Théorie des Graphes

Université Libanaise
Faculté des Sciences
License Informatique
2ème année – S3

Syllabus

1. Concepts introductifs
2. Introduction aux graphes et à leurs utilisations
3. Arbres et graphes bipartis
4. Distance et connexité
5. Graphes Eulériens et Hamiltoniens
6. Coloration des graphes
7. Matrices
8. Algorithmes sur les graphes
9. Graphes planaires
10. Digraphes et réseaux

Arbres et graphes bipartis

Semaine 3

Avant de commencer

- Classe de graphes très importante: les arbres.
- Un graphe avec la propriété qu’aucun de ses sous-graphes n'est un cycle est appelé **acyclique** .
- Un graphe acyclique connexe est appelé un **arbre** .
- Les arbres sont la base d'importantes structures de données.
- Un arbre est un type spécial de graphe biparti.

Plan

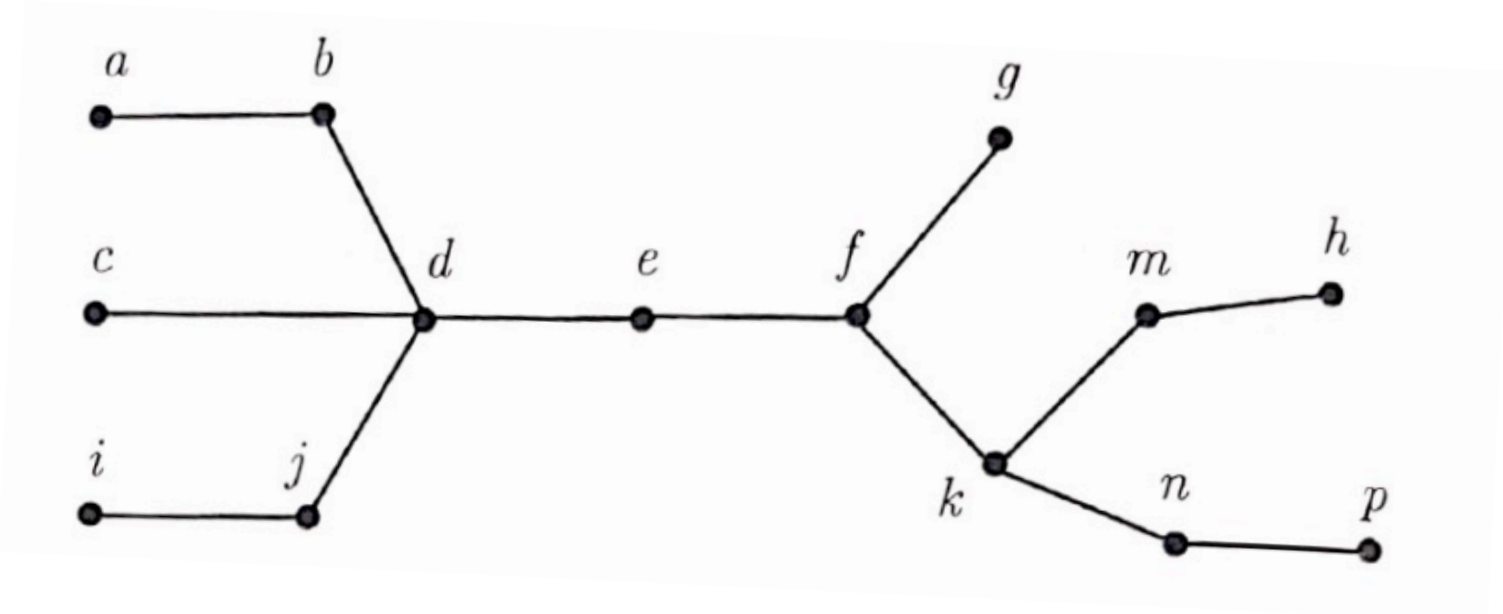
- Propriétés des arbres
- Arbres couvrants minimum
- Une caractérisation des graphes bipartis
- Correspondances et affectations de tâches

Propriétés des arbres

- Vous avez déjà rencontré deux classes importantes d'arbres:
  1. étoiles
  2. chemins
- Le chemin P_n peut être défini comme un arbre sur des sommets, tel que that $\Delta(T) \leq 2$. (Rappelez-vous que Δ signifie le degré maximum.)
- L'étoile $K_{1,n}$ peut être définie comme l'arbre de l'ordre $n+1$, tel qu'un seul sommet, v, satisfait $\deg(\mathrm{v}) = n$.
- Puisque $K_{1,n}$ est un arbre, les autres sommets doivent être des sommets d'extrémité (parfois appelés **feuilles** de l'arbre), sinon nous aurions un cycle.

Propriétés des arbres

- Exemple d'arbre avec 14 sommets et 13 arêtes.



Propriétés des arbres
Certaines propriétés

- P1. Chaque arbre non trivial (le graphe trivial K_1 est aussi un arbre) a au moins deux sommets d'extrémité.
- P2. La suppression de n'importe quelle arête d'un arbre le rend non connexe. (Une arête d'un graphe connexe dont la suppression déconnecte le graphe est appelée un **pont** . Ainsi, toutes les arêtes d'un arbre sont des ponts. Incidemment, même si G est non connexe, une arête de G est un pont si sa suppression augmente le nombre de composants de G.)
- P3. Étant donné deux sommets, x et y, d'un arbre, il y a une chaîne élémentaire $x - y$ unique. Par conséquent, cette chaîne est une géodésique. On pourrait dire qu'il n'y a pas d'alternatives de voyage dans un arbre. Comparez cela avec les deux géodésiques entre un premier et un quatrième sommet consécutif sur le cycle C_6.
- P4. Un arbre avec n sommets et q arêtes satisfait $q = n - 1$. Ainsi, un arbre est minimalement connexe.

Propriétés des arbres
Séquences graphiques des arbres

- **Théorème**. Étant donné une séquence S de n entiers positifs $d_1, d_2, \dots, d_n$ tels que $d_1 \geq d_2 \geq \cdots \geq d_n$ et $d_1 + d_2 + \cdots + d_n = 2(n-1)$, alors il existe un arbre dont la séquence de degrés est S.

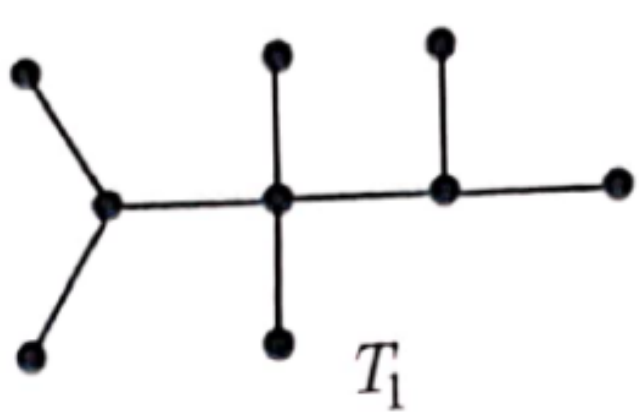
Propriétés des arbres
Arbres non isomorphes

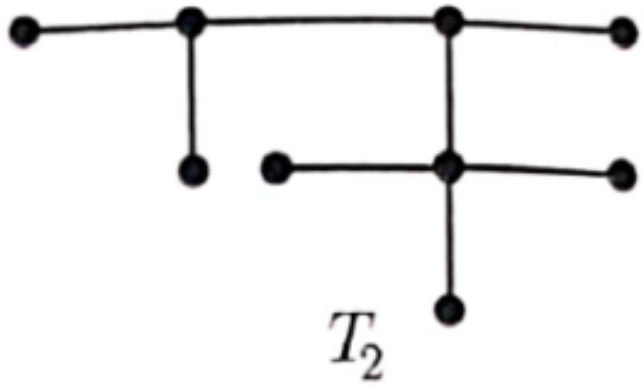
- Pour les arbres, la plus utile de ces techniques est de comparer des séquences de degrés, d'examiner les chemins les plus longs et de regarder les chemins les plus courts entre les sommets d'un degré unique donné.

Propriétés des arbres
Arbres non isomorphes

Exercice

Trouvez une propriété structurelle des arbres T_1 et T_2 qui montre qu'ils ne sont pas isomorphes.

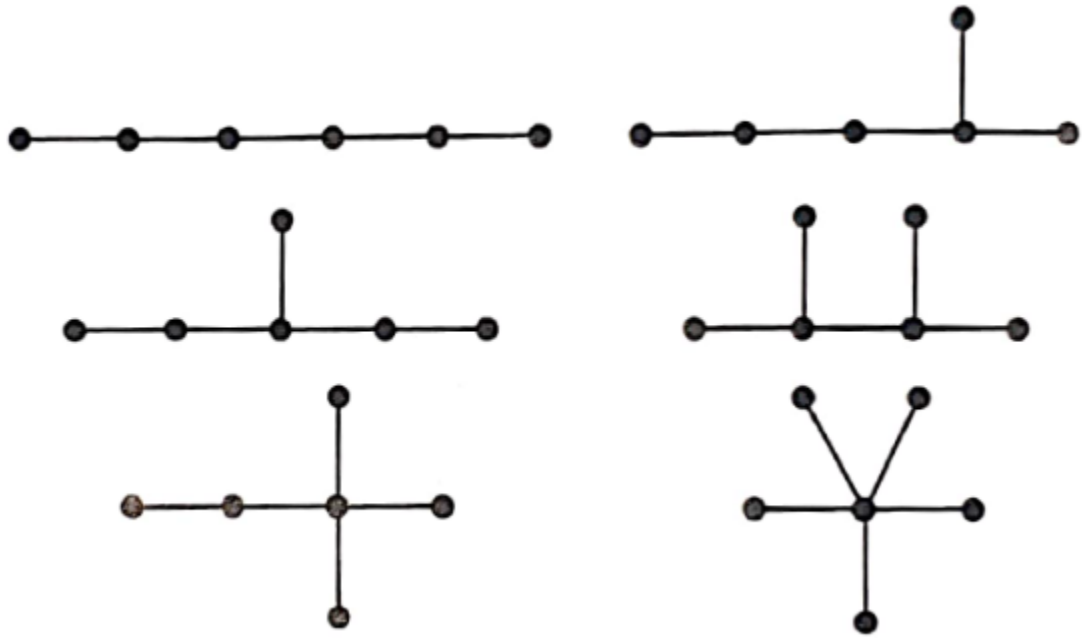




Propriétés des arbres
Arbres non isomorphes

Exercice

Il y a six arbres qui ont six sommets. Dessinez-les.



Propriétés des arbres
Le nombre de feuilles d'un arbre

- Étant donné un arbre T non trivial, définissez n_i comme le nombre de sommets de degré i.
- C'est un fait intéressant que le nombre de feuilles (ou de sommets d'extrémité) d'un arbre (c'est-à-dire n_1) peut être déterminé à partir des nombres n_3, n_4, n_5, ...
- Le nombre de feuilles d'un arbre non trivial est donné par la formule

$$n_1 = 2 + n_3 + 2n_4 + 3n_5 + \cdots = 2 + \sum_{i=3}^{\infty} (i-2)n_i$$

Plan

- Propriétés des arbres
- Arbres couvrants minimum
- Une caractérisation des graphes bipartis
- Correspondances et affectations de tâches

Arbres couvrants minimum

- Étant donné un graphe connexe G, un sous-graphe couvrant de G qui est un arbre est appelé **arbre couvrant** .
- Dans les applications impliquant des arbres couvrants, il existe parfois des nombres appelés **poids** , associés à chaque arête d'un graphe.
- Il faut alors trouver un arbre couvrant pour le graphe pour lequel le total des poids dans l'arbre est minimum.

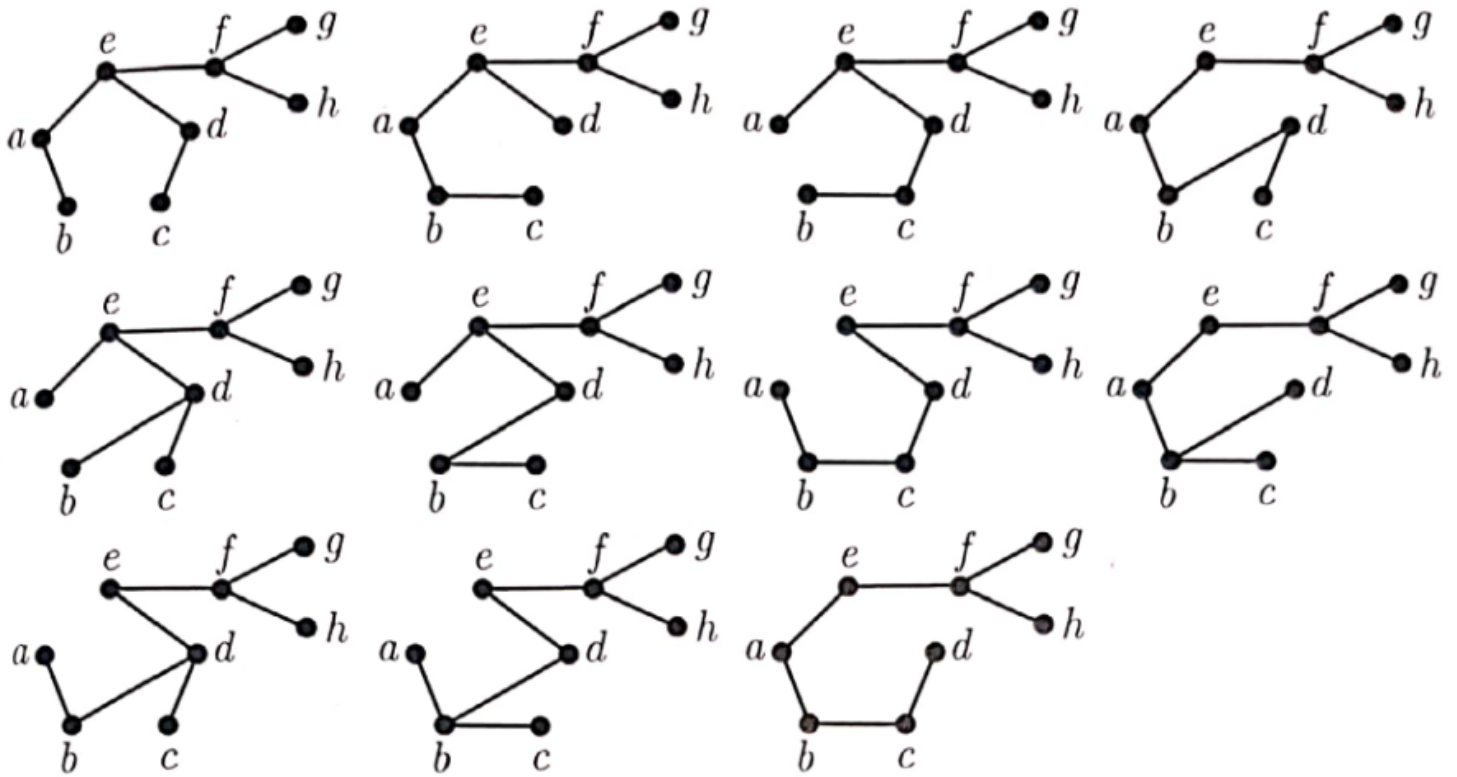
Arbres couvrants minimum
Arbres couvrants

- Puisqu'un arbre avec n sommets a $n-1$ arêtes, pour générer un arbre couvrant d'un graphe connexe G ayant n sommets et q arêtes, nous devons supprimer de G tout sauf $n-1$ arêtes.
- Nous ne pouvons pas le faire au hasard, car un arbre est connexe et n'a pas de cycles.
- Nous devons supprimer $q-(n-1)=q-n+1$ arêtes, dont aucune n'est un pont, pour former un arbre couvrant de G.

Arbres couvrants minimum
Arbres couvrants

Exercice

Trouvez tous les arbres couvrants pour le graphe étiqueté suivant



Arbres couvrants minimum
Sommets k-déficients d'arbres couvrantss

- Un sommet v dans un arbre couvrant T d'un graphe G est appelé **k-déficient**, si son degré satisfait l'équation $deg_G(v) - deg_T(v) = k$.
- L'entier, k, est appelé la **déficience** du sommet v.
- Si k = 0, v est un sommet **préservant les degrés**, c'est-à-dire $deg_G(v) = deg_T(v)$.
- Observez que pour tout sommet **k**-déficient d'un arbre non trivial T, nous avons $k \leq \Delta(G) - 1$, puisque $deg_T(v) > 0$.
- **Théorème** . Soit G un graphe connexe sur n sommets et q arêtes. Il s'en suit que, la somme des déficiences des sommets de tout arbre couvrant de G est $2(q - n + 1)$.

Arbres couvrants minimum
Arbres couvrants de coût minimum

- Problème important: la création d'un arbre couvrant à coût minimum, est souvent appelé **problème du connecteur minimal**.
- Étant donné un graphe G tel que chaque arête a un poids positif, qui représente son coût, le coût d'un arbre couvrant est la somme des poids des arêtes de T.
- Algorithme dû à *Kruskal* : construction d'un arbre couvrant de sorte qu'aucun autre arbre couvrant n'ait un moindre coût. Un tel arbre est appelé un **arbre couvrant minimum**.

Arbres couvrants minimum
Arbres couvrants de coût minimum

- L'algorithme est appelé **glouton** , car il nous dit de procéder à chaque étape en sélectionnant la meilleure (c'est-à-dire la moins chère) arête disponible qui ne forme pas un cycle.
- L'algorithme s'auto-termine, dans le sens que nous nous arrêtons dès que nous obtenons un arbre couvrant, car l'ajout de tout arête restant créera un cycle.
- Il convient de noter qu'il peut y avoir plusieurs arbres couvrants minimales pour un graphe pondéré. En effet, il y a des choix à faire entre des arêtes de coûts égals.

Arbres couvrants minimum
Arbres couvrants de coût minimum

- Nous supposerons qu'un graphe connexe pondéré G et son ordre n ont été stockés, que son nombre d'arêtes q a été compté et que les arêtes ont été pré-triées selon le poids du plus petit au plus grand dans une liste L.
- Les arêtes dont le poids est le même sont stockées à leur place par ordre alphabétique.
- Un ensemble S sera utilisé pour collecter les arêtes de l'arbre couvrant lorsque les arêtes sont sélectionnées.
- À l'étape 1, S est initialisé pour être l'ensemble vide.

Arbres couvrants minimum
Arbres couvrants de coût minimum

- L'algorithme utilise un type spécial de sous-graphe. Pour un ensemble A d'arêtes d'un graphe G, le sous-graphe induit par les arêtes $\langle A \rangle$ est le sous-graphe H avec $E(H) = A$ et $V(H) = \{v : v \text{ est incident avec une arête dans } A\}$.

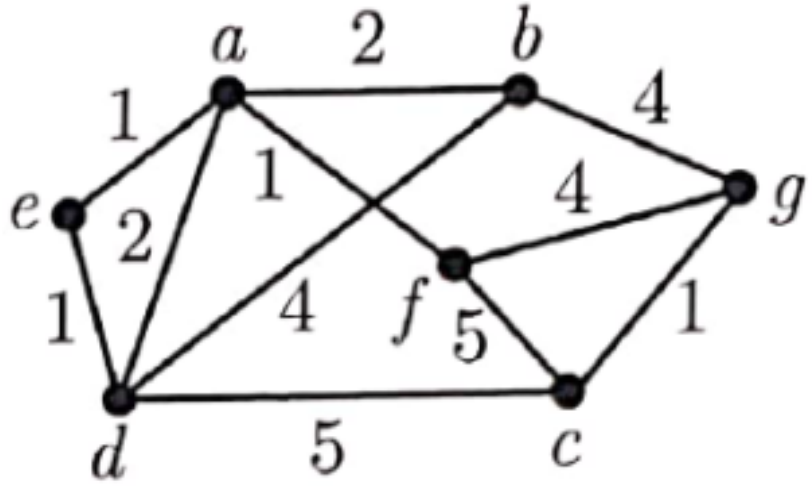
Algorithme de Kruskal

1. $S = \emptyset$
2. Soit e l'arête suivante de la liste triée L pour laquelle $e \notin S$ et le sous-graphe induit par l'arête $\langle S \cup \{e\} \rangle$ est acyclique. Soit $S = S \cup \{e\}$.
3. Si $|S| = n - 1$, arrêter et retourner S. Sinon, revenez à l'étape 2 et continuez le long de la liste L.

Arbres couvrants minimum
Arbres couvrants de coût minimum

Exercice

Appliquez l'algorithme de Kruskal au graphe G pondéré et montrez l'ordre dans lequel les arêtes entrent dans l'ensemble. Ensuite, dessinez l'arbre couvrant minimum résultant pour G.



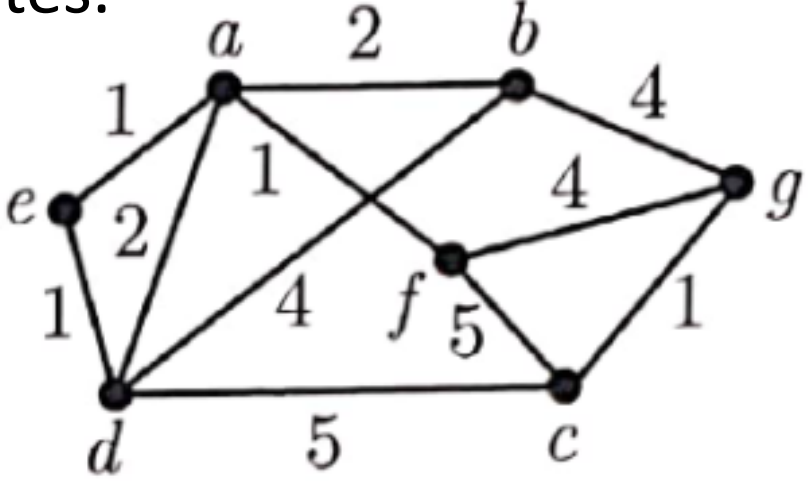
Arbres couvrants minimum
Arbres couvrants de coût minimum

Soltuion

Trier les arêtes selon l'ordre croissant des poids

{ae, af, cg, de, ab, ad, bd, bg, fg, cd, cf}

G a onze arêtes et sept sommets, donc un arbre couvrant pour G a six arêtes.



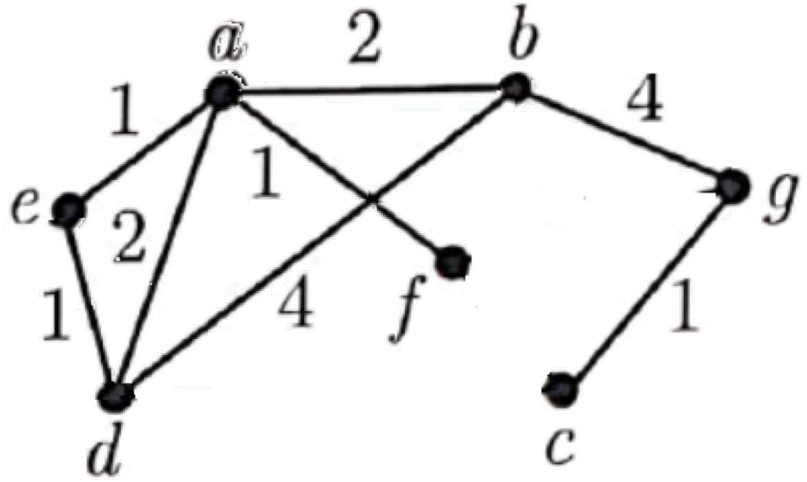
Arbres couvrants minimum
Arbres couvrants de coût minimum

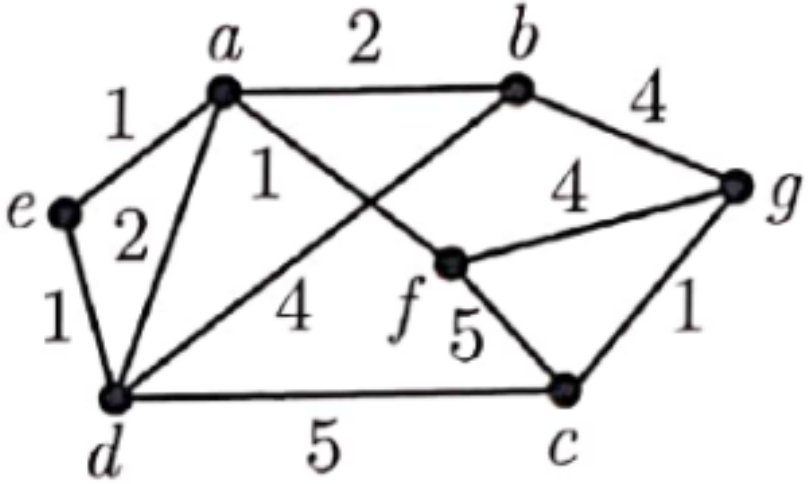
Soltuion

{ae, af, cg, de, ab, ad, bd, bg, fg, cd, cf}

✓ ✓ ✓ ✓ ✓ ✗ ✗ ✓

L'algorithme commence par $S = \emptyset$.





Arbres couvrants minimum
Arbres couvrants de coût minimum

- Il existe d'autres algorithmes pour générer un arbre couvrant minimum. Nous présentons maintenant un algorithme dû à Prim qui est également gourmand, mais contrairement à l'algorithme de Kruskal, le sous-graphe induit par les arêtes est connexé à chaque étape.
- La façon dont l'algorithme fonctionne est de commencer à un sommet, que nous pourrions considérer comme la racine de l'arbre. Il ajoute ensuite successivement de nouveaux sommets pas encore dans l'arbre en sélectionnant un sommet qui peut être attaché à l'arbre en utilisant une arête de poids minimum.

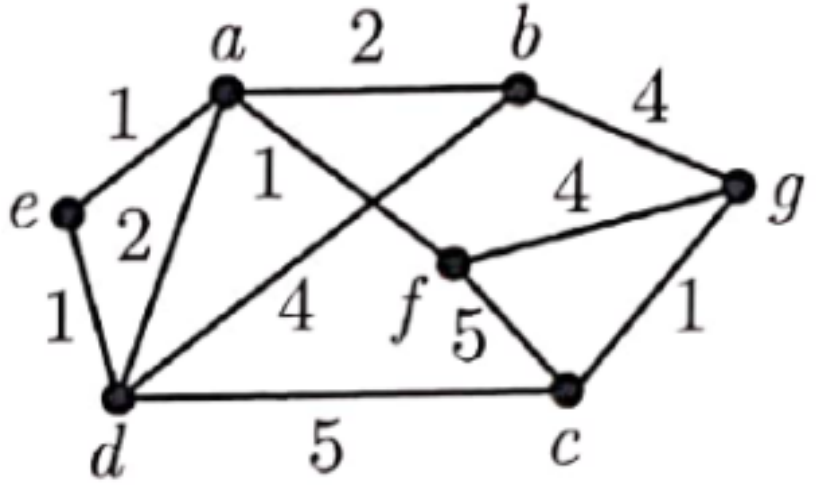
Algorithme de Prim

1. Sélectionnez un sommet v et soit let $V(T) = \{v\}, E(T) = \emptyset$.
2. Parmi tous all $u \notin V(T)$, soit l $e = uw$ une arête de poids minimum joignant u à w, où $w \in V(T)$. Soit $V(T) = V(T) \cup \{u\}$ et $E(T) = E(T) \cup \{uw\}$.
3. Si $|\mathrm{E}(T)| = n - 1$ arrêtez et affichez $E(T)$. Sinon, revenez à l'étape 2 et ajoutez un autre sommet à l'arbre.

Arbres couvrants minimum
Arbres couvrants de coût minimum

Exercice

Appliquez l'algorithme de Prim au graphe pondéré G. Montrez l'arborescence au fur et à mesure de sa croissance pendant l'algorithme lorsque a est utilisé comme racine.



Arbres couvrants minimum
Arbres couvrants de coût minimum

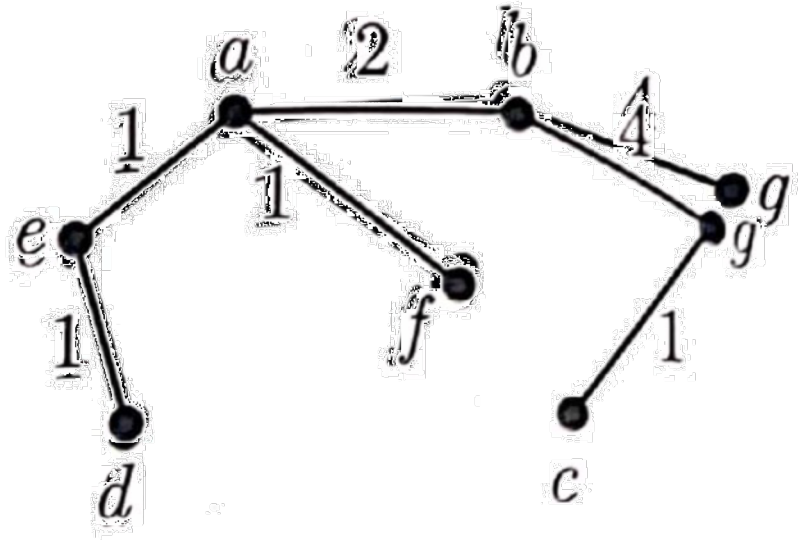
Soltuion

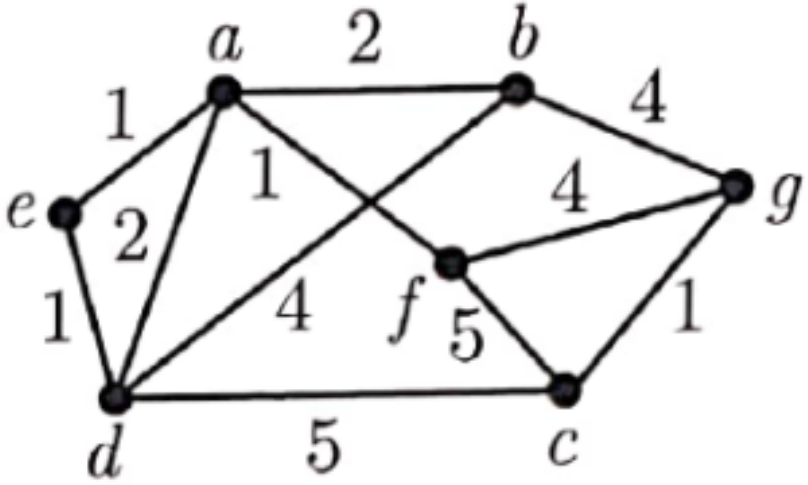
Trier les arêtes selon l'ordre croissant des poids

$v = \text{a},$

$V(T) = \{a\ e\ f\ d\ b\ g\ c\ \}$

$E(T) = \{\ ae\ \ af\ \ de\ \ ab\ \ bg\ \ cg\}$





Arbres couvrants minimum
Arbres couvrants de coût minimum

- Nous mentionnons ici que l'algorithme de Prim appliqué à un graphe connexe où tous les poids sont égaux est équivalent à une technique de recherche importante appelée **recherche en largeur d'abord** (**breadth-first search**) lorsque les liens sont convenablement rompus (généralement par ordre dans les listes de contiguïté).
- La recherche en largeur d'abord sera discutée au cours de la semaine 8 avec une variété d'autres algorithmes.

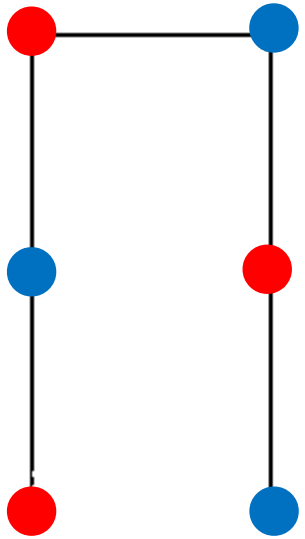
Plan

- Propriétés des arbres
- Arbres couvrants minimum
- Une caractérisation des graphes bipartis
- Correspondances et affectations de tâches

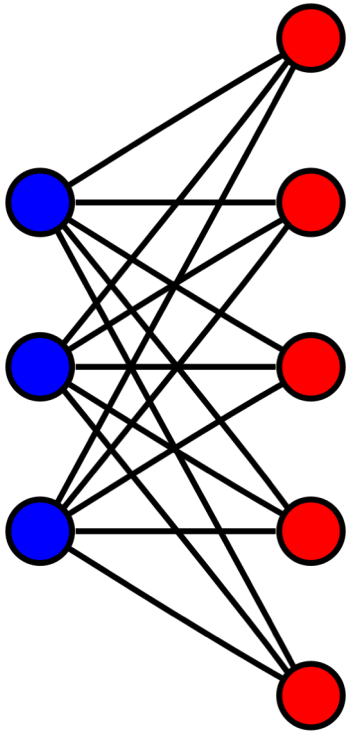
Une caractérisation des graphes bipartis

- Un graphe G est appelé biparti si $\mathrm{V}(G)$ peut être partitionné en deux sous-ensembles non vides V_1 et V_2, de sorte que si xy est une arête de G, alors x et y appartiennent à des sous-ensembles différents.
- Les ensembles V_1 et V_2 sont les **parties** ou **ensembles de bipartitions** de G.

Une caractérisation des graphes bipartis



P_6



$K_{m,n}$

Une caractérisation des graphes bipartis

Nous classons les graphes bipartis comme suit

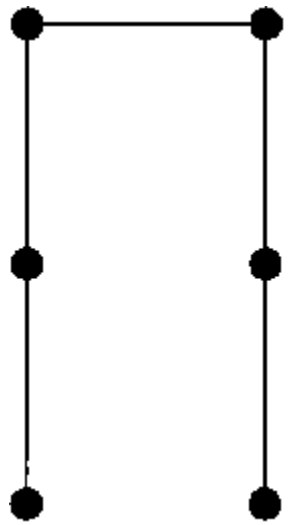
- Soit G biparti et soit son ensemble de sommets soit partitionné en sous-ensembles V_1 et V_2, de sorte que $|V_1| \geq |V_2|$. (le plus grand est V_1)

1. G est appelé **équitable** si $|V_1| = |V_2|$
2. G est **presque équitable** si $|V_1| - |V_2| = 1$
3. G est **biaisé** si $|V_1| - |V_2| \geq 2$

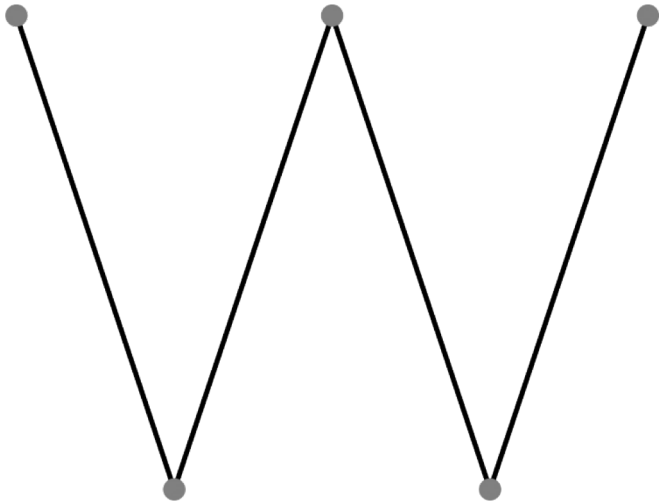
Une caractérisation des graphes bipartis

Exercice

Dites si le graphe suivant est équitable ou presque équitable.



P_6

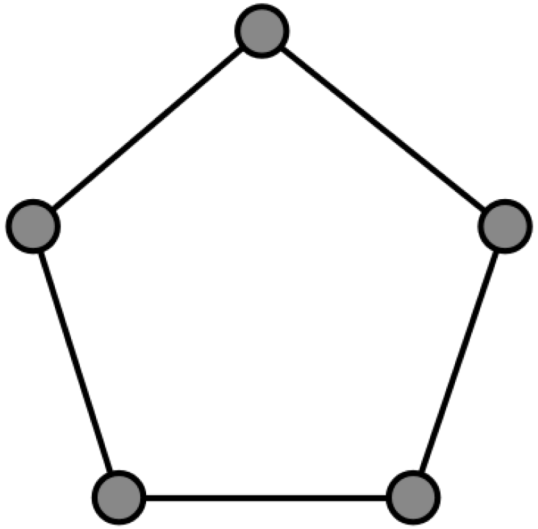


P_5

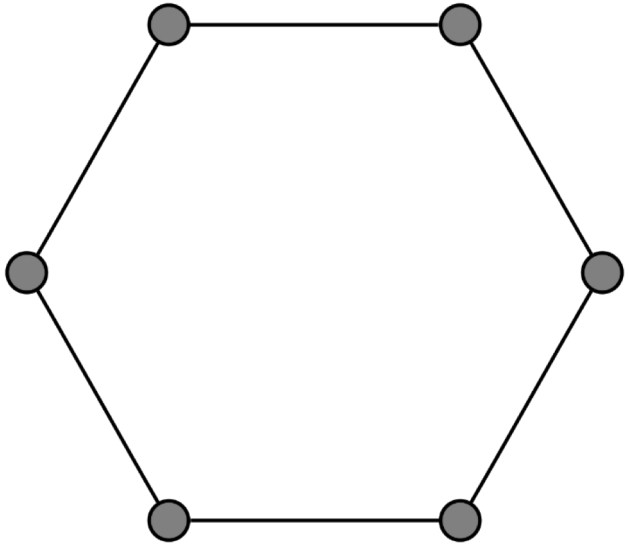
Une caractérisation des graphes bipartis

Exercice

Dites si le graphe suivant est équitable ou presque équitable.



C_5



C_6

Une caractérisation des graphes bipartis Déterminer si G est biparti

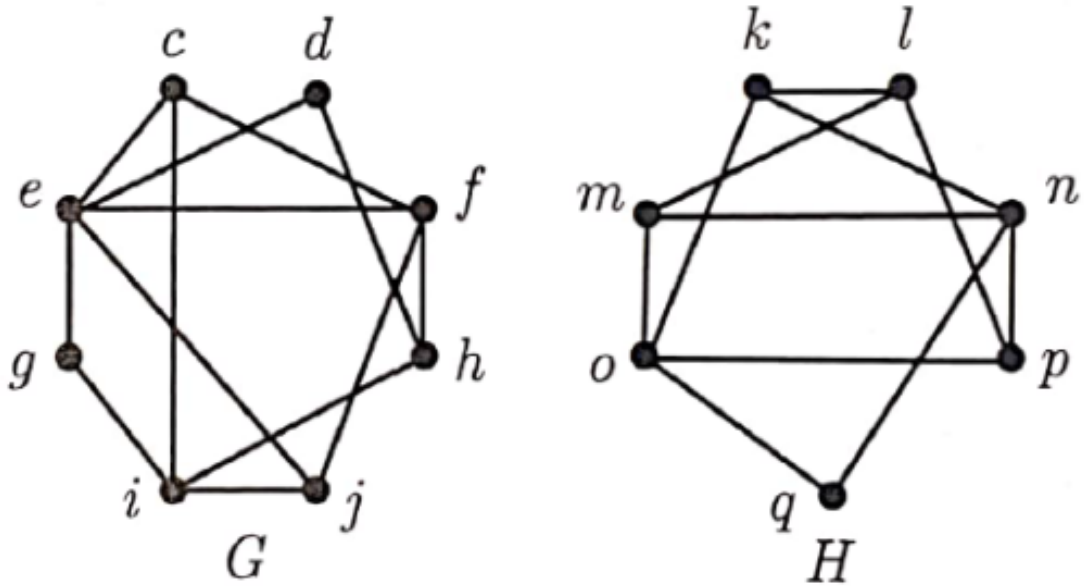
Dans cette procédure, nous nous référons aux étiquettes a et b t comme étant des étiquettes opposées.

1. Étiqueter n'importe quel sommet a
2. Étiqueter tous les sommets adjacents à avec a par l'étiquette b
3. S'il y a des sommets non étiquetés adjacents à un sommet étiqueté v, étiquetez tous les sommets adjacents à v en utilisant l'étiquette opposée à celle qui a été attribuée à v.
4. Répétez l'étape 3 jusqu'à ce qu'il n'y ait pas de sommets sans étiquette adjacents à un sommet étiqueté
5. S'il y a un sommet sans étiquette, sélectionnez un de ces sommets (qui est nécessairement dans une nouvelle composante), étiquetez-le et revenez à l'étape 3
6. Si les sommets adjacents reçoivent des étiquettes distinctes, alors le graphe est biparti. L'ensemble V_1 dans la définition d'un graphe biparti est alors l'ensemble de tous les sommets avec étiquette a; l'ensemble V_2 est l'ensemble de tous les sommets avec étiquette b. Si une paire de sommets adjacents a reçu la même étiquette, alors G n'est pas biparti.

Une caractérisation des graphes bipartis

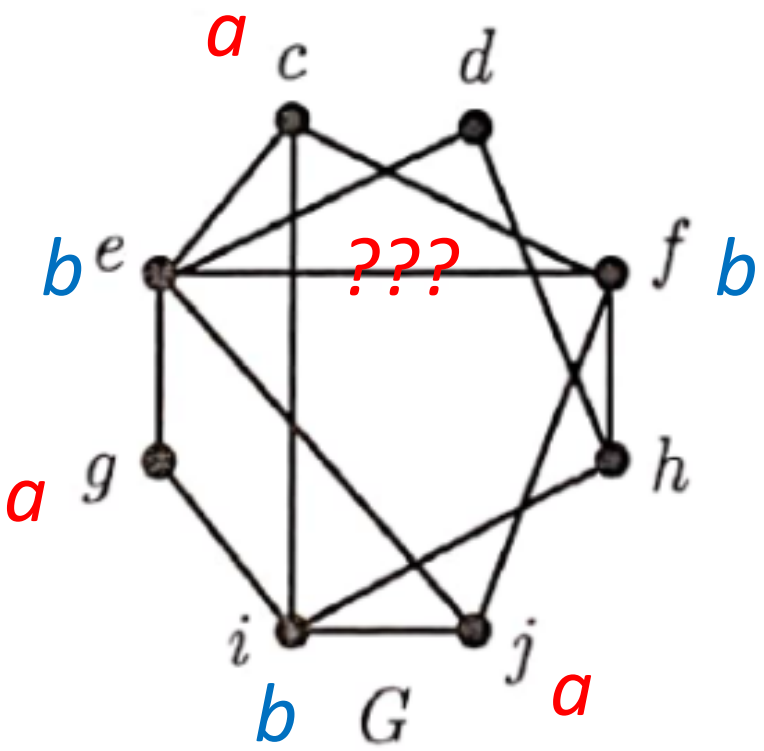
Exercice

Étiquetez les sommets de graphes G et H et déterminez s'ils sont bipartis. Si biparti, redessinez le graphe pour afficher les ensembles de sommets V_1 et V_2.



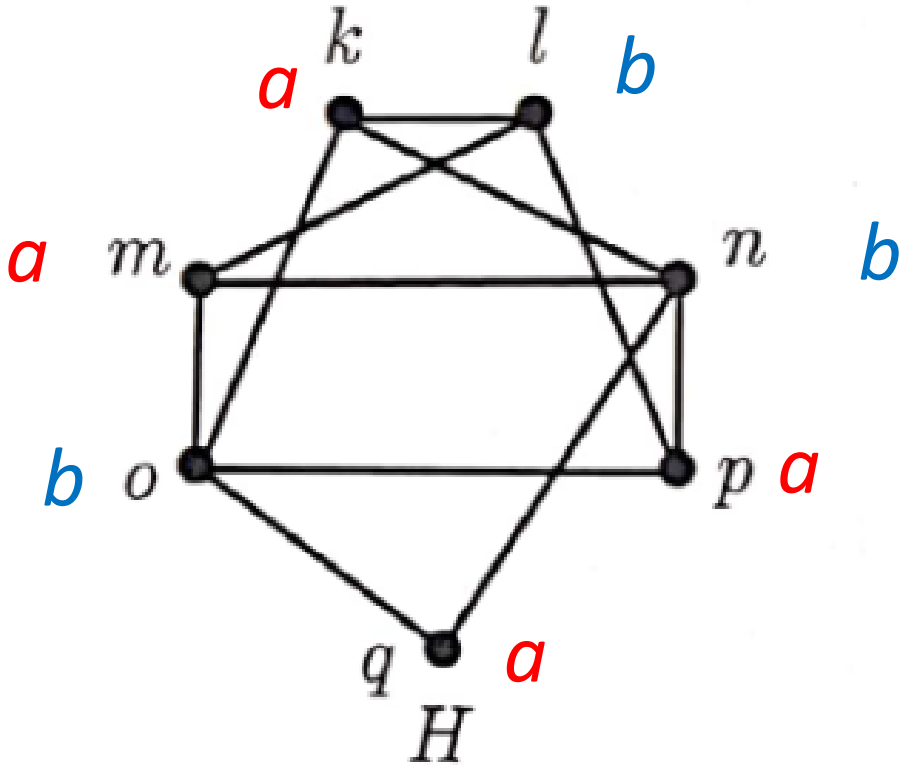
Une caractérisation des graphes bipartis

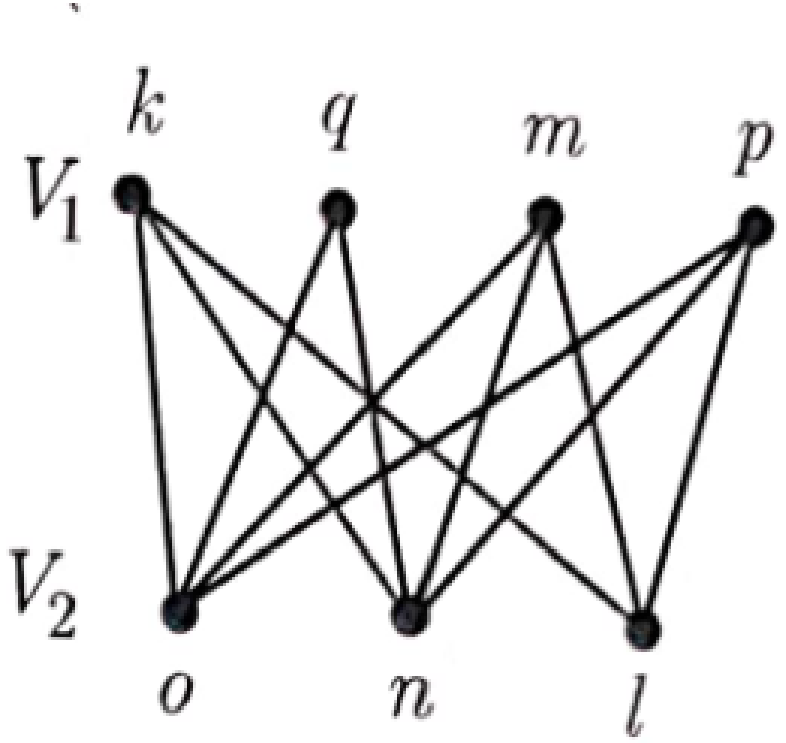
Solution



Une caractérisation des graphes bipartis

Solution





Une caractérisation des graphes bipartis Déterminer si G est biparti

- **Théorème** : Un graphe G est biparti si et seulement si ne contient pas de cycles de longueur impaire.

- Il s'ensuit que les arbres sont des graphes bipartis.

Une caractérisation des graphes bipartis

Exercice

Dessinez tous les graphes bipartis connexes d'ordre 5 (5 sommets).

$K_1 + K_1 + K_1 + \overline{K_2}$ $K_1 + K_1 + \overline{K_2} + K_1$

$K_{1,4}$ P_5 $K_{2,3}$

4 arêtes 5 arêtes 6 arêtes