

Groupe : ASD1_B_A

Estimation du temps

Pour estimer le temps de calcul d'une grande quantité de données, une mesure du temps est effectuée pour 100 valeurs. Puis, par calcul, nous avons obtenu une estimation du temps.

| Quantité de données | Select Sort | QuickSort | Counting Sort |
|--|--------------------|----------------------------|---------------|
| Temps en ms pour 100 éléments | 80'870.55 | 25'073.35 | 8'433.70 |
| Complexité | $O(N^2) = 10\ 000$ | $O(N \cdot \log(N)) = 664$ | $O(N) = 100$ |
| Temps unitaire | 8.087 | 34.760 | 84.337 |
| Temps pour 10 000 [en ns] Complexité × temps unitaire | 808 700 000 | 4 618 808 | 843 370 |
| Temps pour 10 000 [en ms] | 808 ms | 4.6 ms | 0.84 ms |

Comparaison des tris en fonction de la quantité des données

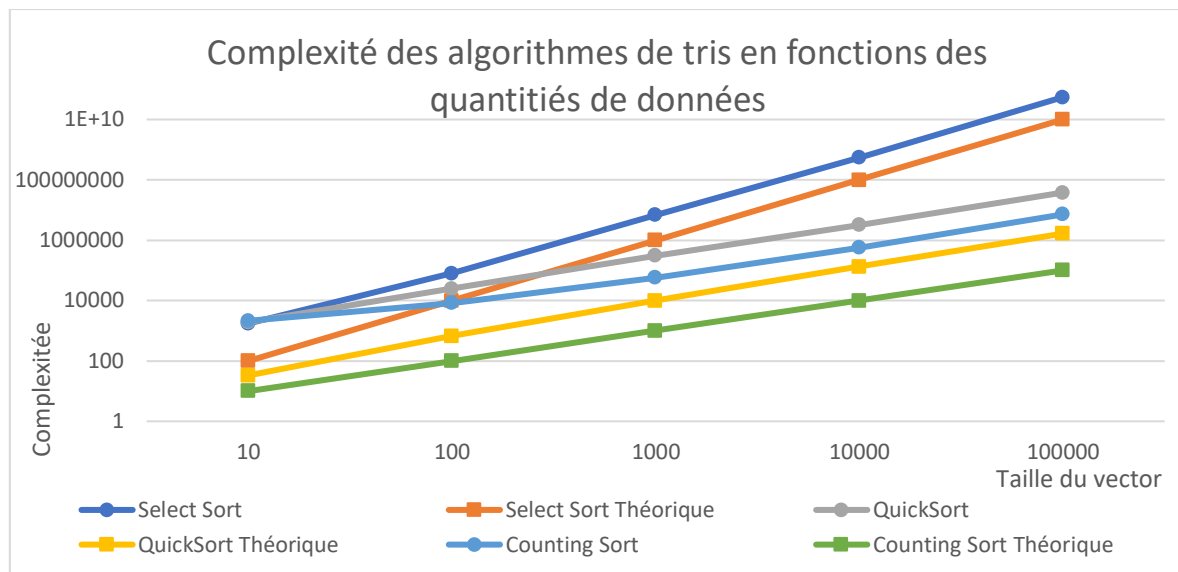
Nous avons comparé la durée moyenne d'exécution de chaque tri avec une même base. Il est possible de voir que la durée d'un tri par sélection est plus rapide sur des tableaux de faible taille. Le Counting Sort, est plus rapide lorsqu'on dépasse une taille de données de plus de 100 éléments.

| Quantité de données | Select Sort | QuickSort | Counting Sort |
|---------------------|-------------------|---------------|---------------|
| 10 | 1'744.25 | 1'930.15 | 2'139.85 |
| 100 | 80'870.55 | 25'073.35 | 8'433.70 |
| 1000 | 6'795'954.35 | 310'400.70 | 58'190.30 |
| 10 000 | 549'318'570.75 | 3'171'283.55 | 565'501.40 |
| 100000 | 54'953'453'459.50 | 37'961'676.25 | 7'165'844.80 |

Tableau 1 - Temps moyen (en nanoseconde) des tris en fonction d'une quantité de données

| Quantité de données | Select Sort | QuickSort | Counting Sort |
|---------------------|----------------|--------------|---------------|
| 10 | 100 | 33.21 | 10 |
| 100 | 10'000 | 664.38 | 100 |
| 1000 | 1'000'000 | 9'965.78 | 1'000 |
| 10000 | 100'000'000 | 132'877.12 | 10'000 |
| 100000 | 10'000'000'000 | 1'660'000.00 | 100'000 |

Tableau 2 - Complexité théorique en fonction d'une quantité de données



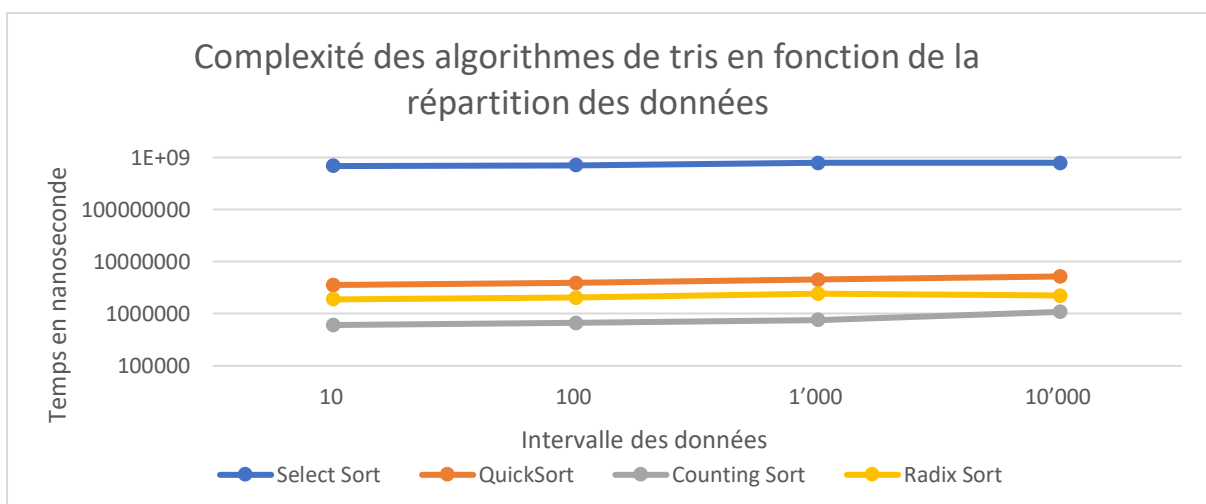
Nous constatons que les lignes des mesures et les lignes des valeurs théoriques sont parallèles deux-à-deux, ce qui confirme la théorie.

Comparaison des tris en fonction des données

Pour les mesures suivantes, nous avons pris un tableau de 10 000 entrées. Le tableau a été rempli de valeur aléatoire entre 0 et 100. L'opération a été répétée pour les intervalles 0-1000 et 0-10'000.

| Intervalle des valeurs | Select Sort | QuickSort | Counting Sort | Radix Sort |
|------------------------|----------------|--------------|---------------|--------------|
| 0 à 10 | 683'470'941.55 | 3'571'885.05 | 608'261.05 | 1'887'417.35 |
| 0 à 100 | 703'739'570.20 | 3'929'509.50 | 661'383.85 | 2'047'123.15 |
| 0 à 1'000 | 790'904'970.75 | 4'555'834.30 | 764'590.65 | 2'431'016.75 |
| 0 à 10'000 | 785'782'223.55 | 5'204'417.75 | 1'087'482.65 | 2'214'717.95 |

Tableau 3 - Temps moyen (en nanoseconde) selon un intervalle des valeurs



Nous constatons que les lignes du Select, QuickSort et Radix sont horizontales, ce qui indique que les algorithmes ne dépendent pas de la répartition des données. Le Counting sort prend un plus de temps pour les grands tableaux, ceci car sa complexité est en $O(n+m)$ avec n la taille du tableau et m le nombre de casier. Hypothèse : pour 100'000 le Radix sort devrait être plus rapide que le Counting sort.