

II.6 Samedi 11 avril ❤️

1- Écrire la fonction `distance(X:list,Y:list)->float` qui renvoie la distance entre les deux listes X et Y.

Solution:

```
1 def distance(X,Y):
2     d=0
3     for i in range(len(X)):
4         d = d + (X[i]-Y[i])**2
5     return sqrt(d)
```

2- Écrire la fonction `Distances(X:list,Z:list[list])->list[float]` qui renvoie la liste des distances entre X et chaque liste de la liste Z en utilisant la fonction précédente.

Solution: Avec la fonction précédente.

```
1 def Distances(X,Z):
2     LD=[] # liste des distances
3     for i in range(len(Z)):
4         dist_i=distance(X,Z[i]) # distance entre la liste Z[i] et X
5         LD.append(dist_i)
6     return LD
```

3- Évaluer la complexité de cette dernière fonction en fonction du nombre d'éléments de Z et de la longueur de X.

Solution: Il y a une boucle de `len(Z)` itérations. Au sein de chaque itération, on appelle la fonction `distance` qui calcule la distance entre X et `Z[i]`. Au sein de `distance` il y a une boucle `for` de `len(X)` itérations, chaque itération est de coût constant.

Ainsi, `Distances` est de complexité $\mathcal{O}(\text{en}(Z) \times \text{len}(X))$