

TD -Recherche d'un élément dans une liste

Recherche d'un élément dans une liste non triée

Le problème est de rechercher la présence d'un élément dans une liste.

En adoptant le paradigme "**diviser pour régner**", l'idée pour résoudre cette question est de **rechercher récursivement** l'élément dans la **première moitié** de la liste et dans la **seconde**, puis de **combiner les résultats** via l'opérateur logique **or**.

En effet, l'élément recherché sera dans la liste s'il est dans la première moitié ou dans la seconde. La condition d'arrêt à la récursivité sera l'obtention d'une liste à un seul élément, car il est alors immédiat de conclure si l'élément recherché appartient à une telle liste ou non.

Voici donc les trois étapes de la résolution de ce problème via la méthode "**diviser pour régner**":

- **Diviser la liste en deux** sous-listes en la "**coupant**" par la moitié.
- **Rechercher** la présence de **l'élément** dans chacune de ces sous-listes. **Arrêter** la récursion lorsque les listes n'ont plus qu'**un seul élément**.
- Combiner avec l'opérateur logique **or** les résultats obtenus.

Voici l'algorithme

```
Données : l est une liste d'entiers  
x l'élément à rechercher  
d est l'indice du 1er élément de la liste (0)  
f est l'indice du dernier élément de la liste  
fonction recherche(l,x,d,f):  
Si d = f alors  
  | renvoyer l[d] == x  
m ← (d + f)//2  
renvoyer recherche((l,x,d,m) ou recherche(l,x,m+1,f)
```

Exercice 1:

Implémenter cet algorithme et le tester sur cette liste et le tester pour quelques valeurs:

L=[95, 28, 36, 52, 85, 56, 34, 59, 17, 26, 16, 25, 69, 98, 4, 85, 81, 48, 11, 57]

Recherche d'un élément dans une liste triée

Le problème est de rechercher la présence d'un élément dans une liste préalablement triée.

L'idée pour résoudre cette question est d'utiliser **une méthode dichotomique**.

La **liste** étant **triée**, après **comparaison** avec l'élément du "**milieu**" il est en effet facile de voir dans quelle moitié peut éventuellement se trouver l'élément cherché. On aura plus alors qu'à recommencer récursivement la recherche.

Le terme dichotomie provient du grec ancien "dikhotomia", signifiant "division en deux parties".

Voici donc les trois étapes de la résolution de ce problème via la méthode "**diviser pour régner**".

- **Diviser** la liste en **deux sous-listes** en la "coupant" par la **moitié**.
- Rechercher **récursivement** la présence de l'élément recherché dans la "bonne" des deux sous-listes après l'avoir comparé à l'élément situé au milieu de la liste.
- Pas de résultats à combiner puisque l'on ne "**travaille**" que sur l'**une des deux sous-listes**.

Exercice 2:

En vous inspirant de ce qui a été vu plus haut, écrire un algorithme et le programme correspondant qui réalise une recherche dichotomique dans un tableau trié. Tester cet algorithme pour quelques valeurs de

T = [4, 11, 16, 17, 25, 26, 28, 34, 36, 48, 52, 56, 57, 59, 69, 81, 85, 85, 95, 98].

L'algorithme

Le programme