

Quelques algorithmes importants sur les listes

Recherche de minimum et maximum :

Recherche d'un maximum ou minimum dans une liste :

On va créer un script qui permette d'afficher le maximum d'une liste donnée de 6 valeurs

Exercice 1. On considère la liste des 6 valeurs [2, 10, 11, 4, 7, 12] et on souhaite afficher sa plus grande valeur, sans utiliser la fonction *max*. Pour cela compléter le script, et tester les tentatives à la main ou avec l'interface :

```
List=[2,10,11,4,7,12]
max=List[0]
for k in range(...):
    if List[k]>max:
        max=...
print(max)
```

On souhaite dans la suite créer un script permettant d'afficher la valeur maximale d'une liste d'une longueur quelconque.

Exercice 2. 1. Compléter le script de la fonction suivante pour qu'il permette à l'utilisateur de créer une liste de nombres entiers dont il choisira la longueur et le contenu terme à terme.

```
def creation_liste(n):
    List=[]
    for k in range(...):
        List=...+[int(input('entre l\'entier numéro suivant'))]
    return List
```

2. Compléter le script suivant pour qu'il retourne la valeur maximale d'une liste

```
def MaximumListe(List):
    Max=List[0]
    for k in range(...):
        if List[k]>Max:
            Max=
    return Max
```

3. Créer maintenant un script qui permette de créer une liste de longueur quelconque et d'en trouver le maximum.

4. Modifier le script précédent pour qu'il donne le minimum.

Recherche du deuxième maximum :

Une liste d'entiers étant donnée, on souhaite trouver la valeur maximale, puis la valeur maximale suivante, c'est à dire une fois enlevée la valeur maximale de la liste, on essaie de trouver la valeur maximale de la liste obtenue.

Exercice 3. 1. Utiliser la fonction **max** afin de créer une fonction prend en entrée, une liste et qui en renvoie la deuxième valeur la plus grande de la liste. Pour cela on créera une nouvelle liste privée de la valeur maximale trouvée avec la fonction **max**

2. L'algorithme suivant, à compléter, permet de parcourir une seule fois la liste et d'en donner les deux maxima.

```
def deux_maxima(List):
    save1=List[0]
    save2=List[1]
    if save1>save2:
        save1,save2=save2,save1
```

```

for i in range(2,len(List)):
    if save2<=List[i]:
        save1,save2=...
    elif save1<List[i]<save2:
        save1=...
return(save1,save2)

```

Recherche de la distance minimale entre deux valeurs :

On se donne une liste de nombre entiers et on souhaite déterminer la distance entre les deux nombres les plus proches.

Exercice 4. On considère la liste suivante $L = [2, 3, 5, 11, 7, 5]$.

1. Compléter le script suivant pour qu'il permette de trouver la distance entre un nombre de la liste et le nombre 4

```

L=[2,3,6,11,7,5]
distance=4-L[0]
for k in range(len(L)):
    if (abs(4-L[k])<distance):
        distance=...
print(distance)

```

2. Compléter le script précédent pour qu'il détermine la distance minimale entre deux nombres de la liste. On intégrera pour cela dans le script une deuxième boucle **for**.

```

L=[2,3,6,11,7,5]
distance=5-2
for k in range(len(L)):
    for i in range(len(L)):
        ...
        ...
print(distance)

```

3. Il est aussi possible d'utiliser la fonction *min* de la librairie **numpy**. Pour cela il suffit de calculer toutes les distances séparant les différents nombres d'une liste et d'en choisir le minimum. Proposer un tel script
4. Créer une fonction qui permette l'entrée d'une liste quelconque de nombres entiers et l'affichage de la distance minimale entre deux nombres de cette liste sans la fonction *min*.

Parcours de listes :

Boucle For avec les listes

Le boucle **for** peut avoir une syntaxe plus compacte avec les listes sous la forme $L=[x \text{ for } x \text{ in } \dots]$

Exercice 5. On considère la fonction suivante qui prend en entrée une liste d'entiers et retourne la liste privée de tout entier égal à un certain entier.

1. Quel entier permet de retirer d'une liste la fonction suivante ?

```

def suppression(List1)
    List2=[x for x in List1 if x!=2]
    return List2

```

2. Modifier le script pour qu'il ne garde que les entiers pairs

Recherche dichotomique dans une liste triée.

On considère une liste de nombres triée et on souhaite savoir si cette liste contient un certain élément.

Exercice 6. Compléter le script afin qu'en sortie, est affiché True ou False selon que la valeur cherchée est dans la liste ou pas.

```
def dichotomie(List, nombre):
    a = 0
    b = len(List) - 1
    while a <= b:
        m = (a + b) // 2
        if List[m] == nombre:
            return ...
        elif List[m] < nombre:
            a = m + 1
        else:
            b = m - 1
    return ...
```

Comptage dans une liste :

On considère une liste et on souhaite la parcourir pour compter les données qui vérifient certaines conditions.

Exercice 7. On considère une liste de personnes composée de couples (nom, age) . On souhaite compter le nombre de personnes dont l'âge est strictement inférieur à 18 ans.

1. Compléter le script afin qu'il permette un tel comptage.

```
def nombre_mineurs(liste_personnes):
    compteur = 0
    for (nom, age) in liste_personnes:
        if ...:
            compteur =
    return compteur
```

2. Modifier le script précédent pour qu'il retourne le nombre de personnes dont l'âge est inférieur égal à 18 et le nombre supérieur à 18 ans mais inférieur ou égal à 60 ans puis enfin le nombre de personnes d'un âge strictement supérieur à 60 ans.