
OPTION INFORMATIQUE

TD n°4 : Nombre d'inversion dans une permutation ou une liste

Une **permutation** $\sigma \in \mathcal{S}_n$ est une application bijective de $\{1, 2, \dots, n\}$ dans $\{1, 2, \dots, n\}$.

On dit qu'une liste est **associée à une permutation** σ si, après tri, cette liste devient $[1; 2; \dots; n]$, le i^e élément de cette liste étant initialement $\sigma(i)$.

Étant donné une permutation σ , on appelle **inversion** un couple (i, j) avec $i < j$ et $\sigma(i) > \sigma(j)$.

Exercice 1. ALGORITHME NAÏF.

1. Décrire un algorithme naïf permettant de calculer le nombre d'inversions dans une liste associée à une permutation.
2. Calculer sa complexité.
3. L'implémenter.

Exercice 2. ALGORITHME FUSION.

1. Adapter l'algorithme de tri par fusion pour calculer le nombre d'inversions dans une liste associée à une permutation de façon plus efficace.
NB : pour l'étape de scission de la liste, il faut couper à la moitié de celle-ci.
2. Quelle est la complexité du tri par fusion ? De l'algorithme obtenu en l'adaptant ?
3. Implémenter l'algorithme obtenu. On pourra recycler son code sur le tri par fusion.