UNIVERSITÉ DE REIMS CHAMPAGNE-ARDENNE

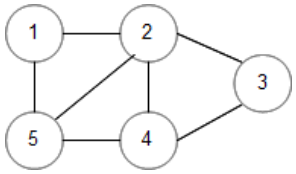
LICENCE Informatique
Info0501 – 2018/2019

Travaux dirigés n° 4

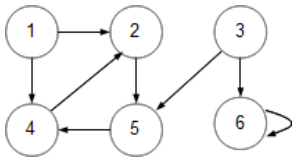
Graphes : représentation et parcours

Exercice 1 (Représentation des graphes)

1°) Soit le graphe non orienté ci-contre, donnez
a) sa taille ;
b) son ordre ;
c) sa représentation par listes d'adjacences ;
d) sa représentation par matrice d'adjacences.



2°) Soit le graphe orienté ci-contre, donnez sa représentation
a) par listes d'adjacences ;
b) par matrice d'adjacences.



3°) En fonction de son nombre de sommets et de son nombre d'arêtes ou d'arcs, donnez la quantité de mémoire nécessaire pour stocker un graphe par

a) listes d'adjacences ;

b) matrice d'adjacences.

4°) Donnez le temps d'exécution nécessaire pour afficher tous les sommets adjacents à un sommet u lorsque le graphe est représenté par

a) listes d'adjacences ;

b) matrice d'adjacences.

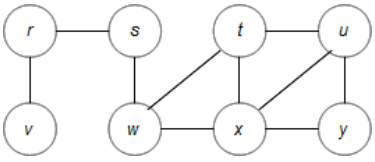
5°) Donnez le temps d'exécution nécessaire pour déterminer si l'arête ou l'arc $(u, v) \in A$ lorsque le graphe est représenté par

a) listes d'adjacences ;

b) matrice d'adjacences.

Exercice 2 (Parcours en largeur)

1°) Soit le graphe ci-contre, illustrez le fonctionnement d'un parcours en largeur démarrant au sommet s en donnant, pour chaque sommet, sa distance par rapport à s (le plus petit nombre d'arêtes comprises entre celui-ci et s). Donnez aussi l'arbre de parcours résultant.

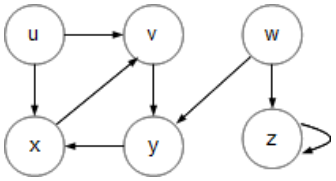


2°) Définissez la procédure `parcours-largeur` qui effectue le parcours en largeur d'un graphe G à partir de s et qui calcule la distance (plus petit nombre d'arcs) entre s et chaque sommet. Donnez-en le temps d'exécution.

3°) Définissez la procédure `afficher-chemin` qui affiche les sommets d'un plus court chemin reliant le sommet s à un sommet v de G, en supposant que `parcours-largeur` a déjà calculé un arbre de parcours en largeur. Donnez-en le temps d'exécution.

Exercice 3 (Parcours en profondeur)

1°) Soit le graphe ci-contre, illustrez le fonctionnement d'un parcours en profondeur en considérant les sommets dans l'ordre lexicographique et en donnant, pour chaque sommet, sa date de découverte et sa date de fin de traitement. Donnez aussi la forêt de parcours résultante.

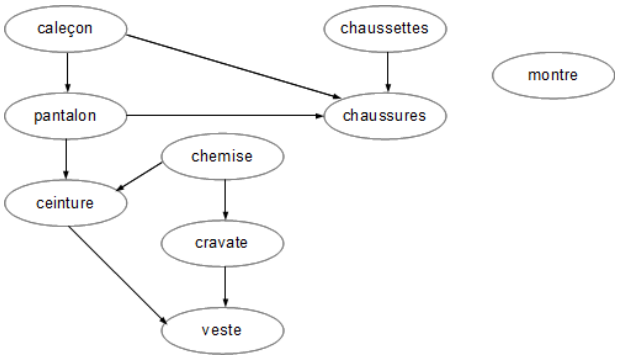


2°) Définissez la procédure `parcours-profondeur` qui effectue le parcours en profondeur d'un graphe G et qui calcule, pour chaque sommet, les dates de découverte et de fin de traitement. Donnez-en le temps d'exécution.

Exercice 4 (Tri topologique)

Les graphes acycliques orientés sont utilisés dans de nombreuses applications pour représenter des précédences entre événements. Le tri topologique d'un graphe acyclique orienté $G = (S, A)$ consiste à ordonner linéairement tous ses sommets de sorte que, si G contient un arc (u, v), u apparaisse avant v dans le tri.

La figure ci-contre donne un exemple de graphe acyclique. Celui-ci a été conçu par le savant professeur Cosinus qui avait de la difficulté à s'habiller le matin. Chaque arc (u, v) signifie que le vêtement u doit être enfilé avant le vêtement v. L'absence d'arc entre deux vêtements signifie que ceux-ci peuvent être enfilés dans n'importe quel ordre. À chaque matin, il réalise un tri topologique de ce graphe afin d'obtenir un ordre permettant de s'habiller correctement.



1°) Illustrez l'application d'un parcours en profondeur à ce graphe en notant, pour chaque sommet, sa date de découverte et sa date de fin de traitement. Déduisez-en un ordre permettant de s'habiller correctement.

2°) Définissez la procédure `tri-topologique` qui retourne une liste contenant tous les sommets triés topologiquement d'un graphe acyclique orienté. Donnez-en le temps d'exécution.

Références

Cormen T. H., Leiserson C. E., Rivest R. L. et Stein C., "Algorithmique" 3e édition, Dunod, 2010.