

Devoir Surveillé Terminal

Tous les algorithmes, fonctions et procédures doivent être donnés en langage algorithmique et en respectant les conventions vues en CM et en TD (pas de C).

Vous pouvez rédiger et rendre vos réponses directement en un seul fichier dans l'un des formats électroniques suivants : txt, doc, docx, odt, pdf, ou bien répondre sur feuille et envoyer des scans/photos en format jpg ou pdf. Si vous envoyez plusieurs scans/photos, vous pouvez les regrouper dans une archive zip.

Assurez-vous que vos documents soient lisibles et de taille raisonnable, puis déposez le résultat dans le dépôt de fichiers prévu à cet effet sur la page Moodle d'Info0501 avant l'heure limite.

1 heure 30 - 30 points

Exercice 1 (Files de priorités, tables de hachage et arbres binaires de recherche - 16 points)

Une file de priorités supporte les opérations **extraire-max**, **insérer** et **augmenter-clé**. On peut implémenter une file de priorités à l'aide d'un tas, mais on peut aussi le faire avec une table de hachage ou un arbre binaire de recherche. Soit les tâches et priorités ci-contre :

1°) En expliquant les éventuelles étapes intermédiaires et en donnant le résultat final, montrez comment construire une table de hachage à partir de ces tâches avec $m = 5$. Précisez comment vous gérez les collisions et l'insertion dans la table de hachage.

Tâche	Priorité
Réviser Info0503	4
Réviser Info0501	8
Réviser Info0502	9
Finir les TP d'Info0501	5
Finir les TP d'Info0503	7
Dormir	1
Manger	3
Prendre une douche	2
Prendre une journée de congé	11
Relaxer	10

2°) En utilisant comme exemple la table de hachage construite à la question précédente, montrez comment on peut réaliser les trois opérations des files de priorités mentionnées ci-haut avec une implémentation sous forme de table de hachage, puis écrivez le pseudo-code des trois fonctions réalisant ces opérations. Dans chaque cas, donnez le temps d'exécution (en notation O) de la fonction en justifiant votre réponse.

3°) En expliquant les éventuelles étapes intermédiaires et en donnant le résultat final, montrez comment construire un arbre binaire de recherche à partir de ces tâches.

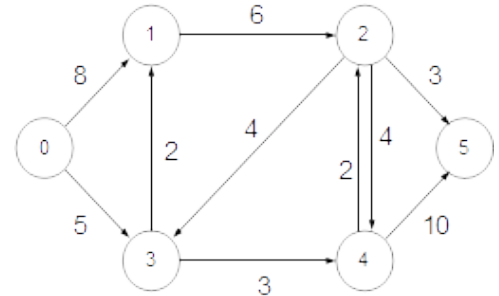
4°) En utilisant comme exemple l'arbre binaire de recherche construit à la question précédente, montrez comment on peut réaliser les trois opérations des files de priorités mentionnées ci-haut avec une implémentation sous forme d'arbre binaire de recherche, puis écrivez le pseudo-code des trois fonctions réalisant ces opérations. Dans chaque cas, donnez le temps d'exécution (en notation O) de la fonction en justifiant votre réponse.

Exercice 2 (Plus courts chemins - 10 points)

Pour les deux questions qui suivent, vous devez détailler le fonctionnement des algorithmes dans les premières itérations et dans la dernière itération qui mène à la terminaison de l'algorithme, mais il n'est pas nécessaire de le faire dans les itérations intermédiaires si celles-ci n'apportent aucun élément nouveau dans le fonctionnement de l'algorithme.

Soit le graphe ci-contre,

1°) En expliquant votre démarche et en montrant comment est utilisée la file de priorités, donnez et illustrez le résultat obtenu (distance(s) et chemin(s)) en appliquant l'algorithme de Dijkstra implémenté par files de priorités à partir du sommet 0. Lorsque les sommets/arcs peuvent être choisis dans un ordre arbitraire, vous les choisirez par ordre croissant d'indice.



2°) En expliquant votre démarche, donnez le résultat obtenu (distance(s) et chemin(s)) en appliquant l'algorithme de Bellman-Ford à partir du sommet 0. Lorsque les sommets/arcs peuvent être choisis dans un ordre arbitraire, vous les choisirez par ordre croissant d'indice.

Exercice 3 (Suppression dans un arbre binaire de recherche - 4 points)

Soit l'arbre binaire de recherche ci-contre :

1°) En expliquant les éventuelles étapes intermédiaires, donnez l'arbre obtenu après les opérations successives suivantes :

- suppression du noeud de clé 2 ;
- suppression du noeud de clé 16.

