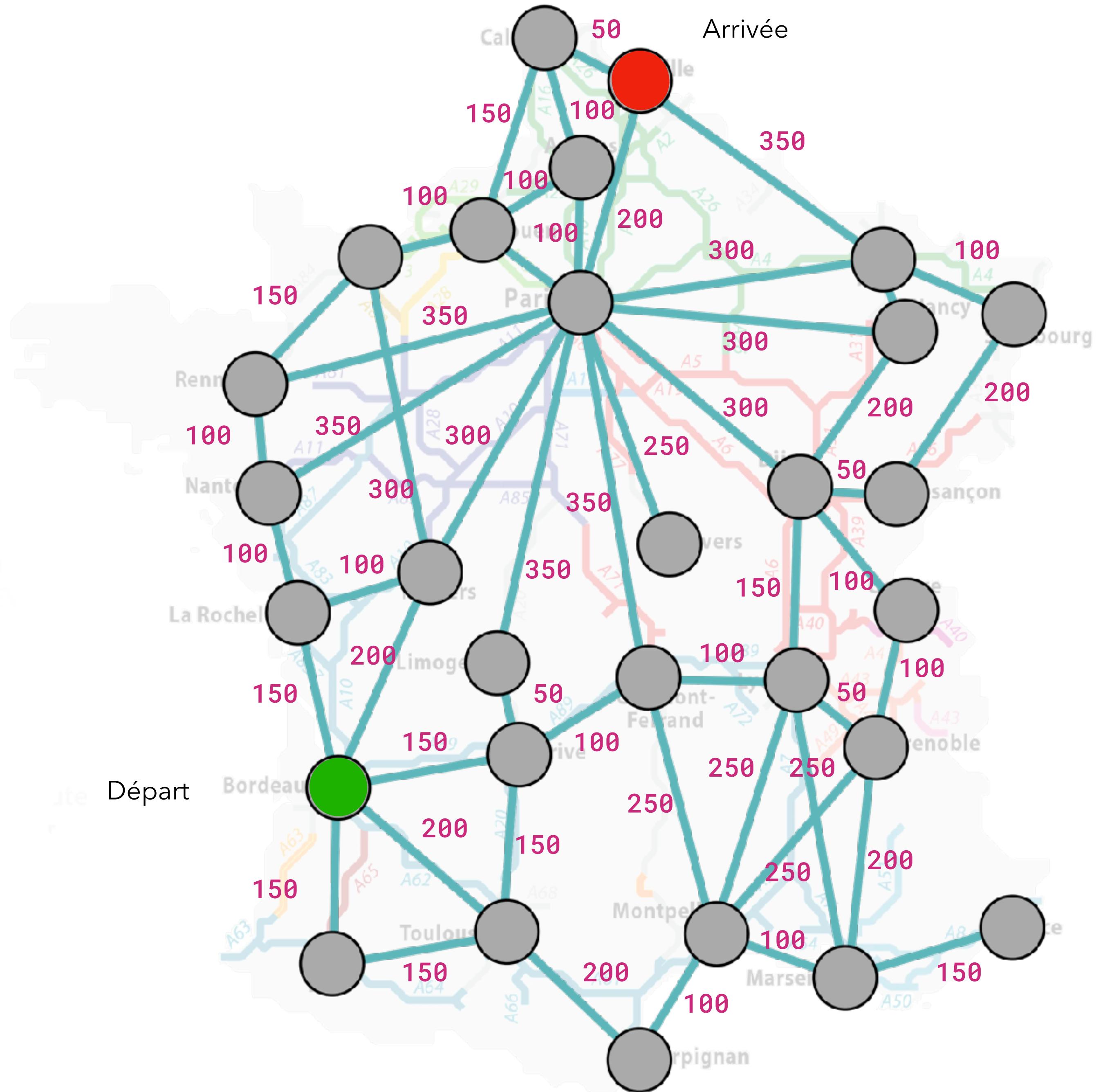
- $150 * (1 + 30/180) = 175$
- $200 * (1 + 0/180) = 200$
- $150 * (1 + 50/180) = 191.67$



D'autres problèmes

Est-ce que la distance est la seule valeur intéressante que l'on voudrait optimiser ?

- ▶ Le chemin le plus rapide;
- ▶ Le chemin qui consomme le moins;
- ▶ Le chemin le moins cher;
- ▶ etc



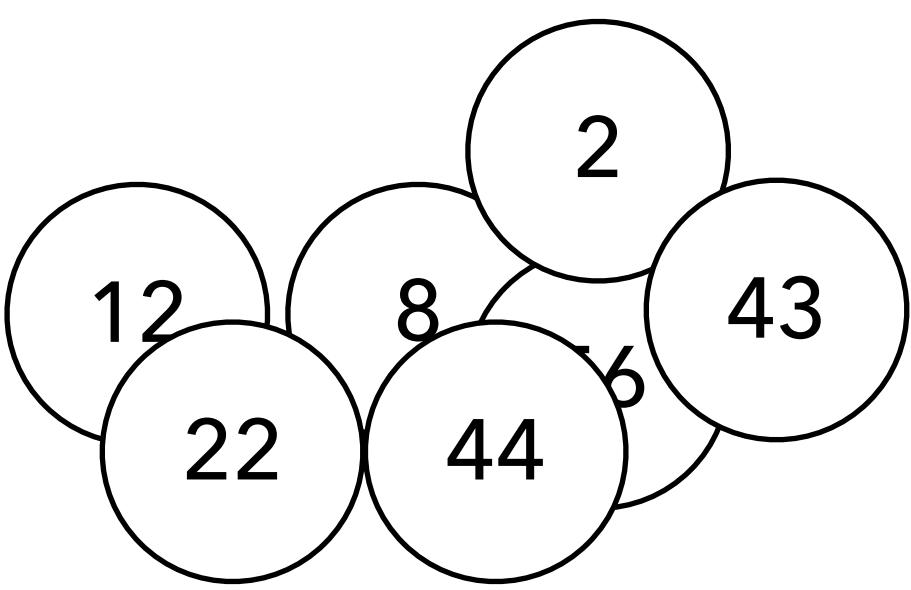
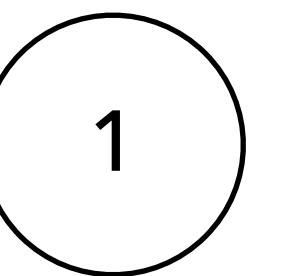
L'importance de la structure des données

Trier à la source

Arbre binaire

Enfant gauche > parent

Enfant droite < parent

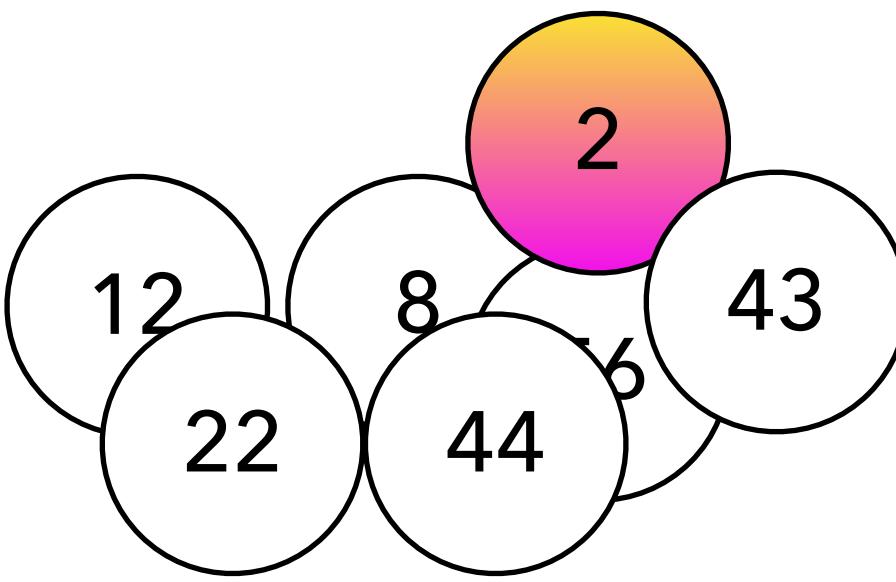
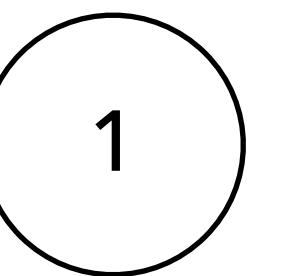


Trier à la source

Arbre binaire

Enfant gauche > parent

Enfant droite < parent

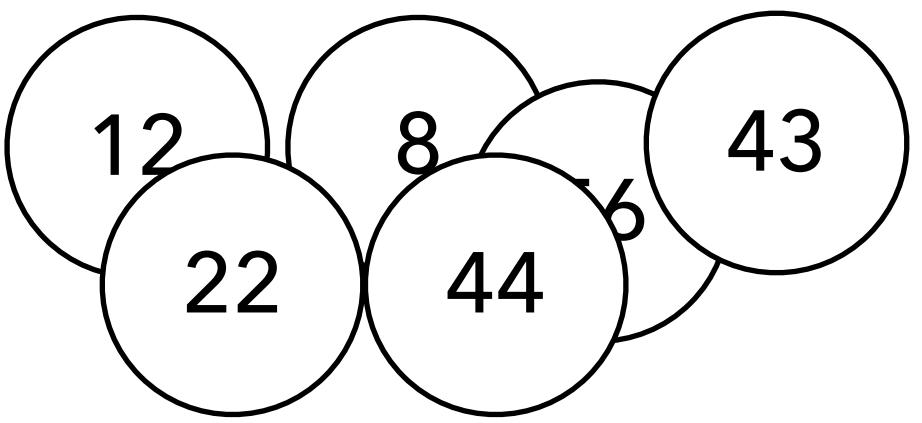
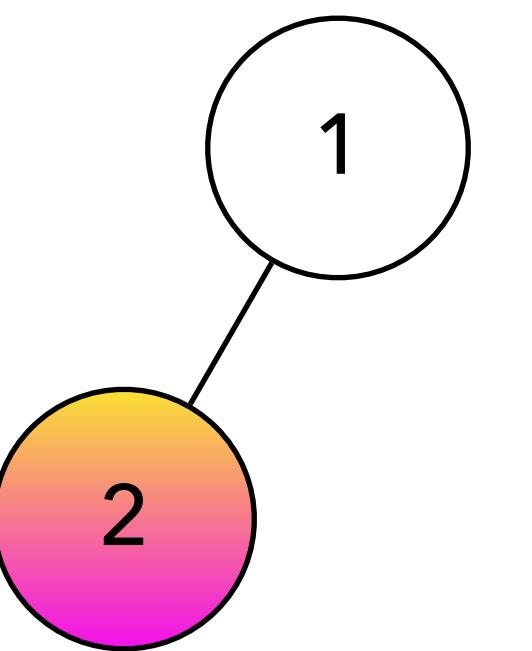


Trier à la source

Arbre binaire

Enfant gauche > parent

Enfant droite < parent

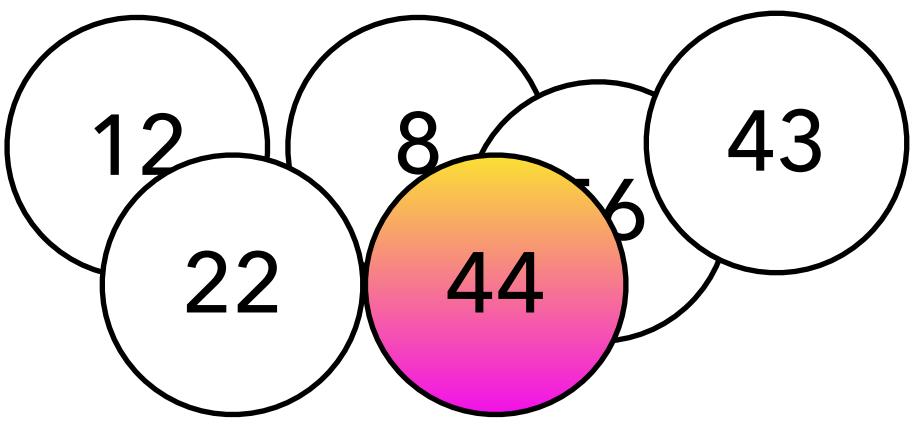
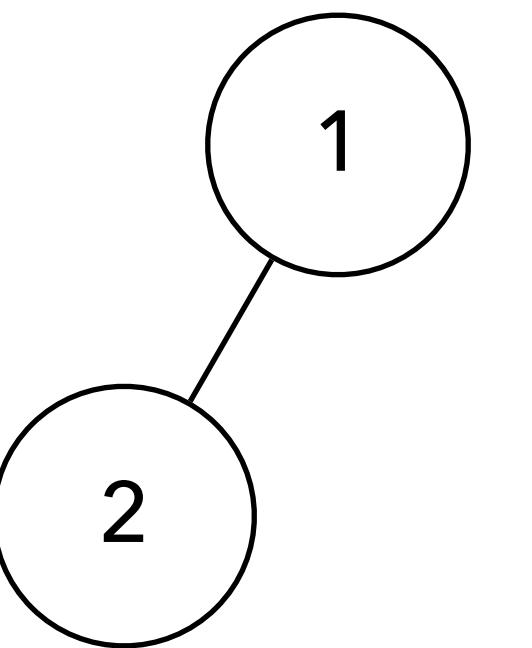


Trier à la source

Arbre binaire

Enfant gauche > parent

Enfant droite < parent



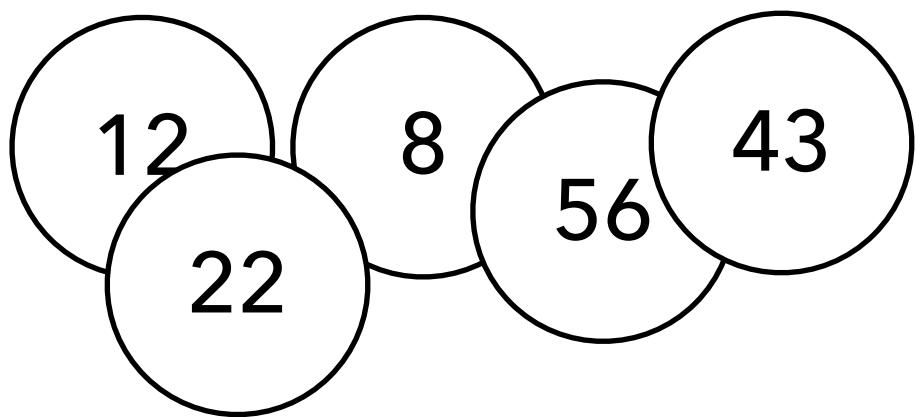
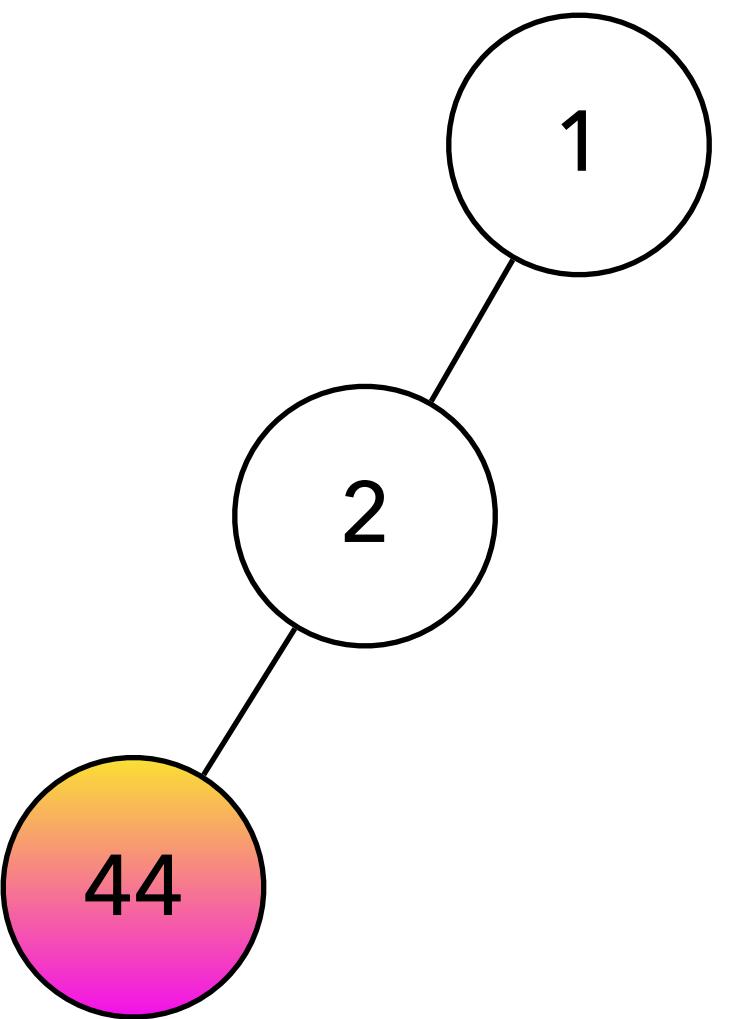
Trier à la source

Arbre binaire

Enfant gauche > parent

Enfant droite < parent

✗ Il faut respecter la structure
de l'arbre binaire



Trier à la source

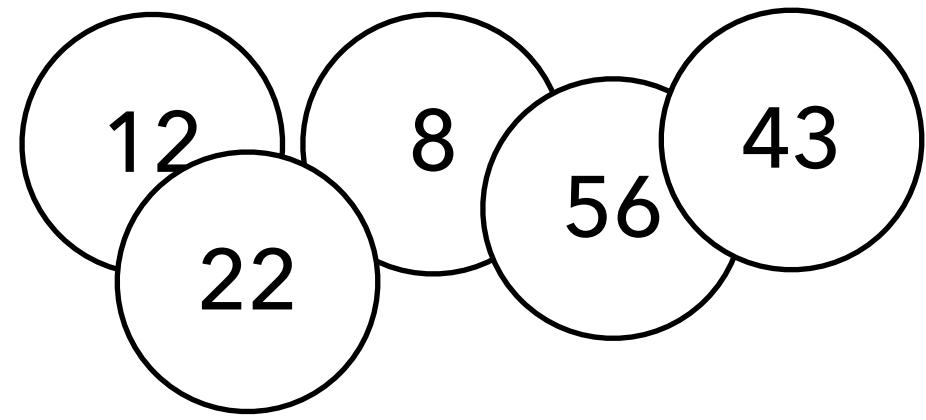
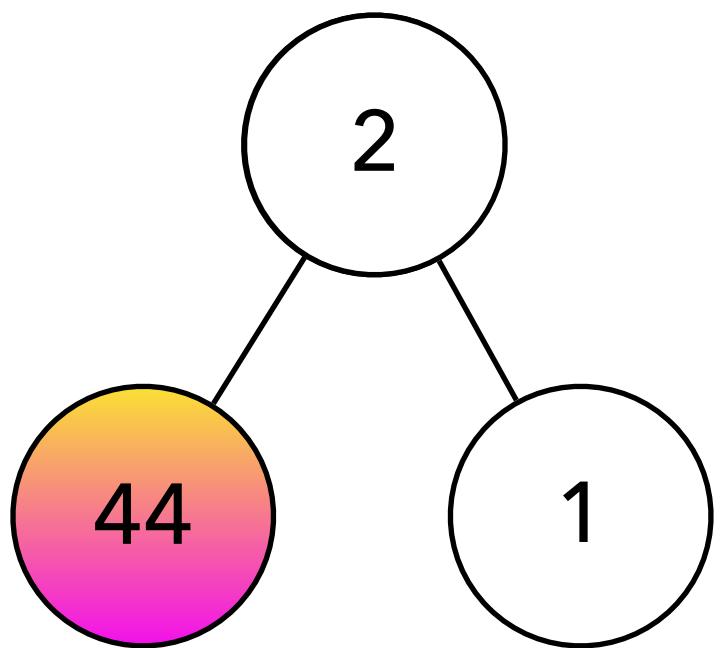
Arbre binaire

Enfant gauche > parent

Enfant droite < parent



Il faut respecter la structure
de l'arbre binaire



Trier à la source

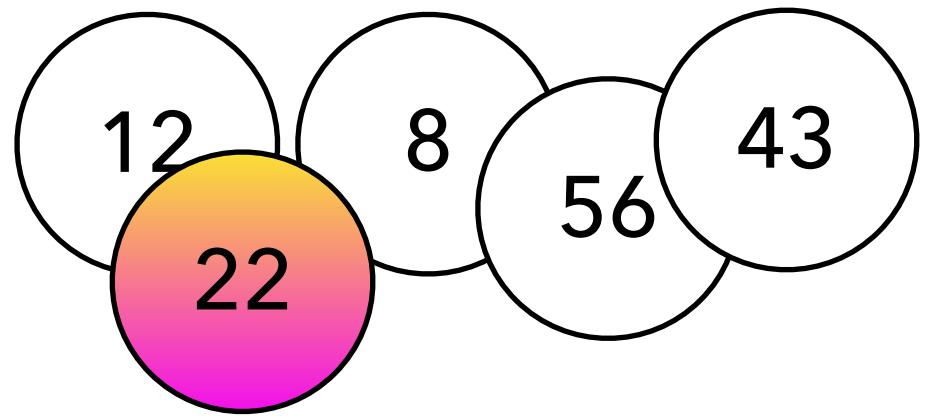
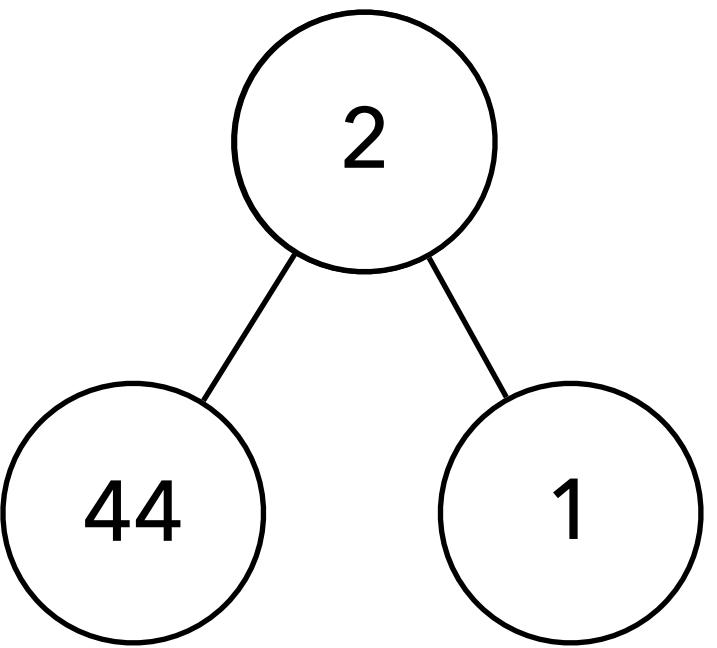
Arbre binaire

Enfant gauche > parent

Enfant droite < parent



Il faut respecter la structure
de l'arbre binaire



Trier à la source

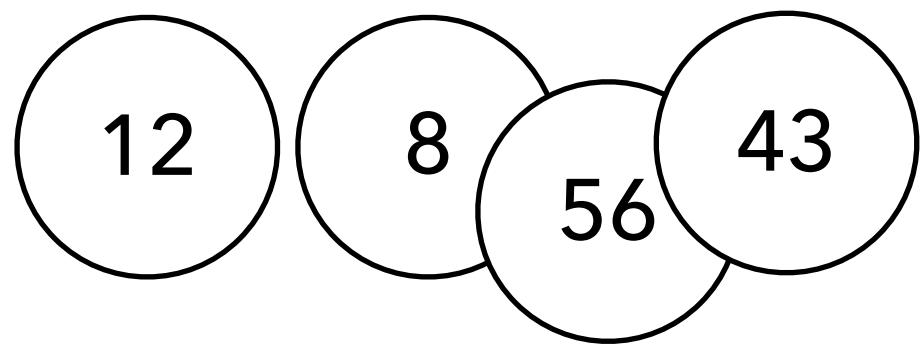
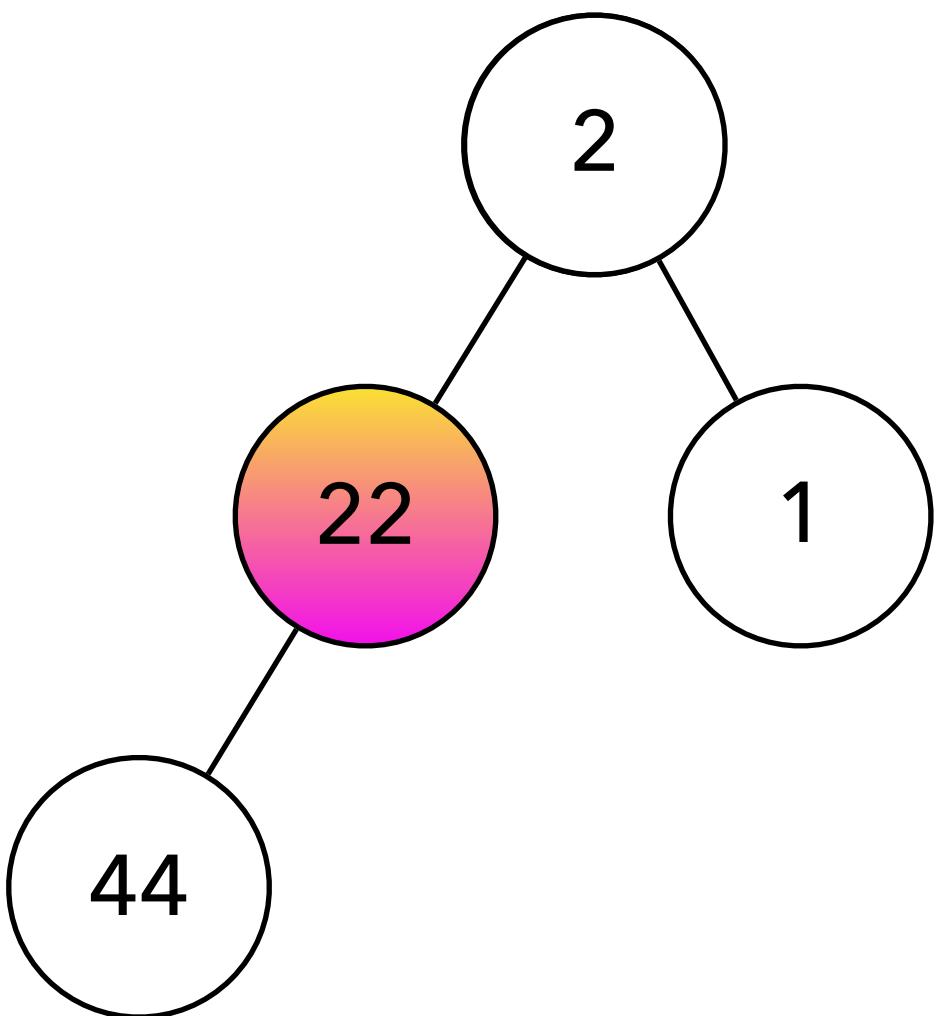
Arbre binaire

Enfant gauche > parent

Enfant droite < parent

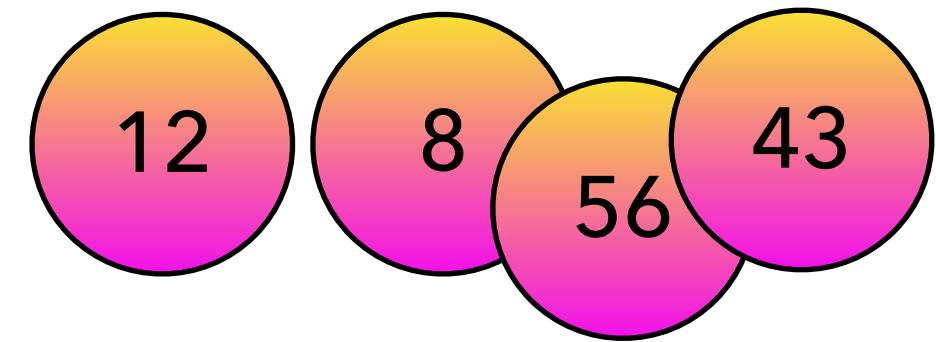


Il faut respecter la structure
de l'arbre binaire



Trier à la source

Arbre binaire

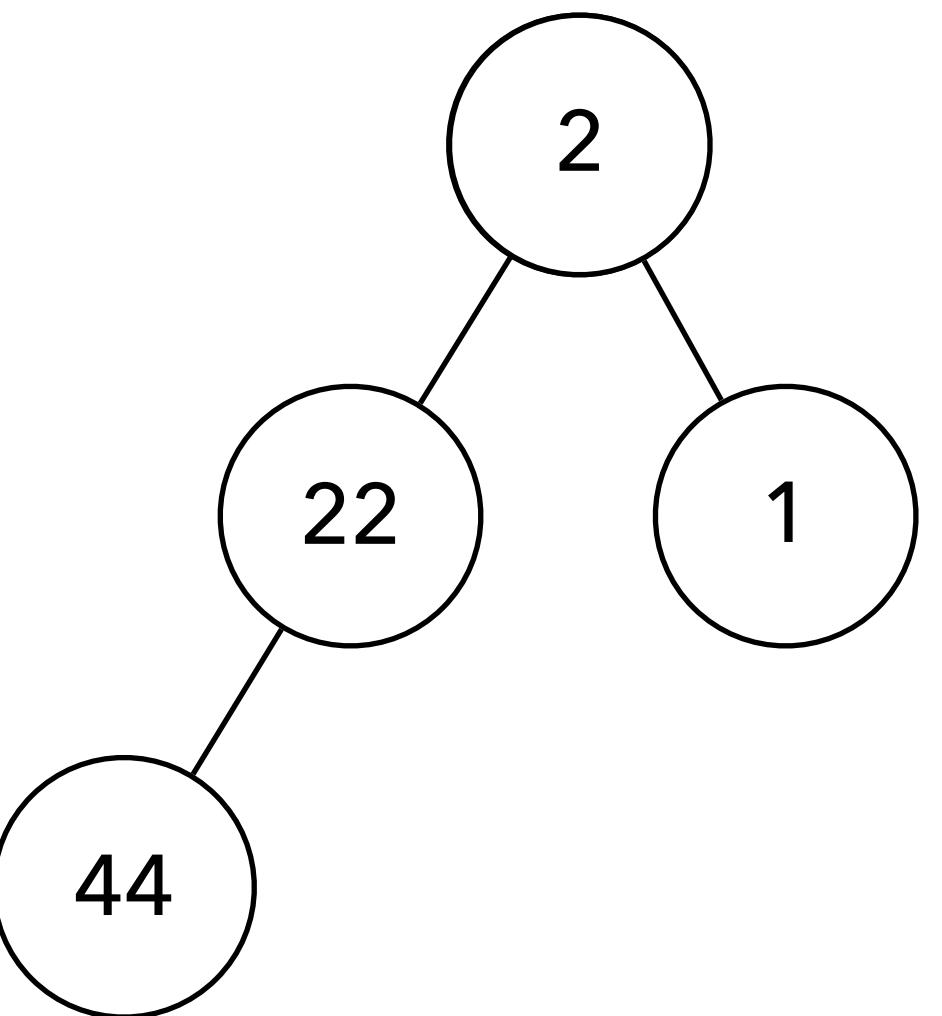


Enfant gauche > parent

Enfant droite < parent



Il faut respecter la structure
de l'arbre binaire



And so on ...

Trier à la source

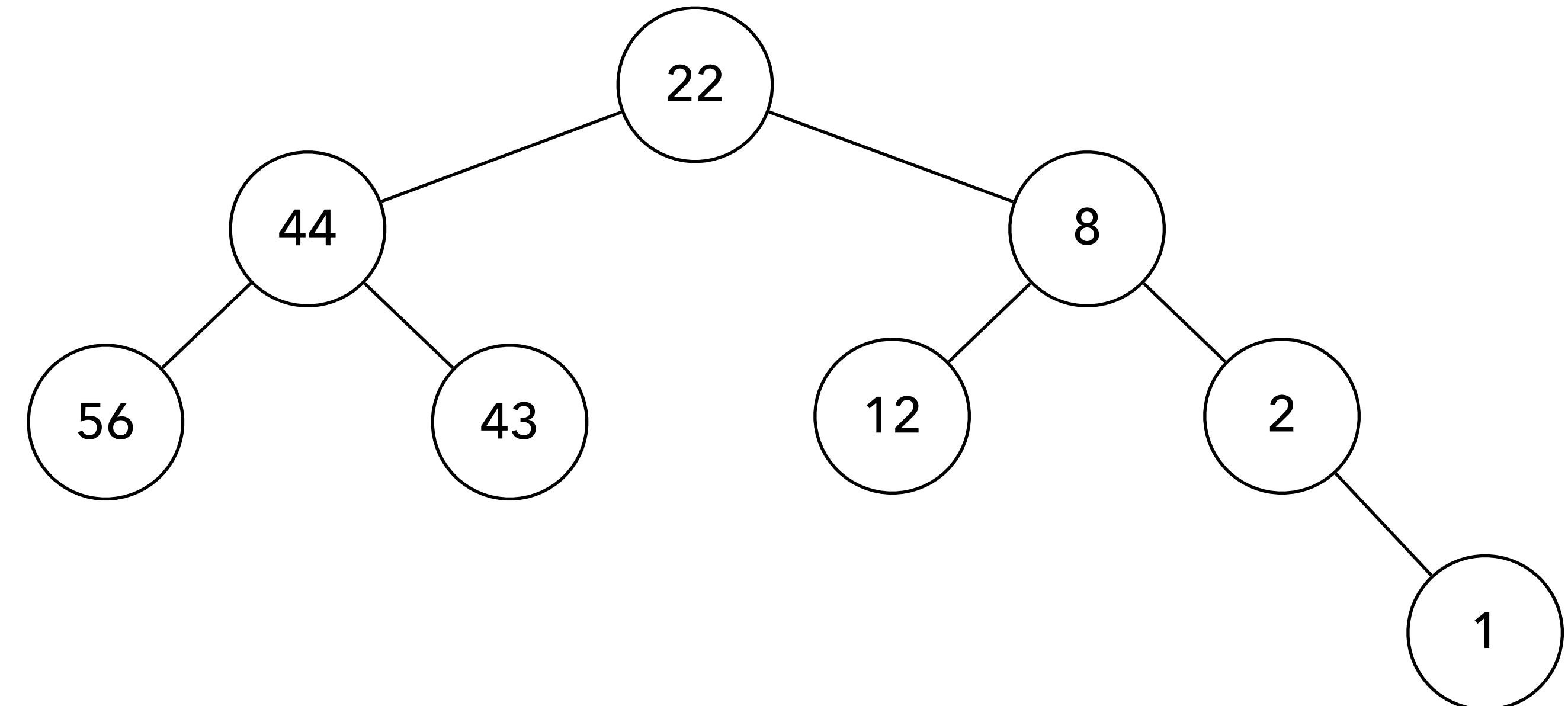
Arbre binaire

Enfant gauche > parent

Enfant droite < parent



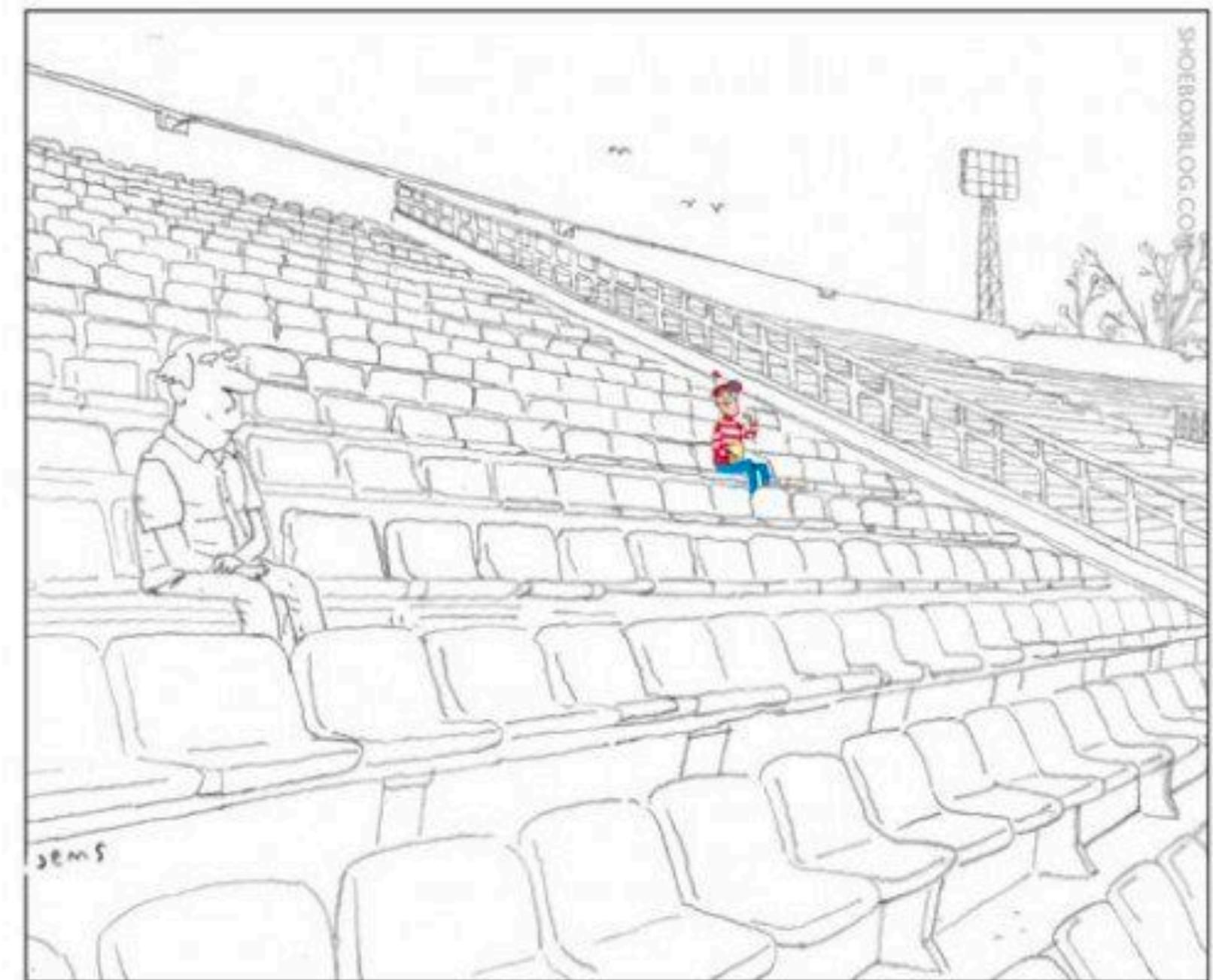
Il faut respecter la structure
de l'arbre binaire



Pourquoi trier les données ?

- Arranger les données suivant une métrique possède deux principaux avantages :
 - Une visualisation plus compréhensible
 - Une recherche plus efficace

Where's Waldo?
FOR DUMMIES



Pourquoi trier les données ?

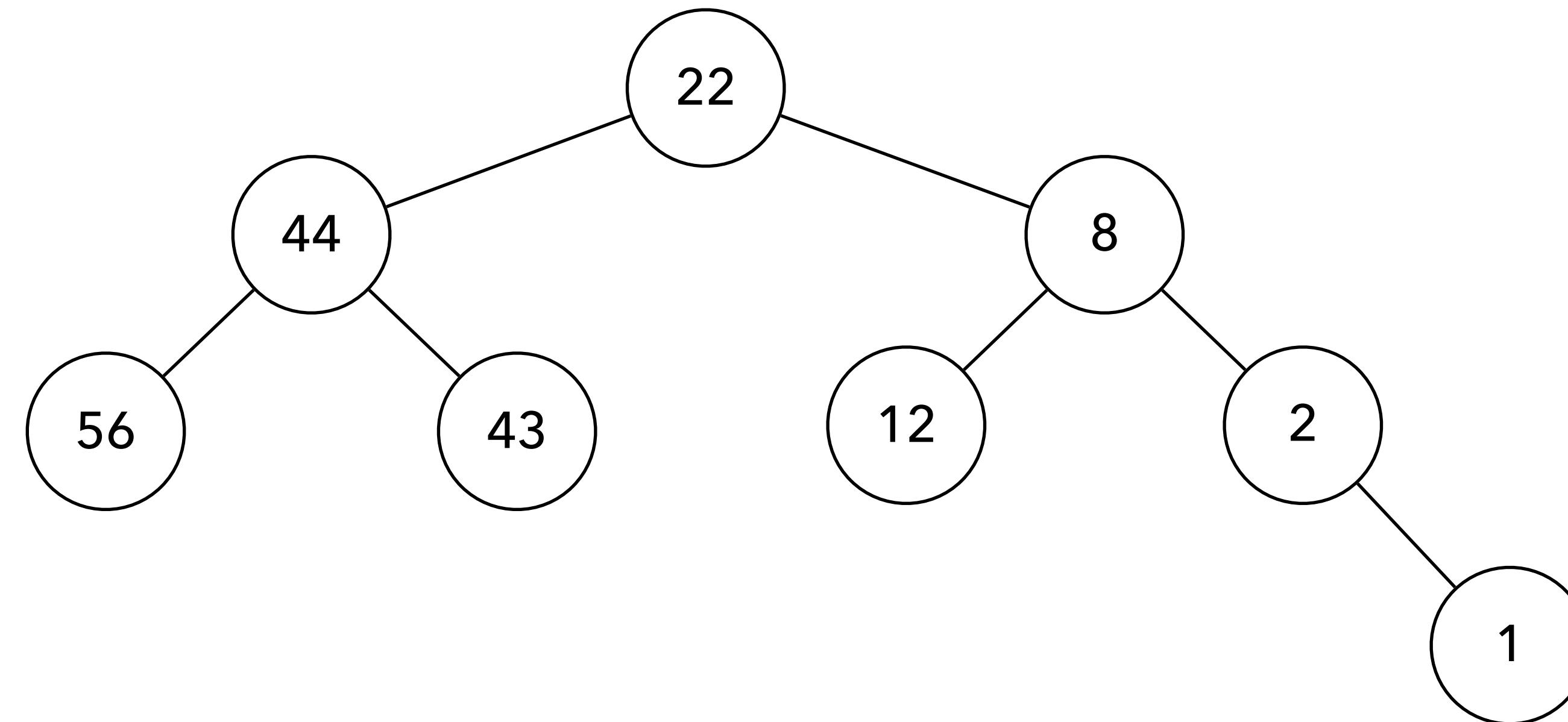
Illustrations et stratégies

- Le dictionnaire
 - Imaginez un dictionnaire où tous les mots seraient dans le désordre
- Un tableau de bord SNCF
 - Si le tri de l'affichage n'est pas fait sur l'horaire des trains et que l'ordre est aléatoire : c'est la pagaille !
- Un album de "Où est Charlie"
 - Si les personnages étaient repartis par couleur sur la page alors la recherche serait plus efficace.

Pourquoi trier les données ?

Illustrations et stratégies

- La Recherche Dichotomique



- Animation : <https://www.codingame.com/learn/binary-search>