

Les k plus proches voisins – un exemple très simple

On positionne des points sur une droite (l'axe des x). On prend un point particulier et on cherche ses plus proches voisins.

Les points sont représentés par une liste d'abscisse : droite = [-10,-8,-7,5,10,13,-5,12,20]

Le point cible : 5

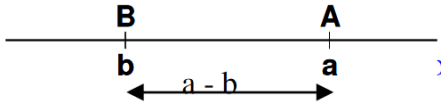
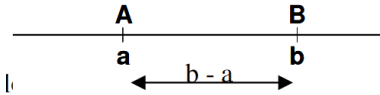
1. Notion de distance et calculs

En [mathématiques](#), la **valeur absolue** (parfois appelée **module**, c'est-à-dire [mesure](#)) d'un [nombre réel](#) est sa valeur numérique sans tenir compte de son [signe](#).

La valeur absolue possède les propriétés suivantes, pour tout réel a

$$|a| \geq 0 ; |a| = |-a| ; |a| = 0 \leftrightarrow a = 0$$

Définition : Soit une droite graduée et les points A et B les points de cette droite aux abscisses respectives a et b . On définit la distance entre les points A et B par :

Cas 1 : $a \geq b$ alors la distance est $a-b$ (≥ 0)	Cas 2 : $a \leq b$ alors la distance est $b-a$ (≥ 0)
	

La distance les séparant A et B est alors de manière générale dans les deux cas : $distance = |a - b|$

La fonction `math.fabs(x)` de la bibliothèque `math` retourne la valeur absolue de x .

On suppose pour la suite que toutes les bibliothèques utiles ont été importées

Question 1

On s'intéresse à un point « cible » dont l'abscisse est aussi désignée par la valeur « cible ». On désire mesurer la distance entre ce point et un autre point quelconque appelé « point » d'abscisse « point ». Donner le code de la fonction « calcul_distance » qui renvoie une liste double de sous liste : [[point_0, distance(point_0/cible)],

1

Les k plus proches voisins – un exemple très simple

2. Classement

Maintenant que l'on possède une liste adapter le tri par sélection pour trier la liste en fonction des distances croissantes

Question 2 Implémenter le code du tri par sélection pour la liste double

Question 3 Ecrire une fonction tri_distance qui renvoi une liste triée par les distances

3. Exploitation affichage des voisins

On désire maintenant afficher les valeurs voisines

Question 4 Combiner les fonction précédentes pour créer la fonction **k_plus_proche(l,c,nb)** où , est liste de départ, cible la valeur cible et nb le nombre de voisin. Cette fonction affiche :

```
voisin 1 : 6; distance 1.0
voisin 2 : 10; distance 5.0
voisin 3 : 12; distance 7.0
```

Les k plus proches voisins – un exemple très simple

