

Algorithmique des Graphes – TD 6

Exercice 1

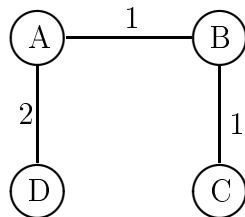
Soit \mathcal{G} un graphe non orienté connexe à n sommets et m arêtes, dont les arêtes sont valuées par des nombres.

Vous employez un jeune programmeur qui, pour construire un arbre recouvrant de poids minimal de \mathcal{G} , vous propose l'algorithme suivant :

```

soit  $\mathcal{A}$  le graphe dont les sommets sont ceux de  $\mathcal{G}$ ,
et qui ne contient aucune arête ;
soit  $s$  un sommet de  $\mathcal{A}$  ;
TANT QUE l'on modifie quelque chose
  parmi les arêtes de  $\mathcal{G}$  d'extrémité  $s$ 
    qui ne sont pas dans  $\mathcal{A}$  et ne créent pas de cycle dans  $\mathcal{A}$ ,
    choisir une arête  $s - s'$  de poids minimal ;
    ajouter l'arête  $s - s'$  à  $\mathcal{A}$  ;
     $s := s'$ 
FIN TANT QUE
  
```

1. Montrez que l'algorithme se termine.
2. En appliquant l'algorithme au graphe suivant :

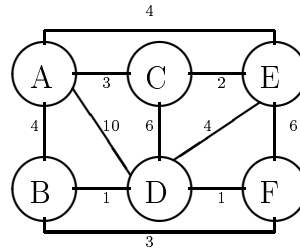


Montrez-lui que le graphe \mathcal{A} obtenu à la fin de l'algorithme peut ne pas être connexe.

3. Votre employé propose la correction suivante : si le graphe \mathcal{A} obtenu à la fin de l'algorithme n'est pas connexe, on recommence l'exécution de la boucle **TANT QUE** à partir d'une feuille de \mathcal{A} (c'est-à-dire un sommet extrémité d'au plus une arête de \mathcal{A}). Cette correction vous satisfait-elle ? Pourquoi ?

Exercice 2

En utilisant l'algorithme de Kruskal, déterminer un arbre recouvrant de poids minimal du graphe non orienté valué G suivant :



Exercice 3

Un **tas** est un arbre binaire dont chaque sommet est indexé par une *clé* (les clés étant prises dans un ensemble totalement ordonné), et qui vérifie les deux propriétés suivantes :

- Ordre vertical (la clé d'un sommet est toujours inférieure ou égale à celles de ses fils)
- Quasi-complétude : l'arbre binaire est complet, sauf peut-être à son niveau le plus bas, qui est rempli en partant de la gauche ; c'est-à-dire que, si les sommets sont numérotés dans l'ordre du parcours en largeur, le numéro du père de tout sommet i est $i/2$.

La structure de tas est une implémentation possible d'une file de priorité.

1. Dessiner tous les tas possibles (à l'ordre près des fils droit et gauche des sommets) dont l'ensemble de clés est $\{1, 2, 3, 4, 5\}$.
2. On considère un tas dont tous les sommets vérifient la propriété d'ordre vertical, sauf éventuellement le sommet d'indice i . Ecrire une procédure qui a pour données un tas et un indice i et qui exécute les réorganisations nécessaires. Quelle est sa complexité ?
3. Donner l'évolution d'un tas lorsque l'on insère la séquence $V = \{8, 3, 7, 9, 4\}$, à partir du tas vide, puis qu'on retire le minimum du tas jusqu'à ce qu'il soit vide.

Ecrire une procédure qui ajoute un élément à un tas et une fonction qui retire du tas l'élément de clé minimale, et retourne cet élément.

Exercice 4

En utilisant l'algorithme de Prim, déterminer un arbre recouvrant de poids minimal du graphe G non orienté valué ci-dessous. Quel rôle joue la structure de tas dans l'implémentation de l'algorithme de Prim ?

