R1.01 - Initiation au développement

Sujet: Rechercher dans un tableau

Instructeur: Joseph AZAR - joseph.azar@univ-fcomte.fr

Problème:

Par exemple, supposons que vous écriviez un programme qui suit les étudiants de votre université. Le programme doit pouvoir ajouter un étudiant, supprimer un étudiant, afficher les données d'un étudiant, etc. Toutes ces opérations nécessitent que vous recherchiez d'abord l'étudiant dans un tableau d'identifiants d'étudiants (même l'opération d'ajout d'un étudiant nécessite une recherche, pour s'assurer que l'étudiant n'est pas déjà dans le tableau).

ETUDIANTS ID

2	4	7	10	11	45	50	59	60	66	69	70	79

Idée

Pour utiliser un tableau, vous devez accéder à ses éléments individuels. Si vous connaissez l'emplacement de l'élément qui vous intéresse, vous accédez simplement à l'élément en mettant l'index de l'élément entre crochets. Mais si vous ne connaissez pas l'emplacement de l'élément, vous devez le rechercher.

Exemple

Etudiant ID 11 $\longrightarrow$ Karl, note = 17

Approches

Approche 1: Recherche séquentielle
Approche 2: Recherche binaire (Recherche dichotomique)

Un premier pseudo-code

Debut

tab_id ← {2, 4, 7, 10, 11, 45, 50, 59, 60, 66, 69, 70, 79}

tab_prenoms ← {"prenoms 1", "prenoms 2", ...}

tab_notes ← {note 1, note 2, ...}

lire(id)

index ← Rechercher (tab_id, id)

Afficher (tab_prenoms, tab_notes, index)

Fin

Recherche sequentielle

Idée

Parcourez le tableau de manière séquentielle et comparez la valeur de chaque élément du tableau avec la valeur recherchée. Lorsque vous trouvez une correspondance, faites quelque chose et retourner.

Algorithme

```
i ←——— 0

Tant que i < nombre d'éléments dans le tableau
   Si tab[i] = la valeur recherchée
        Faire qque chose
        retourner i
   Fin si

   incrementer i

Fin Tant que
retourner -1
```

Recherche binaire/dichotomique

Idée

La recherche binaire est l'autre approche de recherche courante pour une liste de valeurs. Pour que la recherche binaire fonctionne, les éléments du tableau doivent déjà être triés. Supposons que le tableau est dans l'ordre croissant. La recherche binaire compare d'abord la clé avec l'élément au milieu du tableau.
Considérez les trois cas suivants :
1. Si la clé est inférieure à l'élément du milieu, vous devez continuer à rechercher la clé uniquement dans la première moitié du tableau.
2. Si la clé est égale à l'élément du milieu, la recherche se termine par une correspondance.
3. Si la clé est supérieure à l'élément du milieu, vous devez continuer à rechercher la clé uniquement dans la seconde moitié du tableau.

Clairement, la méthode de recherche binaire élimine au moins la moitié du tableau après chaque comparaison. Supposons que le tableau comporte n éléments. Par commodité, soit n une puissance de 2. Après la première comparaison, il reste n/2 éléments pour une recherche ultérieure ; après la deuxième comparaison, il reste (n/2)/2 éléments.

Recherche binaire/dichotomique

Exemple

2	4	7	10	11	45	50	59	60	66	69	70	79

Clé = 11

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]
2	4	7	10	11	45	50	59	60	66	69	70	79
low						milieu						high

[0]	[1]	[2]	[3]	[4]	[5]
2	4	7	10	11	45
low		milieu			high

[3]	[4]	[5]
10	11	45
low	milieu	high

Recherche binaire/dichotomique

Algorithme

low ← 0

high ← tab.length -1

Tant que high >= low Que se passerait-il si nous remplacions (high >= low) par (high > low) ?

  milieu ← (low + high) / 2

  Si clé < tab[milieu]

  high ← milieu - 1

  Si clé = tab[milieu]

    retourner milieu

  Si clé > tab[milieu]

    low ← milieu + 1

retourner -1

Recherche binaire/dichotomique

Exercice

Complétez le tableau suivant :

2	4	7	10	11	45	50	59	60	66	69	70	79

	Low	High	Valeur retournée
binarySearch(tab,11)			
binarySearch(tab,2)			
binarySearch(tab,12)			
binarySearch(tab,1)			
binarySearch(tab,3)			

Recherche binaire/dichotomique

Exercice

1. Lorsque la clé n'est pas trouvée, la méthode renvoie une valeur négative (-1) pour indiquer que la clé n'est pas dans le tableau. Pouvez-vous réfléchir à la façon de modifier la fonction afin qu'elle renvoie le point d'insertion de la clé lorsqu'elle n'est pas trouvée ?

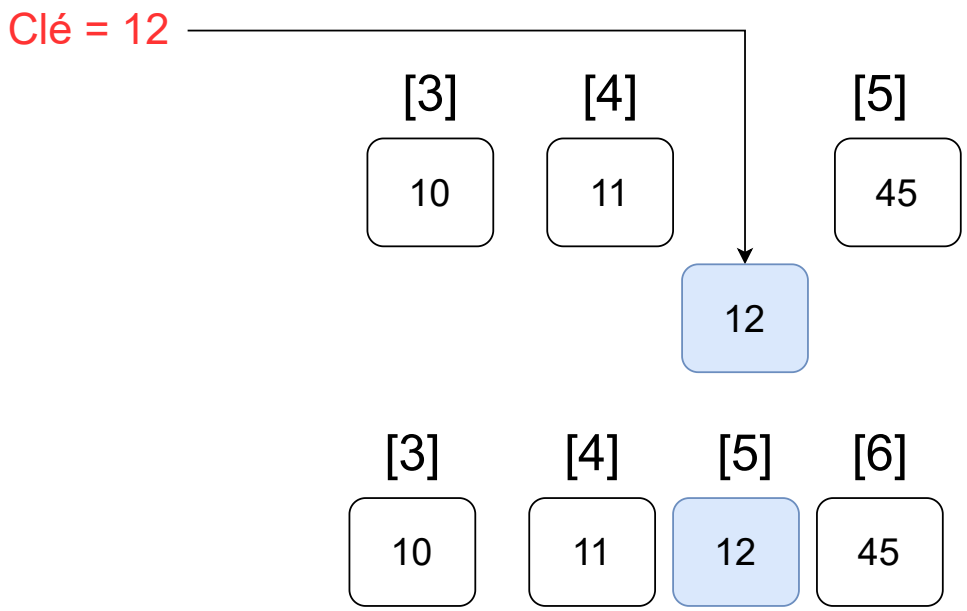
2. Si la méthode de recherche binaire renvoie −4, la clé est-elle dans la liste ? Où la clé doit-elle être insérée si vous souhaitez insérer la clé dans la liste ?

3. Complétez le tableau ci-dessous après les modifications ci-dessus

	Low	High	Valeur retournée
binarySearch(tab,1)			
binarySearch(tab,12)			

4. Écrivez l'algorithme d'insertion d'un élément au point d'insertion renvoyé par la fonction *binarySearch*.

Insérer un élément à une position spécifique dans un tableau



Algorithme

InsererPosition entrée: tab, index, id

sortie: tab_n

1. Créer un nouveau tableau avec une taille supérieure à la taille précédente (N+1)

2. Copiez tous les éléments du tableau précédent dans le nouveau tableau jusqu'à la position index

3. Insérez l'élément id à la position index

4. Insérez le reste des éléments du tableau précédent dans le nouveau tableau après l'element index