TD Algo2 – session 8 – Arbres couvrants de poids minimum

4 avril 2025

Objectifs d'apprentissage :
- trouver des exemples de graphes présentant des propriétés particulières relatives aux arbres couvrants de poids minimum ;
- analyser des propriétés formelles sur les algorithmes manipulant des arbres couvrants de poids minimum ;
- concevoir des algorithmes manipulant ces arbres couvrants.

Les 7 premiers exercices sont essentiels. Plus d'exercices sur `https://algs4.cs.princeton.edu/43mst/`.

Exercice 1 : Énumération

Combien y a t-il d'arbres couvrants de poids minimum dans un graphe complet de 4 sommets avec des pondérations unitaires.

Il y a $n(n-1)/2 = 6$ arêtes. Il faut choisir 3 arêtes parmi 6 : $6 \times 5 \times 4 = 20$. Mais il faut éliminer les 4 cas où il y a un cycle. Il y en a donc 16.

Exercice 2 : Diviser-pour-régner

Le professeur Bizard propose un nouvel algorithme diviser-pour-régner pour le calcul des arbres couvrants de poids minimum, dont voici le principe. Étant donné un graphe $G = (V, E)$, on partitionne l'ensemble V des sommets en deux ensemble V_1 et V_2 tels que $|V_1|$ et $|V_2|$ diffèrent d'au plus 1. Soit E_1 l'ensemble des arêtes qui ne sont incidentes qu'à des sommets de V_1 et soit E_2 l'ensemble des arêtes qui ne sont incidentes qu'à des sommets de V_2. On résout alors récursivement un problème d'arbres couvrants de poids minimum pour chacun des deux sous-graphes $G_1 = (V_1, E_1)$ et $G_2 = (V_2, E_2)$. Enfin, on sélectionne l'arête de poids minimal de E qui traverse la coupe (V_1, V_2) et l'on utilise cette arête pour réunifier en un seul arbre couvrant les deux arbres couvrants minimum résultants. Donner un exemple simple de graphe connexe pour lequel cet algorithme ne calcule pas toujours un arbre couvrant de poids minimum.

Graphe avec 4 sommets : a, b, c et d. $w(a,b) = w(c,d) = 1$, $w(a,c) = w(b,d) = 10$ et $w(a,d) = w(b,c) = 2$. On commence avec $V_1 = \{a,c\}$ et $V_2 = \{b,d\}$. On sera amené à connecter a avec c et b avec d pour un arbre couvrant de poids 21 alors que le poids minimum est 4.

Exercice 3 : Transformation affine

Soit un arbre couvrant de poids minimum. Après l'application d'un facteur multiplicatif positif sur les poids, cet arbre peut-il ne plus être de poids minimum ? Même question si on ajoute une valeur positive constante à tous les poids.

On considère l'arbre construit par l'algorithme de Kruskal. Celui-ci ne considère les poids que pour ordonner les arêtes. Si les poids subissent une transformation affine (facteur multiplicatif plus une constante), l'ordre n'est pas impacté et l'arbre couvrant de poids minimum résultat reste donc le même.

Exercice 4 : Arête de poids minimal

Soit (u, v) une arête de poids minimal unique dans un graphe G. Cette arête (u, v) appartient à un arbre couvrant de poids minimum. Proposer un argument pour expliquer pourquoi.

> C'est la première arête considérée par l'algorithme de Kruskal et comme tous les sommets sont dans des ensembles séparés à ce stade, elle est toujours conservée.

Exercice 5 : Arbre couvrant de poids maximum

Concevoir un algorithme pour trouver un arbre couvrant de poids maximum.

ACM-Maximum(G, w)

pour $(u, v) \in G.E$ **faire**
  $w'(u, v) \leftarrow -w(u, v)$
retourner ACM-Kruskal(G, w')

Exercice 6 : Irrigation

Dans ce problème issu de l'olympiade informatique des États-Unis (*USA Computing Olympiad*), on se situe dans une ville avec N maisons qui ont toutes besoin d'être approvisionnées en eau. Construire un puits dans la maison i coûte $p[i]$ et raccorder la maison i à la maison j coûte $c[i][j]$. Une maison peut recevoir de l'eau soit via son propre puits, soit s'il y a une maison ayant un puits et pour laquelle il existe un chemin depuis la maison (directement ou indirectement). Concevoir un algorithme pour trouver le coût minimal pour approvisionner chaque maison en eau.

Irrigation(c, p)

$G \leftarrow \{0, 1, \ldots, N\}$
pour $i = 1$ **jusqu'à** N **faire**
  pour $j = 1$ **jusqu'à** N **faire**
    si $i \neq j$ **alors**
      $w(i, j) \leftarrow c[i][j]$
pour $i = 1$ **jusqu'à** N **faire**
  $w(0, i) \leftarrow p[i]$
retourner ACM-Kruskal(G, w)

Exercice 7 : Diminution

Étant donné un graphe G et un arbre couvrant de poids minimum T, on suppose que l'on diminue le poids de l'une des arêtes qui n'est pas dans T. Donner un algorithme en temps $O(V)$ qui permet de trouver l'arbre couvrant de poids minimum du graphe ainsi modifié. Même question si l'on augmente le poids.

On va ajouter l'arête, identifier le cycle créé et enlever l'arête de poids maximum sur ce cycle. On prouverait la validité de l'approche par l'absurde.

ACM-Diminuer(G, w, T, e)

pour $(u, v) \in T$ **faire**
  $u.neigh \leftarrow u.neigh \cup \{v\}$
  $v.neigh \leftarrow v.neigh \cup \{u\}$
Partir d'un sommet arbitraire et initialiser récursivement $v.parent$ en parcourant $v.neigh$
Parcourir chaque sommet de e vers la racine
Le cycle est l'ensemble des arêtes rencontrées une fois plus e
On supprime l'arête de poids maximum parmi ces arêtes

Si l'arête n'était pas dans T, il n'est pas possible qu'elle y soit si son poids augmente.

Exercice 8 : Arête minimale

À chaque itération de l'algorithme `ACM-Générique`, on rajoute une arête minimale qui traverse une coupe. Considérons toutes les arêtes minimale qui sont minimales pour au moins une coupe quelconque. Donner un exemple simple d'un graphe connexe pour lequel cet ensemble ne forme pas un arbre couvrant de poids minimum.

Indice : il suffit qu'une seule des propriétés d'un arbre couvrant de poids minimum ne soit pas respectée.

> Trois sommets a, b et c avec $w(a,b) = w(b,c) = w(a,c) = 1$.

Exercice 9 : Réciproque

S'il existe une arête minimale unique traversant chaque coupe du graphe, alors un graphe possède un arbre couvrant de poids minimum unique. Cependant, la réciproque n'est pas vraie. Donner un exemple simple de graphe connexe pour lequel l'arbre couvrant de poids minimum est unique, mais pour lequel il n'existe pas une arête minimale unique pour chaque coupe.

> Trois sommets a, b et c avec $w(a,b) = w(b,c) = 1$ (pas d'arête entre a et c). Pour la coupe $(\{b\},\{a,c\})$, les deux arêtes sont minimales et doivent être présentes toutes les deux dans l'arbre couvrant.

Exercice 10 : Arête de poids maximal

Soit e une arête de poids maximal d'un cycle de $G = (V,E)$. Il existe un arbre couvrant de poids minimum de $G' = (V, E \setminus e)$ qui est également un arbre couvrant de poids minimum de G. En d'autres termes, il existe un arbre couvrant de poids minimum de G qui ne contient pas e. Proposer un argument pour expliquer pourquoi.

> Comme e est sur un cycle, toutes les arêtes de ce cycle ne peuvent pas toutes être présentes dans un arbre couvrant de poids minimum. Comme e est celle de poids maximum, il est idéal de ne pas la conserver.

Exercice 11 : Suppression

Soit T un arbre couvrant de poids minimum associé au graphe G. Supposons que l'on enlève de G une arête qui ne déconnecte pas G. Décrire comment trouver l'arbre couvrant de poids minimum du nouveau graphe en temps $O(E)$.

> Si l'arête n'est pas dans T, alors T reste un arbre couvant de poids minimum pour le nouveau graphe. Sinon, la suppression d'une arête sépare l'arbre couvrant en deux composantes.
>
> ACM-Supprimer(G, w, T, e)
>
> **si** $e \in T$ **faire**
>   **pour** $(u,v) \in T$ **faire**
> $u.neigh \leftarrow u.neigh \cup \{v\}$
> $v.neigh \leftarrow v.neigh \cup \{u\}$
>   Partir d'un sommet arbitraire et initialiser récursivement $v.parent$ en parcourant $v.neigh$
> $T \leftarrow T \setminus \{e\}$
>   Parcourir chaque sommet de e vers les 2 racines
>   Partir des 2 racines et initialiser récursivement $v.composante$ en parcourant $v.neigh$
>   Ajouter l'arête de poids minimum avec une extrémité dans chaque composante