

# Algorithmes de tri et de recherche dans une liste

## 1 Tri par sélection

C'est le plus simple des algorithmes de tri : on cherche le minimum de la liste, qu'on échange avec le premier, puis le second plus petit, qu'on échange avec le deuxième, *etc.*

Programmer en Python une fonction `triSelection` qui trie une liste avec cette méthode.

Pour trier une liste de taille  $n$ , compter combien cet algorithme fait de comparaisons.

## 2 Tri par insertion

C'est le tri qu'on utilise pour trier un jeu de carte : si on a déjà trié une partie de la liste, on regarde l'élément suivant, et on l'insère si besoin dans la partie déjà triée à sa place.

Programmer en Python une fonction `triInsertion` qui trie une liste avec cette méthode.

Pour trier une liste de taille  $n$ , compter combien cet algorithme fait de comparaison dans le pire et dans le meilleur cas.

## 3 Tri à bulles

Dans cet algorithme, on fait remonter les gros éléments de la liste comme des bulles dans une flûte de champagne. En commençant au début de la liste, on échange les paires d'éléments consécutifs qui sont dans le mauvais ordre, puis on recommence. Si on parcourt la liste entière sans faire aucun échange, c'est alors qu'elle est triée.

Programmer en Python une fonction `triBulles` qui trie une liste avec cette méthode.

## 4 Recherche dans une liste par dichotomie

Pour trouver un élément dans une liste, il faut dans le pire case regarder tous ses éléments. En revanche, si la liste est déjà triée, on peut regarder l'élément au milieu de la liste, et selon sa valeur, chercher dans le début ou la fin de la liste. Il suffit de réitérer jusqu'à trouver l'élément cherché.

## 5 Médiane d'une liste

La médiane d'une liste est une valeur telle qu'au moins la moitié des valeurs de la liste est inférieure, et au moins la moitié des valeurs de la liste est supérieure.

Écrire une fonction Python qui calcule la médiane d'une liste.