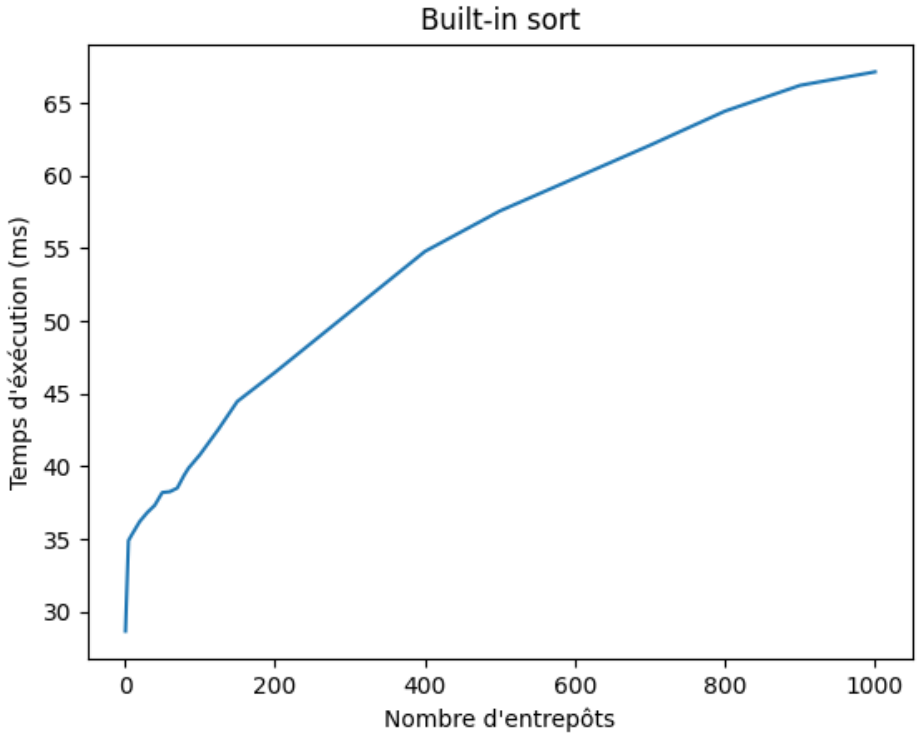
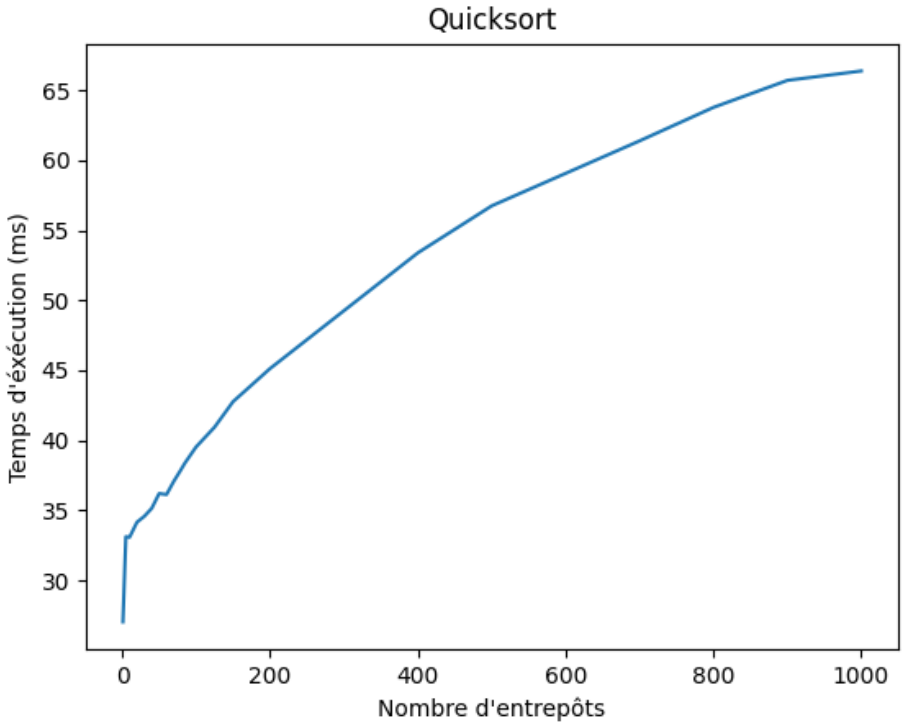
Analyses des différentes méthodes de tri utilisées :

Analyse empirique de Built-in sort :



Son analyse asymptotique ne nous est pas demandée dans le TP

Analyse empirique de quick sort :



Analyse asymptotique de quick sort :

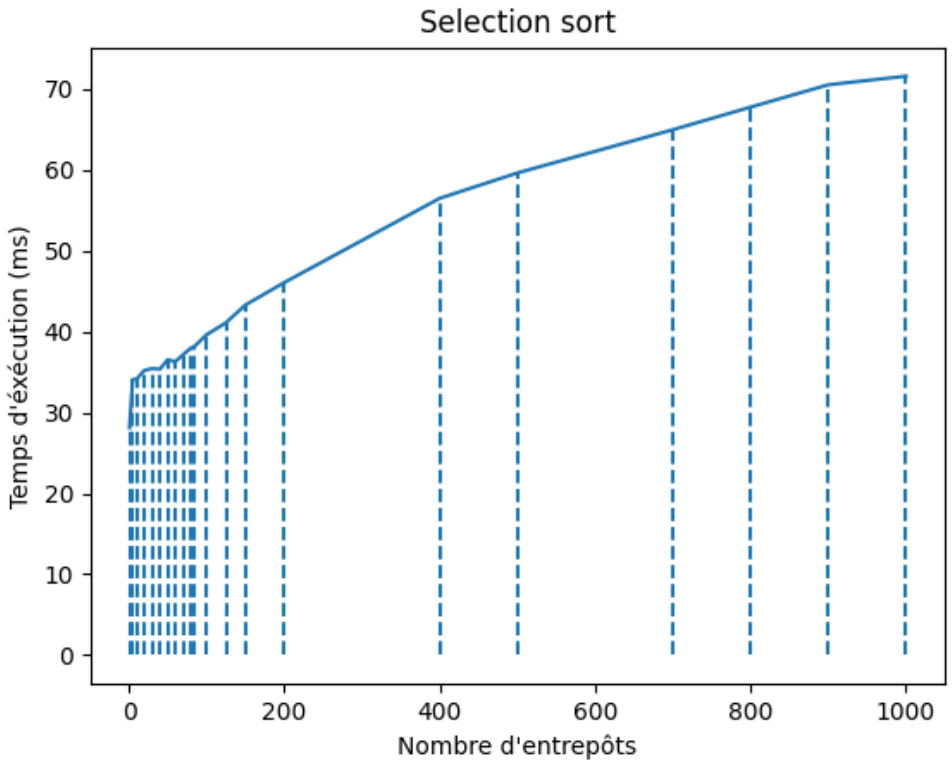
La complexité temporelle de cet algorithme est de O(nlogn) car :
-Fonction run(String [] fileNames) : prend O(n) à cause de la boucle for(Warehouse w : warehouses)
-Fonction getTruckInfoFromFile(String filename) : prend O(1)
-Fonction getWarehoursesFromFile(String filename): prend O(n) à cause du while et for loops
-Fonction getCoordsFromString(String coordsString) : prend O(1)
-Fonction swap: O(1)
-Fonction partition: prend O(n) à cause de la boucle for

-Fonction quickSort : prend O(nlogn) car on utilise partition O(n) puis on appelle recurssivement quicksort sur la première moitié de la liste puis sur la deuxième moitié de la liste O(logn)

-Fonction collectBoxes : prend O(n) à cause du while
-Fonction writeResult : prend O(n) à cause du for loop

D'où la complexité temporelle est de :
--> O(n)+O(1)+O(n)+ O(1)+ O(1)+O(n)+O(nlogn)+ O(n)+ O(n)
--> O(5n+nlogn)
--> O(nlogn) car la fonction nlogn croît plus rapidement que la fonction n

Analyse empirique de selection sort :



Analyse asymptotique de selection sort :

La complexité temporelle de cet algorithme est de O(n^2) car :

-Fonction run(String [] fileNames) : prend O(n) à cause de la boucle for(Warehouse w : warehouses)
-Fonction getTruckInfoFromFile(String filename) : prend O(1)
-Fonction getWarehoursesFromFile(String filename): prend O(n) à cause du while et for loops
-Fonction getCoordsFromString(String coordsString) : prend O(1)

-Fonction selectionSort(ArrayList<Warehouse warehouses): prend O(n^2) car on a une boucle for imbriquée dans une autre boucle for

-Fonction collectBoxes : prend O(n) à cause du while
-Fonction writeResult : prend O(n) à cause du for loop

D'où la complexité temporelle est de :
--> O(n)+O(1)+O(n)+ O(1)+ O(n^2)+ O(n)+ O(n)
--> O(4n+n^2)
--> O(n^2) car la fonction n^2 croît plus rapidement que la fonction n