

Département d'Informatique

L3 Informatique

Examen final Algorithmique 4 Mai 2016

Durée : 1 Heure 30. Cours, documents autorisés

Exercice1 : Arbre_B

4pts

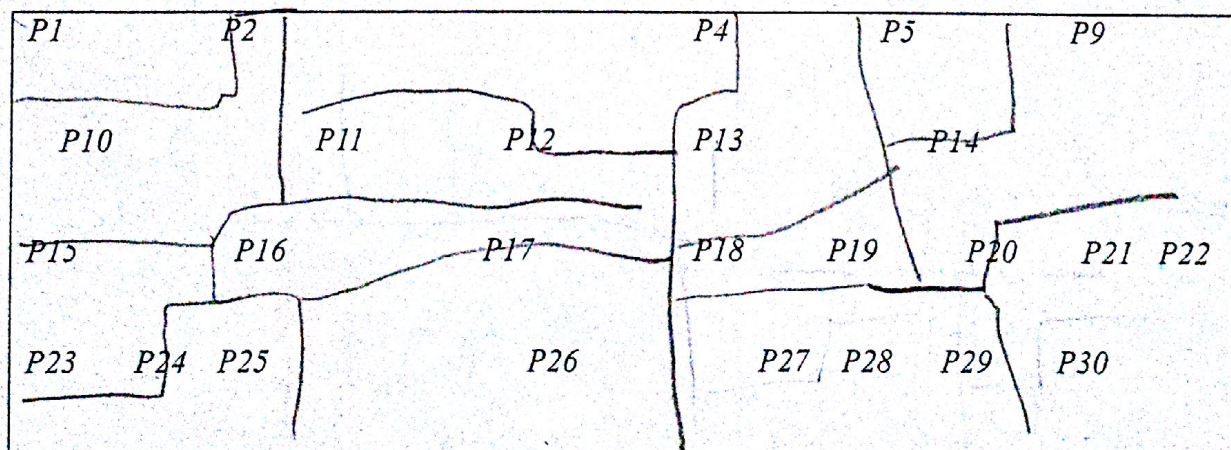
Soit le fichier séquentiel suivant (on ne donne pour chaque article du fichier que la clé sur laquelle on souhaite construire l'arbre_B) : 1 15 3 12 6 4 11 7 2 5 14 8 9 17 10 13 16

- . Donnez l'arbre_B d'ordre 2 résultant après l'insertion de tous les articles du fichier séquentiel. Que devient cet arbre s'il était d'ordre 3 ?
- . Combien de nœuds différents (racine et feuilles comprises) doit-on parcourir dans l'index pour répondre à une requête qui cherche des articles dont la clé appartient à l'intervalle $[5,10]$?
- . Que devient l'arbre d'ordre 2 après suppression de la clé 2 ? (commentez et justifiez brièvement la suppression).

Exercice2 : Kd Arbre

5pts

. Partitionnez l'espace 2D de la figure ci dessous en tenant compte du principe de fonctionnement de l'algorithme de construction d'un Kd_Arbre (Redessinez cette figure sur votre copie, utilisez des couleurs, légendes et commentez si nécessaire l'arbre kd obtenu)



25

Etant donné un point P et un échantillon de points P_1, P_2, \dots, P_n , on souhaite écrire un algorithme de recherche du point P_i le plus de P . Pour les points sont stockés à l'aide d'un Kd_Arbre. On suppose connues les fonctions suivantes :

fonction *intersect* (b :boite, c :cercle) retourner boolean ; // retourne vrai si la boite b coupe le cercle c ;
fonction *feuille* (a :kdarbre) retourner boolean ; // retourne vrai si le nœud de l'arbre kd est une feuille.
fonction *le_point* (a :kdarbre) retourner point ; // retourne le point stocké dans la feuille a ;
fonction *la_boite* (a :kdarbre) retourner boite ; // retourne la boite englobante associée au nœud a ;
fonction *construire_cercle* (r :reel, c :point) retourner cercle ; // construit le cercle de rayon r et de centre c ;

. Ecrire une fonction récursive qui prend en entrée un arbre a de type $kdarbre$ et un point p puis qui renvoie le point le plus proche de p .

Exercice3 : Arbre binaire de recherche

4pts

On suppose connues la procédure d'insertion d'un élément dans un arbre binaire de recherche ainsi que la procédure de parcours *infixé* d'un arbre binaire.

. Ecrire une nouvelle méthode de tri d'un vecteur d'entiers en utilisant ces deux procédures

Exercice4 : Géométrie Algorithmique

7 pts

. Expliquer en quelques lignes la phrase extraite du cours : « Sometimes degenerate cases can be handled by adding "special cases" without increasing algorithmic complexity (but usually this increases "implementation complexity") »

. Expliquer en quelques lignes à quoi correspond une « enveloppe convexe » et donner des exemples d'applications

. Ecrire un algorithme de type « Diviser pour régner » permettant de calculer cette enveloppe convexe, illustré par les schémas ci-dessous, et expliquer quelle serait sa performance théorique.

