

CI 2 : ALGORITHMIQUE & PROGRAMMATION

CHAPITRE 2 – INTRODUCTION À L'ALGORITHMIQUE – COMPLÉMENTS

Exercice 1 – Recherche dans un tableau désordonné

Question – 1 *Étant donné un tableau de nombres, écrire la fonction permettant de savoir si un nombre est présent ou non dans le tableau. On utilisera pour cela la structure while.*

Question – 2 *Étant donné un tableau de nombres, écrire la fonction permettant de savoir si un nombre est présent ou non dans le tableau. On utilisera pour cela la structure for.*

Question – 3 *Étant donné un tableau de nombres, écrire la fonction renvoyant le premier indice où une valeur apparaît dans le tableau.*

Exercice 2 – Recherche dans un tableau ordonné – Divide and conquer

Lorsqu'un tableau est trié, les méthodes précédentes restent envisageables pour déterminer si une valeur appartient au tableau. En revanche, elles imposent de parcourir toutes les valeurs du tableau ce qui peut être très coûteux en temps de calcul si le tableau est très grand. Pour améliorer les performances de l'algorithme de recherche, on peut utiliser un algorithme du type *Divide and conquer* – Diviser pour régner.

La méthode consiste en découper le tableau en 2. On regarde ensuite dans quelle partie du tableau est susceptible de se trouver la valeur recherchée. On redivise en 2 la partie de tableau considérée et on regarde dans quelle partie du tableau est susceptible de se trouver la valeur recherchée *etc.*

On donne le programme suivant :

python

```
def recherche_dichotomique(x, a):
    g, d = 0, len(a)-1
    while g <= d:
        m = (g + d) // 2
        if a[m] == x:
            return m
        if a[m] < x:
            g = m+1
        else:
            d = m-1
    return None
```

Question *Décrire le fonctionnement du programme donné.*

Exercice 3 – Recherche d'un mot dans un texte

L'algorithme suivant permet de rechercher un mot dans un texte.

python

```
def recherche_mot(m, t):
    for i in range(1 + len(t) - len(m)):
        j = 0
        while j < len(m) and m[j] == t[i + j]:
```



```
j += 1
if j == len(m):
    return i
return None
```

Question Décrire le fonctionnement du programme donné.

Exercice 4

Question Concevoir un algorithme vérifiant qu'une suite est croissante jusqu'à un certain rang.