Arbres et arborescences

D1 : Un arbre est un graphe connexe et sans cycle.

R1 : L'orientation est sans importance.

R2 : Un arbre est un graphe simple sans boucle.

T1 : Soit $G = (X, A)$ un graphe sur $n = |X|$.

1- Si G est connexe, $|A| \geq n - 1$

2- Si G est sans cycle, $|A| \leq n - 1$

Corollaire : Un arbre de n sommets comporte $(n - 1)$ arcs.

Arbres et Arborescences

T 2 : **Caractérisation des arbres**

Soit $G = (X, A)$ un graphe sur $n = | X | \geq 2$ Sommets. Les propriétés suivantes sont équivalentes et caractérisent un arbre :

a - G est connexe et sans cycle .

b - G est sans cycle et comporte (n - 1) arcs.

c - G est connexe et comporte (n - 1) arcs.

d - G est sans cycle et maximal pour cette propriété (si on ajoute un arc à G, on crée un cycle)

e - G est connexe et minimal pour cette propriété (si on supprime un arc de G, il n'est plus connexe)

f - Tout couple de sommets de G est relié par une chaîne élémentaire unique.

Corollaire : Tout graphe connexe $G = (X, A)$ possède un graphe partiel qui est un arbre.

Arbres et Arborescences

T3 : Un arbre G = (X , A) sur $n \geq 2$ sommets admet au moins 2 sommets **pendants** (sommets de degré 1).

D 2 : On appelle **forêt** un graphe dont chaque composante connexe est un arbre, c'est à dire un graphe sans cycle.

R 3 : Une forêt à p composantes connexes possède (n - p) arcs.

Arbres et arborescences

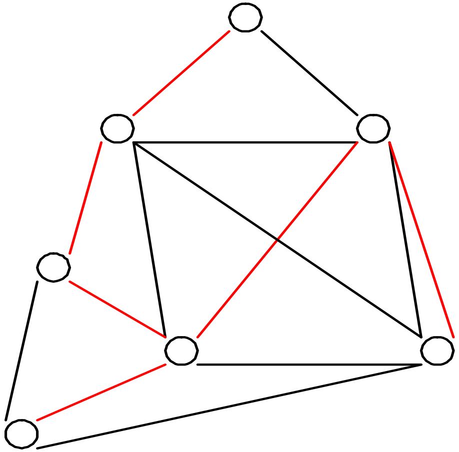
T4 : Soit $G = (X , A)$ un arbre, si on ajoute à A un arc u , le graphe $G' = (X , A \cup \{u \})$ contient un cycle et un seul.

D 3 : L'unique cycle obtenu en ajoutant une arête à un arbre partiel T d'un graphe connexe G est appelé **cycle fondamental** relativement à T.

R4 : Dans un graphe connexe G de n sommets et m arêtes, il existe par rapport à un arbre partiel T de G $(m - n + 1)$ cycles fondamentaux. Toute fois G peut contenir beaucoup plus que $(m-n +1)$ cycles différents .

Arbres et arborescences

Exemple :



Arbres et Arborescences
Problème de l'arbre de poids minimum

Position du problème

Soit G = (X, A) un graphe connexe et **p** une application de A dans **R** qui associe un **poids** (ou longueur) à chaque arc de G. Le problème de l'arbre de poids minimum de G consiste en la recherche d'un graphe partiel qui soit un arbre et pour lequel la somme des poids des arcs est minimum.

Algorithmes pour construire un arbre de poids minimum

- Algorithme de **Kruskal**
- Algorithme de **Prim**

Arbres et Arborescences
Problème de l'arbre de poids minimum

Algorithme de Kruskal :

0) poser $T = \emptyset$, $i = 1$
1) choisir un arc e_i de poids minimum dans E- T
ne déterminant aucun cycle avec des arcs de T
2) faire $T = T \cup \{ e_i \}$, $i = i + 1$
3) si $i < n$ alors aller en (1) sinon stop.

R5 : D'un point de vue informatique, cet algorithme requiert

- ✓ Qu'on trie initialement les arcs par ordre croissant de leur poids.
- ✓ Qu'on mette au point une procédure permettant de dire si un arc e détermine un cycle avec un sous ensemble d'arcs de U.

Arbres et Arborescences
Problème de l'arbre de poids minimum

D4 : Contraction d'un graphe G selon un de ses arcs **v** → $C_v(G)$.

T5 : Soit G = (X , A) un graphe et v = (x , y) ∈ A . Une condition nécessaire et suffisante pour que G soit un arbre est que C_v (G) soit un arbre.

T6: Soit G = (X , A) un graphe connexe, p : U --------> IR, et x ∈ X et u_x un arc de G tel que : $p(u_x) = \text{Min}\ [\ p(u)\ ]$

$\{ u \in \mathbf{A}$; u est adjacent à x et u n'est pas une boucle $\}$

Alors il existe T ⊂ A tel que (X,T) soit un arbre de poids minimum et que $u_x \in T$

Arbres et Arborescences
Problème de l'arbre de poids minimum

Algorithme de Prim :

O) Poser $T = \emptyset$;

1) - Si G ne comporte qu'un sommet, terminer (X, T) est un arbre de poids minimum.
   - Si G comporte plus d'un sommet, soit x un sommet de G, aller en 2).

2) Soit v un arc de G adjacent à x et n'est pas une boucle tel que :

$$p(v) = \mathrm{Min}\,(p(u))$$

pour $u \in A$; u adjacent à x et u n'est pas une boucle

- Poser $T := T \cup \{ v \}$, $G := C_v(G)$;
- Aller en (1) ;

Arbres et arborescences

D5 : Un **arbre binaire** T de n sommets ($n \geq 3$) est un arbre comportant un seul sommet de degré 2 (appelé **racine**) et dont tous les autres sommets sont de degré 1 ou 3.

Terminologie pour un arbre binaire : **sommet interne**, **niveau** d'un sommet, **profondeur** d'un arbre binaire.

Lemme 1 : Le nombre n de sommets de tout arbre binaire T est impair.

Lemme 2 : Le nombre p de sommets pendants de Tout arbre binaire T de n sommets est $p = (n + 1)/2$.

R6 : Il y a toujours exactement un sommet interne de moins que de sommets pendants dans un arbre binaire.

Arbres et arborescences

D6 : Un sommet **a** d'un graphe $G = (X , A)$ est une " **racine** " s'il existe dans G un chemin joignant **a** à x , pour tout $x \in X$.

D7 : Un graphe $G = (X , U)$ sur $n \geq 2$ sommets est une arborescence de racine **a** si :

- a est une racine de G
- G est un arbre

Remarques :

- Une arborescence est un arbre mais la réciproque est fausse en général.
- Le concept d'arborescence est essentiellement orienté.
- Un graphe orienté connexe peut ne pas avoir d'arborescence partielle.

Th7 : **Caractérisation des arborescences** .

Série : 2

<u>Exercice 1</u> :

1- En partant de la définition d'un arbre c'est à dire un graphe connexe et sans cycle, Montrer que tout arbre admet au moins 2 sommets pendants.

2- Montrer par récurrence qu'un graphe connexe sans cycle sur n sommets possède (n-1) arcs.

a- En utilisant le résultat de la question 1).

b- Sans utiliser le résultat de la question 1).

<u>Exercice 2</u> :

Soit G = (X, E) un graphe pondéré (donc avec une fonction poids définie en chaque arête), et T un sous ensemble propre de l'ensemble X des sommets de G. Montrer que (s, t) est une arête de poids minimum telle que s ∈ T et t ∈ **X**-T, alors il existe un arbre de G de poids minimum contenant (s, t).

Série : 2

<u>Exercice 3</u> : Graphe biparti et Coloration de graphe

Un graphe non orienté **biparti** (X, Y, E) est un graphe dans lequel l'ensemble des sommets X **U** Y peut être partitionné en deux classes X et Y de telle sorte que chaque arête ait une de ses extrémités dans une classe et l'autre extrémité dans l'autre.

On appelle k-coloration d'un graphe non orienté sans boucle G = (X,E) une partition de ses sommets en k sous ensembles (X_1, X_2,…, X_k) (coloriés avec les couleurs 1, 2, …, k de telle sorte que 2 sommets adjacents ne soient pas coloriés avec la même couleur. Un graphe est dit k-coloriable s'il est possible de le colorier avec k couleurs. Le **nombre chromatique** d'un graphe G, γ **(G)** est le nombre k minimum pour lequel G est k-coloriable.

Montrer que pour un graphe non orienté, les conditions suivantes sont équivalentes :

1- γ (G) = 2
2- G est biparti
3- G n'a pas de cycle de longueur impaire.

Série : 2

Exercice 4 :

Huit étudiants : E1, E2, E3, E4, E5, E6, E7, E8 doivent passer certaines épreuves parmi les suivantes : M1, M2, M3, M4, M5. On désire que tous les étudiants qui doivent subir une même épreuve le fassent simultanément. Chaque étudiant ne peut se présenter qu'à une épreuve au plus chaque jour. Voici la liste des épreuves que doit passer chaque étudiant :

E1 : M1, M4 ; **E2** : M1, M5 ; **E3** : M2, M3 ; **E4** : M2 ; **E5** : M3, M4 ;
E6 : M3, M4, M5 ; **E7** : M3, M5 ; **E8** : M4, M5.

Formuler le problème qui consiste à chercher le nombre minimum de jours nécessaires à l'organisation de toutes les épreuves comme un problème de coloration dans un graphe qu'on déterminera. Donner alors une solution à ce problème.