PyRat : cours 4

L'équipe PyRat

Jalons

Séance 1	...
Séance 2	...
Séance 3	Ramasser un unique morceau de fromage dans le labyrinthe
Séance 4	...
Séance 5	...
Séance 6	Ramasser plusieurs morceaux de fromage dans le labyrinthe
Séance 7	Gagner contre un adversaire

Au programme aujourd'hui

Voyageur de commerce

- On a vu comment trouver le plus court chemin de i à j dans un graphe pondéré
- Maintenant on veut minimiser la longueur d'un chemin passant par plusieurs sommets
- Ce nouveau problème est autrement plus compliqué que le premier
- On l'appelle le problème du *voyageur de commerce*

Intuitivement

- Il est difficile de faire mieux que de tout tester
- En pratique on ne peut tout tester que quand le nombre de bouts de fromage est très faible

Le voyageur de commerce

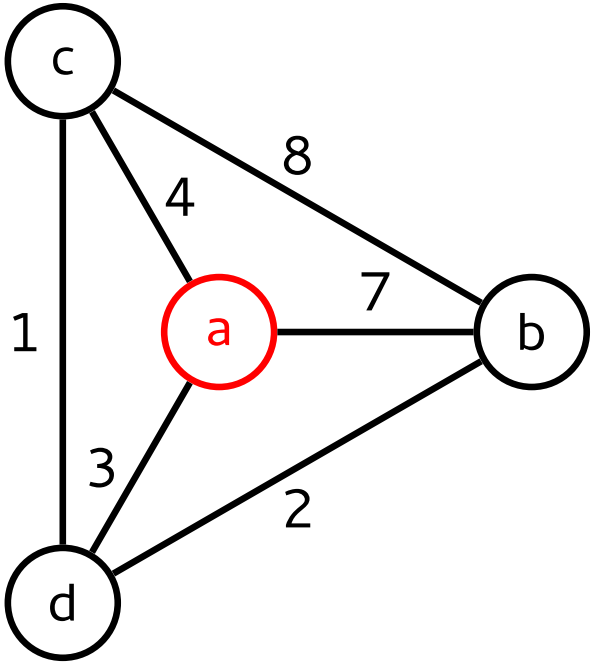
Formalisation

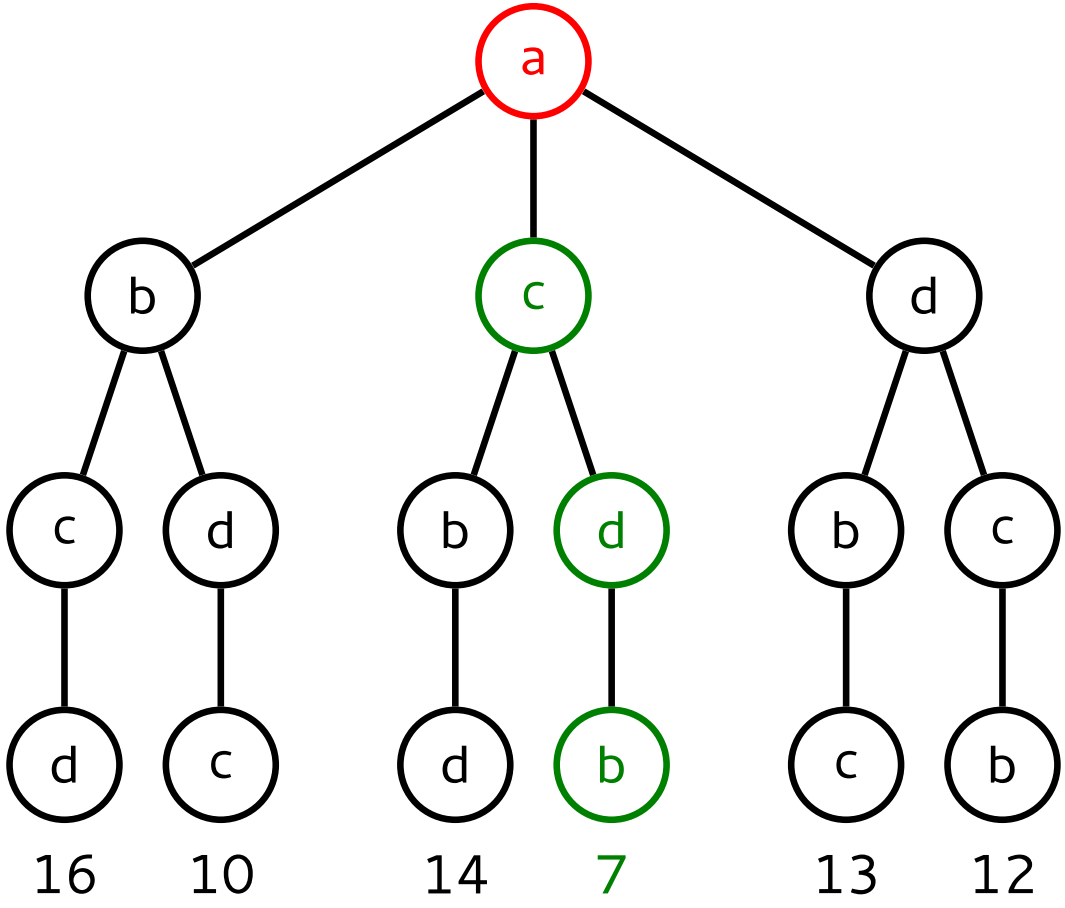
Le problème que l'on va regarder est connu sous le nom de problème du "voyageur de commerce"
Ce problème s'énonce de la façon suivante :

- Soit un graphe complet $\mathcal{G} = \langle V, E \rangle$, pondéré positivement et symétrique
- L'objectif est de trouver un chemin passant une fois par tous les sommets dans V, de longueur minimale, en partant d'un sommet donné
- Le problème est bien défini car il n'y a qu'un nombre fini de tels chemins $((|V| - 1)!)$

Exercice : trouver un exemple de graphe complet pour lequel tous les chemins possibles ont la même longueur. Construire au contraire un exemple où un chemin est significativement moins long que tous les autres.

Exemple avec un petit graphe





Méta-graphe

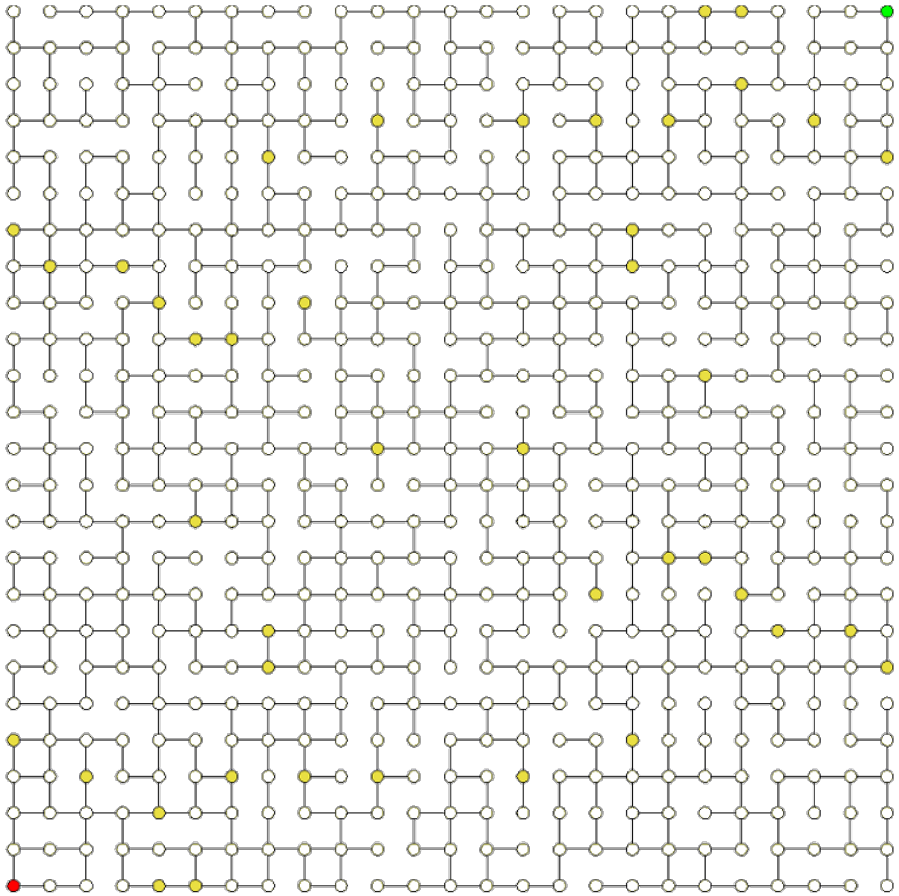
Formalisation

Il va falloir passer du labyrinthe à un graphe complet répondant aux critères du voyageur de commerce

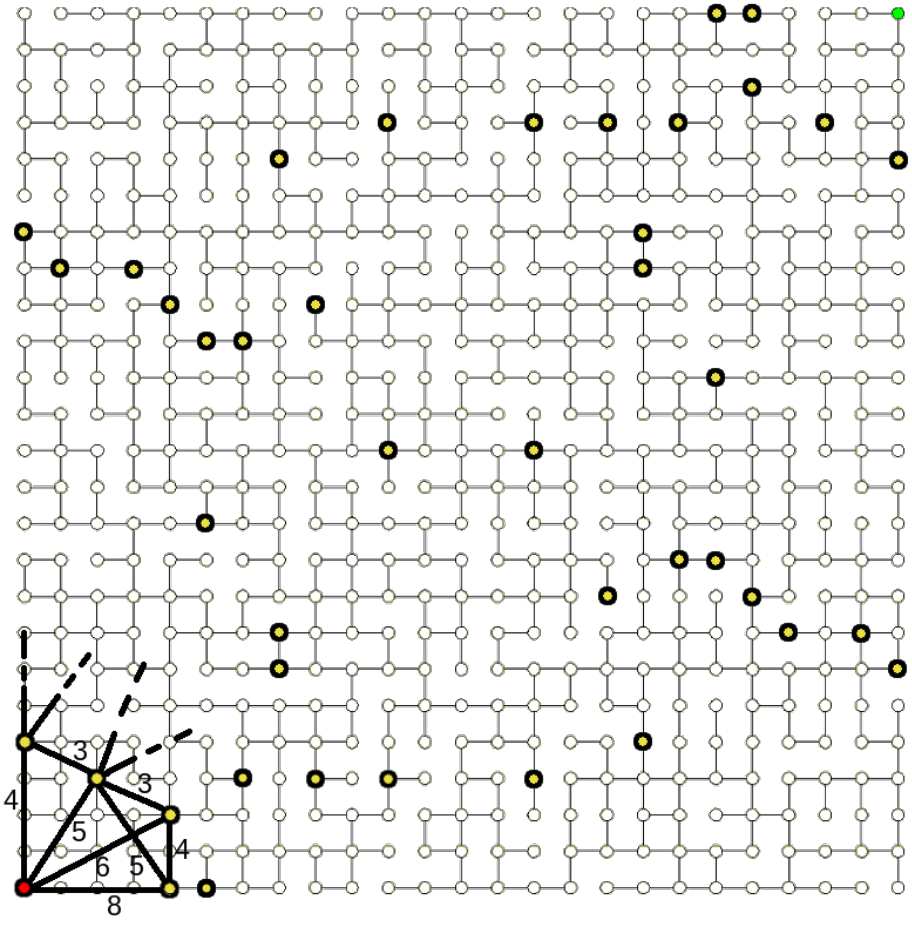
Méta-graphe

- On propose d'extraire du labyrinthe un méta-graphe
- Ce méta-graphe contient autant de sommets qu'il y a de fromages dans le labyrinthe + 1 sommet symbolisant la position de départ
- Le poids de l'arête reliant deux sommets correspond à la longueur du chemin le plus court dans le labyrinthe pour relier les cases respectives

Méta-graphe



Méta-graphe

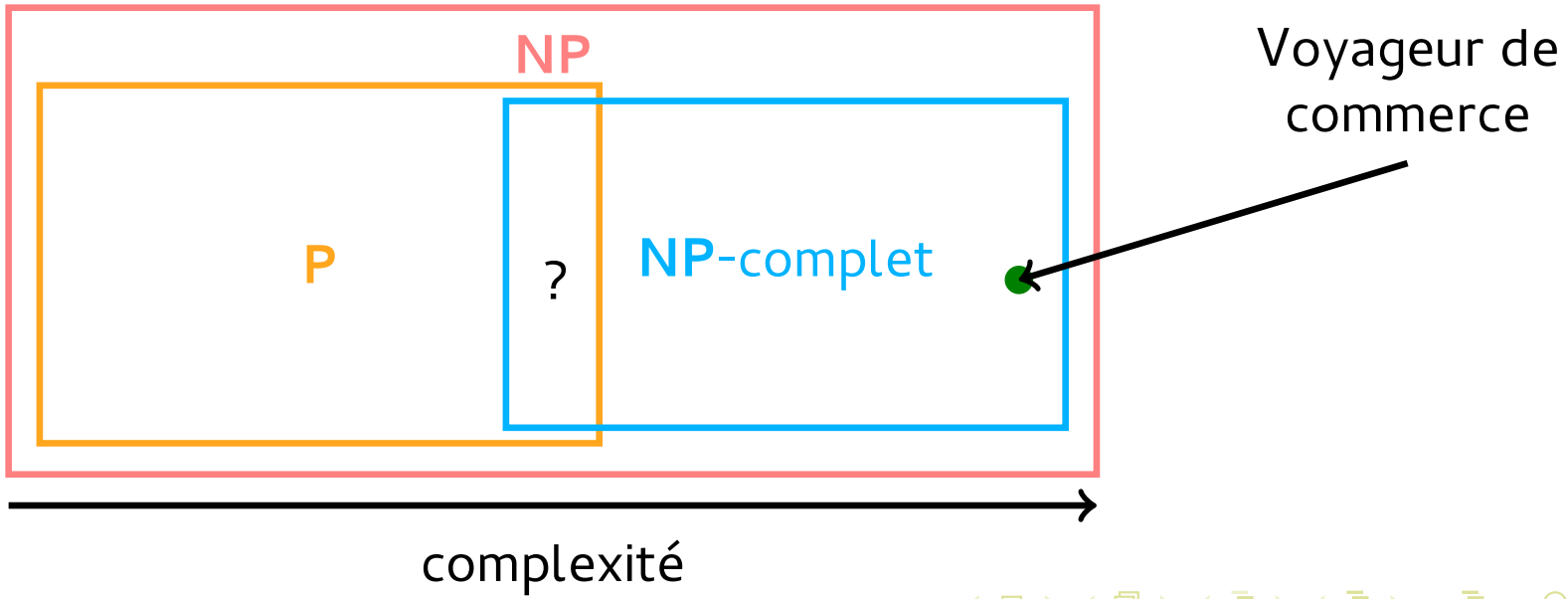


Complexité

Complexité d'un problème

- On a déjà parlé de la complexité d'un algorithme
- La complexité d'un problème est la complexité du moins complexe des algorithmes permettant de le résoudre

Quelques classes de complexité



Exercice

- Énoncer les problèmes vus en cours qui sont dans **P**

Algorithme

Algorithme exhaustif naïf

```
mieux = infinity

def exhaustif(restants, sommet, chemin, poids, graphe):
    si restants est vide:
        si poids < mieux:
            mieux = poids
            meilleur_chemin = chemin
    sinon:
        pour tout sommet i dans restants:
            exhaustif(restants - i, i, chemin + i, \\
                              poids + graphe[sommet][i],graphe)

exhaustif(liste_sommets - sommet_départ, sommet_départ, \\
                                    liste_vide, 0, graphe)
```

Complexité de l'algorithme exhaustif

Complexité

De l'ordre du nombre de chemins possibles, soit $\Theta((|V|-1)!)$

Illustration

Supposons disposer d'un calculateur qui traite 10^9 chemins par seconde

$\lvert V\rvert - 1$	10	15	20	30
temps	4ms	22min.	77ans	8.10^{13} siècles

À vous de jouer

- Jusqu'à combien de bouts de fromage arrivez-vous à répondre au problème ?