Rapport TP Statistiques

Ambrosino-Ielpo Gwenaël (11212798) Busac Pascal (11422244)

June 19, 2019

1 Rappel du but de l'étude

Dans cette étude nous nous proposons de comparer les performances de deux algorithmes de tri, en se basant sur une liste de tableaux représentatifs de la classe des problèmes de tri de tableau. Il s'agit de comparer les temps d'exécution sur un ensemble de tableaux de taille différentes.

2 Méthode utilisée

2.1 Présentation des hypothèses:

Nous nous basons sur un seuil de signification alpha de 0.05.

➔ H1: L'algorithme tri bulle a un temps d'exécution moyen supérieur ou égal au temps d'exécution du tri par insertion.

➔ H0: les deux algorithmes ont des temps d'exécution moyens équivalent.

2.2 Données:

Nous avons décidé de tester nos deux algorithmes sur 1000 instances représentatives des problèmes de tri de tableaux, pour une taille allant de 1000 à 26000 celles-ci sont notées pi pour i allant de 1 à 1000. (voir protocole ci-dessous)

Protocole expérimental:

➔ Indicateur de performance: temps CPU

➔ Paramètres/factors: tableau de tableaux défini par array-size et array-list-size (la taille du tableau et le nombre de tableau)

➔ Levels: on fait varier array-size qui prendra pour valeur 1000 à 26000 par pas de 25 et array-list-size sera constant avec une taille de 10.

➔ Propriété de l'environnement:

- Proc: Intel(R) Core(TM) i5-3210 CPU @ 2.50GHz
- Compilateur: GCC version 7.3.0
- RAM: 8Go DDR3 1600 MHz
- OS: Lubuntu 18.04

2.3 Remarques:

Remarque 1:

Afin de limiter les biais de mesures, nous avons préalablement vérifié les précautions suivantes:

- Nous avons effectué des tests sur les algorithmes pour vérifier qu'ils ne contenaient pas de bogues.
- Le temps mesuré correspond au temps "user".
- Nous avons limité la charge système lors des mesures.
- Nous avons mesuré seulement la résolution du problème en elle-même.

Remarque 2:

Les données obtenues se situent dans le fichier data_stat.csv.

Remarque 3:

L'ensemble de nos résultats a été trouvé en utilisant le logiciel RStudio.

3 Tests:

3.1 Indicateurs statistiques descriptifs:

Avant d'effectuer des tests statistiques, nous présentons les valeurs des indicateurs de statistiques descriptifs (moyenne, écart-type, médiane, variance...) La colonne A correspond aux résultats du tri par insertion et la colonne B, à ceux du tri à bulle.

A	B
Min.: 0.00	Min.: 0.10
1st Qu.: 0.90	1st Qu.: 6.20
Median: 3.00	Median: 26.85
Mean: 3.89	Mean: 37.05
3rd Qu.: 6.40	3rd Qu.: 62.35
Max.: 13.00	Max.: 116.50

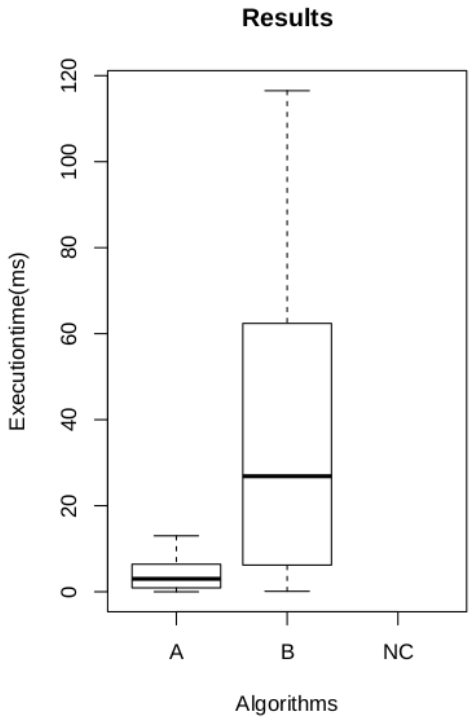

Figure 1: Boxplot

On vérifie bien que la moyenne correspondant à l'algorithme de tri bulle est supérieur à celle de tri par insertion. Pour vérifier si cette différence est significative, nous allons effectuer des tests statistiques appariés et bilatéraux. (On remarque par ailleurs que l'écart interquartile et l'étendue sont beaucoup plus petits pour le tri par insertion.)

3.2 Tests paramétrique

Nous allons dans un premier temps vérifier si les échantillons suivent une loi normale en effectuant un test de normalité.

Test de normalité:

PPCC:
Probability Plot Correlation Coefficient Test

Algorithme A: ppcc = 0.95264, p-value < 2.2e-16
Algorithme B: ppcc = 0.94307, p-value < 2.2e-16

On a p-value < 0.05 dans les deux cas donc on peut raisonnablement considérer que les échantillons suivent une loi normale.

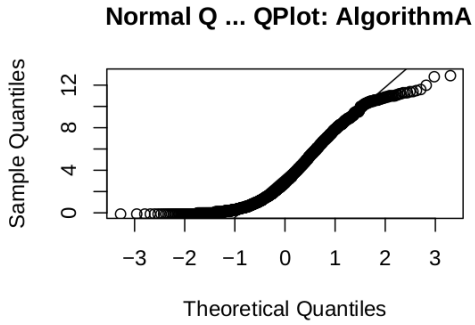

Figure 2: QQplot A

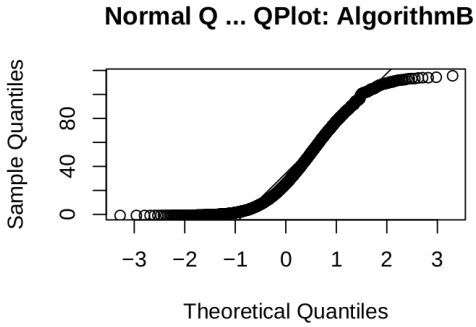

Figure 3: QQplot B

Test de l'égalité des variances:

On vient de voir que les échantillons suivent une loi normale, donc on utilise le test de fisher pour tester l'égalité des variances.

F test pour comparer les deux variances:

F = 0.0096828, num df = 999, denom df = 999, p-value < 2.2e-16

On constate que la p-value est inférieur à alpha pour l'hypothèse nulle suivante "les deux variances sont différentes", on peut donc rejeter l'hypothèse selon laquelle les deux variances sont différentes, et on peut donc raisonnablement considérer les variances comme étant égales.

Test de student:

On a vu précédemment que les deux échantillons suivent une loi normale et que leurs variances sont égales. On effectue donc maintenant un test de T-Student pour poursuivre l'étude.

Paired t-test

t = -34.083, df = 999, p-value < 2.2e-16
mean of the differences: -33.1637

On obtient une p-value inférieure à alpha, donc on peut rejeter H0 (on peut affirmer que les performances entre les deux algorithmes sont différentes).

4 Bonus:

Nous utilisons les tests statistiques pour montrer que la complexité en temps des deux algorithmes de tri est proportionnelle à la longueur du tableau au carré. Première étape: on vérifie que les échantillons représentant le carré

des tailles sont issus d'une distribution approximativement normal. ppcc = 0.95106, n = 1000, p-value < 2.2e-16 On a P value > 0.05 on peut donc raisonnablement considérer que les échantillons suivent une loi normale.

On peut donc utiliser le test de corrélation de Pearson:

Algorithme A:

t = 524.57, df = 998, p-value < 2.2e-16
cor = 0.9981915

On a p-value inférieur à alpha, on peut donc considérer qu'il y a corrélation linéaire entre la longueur du tableau au carré et la complexité en temps.

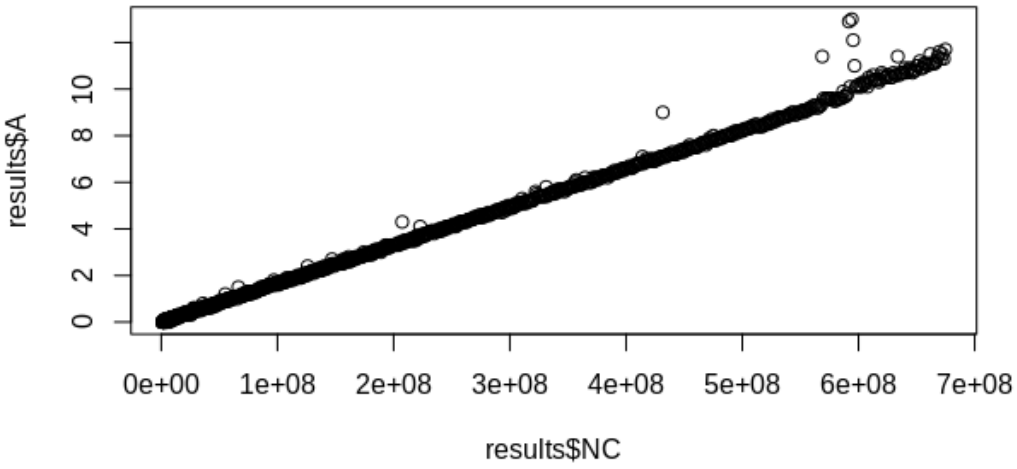

Figure 4: Pearson algorithme A

Algorithme B:

t = 636.81, df = 998, p-value < 2.2e-16
cor = 0.9987718

On a p-value inférieur à alpha, on peut donc considérer qu'il y a corrélation linéaire entre la longueur du tableau au carré et la complexité en temps.

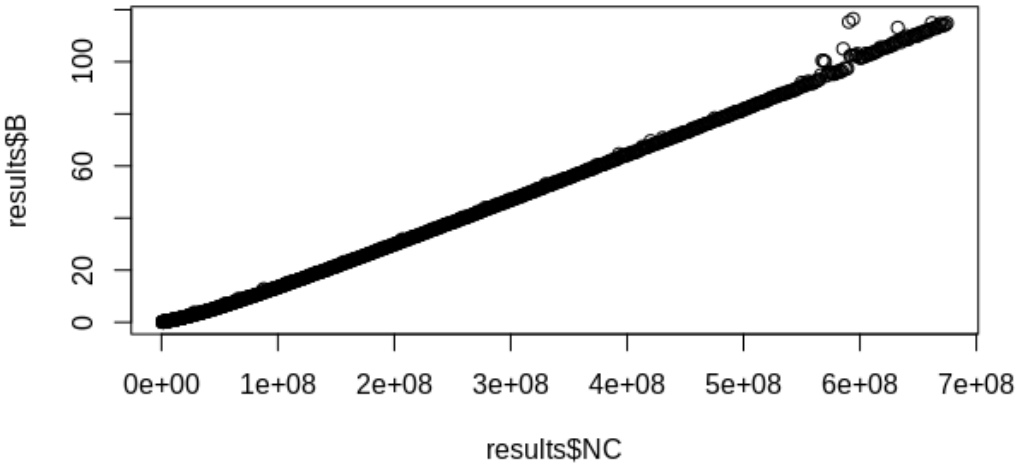

Figure 5: Pearson algorithme B

5 Conclusion:

On peut considérer que les performances de l'algorithme A suit une loi normale, (PPCC = 0.95264 p > alpha) ainsi que celles de l'algorithme B (PPCC = 0.94307, p > alpha). De plus les variantes peuvent être considérées égale (F = 0.0096828, p < alpha).

De plus, on peut affirmer que les performances entre l'algorithme A (M=3.89, SD=XX) et l'algorithme B (M=37.05, SD=XX) sont différentes (t(1000)=524.57, p> alpha) donc peut affirmer que l'algorithme A (tri par insertion) est plus

performant que l'algorithme B (tri bulle) dans la classe de problème de tri de tableaux que nous avons considérés.

De plus, on peut considérer que la complexité en temps des deux algorithmes de tri est proportionelle à la longueur du tableau au carré (Algorithme A: r =0.9981915 , p-value = 2.2e-16, Algorithme B: r = 0.9987718 , p-value $<$ 2.2e-16)

6 Limite de notre méthode:

Il nous semble important de préciser que le résultat que nous obtenons sur le tri par insertion et le tri à bulle concerne une plage de taille de tableau (Allant de 1000 à 26000 par pas de 25).

Ainsi, on peut tout a fait imaginer des résultats de performances différents pour des tailles fixes : l'algorithme de tri à bulle pourrait être plus performant pour des tableaux de petites tailles tout en étant moins performant en moyenne sur un ensemble de tableaux de tailles différentes).

Notre étude ne permet pas par exemple de penser que le tri par insertion est meilleur que le tri à bulle pour toutes les tailles de tableau allant de 1000 à 26000).

Il faudrait donc compléter notre étude par des comparaisons de performances sur des tailles fixes pour pouvoir affiner nos résultats.

De plus, étant donné que tout un ensemble de facteurs peuvent altérer les mesures de manière non contrôlée, il nous semble important de présenter certains points qu'il serait intéressant d'étudier:

- ➔ il serait intéressant de refaire les tests sur différentes plate-formes (plus rapide ou plus lente)
- ➔ d'étudier la manière de choisir le plus haut niveau d'optimisation et la charge du système la plus faible possible
- ➔ il faudrait pouvoir refaire des tests avec plusieurs générateurs de nombres pseudo-aléatoire, car ils produisent parfois des pattern aléatoires qui peuvent biaiser les résultats.

➔ il faudrait vérifier si l'implémentation de l'algorithme ne joue pas un rôle dans les performances obtenues, ce qui pourrait biaiser le résultat (une moins bonne performance serait du à l'implémentation de l'algorithme plutôt qu'à la performance elle-même) on sait par exemple qu'il y a une différence entre les pré-incrémentation et post-incrémentation, exploitation de la mémoire cache...