Algorithmique des Graphes
L3 informatique

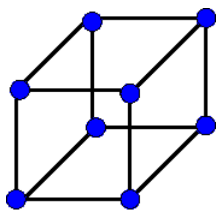
14 janvier 2019

Vous êtes invités à remettre une copie claire, concise, sans rature ni surcharge. Il est par ailleurs inutile de recopier l'énoncé...La note finale tiendra compte de la présentation générale de la copie.

Q 1. Que signifient les l'acronymes ACM, DAG ?

Q 2. Soit $G(S, A)$ un graphe orienté. Pour $x \in S$, on note $\Gamma(x)$ l'ensemble des sommets voisins de x. Compléter la formule :

$$\Gamma(x) = \{y \in S \mid \ldots\ldots\ldots\}.$$



Le cube.

Q 3. Le cube est-il Eulérien, Hamiltonien, planaire ?

Q 4. Que sont les faces d'un graphe planaire ? Rappeler la relation vérifiée par les nombres de sommets, arêtes et faces d'un graphe planaire.

Q 5. On écrit $x \circlearrowleft y$ pour dire que les sommets s et y sont sur un même circuit. Montrer que $\circlearrowleft$ est une relation d'équivalence. Que sont les classes d'équivalences ?

Q 6. Donner un exemple de problème pour chacune des classes : P, NP-complet et NP-dur.

Q 7. Quel algorithme de recherche d'arbre couvrant minimal s'apparente à l algorithme de Dijkstra ?

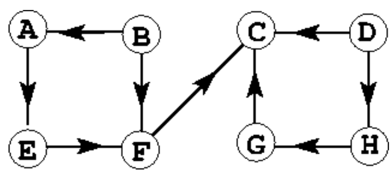
Q 8. L'approximation de Christophides s'appuie sur un problème polynomial que nous n'avons pas pu détailler en cours : lequel ?

Q 9. Soit $G(S, A)$ un graphe orienté. Un noyau du graphe est une partie $K \subseteq S$ tel que :

$$\forall x \in K,\ \Gamma(x) \cap K = \emptyset \quad \forall x \notin K,\ \Gamma(x) \cap K \neq \emptyset$$

a. Donner un exemple de graphe d'ordre 3 sans noyau.

b. Un graphe orienté acyclique fini possède un noyau. Expliquer pourquoi.



Le graphe Λ.

Q 10. On considère le graphe Λ.

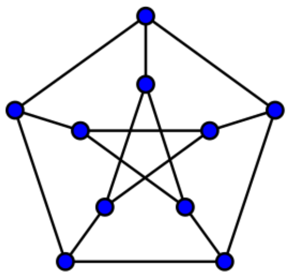
a. Effectuer un tri topologique en traçant les temps de traitement du parcours en profondeur réalisé.

b. En déduire la fonction de Grundy.

Q 11. Préciser le nombre de composantes connexes d'un graphe obtenu à partir d'un arbre par suppression d'un sommet de degré r. Justifier soigneusement la réponse.

Q 12. Lister trois structures de données avancées rencontrées dans le cours en précisant un algorithme dans lesquel elles ont été utilisées.

Q 13. Comment peut-on prouver que le problème du cycle Eulérien est plus facile que le problème du cycle Hamiltonien ? Préciser les complexités de ces problème.



Graphe de Petersen.

Q 14. Montrer qu'un graphe est 2-coloriable ssi il ne possède pas de cycle de longueur impair.

Q 15. On applique l'algorithme de Kruskal sur le graphe de Petersen pondéré par longeur des arêtes. Dessiner les graphes intermédiaires.

Q 16. Montrer que le graphe de Petersen n'est pas planaire.

Q 17. Un graphe non orienté est implanté en langage C par liste d'adjacence. Écrire une fonction `int adj(int x, int y)` qui renvoie vrai si les sommets x et y sont adjacents, faux sinon.

```c
typedef struct _li_{
    int s;
    struct _li_ * svt;
} enrlistint , *listint ;

typedef struct {
      int       nbs;
      listint   *adj;
} graphe;
```

Q 18. Écrire une fonction `int nbc( graphe G )` qui retourne le nombre de composantes de connexes du graphe G.

Q 19. Un point d'articulation d'un graphe non orienté est un sommet x tel que le sous-graphe induit par suppression de x possède strictement plus de composantes connexes que le graphe original. En particulier, si le graphe était connexe avant de retirer ce sommet, il devient donc non connexe.

a. Quels sont les points d'articulation de la version non-orientée du graphe Λ.

b. Quels sont les points d'articulation d'un arbre ?

c. Décrire un algorithme pour déterminer les points d'articulation d'un graphe.

d. Préciser le temps de calcul.