

Partie IV – Arbres et forêts

Graphes et Algorithmes – GRAAL

Gaëtan Staquet
gaetan.staquet@ec-nantes.fr

École Centrale de Nantes – LS2N
S508

Janvier à mars 2026

1. Arbres
2. Arbre couvrant de poids minimum
3. Un petit tour en forêt

1. Arbres

1. Arbres orientés et non orientés
2. Forêts
3. Isthmes

2. Arbre couvrant de poids minimum

3. Un petit tour en forêt

Des graphes particuliers

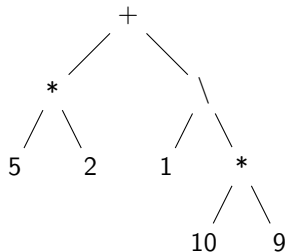


Figure IV.1 – Une expression arithmétique.

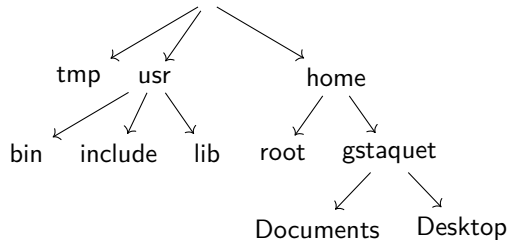


Figure IV.2 – Une arborescence de fichiers.

Réseau de communication

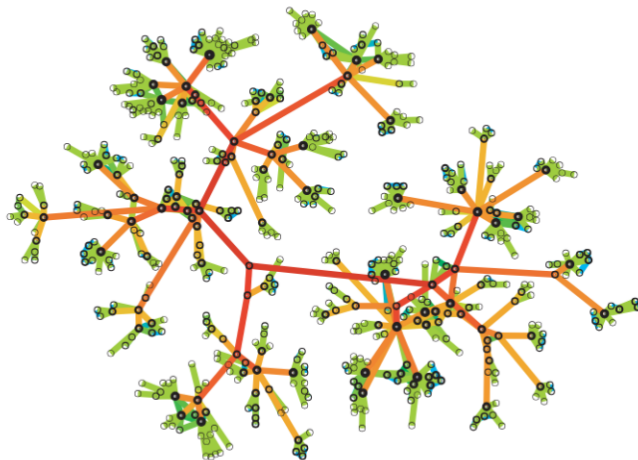


Figure IV.3 – Visualisation d'un réseau de communication mobile centré sur un individu ONNELA et al., « Structure and tie strengths in mobile communication networks », 2007.

Représentation des connaissances

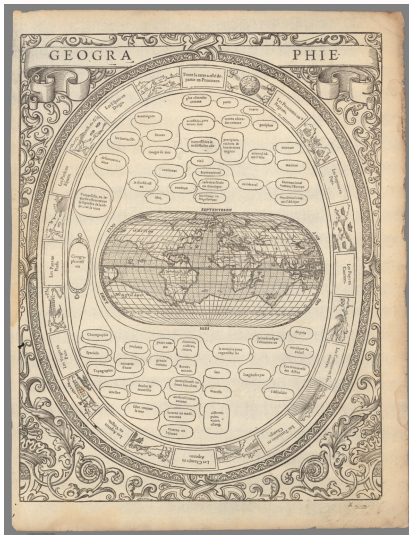


Figure IV.4 – *Géographie*, d'après Christophe de Savigny, *Tableaux accomplis*, 1587. Cette photographie provient de David Rumsey Map Collection, David Rumsey Map Center, Stanford Libraries.

Système figuré des connaissances

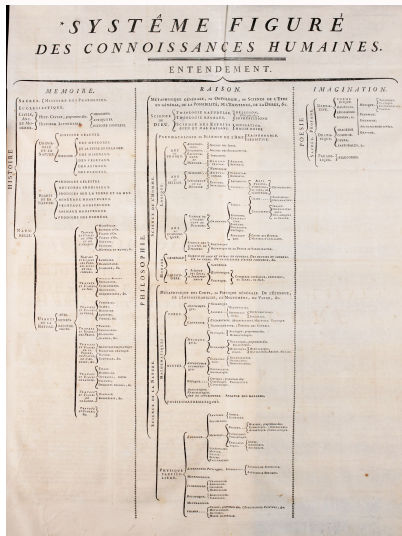


Figure IV.5 – *Système figuré des connaissances humaines*, d'après Diderot et d'Alembert, *Encyclopédie*, 1751.

Problème des 4 reines

On a un échiquier de 4×4 cases. On veut placer 4 reines telles qu'elles ne peuvent pas s'attaquer. Autrement dit, pour toute paire de reines distinctes, elles ne peuvent pas être sur la même ligne, la même colonne ou la même diagonale.

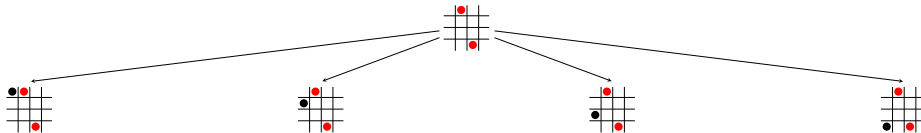
Problème des 4 reines

On a un échiquier de 4×4 cases. On veut placer 4 reines telles qu'elles ne peuvent pas s'attaquer. Autrement dit, pour toute paire de reines distinctes, elles ne peuvent pas être sur la même ligne, la même colonne ou la même diagonale.



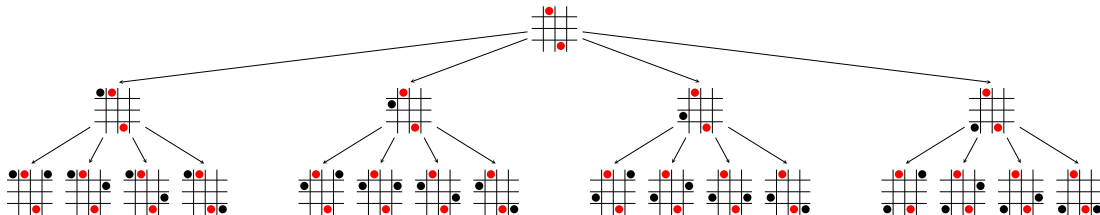
Problème des 4 reines

On a un échiquier de 4×4 cases. On veut placer 4 reines telles qu'elles ne peuvent pas s'attaquer. Autrement dit, pour toute paire de reines distinctes, elles ne peuvent pas être sur la même ligne, la même colonne ou la même diagonale.



Problème des 4 reines

On a un échiquier de 4×4 cases. On veut placer 4 reines telles qu'elles ne peuvent pas s'attaquer. Autrement dit, pour toute paire de reines distinctes, elles ne peuvent pas être sur la même ligne, la même colonne ou la même diagonale.



Arbre non orienté

Définition IV.1 (Arbre non orienté). Un **arbre non orienté** est un graphe non orienté connexe et acyclique.

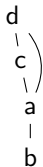


Figure IV.6 – Pas un arbre : le graphe a un cycle.

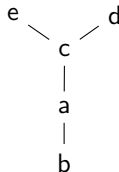


Figure IV.7 – Un arbre non orienté.

Exercice IV.2. Soit un arbre $\mathcal{T} = (V, E)$ non orienté et fini. Montrez que, si $|V| \geq 2$, alors il existe au moins deux sommets de degré 1.

Indice : prenez une chaîne élémentaire¹ de longueur maximale et regardez ses extrémités.

1. Chaque sommet apparaît au plus une fois.

Arbre orienté

Définition IV.3 (Arbre). Un **arbre orienté** est un tuple $\mathcal{T} = (V, E, r)$ où

- ▶ (V, E) est un graphe orienté **acyclique** ;
- ▶ $r \in V$ est un sommet distingué, appelé **racine**, à partir duquel on peut atteindre n'importe quel autre sommet ;
- ▶ le degré entrant de la racine est zéro, *i.e.*, $\rightarrow\delta(r) = 0$; et
- ▶ pour tout sommet $v \neq r$, le degré entrant de v est égal à un, *i.e.*, $\rightarrow\delta(v) = 1$.

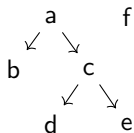


Figure IV.8 – Pas un arbre : il n'existe pas de chemin de a vers f .

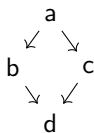


Figure IV.9 – Pas un arbre : f a un degré entrant > 1 .

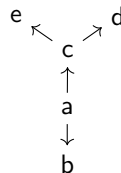
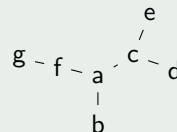


Figure IV.10 – Un arbre orienté.

Arbres – Exercices

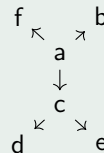
Exercice IV.4.

Est-ce que le graphe à côté est un arbre non orienté ?
Si on construit le graphe orienté correspondant, a-t-on un arbre orienté ?



Exercice IV.5.

Est-ce que le graphe à côté est un arbre avec racine c ? Est-ce qu'un autre sommet pourrait servir de racine ?



Arbres – Exercices

Exercice IV.6. Prouvez que, pour tout arbre non orienté $\mathcal{T} = (V, E)$ fini avec $|V| \geq 1$, on a $|E| = |V| - 1$.

Indice : par récurrence sur $|V|$.

Exercice IV.7. Montrez que, pour tout graphe non orienté $\mathcal{G} = (V, E)$, \mathcal{G} est un arbre si et seulement si, pour toute paire de sommets u et v quelconques, il existe une unique chaîne élémentaire de u à v .

Indice pour \Rightarrow : qu'est-ce qui justifie l'existence d'une chaîne (élémentaire) ? Si on suppose qu'il existe deux chaînes distinctes de u à v , que peut-on conclure ?

Indice pour \Leftarrow : que se passerait-il si on avait un cycle ?

Exploiter la structure

Étant donné la structure particulière des arbres, plusieurs problèmes génériques ont des solutions plus efficaces pour les arbres que pour les graphes :

- ▶ fermeture transitive ;
- ▶ parcourir les sommets ;
- ▶ trouver un chemin optimal ;
- ▶ *etc.*

Ici, on va plutôt s'intéresser à des propriétés spécifiques des arbres.

Forêt non orientée

Définition IV.8 (Forêt non orientée). Une **forêt non orientée** est un graphe non orienté acyclique.

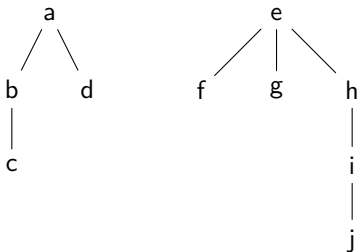


Figure IV.11 – Une forêt composée de deux arbres.

Forêt non orientée

Définition IV.8 (Forêt non orientée). Une **forêt non orientée** est un graphe non orienté acyclique.

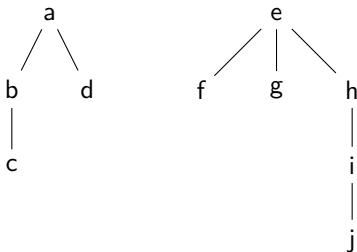


Figure IV.11 – Une forêt composée de deux arbres.

Les composantes connexes d'une forêt non orientée sont des arbres.

Isthme

Définition IV.9 (Isthme). Un **isthme** d'un graphe non orienté $\mathcal{G} = (V, E)$ est une arête e telle que $\mathcal{G}' = (V, E \setminus \{e\})$ a une composante connexe de plus que \mathcal{G} .

Autrement dit, dans \mathcal{G}' , les extrémités de e ne sont plus reliées par une chaîne.

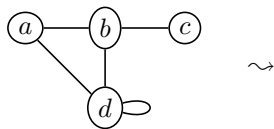


Figure IV.12 – $\{b, d\}$ n'est pas un isthme.

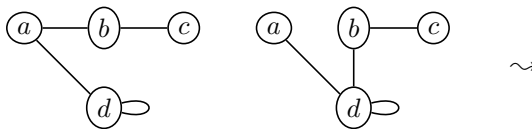


Figure IV.13 – $\{b, d\}$ est un isthme.

Isthme – Exercices

Exercice IV.10. Décrivez un algorithme qui liste tous les isthmes d'un graphe non orienté donné.
Rappel : pour les graphes non orientés, un DFS/BFS étendu suffit à identifier les composantes connexes.

Lemme IV.11. Une arête d'un graphe non orienté est un isthme si et seulement si elle n'appartient pas à un cycle.

Exercice IV.12. À partir du lemme précédent, montrez que toute arête d'un arbre non orienté est un isthme.

Caractérisation des arbres

Théorème IV.13. Les conditions suivantes pour un graphe non orienté $\mathcal{G} = (V, E)$ sont équivalentes :

- ▶ \mathcal{G} est un arbre.
- ▶ \mathcal{G} est connexe et $|E| = |V| - 1$.
- ▶ \mathcal{G} est acyclique et $|E| = |V| - 1$.
- ▶ \mathcal{G} est connexe et toute arête est un isthme.
- ▶ Deux sommets quelconques de \mathcal{G} sont reliés par une chaîne élémentaire unique.

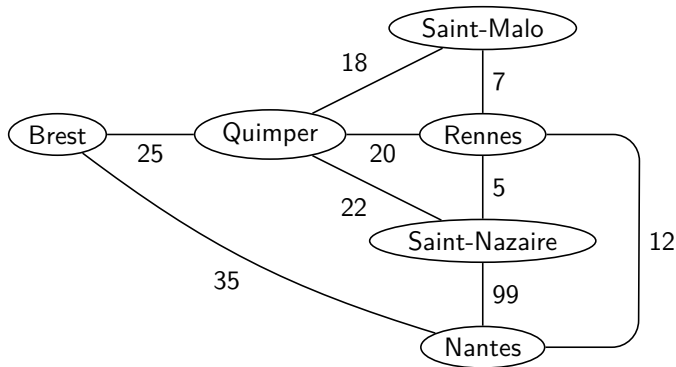
1. Arbres

2. Arbre couvrant de poids minimum

1. Problème : couvrir tous les sommets
2. Arbre couvrant
3. Arbre couvrant de poids minimum

3. Un petit tour en forêt

Connecter des villes



On veut trouver comment connecter toutes les villes en minimisant le coût total des arêtes choisies.

Graphe pondéré

Définition IV.14. Un **graphe pondéré** est un tuple $\mathcal{G} = (V, E, p)$ où

- ▶ (V, E) est un graphe (orienté ou non) et
- ▶ $p : E \rightarrow \mathbb{R}$ est une fonction qui associe un poids à chaque arc.

Le problème de l'arbre couvrant de poids minimum

Pour un graphe pondéré $\mathcal{G} = (V, E, p)$, notons $p(\mathcal{G}) = \sum_{e \in E} p(e)$.

Le problème de l'arbre couvrant de poids minimum. Soit un graphe **non orienté** pondéré **connexe** $\mathcal{G} = (V, E, p)$ avec $p : E \rightarrow \mathbb{R}^{>0}$ (*i.e.*, que des valeurs **strictement positives**). On veut trouver un sous-graphe $\mathcal{G}' = (V', E', p)$ de \mathcal{G} tel que

- ▶ \mathcal{G}' est couvrant (*i.e.*, $V' = V$),
- ▶ \mathcal{G}' est connexe et
- ▶ le poids total des arêtes de \mathcal{G}' est minimum, *i.e.*,

$$p(\mathcal{G}') = \sum_{e \in E'} p(e) \text{ est minimum.}$$

Exercice IV.15. Argumentez qu'une solution \mathcal{G}' est nécessairement un arbre.

Indice : par l'absurde.

Arbre couvrant

Définition IV.16 (Arbre couvrant). Soit $\mathcal{G} = (V, E)$ un graphe non orienté connexe. Un **arbre couvrant** de \mathcal{G} est un sous-graphe $\mathcal{T} = (V', E')$ tel que

- ▶ \mathcal{T} est un arbre (*i.e.*, un graphe connexe et acyclique),
- ▶ \mathcal{T} est un sous-graphe couvrant (*i.e.*, $V' = V$).

Arbre couvrant

Définition IV.16 (Arbre couvrant). Soit $\mathcal{G} = (V, E)$ un graphe non orienté connexe. Un **arbre couvrant** de \mathcal{G} est un sous-graphe $\mathcal{T} = (V', E')$ tel que

- ▶ \mathcal{T} est un arbre (*i.e.*, un graphe connexe et acyclique),
- ▶ \mathcal{T} est un sous-graphe couvrant (*i.e.*, $V' = V$).



Figure IV.14 – Un graphe non orienté. Les arêtes en **gras** forment un arbre couvrant.

Arbre couvrant

Définition IV.16 (Arbre couvrant). Soit $\mathcal{G} = (V, E)$ un graphe non orienté connexe. Un **arbre couvrant** de \mathcal{G} est un sous-graphe $\mathcal{T} = (V', E')$ tel que

- ▶ \mathcal{T} est un arbre (*i.e.*, un graphe connexe et acyclique),
- ▶ \mathcal{T} est un sous-graphe couvrant (*i.e.*, $V' = V$).



Figure IV.14 – Un graphe non orienté. Les arêtes en **gras** forment un arbre couvrant.

Arbre couvrant

Définition IV.16 (Arbre couvrant). Soit $\mathcal{G} = (V, E)$ un graphe non orienté connexe. Un **arbre couvrant** de \mathcal{G} est un sous-graphe $\mathcal{T} = (V', E')$ tel que

- ▶ \mathcal{T} est un arbre (*i.e.*, un graphe connexe et acyclique),
- ▶ \mathcal{T} est un sous-graphe couvrant (*i.e.*, $V' = V$).



Figure IV.14 – Un graphe non orienté. Les arêtes en **gras** forment un arbre couvrant.

Arbre couvrant

Définition IV.16 (Arbre couvrant). Soit $\mathcal{G} = (V, E)$ un graphe non orienté connexe. Un **arbre couvrant** de \mathcal{G} est un sous-graphe $\mathcal{T} = (V', E')$ tel que

- ▶ \mathcal{T} est un arbre (*i.e.*, un graphe connexe et acyclique),
- ▶ \mathcal{T} est un sous-graphe couvrant (*i.e.*, $V' = V$).



Figure IV.14 – Un graphe non orienté. Les arêtes en **gras** forment un arbre couvrant.

Exercice IV.17. Justifiez qu'un graphe **connexe** a (au moins) un arbre couvrant.

Indice : les arêtes non isthmes à la rescousse.

Arbre couvrant

Définition IV.16 (Arbre couvrant). Soit $\mathcal{G} = (V, E)$ un graphe non orienté connexe. Un **arbre couvrant** de \mathcal{G} est un sous-graphe $\mathcal{T} = (V', E')$ tel que

- ▶ \mathcal{T} est un arbre (*i.e.*, un graphe connexe et acyclique),
- ▶ \mathcal{T} est un sous-graphe couvrant (*i.e.*, $V' = V$).



Figure IV.14 – Un graphe non orienté. Les arêtes en **gras** forment un arbre couvrant.

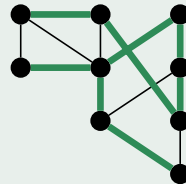
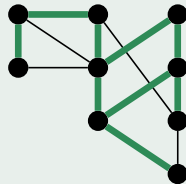
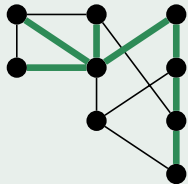
Exercice IV.17. Justifiez qu'un graphe **connexe** a (au moins) un arbre couvrant.

Indice : les arêtes non isthmes à la rescousse.

Les arbres couvrants sont utiles pour s'assurer qu'on ne tourne pas en rond dans un réseau (Spanning Tree Protocol, protocole à état de liens, *etc.*).

Arbre couvrant – Exercices

Exercice IV.18. Pour chaque figure suivante, est-ce que les arêtes mises en évidence forment un arbre couvrant ? Justifiez.



Arbre couvrant – Quelques résultats

Proposition IV.19. Soient un graphe non orienté \mathcal{G} , un arbre couvrant \mathcal{T} de \mathcal{G} et une arête e de \mathcal{G} qui n'appartient pas à \mathcal{T} . Le graphe couvrant obtenu en **ajoutant** e à \mathcal{T} contient un **seul cycle élémentaire**.

Ébauche de preuve.

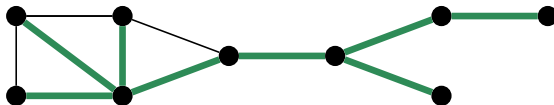
- ▶ Ajouter e à \mathcal{T} fait qu'on obtient un cycle (vu que \mathcal{T} est un arbre).
- ▶ Par l'absurde, supposons qu'il existe deux cycles élémentaires distincts (v_1, \dots, v_n, v_1) et (u_1, \dots, u_m, u_1) qui passent tous deux par l'arête e .
- ▶ Il existe i tel que $(v_i, v_{(i+1) \bmod n}) = e$ et j tel que $(v_j, v_{(j+1) \bmod m}) = e$. Donc, $v_i = v_j$.
- ▶ Il existe deux chaînes élémentaires qui partent de v_i , qui finissent en $v_{i+1 \bmod n}$ et qui ne passent pas par l'arête e (vu que le graphe est non orienté, on suit les arêtes des cycles élémentaires dans l'autre sens).
- ▶ Ces deux chaînes élémentaires sont distinctes (vu que les cycles sont distincts). Ceci est une contradiction avec le fait que \mathcal{T} soit un arbre. On en conclut qu'on ne peut pas avoir deux cycles élémentaires distincts. □

Arbre couvrant – Quelques résultats

Lemme IV.20 (lemme d'échange). Soient un graphe non orienté \mathcal{G} , un arbre couvrant \mathcal{T} de \mathcal{G} , une arête e de \mathcal{G} qui n'appartient pas à \mathcal{T} et une arête f du cycle obtenu en ajoutant e à \mathcal{T} . Alors, le graphe obtenu en **ajoutant** e et en **retirant** f à \mathcal{T} est un **arbre couvrant** de \mathcal{G} .

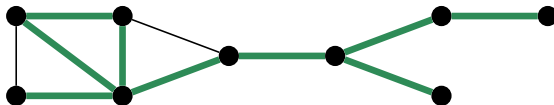
Arbre couvrant – Quelques résultats

Lemme IV.20 (lemme d'échange). Soient un graphe non orienté \mathcal{G} , un arbre couvrant \mathcal{T} de \mathcal{G} , une arête e de \mathcal{G} qui n'appartient pas à \mathcal{T} et une arête f du cycle obtenu en ajoutant e à \mathcal{T} . Alors, le graphe obtenu en **ajoutant** e et en **retirant** f à \mathcal{T} est un **arbre couvrant** de \mathcal{G} .



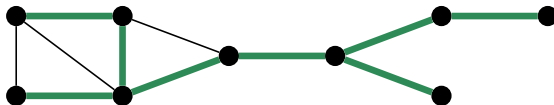
Arbre couvrant – Quelques résultats

Lemme IV.20 (lemme d'échange). Soient un graphe non orienté \mathcal{G} , un arbre couvrant \mathcal{T} de \mathcal{G} , une arête e de \mathcal{G} qui n'appartient pas à \mathcal{T} et une arête f du cycle obtenu en ajoutant e à \mathcal{T} . Alors, le graphe obtenu en **ajoutant** e et en **retirant** f à \mathcal{T} est un **arbre couvrant** de \mathcal{G} .



Arbre couvrant – Quelques résultats

Lemme IV.20 (lemme d'échange). Soient un graphe non orienté \mathcal{G} , un arbre couvrant \mathcal{T} de \mathcal{G} , une arête e de \mathcal{G} qui n'appartient pas à \mathcal{T} et une arête f du cycle obtenu en ajoutant e à \mathcal{T} . Alors, le graphe obtenu en **ajoutant** e et en **retirant** f à \mathcal{T} est un **arbre couvrant** de \mathcal{G} .



Arbre couvrant dans un graphe pondéré

Soit un graphe pondéré $\mathcal{G} = (V, E, p)$ avec $p : E \rightarrow \mathbb{N}$. Un arbre couvrant est alors un triplet $\mathcal{T} = (V', E', p)$ tel que (V', E') forme un arbre couvrant de (V, E) .

2. Pour rappel, pour un graphe $\mathcal{G} = (V, E, p)$, $p(\mathcal{G}) = \sum_{e \in E} p(e)$.

Arbre couvrant dans un graphe pondéré

Soit un graphe pondéré $\mathcal{G} = (V, E, p)$ avec $p : E \rightarrow \mathbb{N}$. Un arbre couvrant est alors un triplet $\mathcal{T} = (V', E', p)$ tel que (V', E') forme un arbre couvrant de (V, E) .

Notons $\mathcal{T}_{\mathcal{G}}$ l'ensemble des arbres couvrants de \mathcal{G} . On peut ordonner $\mathcal{T}_{\mathcal{G}}$ en fonction des valeurs $p(\mathcal{T})$ pour chaque $\mathcal{T} \in \mathcal{T}_{\mathcal{G}}$.²

2. Pour rappel, pour un graphe $\mathcal{G} = (V, E, p)$, $p(\mathcal{G}) = \sum_{e \in E} p(e)$.

Arbre couvrant dans un graphe pondéré

Soit un graphe pondéré $\mathcal{G} = (V, E, p)$ avec $p : E \rightarrow \mathbb{N}$. Un arbre couvrant est alors un triplet $\mathcal{T} = (V', E', p)$ tel que (V', E') forme un arbre couvrant de (V, E) .

Notons $\mathcal{T}_{\mathcal{G}}$ l'ensemble des arbres couvrants de \mathcal{G} . On peut ordonner $\mathcal{T}_{\mathcal{G}}$ en fonction des valeurs $p(\mathcal{T})$ pour chaque $\mathcal{T} \in \mathcal{T}_{\mathcal{G}}$.²

Deux arbres couvrants \mathcal{T} et \mathcal{T}' sont dits **voisins** s'il existe deux arêtes e et f telles que

- ▶ ajouter e et retirer f de \mathcal{T} donne \mathcal{T}' et
- ▶ ajouter f et retirer e de \mathcal{T}' donne \mathcal{T} .

2. Pour rappel, pour un graphe $\mathcal{G} = (V, E, p)$, $p(\mathcal{G}) = \sum_{e \in E} p(e)$.

Arbre couvrant dans un graphe pondéré

Soit un graphe pondéré $\mathcal{G} = (V, E, p)$ avec $p : E \rightarrow \mathbb{N}$. Un arbre couvrant est alors un triplet $\mathcal{T} = (V', E', p)$ tel que (V', E') forme un arbre couvrant de (V, E) .

Notons $\mathcal{T}_{\mathcal{G}}$ l'ensemble des arbres couvrants de \mathcal{G} . On peut ordonner $\mathcal{T}_{\mathcal{G}}$ en fonction des valeurs $p(\mathcal{T})$ pour chaque $\mathcal{T} \in \mathcal{T}_{\mathcal{G}}$.²

Deux arbres couvrants \mathcal{T} et \mathcal{T}' sont dits **voisins** s'il existe deux arêtes e et f telles que

- ▶ ajouter e et retirer f de \mathcal{T} donne \mathcal{T}' et
- ▶ ajouter f et retirer e de \mathcal{T}' donne \mathcal{T} .

Lemme IV.21. Dans $\mathcal{T}_{\mathcal{G}}$, un minimum local est un minimum global.

2. Pour rappel, pour un graphe $\mathcal{G} = (V, E, p)$, $p(\mathcal{G}) = \sum_{e \in E} p(e)$.

Arbre couvrant de poids minimum

Le problème généralisé. Soit un graphe **non orienté** pondéré **connexe** $\mathcal{G} = (V, E, p)$ avec $p : E \rightarrow \mathbb{N}^{>0}$ (i.e., que des valeurs **strictement positives**). On veut trouver un sous-graphe $\mathcal{G}' = (V', E', p)$ de \mathcal{G} tel que

- ▶ \mathcal{G}' est couvrant (i.e., $V' = V$),
- ▶ \mathcal{G}' est connexe et
- ▶ le poids total des arêtes de \mathcal{G}' est minimum, i.e.,

$$p(\mathcal{G}') = \sum_{e \in E'} p(e) \text{ est minimum.}$$

On cherche donc un **arbre couvrant de poids minimum**.

Algorithme naïf

Le lemme précédent nous donne un algorithme d'optimisation naïf.

- ▶ Construire un arbre couvrant \mathcal{T} sans se soucier du poids.
- ▶ Prendre une arête e qui n'appartient pas à \mathcal{T} et une arête f qui appartient au cycle qui apparaît en ajoutant e à \mathcal{T} . Notons $\mathcal{T} + e - f$ l'arbre couvrant obtenu en retirant en plus f .
- ▶ Si $p(\mathcal{T} + e - f) < p(\mathcal{T})$, on recommence avec $\mathcal{T} + e - f$.
- ▶ Sinon, on teste avec une autre paire (e, f) .

Algorithme naïf

Le lemme précédent nous donne un algorithme d'optimisation naïf.

- ▶ Construire un arbre couvrant \mathcal{T} sans se soucier du poids.
- ▶ Prendre une arête e qui n'appartient pas à \mathcal{T} et une arête f qui appartient au cycle qui apparaît en ajoutant e à \mathcal{T} . Notons $\mathcal{T} + e - f$ l'arbre couvrant obtenu en retirant en plus f .
- ▶ Si $p(\mathcal{T} + e - f) < p(\mathcal{T})$, on recommence avec $\mathcal{T} + e - f$.
- ▶ Sinon, on teste avec une autre paire (e, f) .

Il faut tester toutes les paires (e, f) possibles. C'est trop lent.³ On peut faire mieux !

3. Pour un graphe à 20 sommets et toutes les arêtes possibles, il faudrait plusieurs siècles...

Algorithme de Kruskal

Nécessite : \mathcal{G} est un graphe non orienté connexe

```

1 : procédure KRUSKAL( $\mathcal{G} = (V, E, p)$ )
2 :    $F \leftarrow E$ 
3 :    $A \leftarrow \emptyset$ 
4 :   tant que  $|A| < |V| - 1$  faire
5 :     Trouver  $e \in F$  tel que  $p(e)$  est minimum
6 :      $F \leftarrow F \setminus \{e\}$ 
7 :      $\mathcal{G}_{A \cup \{e\}} \leftarrow$  le sous-graphe couvrant de  $\mathcal{G}$  avec comme arêtes  $A \cup \{e\}$ 
8 :     si  $\mathcal{G}_{A \cup \{e\}}$  est acyclique alors
9 :        $A \leftarrow A \cup \{e\}$ 
10 :   retourner le sous-graphe couvrant de  $\mathcal{G}$  avec les arêtes de  $A$ 

```

L'algorithme de Kruskal est un algorithme **glouton** : on prend le meilleur choix maintenant, sans se soucier du futur. Pour l'arbre couvrant de poids minimum, on trouve bien une réponse optimale.⁴

Exemple : au tableau.

4. Il y a d'autres problèmes pour lesquels un algorithme glouton ne retourne pas toujours une solution optimale.

Kruskal – Résultats

Lemme IV.22. Le sous-graphe retourné est un arbre couvrant de poids minimum.

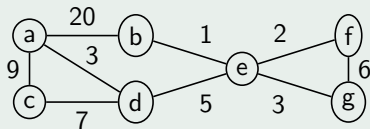
Ébauche de preuve.

- ▶ Le sous-graphe est acyclique et connexe. Il couvre tous les sommets par construction. On a donc bien un arbre couvrant \mathcal{T} .
- ▶ Pour montrer que \mathcal{T} est un arbre couvrant de poids minimum, il suffit de montrer qu'il s'agit d'un minimum local dans $\mathcal{T}_{\mathcal{G}}$. Par contradiction, supposons qu'il n'est pas un minimum local.
- ▶ Alors, il existe un voisin \mathcal{T}' de \mathcal{T} tels que $p(\mathcal{T}') < p(\mathcal{T})$. Autrement dit, il existe f dans \mathcal{T} et e qui n'appartient pas à \mathcal{T} tels que $\mathcal{T}' = \mathcal{T} + e - f$ est un arbre couvrant. Remarquons que \mathcal{T}' est acyclique.
- ▶ Alors, $p(e) < p(f)$ et l'algorithme a choisi f . Or, on doit prendre l'arête de poids minimal qui n'induit pas de cycle. Comme \mathcal{T}' est acyclique, e était un candidat. On a donc une contradiction et \mathcal{T}' ne peut pas exister. □

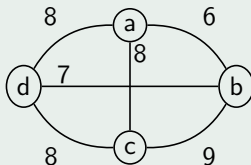
Lemme IV.23. On peut implémenter cet algorithme avec une complexité $\mathcal{O}(|E| \log |V|)$.

Kruskal – Exercices

Exercice IV.24. Appliquez l'algorithme de Kruskal sur le graphe suivant.

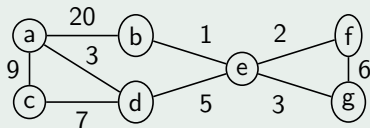


Exercice IV.25. Appliquez l'algorithme de Kruskal sur le graphe suivant.

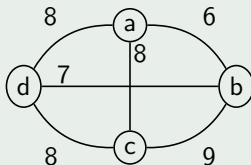


Kruskal – Exercices

Exercice IV.24. Appliquez l'algorithme de Kruskal sur le graphe suivant.



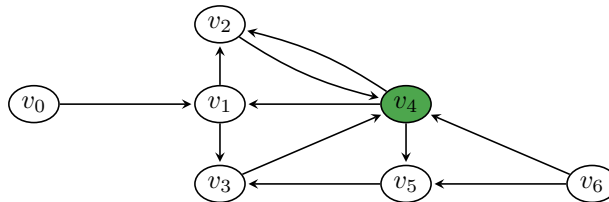
Exercice IV.25. Appliquez l'algorithme de Kruskal sur le graphe suivant.



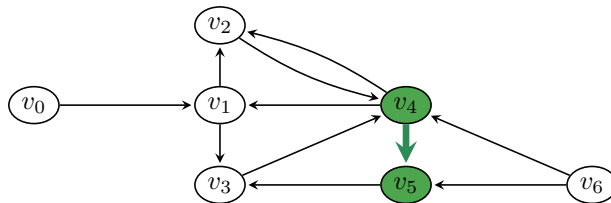
Plusieurs résultats sont possibles. Il n'existe **pas nécessairement** une **unique** solution.

1. Arbres
2. Arbre couvrant de poids minimum
3. Un petit tour en forêt

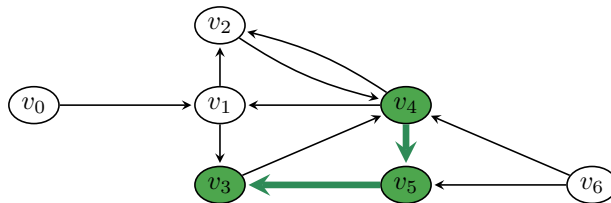
Le DFS est un arbre



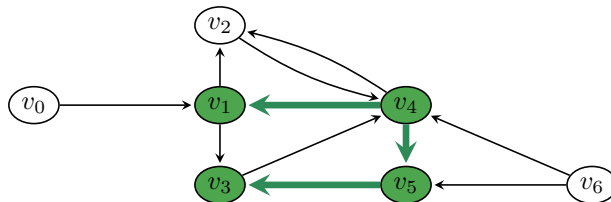
Le DFS est un arbre



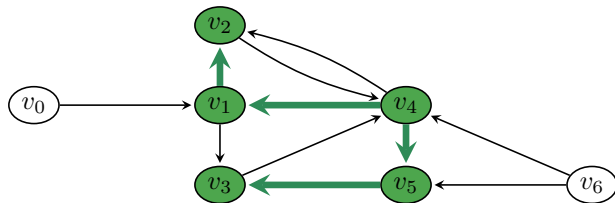
Le DFS est un arbre



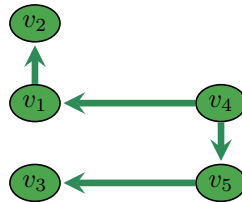
Le DFS est un arbre



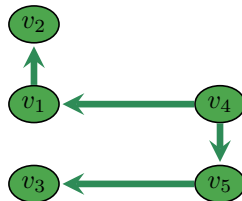
Le DFS est un arbre



Le DFS est un arbre



Le DFS est un arbre



Un DFS à partir d'un nœud v donne **toujours** un arbre orienté de racine v .

Théorie des jeux et minimax

En AGATH :

- ▶ les jeux sous forme extensive ;
- ▶ algorithme minimax.

Question IV.26. Est-ce que vous avez d'autres exemples ?

Arbre binaire de recherche

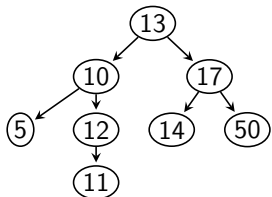


Figure IV.15 – Un arbre binaire de recherche équilibré.

Problème	Complexité en moyenne	Complexité au pire cas
Espace mémoire		
Tester si une valeur est présente		
Ajouter une valeur		
Supprimer une valeur		

Arbre binaire de recherche

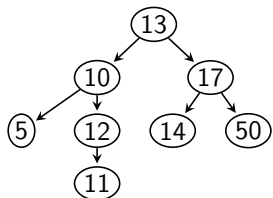


Figure IV.15 – Un arbre binaire de recherche équilibré.

Problème	Complexité en moyenne	Complexité au pire cas
Espace mémoire	$\mathcal{O}(V)$	
Tester si une valeur est présente	$\mathcal{O}(\log V)$	
Ajouter une valeur	$\mathcal{O}(\log V)$	
Supprimer une valeur	$\mathcal{O}(\log V)$	

Arbre binaire de recherche

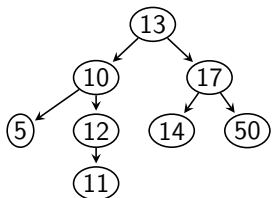


Figure IV.15 – Un arbre binaire de recherche équilibré.



Figure IV.16 – Un arbre binaire de recherche dégénéré.

Problème	Complexité en moyenne	Complexité au pire cas
Espace mémoire	$\mathcal{O}(V)$	$\mathcal{O}(V)$
Tester si une valeur est présente	$\mathcal{O}(\log V)$	$\mathcal{O}(V)$
Ajouter une valeur	$\mathcal{O}(\log V)$	$\mathcal{O}(V)$
Supprimer une valeur	$\mathcal{O}(\log V)$	$\mathcal{O}(V)$

Équilibrage d'arbre binaire de recherche

Il est donc important de s'assurer que l'arbre reste **équilibré** pendant sa construction :

la longueur du plus long chemin à gauche ne doit pas être trop éloignée de la longueur du plus long chemin à droite.

Vous avez déjà vu les arbres AVL dans le cours d'Algorithmique avancée. Il existe d'autres types d'arbres qui s'équilibrent automatiquement.

Équilibrage d'arbre binaire de recherche

Il est donc important de s'assurer que l'arbre reste **équilibré** pendant sa construction :

la longueur du plus long chemin à gauche ne doit pas être trop éloignée de la longueur du plus long chemin à droite.

Vous avez déjà vu les arbres AVL dans le cours d'Algorithmique avancée. Il existe d'autres types d'arbres qui s'équilibrent automatiquement.

Dans le TP d'aujourd'hui :

- ▶ Implémentation des arbres orientés.
- ▶ Arbre couvrant de poids minimum pour un graphe orienté.
- ▶ Arbres rouge-noir.