

# Recherche d'extremum

## ① MÉMO

### Recherche du maximum

**Principe** Parcourir toute la liste en conservant le plus grand élément rencontré jusqu'ici.

**Initialisation** Toujours avec  $L[0]$ , jamais avec  $0$  (faux pour une liste de négatifs).

**Complexité**  $O(n)$  : un seul parcours, une comparaison par élément.

### Recherche du minimum

La même idée, en remplaçant la comparaison  $>$  par  $<$ .

On peut aussi chercher l'**indice** de l'extremum en retenant  $i\_max$  (ou  $i\_min$ ) au lieu de la valeur elle-même.

### Pièges fréquents

- Initialiser avec  $maxi = 0$  au lieu de  $maxi = L[0]$  : échoue si la liste ne contient que des négatifs.
- Écrire  $maxi = L[i]$  au lieu de mettre à jour conditionnellement.
- Oublier que la fonction doit aussi fonctionner sur une liste à un seul élément.

## ② EXEMPLES

### Programme 1 · Recherche du maximum

```
1 def maximum(L):
2     maxi = L[0]
3     for x in L:
4         if x > maxi:
5             maxi = x
6     return maxi
7
8 print(f"max = {maximum([3, -1, 7, 2])}")
9 # max = 7
```

### Programme 2 · Maximum avec trace

```
1 def maximum_trace(L):
2     maxi = L[0]
3     print(f"init : maxi = {maxi}")
4     for x in L[1:]:
5         if x > maxi:
6             print(f" {x} > {maxi} -> maxi = {x}")
7             maxi = x
8         else:
9             print(f" {x} <= {maxi} -> pas de changement")
10    return maxi
```

```
11 maximum_trace([12.5, 14.0, 11.8, 15.3, 16.1, 14.5])
12
```

### Programme 3 · Indice du maximum

```
1 def indice_maximum(L):
2     i_max = 0
3     for i in range(1, len(L)):
4         if L[i] > L[i_max]:
5             i_max = i
6     return i_max
7
8 jours = ["lun", "mar", "mer", "jeu", "ven", "sam", "dim"]
9 T = [12.5, 14.0, 11.8, 15.3, 13.7, 16.1, 14.5]
10 i = indice_maximum(T)
11 print(f"Jour le plus chaud : {jours[i]} ({T[i]} °C)")
```

## ③ EXERCICES

### Exercice 1 *Minimum et indice* (2 points)

1. Écrire une fonction `minimum(L)` qui renvoie le plus petit élément.
2. Écrire une fonction `indice_minimum(L)` qui renvoie l'indice du plus petit élément (le premier s'il y a plusieurs minimums).

**Exercice 2** *Les deux extremums en un parcours* (3 points) Écrire une fonction `extremums(L)` qui renvoie le tuple (minimum, maximum) en effectuant **un seul parcours** de la liste (une seule boucle `for`).

**Exercice 3** *Amplitude thermique* (3 points) On donne une liste de relevés de température sur une journée (un relevé par heure). Écrire une fonction `amplitude(T)` qui renvoie l'amplitude thermique (écart entre le maximum et le minimum) en un seul parcours.

Tester avec `T = [14.2, 13.8, 13.1, 12.7, 12.5, 13.0, 14.8, 17.3, 19.1, 20.5, 21.2, 21.8, 22.1, 21.6, 20.9, 19.5, 18.0, 16.8, 15.5, 14.9, 14.3, 14.0, 13.7, 13.4]`.

**Exercice 4** *Les deux plus grands éléments* (4 points) Écrire une fonction `deux_plus_grands(L)` qui renvoie les deux plus grands éléments d'une liste de taille  $\geq 2$  en un seul parcours (sans trier la liste). Renvoyer un tuple (max1, max2) avec `max1 >= max2`.

## SOLUTIONS DES EXERCICES

### Corrigé de l'exercice 1.

```
1 def minimum(L):
2     mini = L[0]
3     for x in L:
4         if x < mini:
5             mini = x
6     return mini
7
8 def indice_minimum(L):
9     i_min = 0
10    for i in range(1, len(L)):
11        if L[i] < L[i_min]:
12            i_min = i
13    return i_min
```

**Vérification :** `minimum([-3, -7, -1, -5])` renvoie -7; `indice_minimum([3, -1, 7, 2])` renvoie 1.

### Corrigé de l'exercice 2.

```
1 def extremums(L):
2     mini = L[0]
3     maxi = L[0]
4     for x in L:
5         if x < mini:
6             mini = x
7         if x > maxi:
8             maxi = x
9     return mini, maxi
```

**Vérification :** `extremums([3, 7, 2, 9, 1])` renvoie (1, 9).

**Remarque :** on utilise deux `if` indépendants et non `elif`, car un même élément pourrait initialiser les deux variables (cas de la liste à un élément).

### Corrigé de l'exercice 3.

```
1 def amplitude(T):
2     mini = T[0]
3     maxi = T[0]
4     for t in T:
5         if t < mini:
6             mini = t
7         if t > maxi:
8             maxi = t
9     return maxi - mini
```

**Résultat :** `amplitude(T) = 22.1 - 12.5 = 9.6 °C`.

C'est une application directe de `extremums` suivie d'une soustraction.

### Corrigé de l'exercice 4.

```
1 def deux_plus_grands(L):
2     if L[0] >= L[1]:
3         max1, max2 = L[0], L[1]
4     else:
5         max1, max2 = L[1], L[0]
6     for i in range(2, len(L)):
7         if L[i] >= max1:
8             max2 = max1
9             max1 = L[i]
10        elif L[i] > max2:
11            max2 = L[i]
12    return max1, max2
```

**Principe :** on maintient deux variables. Quand un élément dépasse `max1`, l'ancien `max1` descend dans `max2`. Quand il est entre les deux, il remplace seulement `max2`.

**Vérification :** `deux_plus_grands([3, 7, 2, 9, 1])` renvoie (9, 7); `deux_plus_grands([5, 5])` renvoie (5, 5).