

Laboratoire 3 – tris

Objectifs ASD

- Se familiariser avec le tri par comptage
- Mise en œuvre de plusieurs tris vu au cours.
- Comparer graphiquement les complexités théoriques calculées avec celles mesurées empiriquement.

Objectifs C++

- Coder de manière générique avec template

But

Le but du laboratoire est de mettre en œuvre en C++ des algorithmes de tri et d'en comparer la complexité.

Consignes

Vous allez mettre en œuvre trois algorithmes de tri, de complexité $O(n^2)$, $O(n * \log_2(n))$ (en moyenne) et $O(n)$.

- L'algorithme en $O(n^2)$ sera le tri par sélection (***selection sort***).
- L'algorithme en $O(n * \log_2(n))$ sera le tri rapide (***quick sort***).
- L'algorithme en $O(n)$ sera le tri par comptage, également appelé le tri casier (***counting sort***)

Les pseudocodes correspondant aux deux premiers algorithmes se trouvent dans le support de cours. Pour celui de tri casier vous pouvez consulter Wikipédia anglophone

https://en.wikipedia.org/wiki/Counting_sort

(Attention, la page francophone de cet article est de nettement moins bonne qualité !)

D'une manière générale nous vous encourageons à utiliser en priorité la version anglophone et à apprendre le nom des algorithmes en anglais.

Mesure du temps de calcul

Vous allez estimer de manière empirique le temps nécessaire à l'exécution des trois algorithmes sur :

1. des séries de données de taille différente de même distribution, ou alors
2. des séries de données de même taille et de distribution différente.

Mesurez le temps nécessaire pour trier, dans la mesure du possible :

1. n entiers positifs générés aléatoirement entre 1 et 100 (inclus), avec $n = \{10^m \mid m \in [1, 2, \dots, 6]\}$,

2. $n = 100'000 = 10^6$ entiers positifs générés aléatoirement entre 1 et k (inclus), avec $k = \{10^m \mid m \in [1, 2, \dots, 6]\}$. Pour le tri par comptage vous devrez peut-être le faire en deux fois, voir chiffre par chiffre.

Si ce n'est pas possible raisonnablement pour un algorithme ou une taille, indiquez-le.

Respectez les consignes suivantes dans les évaluations de trois algorithmes :

- Pour pouvoir comparer les trois algorithmes, pour une taille de donnée fixée, il faut donner en entrée la même série de données aux trois algorithmes dans les évaluations.
- Il est nécessaire d'effectuer un certain nombre de simulations pour les différentes mesures (par exemple, 30 simulations pour une taille de données fixée, vous préciserez le nombre choisi autre que 30 dans votre rapport).
- Le temps d'exécution d'un algorithme pour une taille de données fixée sera la moyenne des temps d'exécutions de l'algorithme pour les différentes simulations.
- Pour obtenir des résultats réalistes, une nouvelle série des données doit être générée pour chaque simulation.

Présentation des résultats

Tout d'abord, il faut tester les différentes implémentations de tri avec les codechecks et générer **les trois fichiers zip** via codecheck (bouton « *Download Report* »).

Il faut ensuite écrire le code source de vos tests de comparaison dans **un seul fichier labo3.cpp** contenant les tris et le programme d'évaluation de trois tris.

Enfin, il faut rédiger **un rapport de maximum 3 pages** qui contient les résultats de simulation présentés dans un tableau et dans un graphique. N'oubliez pas d'utiliser l'échelle appropriée dans les deux graphiques.

Montrez sur la même figure, les graphes obtenus par l'évaluation empirique et par la formule de complexité théorique. Les courbes obtenues correspondent-elles à la complexité théorique de chaque algorithme ? Donnez des explications.

Le rendu de **ces 5 fichiers** se fait sur Cyberlearn en un seul zip en respectant les règles de rendu indiquées dans le fichier « *consignes.pdf* » se trouvant sur Cyberlearn.

Mesure du temps en C++

Voir le labo 1.

Contacts

- Voir les slides du premier cours ou espace cyberlearn HES-SO (moodle).