

Tri par Insertion

Commentaire du main :

En premier on crée un vecteur x qui contiendra les différentes tailles d'ensemble testés.

Ensuite il y a une première boucle qui va itérer chacune des tailles d'ensemble prévus.

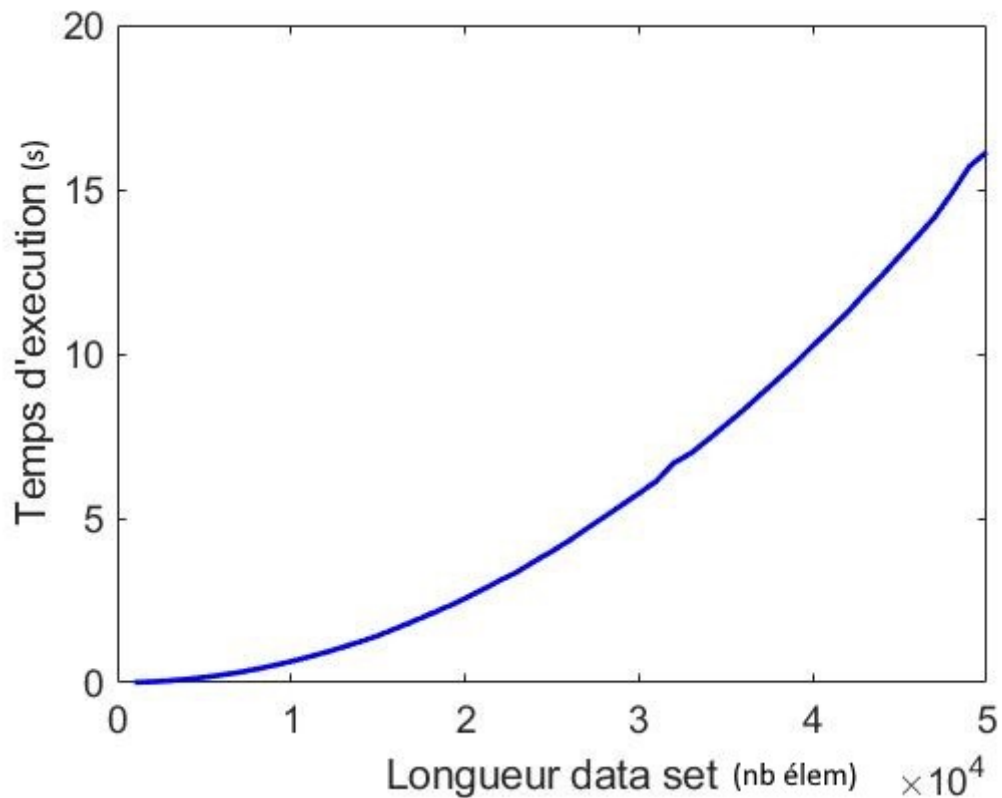
Puis une deuxième boucle qui va faire un certain nombre de fois la mesure du temps.

Notes : randi([10 50],1,N) crée une matrice de 1 par N de nombre entiers aléatoires entre 10 et 50.

tic démarre un timer.

toc renvoie la valeur du timer.

Finalement on affiche nos résultats sur un graphe avec plot.



Après une série de test variant le nombre d'éléments du set de données de 1000 à 100 000 par incrément de 1000, on constate que la forme de la courbe à l'allure de la fonction carré ($x \rightarrow x^2$).

Code source :

```
main.m x
1
2 NB_TEST = 10;
3 WIDTH = 50;
4 INCR = 1000;
5 N = 0;
6
7 x = 1:WIDTH;
8 x = x.*INCR;
9 y = [];
10
11 for k = 1:WIDTH
12
13     N = N + INCR;
14
15     time = 0;
16     for i = 1:NB_TEST
17
18         Liste = randi([10 50],1,N);
19
20         tic
21         Tri = tri_insertion(Liste);
22         time = time + toc;
23     end
24
25     y(k) = time/NB_TEST;
26
27 k
28 end
29
30 figure(1)
31 plot(x,y,'-b','linewidth',2) %affichage de x1 en fonction de t. La courbe en bleue continue d'épaisseur 3
32 xlabel("Longueur data set (nb elem)", 'fontsize',14) %label sur l'axe x avec une police de 14
33 ylabel("Temps d'execution (s)", 'fontsize',14) % idem sur l'axe y
34 set(gca,'fontsize',14) % police des axes en 14
35 set(gcf,'color','white') %couleur de la fenêtre en blanc

tri_insertion.m x
1
2 function [Out] = tri_insertion(L)
3
4 n = length(L);
5
6 for j = 2:n
7
8     x = L(j);
9     i = j-1;
10    while(i>0 && L(i) > x)
11
12        L(i+1) = L(i);
13        i=i-1;
14
15    end
16
17    L(i+1)=x;
18
19 end
20
21 Out = L;
22
23 end
```