

# Auto-évaluation du TP1 :

## res\_0

1	Truck position: (45.5532,-73.5968)		
2	Distance:0	Number of boxes:0	Position:(45.5532,-73.5968)
3	Distance:678.5	Number of boxes:0	Position:(45.5593,-73.5966)
4	Distance:839.6	Number of boxes:0	Position:(45.5569,-73.6062)
5	Distance:1321.0	Number of boxes:0	Position:(45.5487,-73.5811)
6	Distance:1321.3	Number of boxes:0	Position:(45.565,-73.5988)
7	Distance:2078.6	Number of boxes:0	Position:(45.5352,-73.5896)
8	Distance:2081.5	Number of boxes:0	Position:(45.5686,-73.612)
9	Distance:2221.0	Number of boxes:42	Position:(45.5412,-73.6196)

## res\_1

1	Truck position: (45.5766,-73.5695)		
2	Distance:0	Number of boxes:0	Position:(45.5766,-73.5695)
3	Distance:688.6	Number of boxes:0	Position:(45.5708,-73.5726)
4	Distance:1074.6	Number of boxes:0	Position:(45.5717,-73.5576)
5	Distance:1630.1	Number of boxes:0	Position:(45.5736,-73.59)
6	Distance:1640.7	Number of boxes:0	Position:(45.5641,-73.5807)
7	Distance:2795.4	Number of boxes:0	Position:(45.5644,-73.5381)
8	Distance:3362.4	Number of boxes:72	Position:(45.5554,-73.6003)

## res\_2

1	Truck position: (45.4665,-73.5444)		
2	Distance:0	Number of boxes:0	Position:(45.4665,-73.5444)
3	Distance:966.8	Number of boxes:0	Position:(45.4581,-73.5476)
4	Distance:1589.1	Number of boxes:0	Position:(45.4807,-73.5421)
5	Distance:1603.7	Number of boxes:0	Position:(45.4534,-73.5358)
6	Distance:2491.8	Number of boxes:0	Position:(45.4773,-73.5724)
7	Distance:2862.8	Number of boxes:0	Position:(45.4544,-73.5768)
8	Distance:2944.5	Number of boxes:0	Position:(45.4437,-73.5252)
9	Distance:3013.6	Number of boxes:0	Position:(45.4678,-73.583)
10	Distance:3192.5	Number of boxes:0	Position:(45.4393,-73.5575)
11	Distance:3498.7	Number of boxes:0	Position:(45.4549,-73.5861)
12	Distance:4013.2	Number of boxes:0	Position:(45.449,-73.4994)
13	Distance:4108.4	Number of boxes:0	Position:(45.5012,-73.5625)
14	Distance:4427.4	Number of boxes:0	Position:(45.4978,-73.5795)
15	Distance:4576.0	Number of boxes:0	Position:(45.4415,-73.591)
16	Distance:4601.3	Number of boxes:0	Position:(45.4936,-73.589)
17	Distance:4844.6	Number of boxes:0	Position:(45.5014,-73.5816)
18	Distance:5017.3	Number of boxes:0	Position:(45.431,-73.5841)
19	Distance:5092.8	Number of boxes:0	Position:(45.5021,-73.5033)
20	Distance:5441.8	Number of boxes:0	Position:(45.5152,-73.5375)
21	Distance:5509.9	Number of boxes:0	Position:(45.4297,-73.5917)
22	Distance:5593.7	Number of boxes:0	Position:(45.5108,-73.5784)
23	Distance:5910.9	Number of boxes:0	Position:(45.447,-73.6149)
24	Distance:5946.5	Number of boxes:0	Position:(45.4939,-73.6099)
25	Distance:6103.9	Number of boxes:0	Position:(45.4971,-73.6094)
26	Distance:6225.2	Number of boxes:0	Position:(45.4466,-73.619)
27	Distance:6360.0	Number of boxes:0	Position:(45.5204,-73.5717)
28	Distance:6625.5	Number of boxes:0	Position:(45.4384,-73.6193)
29	Distance:6983.8	Number of boxes:0	Position:(45.5293,-73.5457)
30	Distance:7073.8	Number of boxes:0	Position:(45.4447,-73.6296)
31	Distance:7338.1	Number of boxes:0	Position:(45.5147,-73.6087)
32	Distance:7428.0	Number of boxes:0	Position:(45.531,-73.5196)
33	Distance:7452.2	Number of boxes:0	Position:(45.4786,-73.6384)
34	Distance:7584.4	Number of boxes:0	Position:(45.4616,-73.6414)
35	Distance:7653.3	Number of boxes:0	Position:(45.5267,-73.592)
36	Distance:7706.6	Number of boxes:0	Position:(45.5308,-73.5813)
37	Distance:7801.3	Number of boxes:0	Position:(45.4324,-73.6318)
38	Distance:7848.1	Number of boxes:0	Position:(45.5215,-73.6075)
39	Distance:8112.9	Number of boxes:0	Position:(45.5124,-73.6253)
40	Distance:8129.7	Number of boxes:0	Position:(45.5017,-73.6358)
41	Distance:8455.5	Number of boxes:0	Position:(45.5417,-73.5283)
42	Distance:8485.0	Number of boxes:0	Position:(45.4892,-73.6483)
43	Distance:8515.6	Number of boxes:0	Position:(45.5416,-73.523)
44	Distance:8732.6	Number of boxes:0	Position:(45.5344,-73.6007)
45	Distance:8784.9	Number of boxes:0	Position:(45.5454,-73.5386)
46	Distance:8833.5	Number of boxes:0	Position:(45.5425,-73.5774)
47	Distance:9108.1	Number of boxes:0	Position:(45.5465,-73.5193)
48	Distance:9156.3	Number of boxes:0	Position:(45.5453,-73.5103)
49	Distance:9289.8	Number of boxes:0	Position:(45.5469,-73.512)
50	Distance:9450.0	Number of boxes:0	Position:(45.5,-73.6558)
51	Distance:9622.5	Number of boxes:0	Position:(45.553,-73.548)
52	Distance:9914.1	Number of boxes:0	Position:(45.509,-73.6562)
53	Distance:10076.8	Number of boxes:0	Position:(45.5552,-73.5179)
54	Distance:10962.7	Number of boxes:0	Position:(45.555,-73.6064)
55	Distance:11151.6	Number of boxes:0	Position:(45.5645,-73.514)
56	Distance:11706.4	Number of boxes:0	Position:(45.57,-73.5169)
57	Distance:11752.5	Number of boxes:0	Position:(45.5561,-73.6244)
58	Distance:11790.2	Number of boxes:0	Position:(45.5441,-73.6475)
59	Distance:11794.5	Number of boxes:0	Position:(45.5719,-73.5614)
60	Distance:11845.2	Number of boxes:0	Position:(45.5484,-73.6416)
61	Distance:11951.9	Number of boxes:0	Position:(45.5706,-73.5826)
62	Distance:12408.1	Number of boxes:0	Position:(45.5749,-73.5066)
63	Distance:13719.2	Number of boxes:0	Position:(45.5617,-73.6564)

### res\_3

1	Truck position: (45.4431,-73.619)		
2	Distance:0	Number of boxes:0	Position:(45.4431,-73.619)
3	Distance:891.3	Number of boxes:14	Position:(45.4407,-73.6299)

### res\_4

1	Truck position: (45.554,-73.5363)		
2	Distance:0	Number of boxes:0	Position:(45.554,-73.5363)
3	Distance:1043.2	Number of boxes:0	Position:(45.548,-73.526)
4	Distance:1375.7	Number of boxes:0	Position:(45.5523,-73.5188)
5	Distance:2069.7	Number of boxes:0	Position:(45.5638,-73.5589)
6	Distance:2550.6	Number of boxes:22	Position:(45.5376,-73.5134)

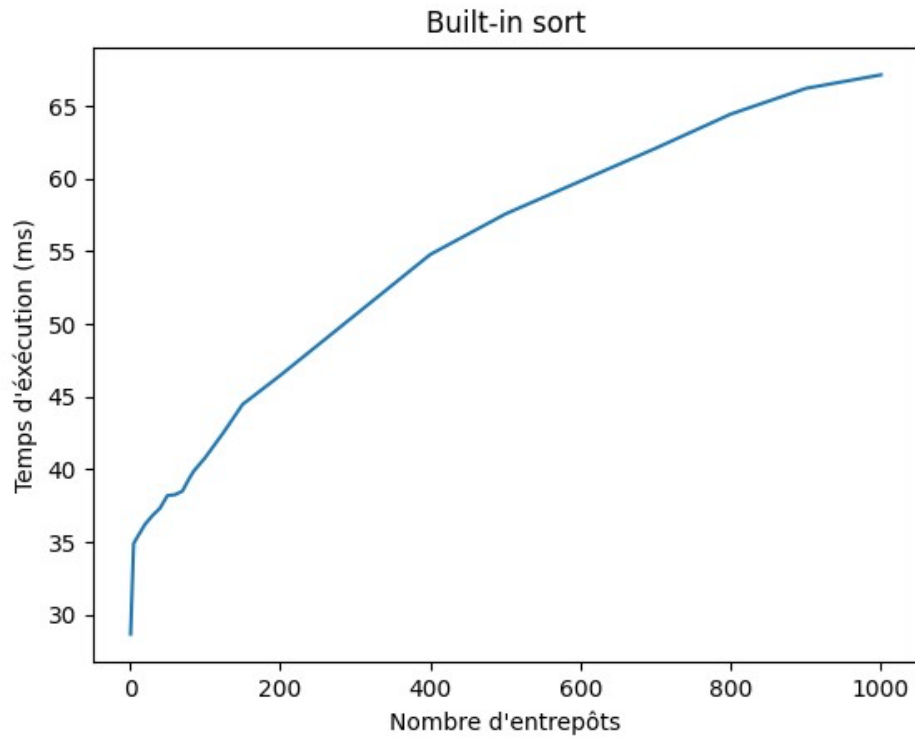
### res\_5

1	Truck position: (45.5449,-73.5891)		
2	Distance:0	Number of boxes:0	Position:(45.5449,-73.5891)
3	Distance:1063.7	Number of boxes:0	Position:(45.5354,-73.5875)
4	Distance:1439.9	Number of boxes:0	Position:(45.5321,-73.5863)
5	Distance:1802.8	Number of boxes:0	Position:(45.5495,-73.6113)
6	Distance:2260.2	Number of boxes:0	Position:(45.5275,-73.6041)
7	Distance:2263.8	Number of boxes:0	Position:(45.56,-73.6086)
8	Distance:2279.9	Number of boxes:0	Position:(45.5317,-73.6115)
9	Distance:2620.1	Number of boxes:0	Position:(45.5572,-73.6178)
10	Distance:2648.8	Number of boxes:0	Position:(45.526,-73.6098)
11	Distance:2802.1	Number of boxes:0	Position:(45.5338,-73.5568)
12	Distance:2935.6	Number of boxes:0	Position:(45.5685,-73.5722)
13	Distance:2977.6	Number of boxes:0	Position:(45.5715,-73.5847)
14	Distance:2983.4	Number of boxes:0	Position:(45.5443,-73.5508)
15	Distance:3066.4	Number of boxes:0	Position:(45.5247,-73.6159)
16	Distance:3092.0	Number of boxes:0	Position:(45.5632,-73.619)
17	Distance:3261.4	Number of boxes:0	Position:(45.5332,-73.6275)
18	Distance:3386.4	Number of boxes:0	Position:(45.5224,-73.5598)
19	Distance:3526.9	Number of boxes:16	Position:(45.5226,-73.6213)

**Le programme fonctionne parfaitement, les screen-shots présentés dans l'auto-évaluation ci-dessus correspondent aux résultats fournis dans l'énoncé du TP1**

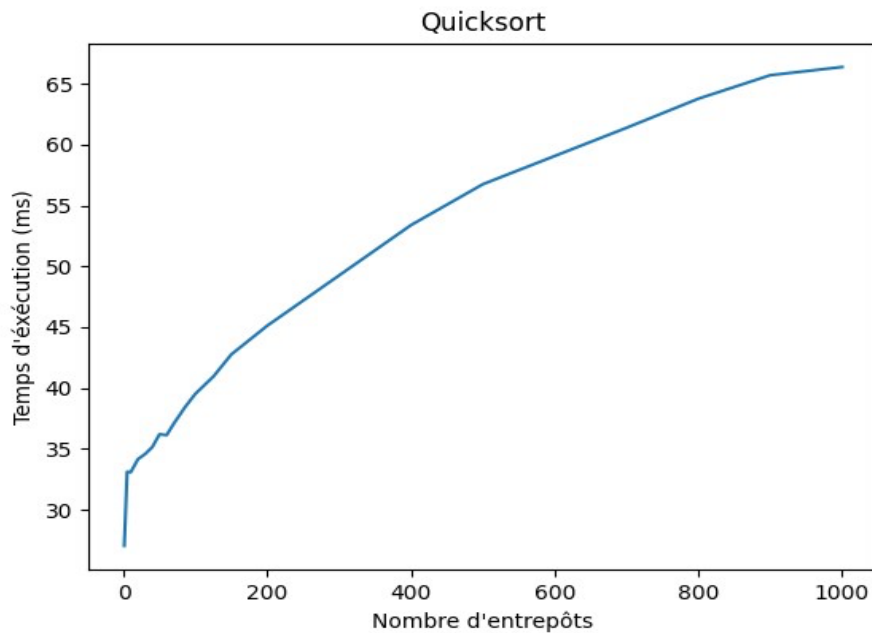
## ***Analyses des différentes méthodes de tri utilisées :***

### **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 (Récap de ce qui a été mis en commentaires dans le code Java):

La complexité temporelle de cet algorithme est de  **$O(n \log n)$**  car :

- Fonction `run(String [] fileNames)` : prend  $O(n)$  à cause de la boucle `for(Warehouse asymptotique : warehouses)`
- Fonction `getTruckInfoFromFile(String filename)` : prend  $O(1)$
- Fonction `getWarehousesFromFile(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  $O(n \log n)$  car on utilise `partition`  $O(n)$  puis on appelle récursivement `quicksort` sur la première moitié de la liste puis sur la deuxième moitié de la liste  $O(\log n)$
  
- Fonction `collectBoxes` : prend  $O(n)$  à cause du `while`
- Fonction `writeResult` : prend  $O(n)$  à cause du `for` loop

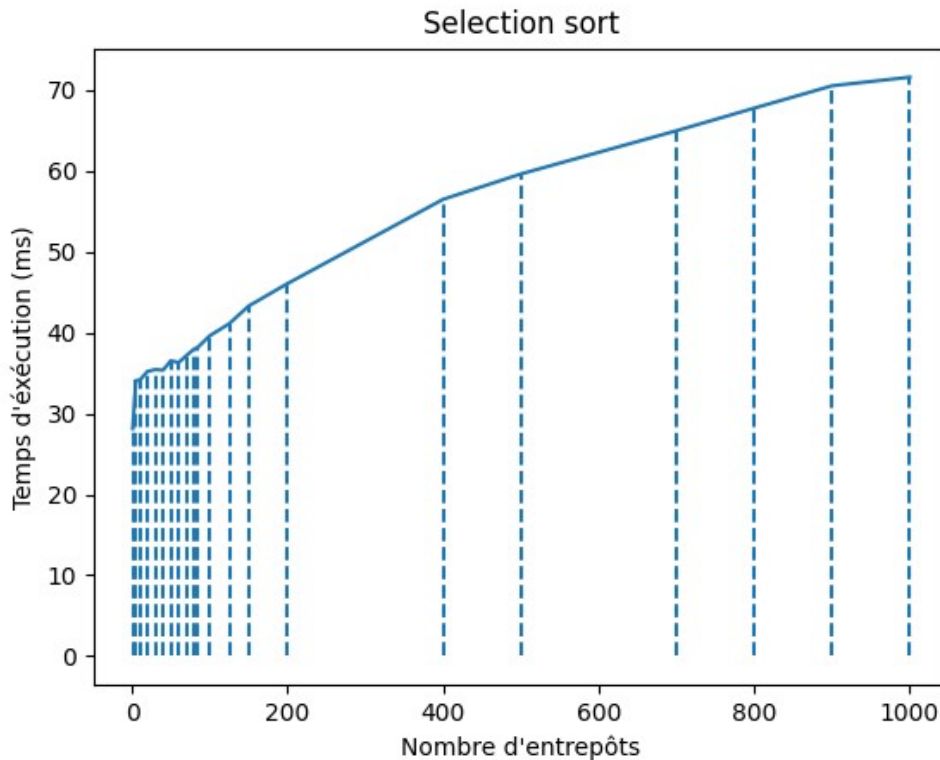
D'où la complexité temporelle est de :

-->  $O(n) + O(1) + O(n) + O(1) + O(1) + O(n) + O(n \log n) + O(n) + O(n)$

-->  $O(5n + n \log n)$

-->  $O(n \log n)$  car la Fonction  $n \log n$  croît plus rapidement que la Fonction  $n$

## Analyse empirique de selection sort :



## Analyse asymptotique de selection sort (Récap de ce qui a été déjà mis en commentaires dans le code Java):

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 asymptotique : warehouses)`
- Fonction `getTruckInfoFromFile(String filename)` : prend  $O(1)$
- Fonction `getWarehousesFromFile(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  $n$