

Chapitre XII - Algorithmique

I - Recherche dans un tableau

Exemple :

Appliquer l'algorithme à la liste [2, 3, 8, 10, 15, 19, 21, 23, 25, 29, 32, 35] et à la valeur $x=22$.

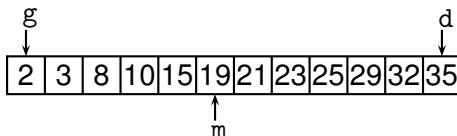
Donner les étapes successives et les valeurs prises par les différentes variables (comme dans l'exemple). Faire les schémas comme ceux de droite permet de ne pas oublier certains tests.

Que renvoie la fonction rechercheDicho dans ce cas ?

Recherche dichotomique

```
def rechercheDicho(liste, x):  
    g=0  
    d=len(liste)-1  
    while g<=d:  
        m=(g+d)//2  
        if x==liste[m]:  
            return m  
        elif x<liste[m]:  
            d=m-1  
        else :  
            g=m+1
```

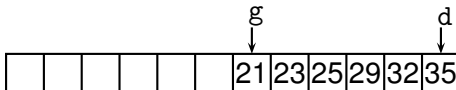
Étape 1 : $g=0$, $d=11$, ainsi $g \leq d$ vaut True et on rentre dans la boucle.
 $m=5$ comme ci-dessous :



Recherche dichotomique

```
def rechercheDicho(liste, x):  
    g=0  
    d=len(liste)-1  
    while g<=d:  
        m=(g+d)//2  
        if x==liste[m]:  
            return m  
        elif x<liste[m]:  
            d=m-1  
        else :  
            g=m+1
```

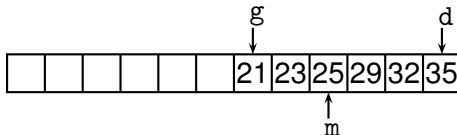
Comme $x > \text{liste}[m]$, g devient $m+1$ (soit 6) et la recherche se limite à :



Recherche dichotomique

```
def rechercheDicho(liste, x):  
    g=0  
    d=len(liste)-1  
    while g<=d:  
        m=(g+d)//2  
        if x==liste[m]:  
            return m  
        elif x<liste[m]:  
            d=m-1  
        else :  
            g=m+1
```

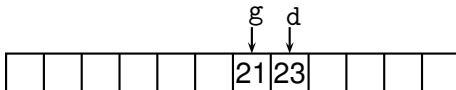
Étape 2 : $g=6$, $d=11$, ainsi $g \leq d$ vaut True
 $m=8$ comme ci-dessous :



Recherche dichotomique

```
def rechercheDicho(liste, x):  
    g=0  
    d=len(liste)-1  
    while g<=d:  
        m=(g+d)//2  
        if x==liste[m]:  
            return m  
        elif x<liste[m]:  
            d=m-1  
        else :  
            g=m+1
```

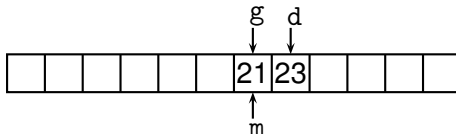
Comme $x < \text{liste}[m]$, d devient $m-1$ (soit 7) :



Recherche dichotomique

```
def rechercheDicho(liste, x):  
    g=0  
    d=len(liste)-1  
    while g<=d:  
        m=(g+d)//2  
        if x==liste[m]:  
            return m  
        elif x<liste[m]:  
            d=m-1  
        else :  
            g=m+1
```

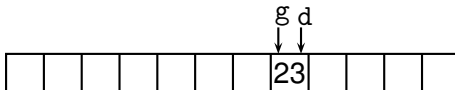
Étape 3 : $g=6$, $d=7$, ainsi $g \leq d$ vaut True
 $m=6$ comme ci-dessous :



Recherche dichotomique

```
def rechercheDicho(liste, x):  
    g=0  
    d=len(liste)-1  
    while g<=d:  
        m=(g+d)//2  
        if x==liste[m]:  
            return m  
        elif x<liste[m]:  
            d=m-1  
        else :  
            g=m+1
```

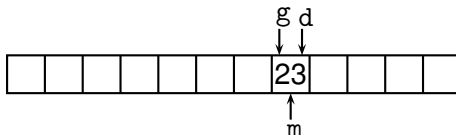
Comme $x > \text{liste}[m]$, g devient $m+1$ (soit 7) :



Recherche dichotomique

```
def rechercheDicho(liste, x):  
    g=0  
    d=len(liste)-1  
    while g<=d:  
        m=(g+d)//2  
        if x==liste[m]:  
            return m  
        elif x<liste[m]:  
            d=m-1  
        else :  
            g=m+1
```

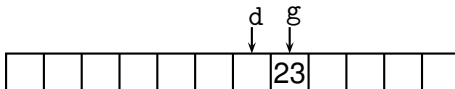
Étape 4 : $g=7$, $d=7$, ainsi $g \leq d$ vaut True
 $m=7$ comme ci-contre :



Recherche dichotomique

```
def rechercheDicho(liste, x):  
    g=0  
    d=len(liste)-1  
    while g<=d:  
        m=(g+d)//2  
        if x==liste[m]:  
            return m  
        elif x<liste[m]:  
            d=m-1  
        else :  
            g=m+1
```

Comme $x < \text{liste}[m]$, d devient $m-1$ (soit 6) :



Or à cet instant, la condition $g \leq d$ renvoie False, donc on sort de la boucle. La fonction ne renvoie donc rien (None) en python ou -1 en javascript.