

Lectures et exercices

A faire :

Relire le cours (livre)- Faire les exercices du livre (1,5/2h)

Pratique (4h) :

- Programmer un tri par insertion (sans copier sur internet)
- Faire des essais avec des jeux de données de taille et de contenus différents et mesurer le temps réel d'exécution sur votre machine.
- Présenter les résultats (graphiques c'est mieux avec analyse et commentaires)

Rendu (pris en compte dans le CC) :

- Code source+résultats+commentaires/explications ;
- Dans un seul fichier zip, nom_prénom dans le nom du fichier ;
- Sur chamilo, cours CS353, dossier travaux/Tri_Insertion
- Avant le 27/02 minuit
- *Le respect de ces consignes est évalué*
- *La copie, le plagiat sont sanctionnés*

Exercises

2.2-1

Express the function $n^3/1000 - 100n^2 - 100n + 3$ in terms of Θ -notation.

2.2-2

Consider sorting n numbers stored in array A by first finding the smallest element of A and exchanging it with the element in $A[1]$. Then find the second smallest element of A , and exchange it with $A[2]$. Continue in this manner for the first $n - 1$ elements of A . Write pseudocode for this algorithm, which is known as **selection sort**. What loop invariant does this algorithm maintain? Why does it need to run for only the first $n - 1$ elements, rather than for all n elements? Give the best-case and worst-case running times of selection sort in Θ -notation.

2.2-3

Consider linear search again (see Exercise 2.1-3). How many elements of the input sequence need to be checked on the average, assuming that the element being searched for is equally likely to be any element in the array? How about in the worst case? What are the average-case and worst-case running times of linear search in Θ -notation? Justify your answers.

2.2-4

How can we modify almost any algorithm to have a good best-case running time?