



ALGORITHMES DE TRIS

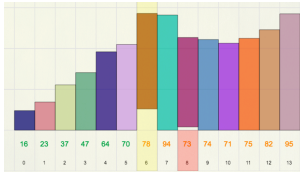
[Marion SZPIEG – Frédéric PEURIERE]

Etablir, implémenter et analyser deux algorithmes de tris de tableaux.

- *Le tri par sélection*
- *Le tri par insertion*

1. LE TRI PAR SELECTION (DU MINIMUM):

DESCRIPTION:



Observez attentivement dans l'animations *algorithmes illustrés* le principe de fonctionnement du tri par sélection. Proposez une description simple de l'exécution de cet algorithme.

CODE PYTHON:

Ecrivez la fonction `tri_Selection(Liste)` qui implémente cet algorithme en PYTHON.

ANALYSE DE COMPLEXITE:

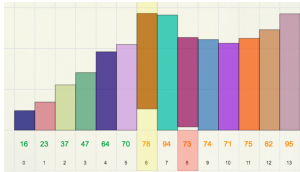
Déterminons d'abord le coût de cet algorithme, on se limite au **nombre de comparaisons effectuées**. Quelle est la *complexité* de l'algorithme de tri par sélection?

RECHERCHE D'UN INVARIANT DE BOUCLE:

Proposez un invariant de boucle pour le *tri par sélection*:

2. LE TRI PAR INSERTION:

DESCRIPTION:



Observez attentivement dans l'animations *algorithmes illustrés* le principe de fonctionnement du tri par insertion. Proposez une description simple de l'exécution de cet algorithme.

CODE PYTHON:

Ecrivez la fonction `tri_Insertion(Liste)` qui implémente cet algorithme en PYTHON.

ANALYSE DE COMPLEXITE:

Déterminez le coût de cet algorithme, dans le **meilleur** puis dans le **pire** des cas. On se limite au **nombre d'échanges effectués**. Quelle est la *complexité* de l'algorithme de tri par insertion?

RECHERCHE D'UN INVARIANT DE BOUCLE:

Proposez un invariant de boucle pour le *tri par insertion*:
