# I $k$-plus proches voisins en dimension 1

Écrire une fonction `int* voisins1D(float x, float* X, int k, int n)` permettant de trouver efficacement les $k$ plus proches voisins de `x` dans l'ensemble de `n` données `X` trié par ordre croissant, où chaque donnée est un réel (en dimension 1). La fonction renvoie un tableau d'entiers contenant les indices des $k$ plus proches voisins de `x` dans `X`.

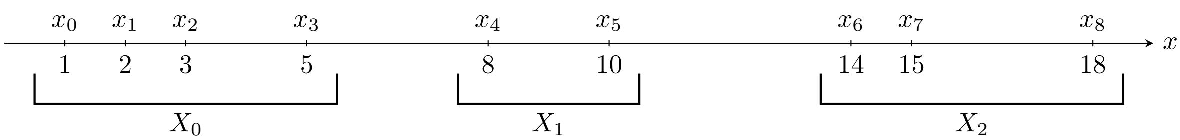
# II Clustering en dimension 1

Soit $K$ un entier et $E$ un ensemble de $N$ réels $x_0 \leq x_1 \leq ... \leq x_{N-1}$. On cherche à déterminer une partition de $[\![0, N-1]\!]$ en $K$ sous-ensembles $\mathcal{P} = \{X_0, X_1, ..., X_{K-1}\}$ non vides tels que l'inertie

$$I(\mathcal{P}) = \sum_{i=0}^{K-1} \sum_{j \in X_i} (x_j - \overline{X_i})^2$$

soit minimum, où $\overline{X_i} = \dfrac{1}{|X_i|} \displaystyle\sum_{j \in X_i} x_j$ est le centre de $X_i$, c'est-à-dire la moyenne des éléments correspondant à la classe $X_i$. Autrement dit, on veut minimiser la somme des carrés des écarts de chaque élément au centre de sa classe.

Par exemple, pour $E = \{1, 2, 3, 5, 8, 10, 14, 15, 18\}$ et $K = 3$, une solution optimale est donnée par la partition suivante, d'inertie environ 19.42 :




## II.1 Préliminaires

1. Comment trouver une solution au problème lorsque $K = N$ ? Lorsque $K = N - 1$ ?
2. Appliquer l'algorithme des $K$-moyennes sur l'ensemble $E$ de l'exemple précédent avec $K = 3$, et en initialisant les centres à $c_0 = 1$, $c_1 = 8$ et $c_2 = 10$. Est-ce que la solution trouvée est optimale ?
3. Appliquer l'algorithme CHA (Clustering Hiérarchique Ascendant) sur l'ensemble $E$ de l'exemple précédent avec $K = 3$, en fusionnant à chaque étape les deux classes dont les centres sont les plus proches.
4. Montrer que l'algorithme de clustering hiérarchique ascendant ne permet pas de résoudre le problème de manière optimale. On pourra prendre $N = 4$ et $K = 2$.

Pour $E = \{x_0, ..., x_{N-1}\}$ et $0 \leq i < j \leq N$, on note $S(i, j)$ la valeur

$$S(i, j) = \sum_{\ell=i}^{j-1} (x_\ell - \mu)^2$$

où $\mu$ est le centre de $\{x_i, x_{i+1}, ..., x_{j-1}\}$.

Pour $n \in [\![0, N]\!]$ et $k \in [\![1, K]\!]$, on note $I(n, k)$ l'inertie minimale possible d'une partition de $\{x_0, ..., x_{n-1}\}$ en $k$ classes non vides.

5. Que vaut $I(n, k)$ si $k = 1$ ?
6. Montrer que pour $n > 0$ et $k > 1$, $I(n, k) = \min\limits_{m=k-1}^{n-1} (I(m, k-1) + S(m, n))$.
7. En déduire une fonction `double inertie(double* E, int N, int K)` qui calcule l'inertie minimale possible d'une partition de $E$ en $K$ classes non vides.
8. Déterminer la complexité temporelle de la fonction précédente.
9. Expliquer comment modifier la fonction précédente pour qu'elle renvoie une partition optimale plutôt que l'inertie minimale.

On cherche à améliorer la complexité temporelle de la solution précédente en effectuant des précalculs. Pour $0 \leq i < j \leq N$, on note $\mu(i, j)$ la moyenne des éléments de $\{x_i, ..., x_{j-1}\}$.

10. Expliquer comment calculer l'ensemble des $S(i, j)$ pour $0 \leq i < j \leq N$ en complexité $\mathrm{O}(N^2)$.