

- Contrôle continu - SUJET A -

- Durée : 1 heure - Aucun document n'est autorisé -

Le barème est indicatif, la note totale est sur 15 points.

Les exercices peuvent être traités dans un ordre quelconque.

Sauf mention contraire, toutes les réponses doivent être justifiées, et notamment, lors du déroulement d'algorithmes, vous indiquerez la mise-à-jour des variables utilisées.

- Exercice 1 - Connexité (6 points) -

On considère que les graphes sont codés par liste d'arêtes.

- Question de cours* : Écrire un algorithme qui prend en entrée un graphe $G = (V, E)$ et qui retourne *FAUX* si G n'est pas connexe ou un tableau *pere* correspondant à un parcours du graphe G .
- Dans le cas où G est connexe, écrire un algorithme qui prend en entrée un sommet r de G ainsi que le tableau *pere* calculé précédemment et qui renvoie un tableau *arbre* correspond aux arêtes d'un arbre couvrant de G .
- Si G est connexe, pourquoi a-t-on $m \geq n - 1$ où n désigne le nombre de sommets de G et m son nombre d'arêtes ?
- Préciser la complexité des algorithmes que vous proposez aux questions a. et b.
- Soit G un graphe connexe. Comment trouver un sommet x de G tel que $G \setminus x$ soit connexe ?
- Écrire un algorithme qui prend en entrée un graphe G connexe et un entier k et qui retourne un ensemble X de sommets de G tel que $|X| = k$ et $G \setminus X$ est connexe. On pourra faire appel à une fonction ARBRE-COUVRANT qui calcule un arbre couvrant de G .

- Exercice 2 - Arbres couvrants de poids minimum (5 points) -

Soit $V = \{v_1, \dots, v_8\}$ un ensemble de noeuds que l'on désire connecter par un arbre. Le poids w de la connexion entre deux sommets est donné par la matrice (symétrique) suivante :

$$(w(v_i, v_j)) = \begin{pmatrix} 0 & 1 & 5 & 2 & 4 & 6 & 8 & 7 \\ . & 0 & 3 & 3 & 8 & 5 & 6 & 8 \\ . & . & 0 & 1 & 7 & 6 & 5 & 4 \\ . & . & . & 0 & 6 & 7 & 5 & 4 \\ . & . & . & . & 0 & 3 & 3 & 2 \\ . & . & . & . & . & 0 & 3 & 7 \\ . & . & . & . & . & . & 0 & 4 \\ . & . & . & . & . & . & . & 0 \end{pmatrix}$$

Le *poids* d'un ensemble F d'arêtes de G est défini comme la somme des poids des arêtes de F .

- Construire un arbre couvrant $T = (V, F)$ de poids minimum. On utilisera l'algorithme de Kruskal, en justifiant les étapes de sélection ou de rejet des arêtes.
- Existe-il un arbre couvrant de G de poids minimum dans lequel le sommet v_3 n'est pas une feuille ? Si oui lequel, sinon justifier. Et existe il un arbre couvrant de G de poids minimum dans lequel v_3 est une feuille ? Si oui lequel, sinon justifier.
- Montrer que si T est un arbre couvrant de poids minimum de G et f une feuille de T , alors $T \setminus f$ est un arbre couvrant de poids minimum de $G \setminus f$.

- d. Écrire un algorithme qui permette de répondre à la question de façon générale : étant donné un graphe G dont les arêtes sont pondérées par une fonction de poids w et un sommet x de G , existe-il un arbre couvrant de G de poids minimum dans lequel x est une feuille ?

- Exercice 3 - g++ ALGO.cc -o ALGO; ./ALGO (4 points)

On se donne l'algorithme suivant :

Algorithme : ALGO

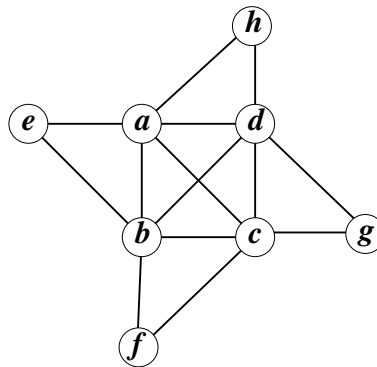
Données : Un graphe $G = (V, E)$ codé par listes de voisin $Vois(x)$ pour $x \in V$.

Résultat : Un ensemble X de sommets de G .

```

1 début
2    $X \leftarrow \emptyset$ ;
3   pour tous les  $x \in V$  faire  $c(x) \leftarrow 0$ ;
4    $i \leftarrow 0$ ;
5   tant que il existe  $v \notin X$  avec  $c(v) = i$  faire
6     Choisir un tel  $v$ ;
7      $i \leftarrow i + 1$ ;
8      $X \leftarrow X \cup \{v\}$ ;
9     pour tous les  $z \in Vois(v)$  faire
10       $c(z) \leftarrow c(z) + 1$ ;
11   fin
12 fin
13 retourner  $X$ ;
14 fin
    
```

- a. Dérouler ALGO sur le graphe G suivant :



- b. Quelle est la complexité de ALGO ? On notera n le nombre de sommets de G et m son nombre d'arêtes.
- c. Sans preuve, dire quelle propriété possède l'ensemble X .
- d. Proposer un autre déroulement de ALGO sur le graphe G , qui retourne un ensemble X de taille différente de celle obtenue à la question a.