

Nom :

Devoir 5

Soit une liste L de N entiers distincts triés par ordre croissant. On souhaite déterminer la présence d'un élément x dans L.

1) Proposer une fonction recherche_simple(x,L) qui renvoie le nombre d'itérations nécessaires pour trouver x et sa position dans L. Le booléen False est renvoyé si x n'est pas présent dans L.

2) Estimer le nombre de comparaisons dans le meilleur cas et dans le pire cas pour trouver x dans L avec le programme précédent.

Meilleur cas :

Pire cas :

3) Proposer une recherche itérative dichotomique d'un élément x dans L. Cette fonction renvoie le nombre d'itérations nécessaires pour trouver x et sa position dans L. Le booléen False est renvoyé si x n'est pas présent dans L.

4) Estimer le nombre de comparaisons dans le meilleur cas et dans le pire cas pour trouver x dans L avec le programme précédent

Meilleur cas :

Pire cas :

Nom :

Devoir 5

Soit une liste L de N entiers distincts triés par ordre croissant. On souhaite déterminer la présence d'un élément x dans L.

1) Proposer une fonction recherche_simple(x,L) qui renvoie le nombre d'itérations nécessaires pour trouver x et sa position dans L. Le booléen False est renvoyé si x n'est pas présent dans L.

2) Estimer le nombre de comparaisons dans le meilleur cas et dans le pire cas pour trouver x dans L avec le programme précédent.

Meilleur cas :

Pire cas :

3) Proposer une recherche itérative dichotomique d'un élément x dans L. Cette fonction renvoie le nombre d'itérations nécessaires pour trouver x et sa position dans L. Le booléen False est renvoyé si x n'est pas présent dans L.

4) Estimer le nombre de comparaisons dans le meilleur cas et dans le pire cas pour trouver x dans L avec le programme précédent

Meilleur cas :

Pire cas :