Université de Paris

L2 informatique - Année 2020–2021

TD d'Éléments d'Algorithmique n° 5

Dans ce TD, "trié" signifie "trié par ordre croissant".
* Les exercices marqués d'une étoile sont à faire à la maison.

Exercice 1. *Dichotomie.*

1. Exécutez l'algorithme de dichotomie récursif vu en cours sur la valeur 42 et chacun des tableaux suivants :

$$[1, 2, 42, 57, 99] \qquad [1, 2, 3, 4, 42] \qquad [1, 2, 3, 57, 99]$$

   Combien de comparaisons a-t-il fallu faire dans chaque cas ?
2. Même question avec l'algorithme itératif.

Exercice 2. *Recherche linéaire et dichotomie.*
On considère l'algorithme de recherche linéaire étudié au premier cours.

1. Combien de tests d'égalité cet algorithme fait-il dans le pire des cas ? Pour quelles données ce cas est-il atteint ?
2. Intuitivement, combien de tests d'égalité cet algorithme fait-il en moyenne lorsque l'élément recherché est trouvé ?

On considère maintenant l'algorithme de recherche dichotomique présenté en cours.

3. On suppose qu'un test d'égalité a le même coût qu'une comparaison. On a un tableau de taille n dans lequel on veut effectuer m recherches. Combien de tests d'égalité et de comparaisons faut-il faire dans le pire des cas :
   (a) si on effectue m recherches linéaires ;
   (b) si on effectue un tri par insertion suivi de m recherches par dichotomie.

Exercice 3. *Diviser pour régner.*
On dispose d'un tas de n pièces $p_0, p_1 \ldots p_{n-1}$ dont exactement une est fausse ; toutes les pièces ont le même poids sauf la pièce fausse, qui est plus légère. On dispose d'une balance à deux plateaux.

1. Combien de pesées sont nécessaires pour déterminer la pièce fausse lorsque $n = 4$? $n = 8$? $n = 9$?
2. Écrivez un algorithme récursif qui permet de déterminer la pièce fausse lorsque le nombre de pièces est une puissance de deux. Combien de pesées fait-il ?
3. Même question dans le cas général.

Exercice 4. *Point fixe*.*
On considère un tableau trié T d'entiers relatifs tous distincts. On dit qu'un indice i est un *point fixe* de T si $T[i] = i$, un *pré-point fixe* si $T[i] \leq i$, et *post-point fixe* si $T[i] \geq i$.

1. Quels sont les pré- et post-points fixes du tableau $T = [-1, 0, 1, 3, 4, 8]$?
2. Quelle propriété satisfont les ensembles de pré- et post-points fixes d'un tel tableau ?
3. Déduisez-en un algorithme itératif efficace qui prend en entrée un tableau trié T d'entiers relatifs tous distincts et retourne vrai si et seulement s'il existe un indice i tel que $T[i] = i$. Combien votre algorithme fait-il de comparaisons dans le cas pire ?