Contrôle continu du 02-Avril 2010. Durée 2 heures

Recherche dichotomique

a) Écrire une méthode `recherche()` qui prend un tableau `tab` d'entiers **triés** et une valeur entière `x` qui retourne l'indice `i` tel que `tab[i] = x` ou `-1` si `x` ne se trouve pas dans `tab`.

b) Donner les complexités en temps dans le cas pire, dans le meilleur de cas et en moyenne de cet algorithme. Justifiez vos réponses.

Question de cours

c) Écrire une méthode **non récursive** `trinonrec()` qui trie un tableau d'entiers dans un ordre croissant.

d) Que signifient les termes **conditions d'arrêt** et **appels récursifs** ? Quelles sont les RÈGLES à suivre pour écrire des programmes récursifs ?

e) Écrire une méthode **récursive** `trirec()` qui trie un tableau d'entiers dans un ordre décroissant. Spécifier alors quelles sont les **conditions d'arrêt** ainsi que les **appels récursifs** dans votre programme. Est-ce-qu'ils suivent les RÈGLES décrites à la question d) ?

Question algorithmique

Dans cet exercice, on cherche à trouver dans un tableau `tab` d'entiers deux entiers `a` et `b` donc la somme vaut `x`.
f) Décrire un algorithme qui va écrire un tableau constitué de toutes les sommes de paire d'entiers de `tab`. Ecrire le programme correspondant en JAVA. Quelle est la complexité de cet algorithme ? Expliquez brièvement comment faire alors pour trouver la valeur `x` ?

g) On commence par trier le tableau `tab` pour obtenir un autre tableau `tabtrié`. Dans ce second algorithme, on considère la somme des deux valeurs extrêmes de `tabtrié`. Si cette somme est plus grande (respectivement plus petite) que `x` alors on enlève la plus grande (respectivement la plus petite) valeur du tableau trié `tabtrié`. On continue jusquà ce qu'on trouve `x` ou non. En supposant que le tri du tableau `tri` soit le tri à bulles, quelle est la complexité de votre algorithme ?