Activité : algorithmes de tris par sélection et par insertion

Nous avons vu qu'un algorithme de recherche d'un élément avait au pire cas un coût linéaire dans un tableau non trié, mais un coût nettement inférieur dans un tableau déjà trié (par exemple avec un algorithme de recherche dichotomique).

Un exemple pratique est la recherche de la définition d'un mot dans un dictionnaire linguistique ou d'un article dans une encyclopédie : la recherche est rapide car les éléments sont triés par ordre alphabétique. Elle serait extrêmement longue si les éléments n'étaient pas triés !

Les algorithmes permettant de trier un tableau (ou une liste) de données sont nombreux. Cette année, nous allons étudier plus spécifiquement deux d'entre eux : le tri par insertion et le tri par sélection.

1 Tri par sélection

1.1 Présentation de l'algorithme

Avant même de coder un algorithme dans un langage de programmation, il faut être capable de le décrire avec des mots.

Une vidéo présente le principe de l'algorithme de tri par sélection. Les éléments sont représentés par des danseurs, qui vont, en dansant, simuler les comparaisons, les affectations, etc. que demandent cet algorithme. Naturellement, la technique de danse ou l'accompagnement musical n'ont aucune importance !

Visionner cette vidéo, pour comprendre le fonctionnement de l'algorithme : lien vers la vidéo.

1.2 Description de l'algorithme

1. Dans la vidéo, quel nom, très simple, est donné au tableau (ou à la liste) ? Regarder au dessus des danseurs.
2. Décrire cet algorithme avec des phrases, en détaillant la suite des instructions des différentes étapes.

1.3 Programmation

3. Programmer en Python une fonction *tri_selection* qui prend en argument une liste non triée et qui trie cette liste.
4. Tester votre fonction en l'appliquant sur les listes suivantes :
   - liste1 $= [4, 2, 8, 10, 6]$
   - liste1 $= [10, 8, 6, 4, 2]$
   - liste3 $= []$
   - liste4 $= [5, 5, 4, 3, 3]$

2 Tri par insertion

Refaire la même étude (et donc répondre aux mêmes questions) pour le tri par insertion.

Visionner cette vidéo, pour comprendre le fonctionnement de l'algorithme : lien vers la vidéo.