Feuille 2 de TD

À faire avant le TD

- Exercice 2.1 (2 premières questions)
- Exercice 2.4

1 Parcours

Exercice 1.1 Parcours en profondeur, itératif

Q 1.1.1 Ecrire un algorithme itératif de parcours en profondeur d'un graphe G à partir d'un sommet donné s qui produit en sortie la liste des sommets dans l'ordre dans lequel ils ont été visités.

Exercice 1.2 Complexité en temps des parcours $\star$

Q 1.2.1 Quelle est la complexité en temps du parcours en profondeur ?

```
PP(G,s)
 1  pour chaque sommet u de G faire
 2    couleur[u] ← BLANC
 3  parcours  ← []
 4  VISITER-PP(s)
 5  retourner parcours

VISITER-PP(u)
 1  couleur[u] ← GRIS // le sommet blanc u vient d'être découvert
 2  ajouter u en queue de la liste parcours
 3  pour chaque voisin v de u faire // on explore l'arête {u, v}
 4    si couleur[v] = BLANC alors
 5      VISITER-PP(v)
 6  couleur[u] ← NOIR // noircir u, car on en a fini avec lui
```

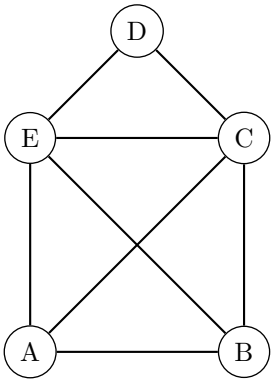
Q 1.2.2 Quel est la complexité en temps du parcours en largeur ?

```
PL(G = (V,E), s)
  1  pour chaque sommet u dans V faire
  2    couleur[u] ← BLANC
  3  couleur[s] ← GRIS
  4  F ← {s}
  5  parcours  ← [] // les sommets dans l'ordre de parcours
  6  // INVARIANT F contient tous les sommets gris
  7  tant que F n'est pas vide faire
  8    u ← défiler F
  9    ajouter u en queue de la liste parcours
 11    pour chaque voisin v de u faire
 12      si couleur[v] = BLANC alors
 13        couleur[v] ← GRIS
 14        enfiler v dans F
 15    couleur[u] ← NOIR
 16  retourner parcours
```

2 Arbres couvrants

Exercice 2.1 Ensembles d'arbres couvrants

On considère le graphe suivant :



Q 2.1.1 Lister l'ensemble des arbres couvrants à partir du sommet A calculés à partir d'un parcours en largeur ?

Q 2.1.2 Lister l'ensemble des arbres couvrants à partir du sommet A calculés à partir d'un parcours en profondeur ?

Q 2.1.3 Construire un algorithme qui permette de produire une liste des arbres couvrants d'un graphe connexe G à partir d'un sommet s (vous pourrez choisir l'une ou l'autre des deux stratégies de parcours).

Exercice 2.2 Connexité[1] $\star$

On considère l'ensemble des graphes $\mathcal{G}$ dont l'ensemble des sommets est $\{a, b, c, d\}$ et tels que pour tout graphe g de $\mathcal{G}$:

- les sous-graphes induits de g par les ensembles de sommets $\{a, b, c\}$, $\{b, c, d\}$ et $\{a, c, d\}$ sont connexes
- le sous-graphe induit de g par les sommets $\{a, b, d\}$ n'est pas connexe

Q 2.2.1 Montrer que g est nécessairement connexe et que le nombre d'arêtes est supérieur ou égal à 3.

Q 2.2.2 Montrer que le nombre d'arêtes est inférieur ou égal à 4.

Q 2.2.3 Déterminer tous les graphes de $\mathcal{G}$ ayant 3 arêtes.

Q 2.2.4 Déterminer tous les graphes de $\mathcal{G}$ ayant 4 arêtes.

Exercice 2.3 Nombre cyclomatique $\star$

Pour un graphe G ayant m arêtes, n sommets et p composantes connexes, on définit :

$$\nu(G) = m - n + p$$

$\nu(G)$ est appelé le nombre cyclomatique.

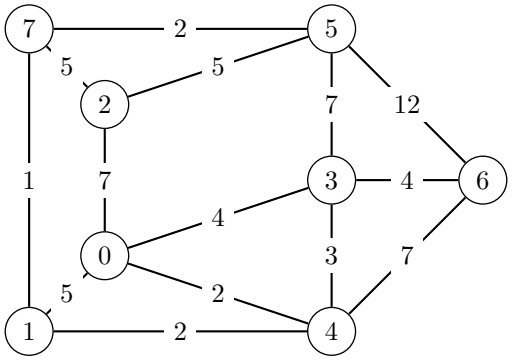
Q 2.3.1 Démontrer que $\nu(G) \geq 0$ pour tout graphe G. De plus, $\nu(G) = 0$ si et seulement si G est sans cycle.

[1] Tiré de *Introduction à la théorie des graphes*, I. Larramendy Valverde et A. Marie-Jeanne, Ellipses

Exercice 2.4 Application de l'algorithme de Prim ⋆

Pour tous avoir la même réponse, on considèrera que les voisins d'un sommet sont fournis dans l'ordre croissant de leur numéro.

Q 2.4.1 Exécutez l'algorithme de Prim sur le graphe suivant :



Q 2.4.2 Proposez une distribution de poids sur les arêtes de ce graphe telle qu'il n'existe qu'un seul arbre couvrant de poids minimum.

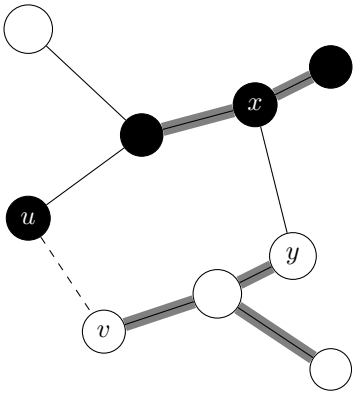
Q 2.4.3 Pouvez-vous donner une propriété sur les poids des arêtes d'un graphe qui implique qu'il n'existe qu'un seul arbre couvrant de poids minimum ?

Exercice 2.5 Preuve de l'algorithme de Prim ⋆

Q 2.5.1 Expliquer pourquoi l'algorithme de Prim se termine.

Théorème 1 *Soit $G = (V, E, w)$ un graphe non orienté pondéré. Soit A un sous-ensemble d'arêtes de E inclus dans un arbre couvrant de poids minimum de G. Soit P un sous-ensemble de sommets de V tel que aucune des arêtes de A ne relie un sommet de P à un sommet de $V - P$. Soit $\{u, v\}$ une arête de poids minimal parmi toutes les arêtes dont une extrêmité est dans P et l'autre extrêmité est dans $V - P$. Alors $\{u, v\}$ est une arête sûre pour A.*

Q 2.5.2 Considérer le schéma suivant[2] : dans l'ACPM T (les arêtes fines hormis $\{u, v\}$), dont A est un sous-ensemble d'arêtes (les arêtes surlignées), mais qui ne contient pas l'arête $\{u, v\}$, quel est le poids d'un nouvel arbre T' si on remplace $\{x, y\}$ par $\{u, v\}$? P est constitué des sommets noirs.



Q 2.5.3 Montrer alors que $w(T') \leq w(T)$

Q 2.5.4 Que dire alors de T' ?

Q 2.5.5 Pourquoi peut-on affirmer que $\{u, v\}$ est une arête sûre pour A ?

Corollaire 1 *Soit $G = (V, E, w)$ un graphe non orienté pondéré. Soit A un sous-ensemble d'arêtes de E inclus dans un arbre couvrant de poids minimum de G. Soit C une composante connexe de $G_A = (V, A)$. Si $\{u, v\}$ est une arête minimale reliant C à une autre composante connexe de G_A, alors $\{u, v\}$ est une arête sûre pour A.*

[2] Tiré de Cormen, section 23

Q 2.5.6 Démontrer ce corollaire.

Q 2.5.7 Conclure sur la preuve de l'algorithme de Prim.

Exercice 2.6 Autour des arbres couvrants de poids minimum $\star$

Q 2.6.1 Démontrez le lemme suivant :

Lemme 1 *Soit G un graphe pondéré non-orienté connexe et T un arbre couvrant. Si T est de poids minimum alors il contient au moins une arête de G de poids minimum.*

Q 2.6.2 Écrivez la réciproque de cette propriété.

Q 2.6.3 La réciproque est-elle vraie ? Si oui, démontrez-la. Si non donnez un contre-exemple.

Q 2.6.4 Démontrez le lemme suivant :

Lemme 2 *Soit G un graphe pondéré non-orienté connexe et C un cycle de G. Soit f une arête de poids maximum du cycle C. Un arbre couvrant de poids minimum ne contient pas f.*