

Laboratoire 1 – complexité

Objectifs ASD

- Se familiariser avec le calcul des complexités.
- Estimer empiriquement la complexité d'une fonction :
 - En comptant des opérations
 - En mesurant le temps d'exécution.
- Comparer graphiquement les complexités théoriques calculées avec celles mesurées empiriquement.
- Identifier le meilleur cas, le pire cas et le cas moyen quand le temps d'exécution dépend de données d'entrée.

Objectifs C++

- Réviser la classe vector de la STL.
- Réviser les itérateurs en C++.
- Comprendre les informations qu'on trouve dans la documentation de la STL concernant la complexité des méthodes utilisées.

Consignes

Ce laboratoire se déroule sur une période de deux semaines. Vous devez rendre un rapport et le code qui vous a servi à mesurer ou estimer les complexités.

Différentes fonctions C++ vous sont fournies dans le fichier .cpp en annexe.

Pour chacune des ces fonctions, estimez **théoriquement** leur complexité. Lorsque ces fonctions utilisent une méthode de la STL, utilisez sa documentation (via cplusplus.com par exemple) pour en connaître la complexité.

Pour chacune de ces fonctions, les commentaires de l'en-tête précisent également sur quelle base en estimer **empiriquement** la complexité. Soit en comptant un nombre d'opérations, soit en mesurant un temps d'exécution.

Pour l'estimation empirique de la complexité, créez un programme de test qui appelle les différentes fonctions avec différentes tailles de jeux de données (N) et différents paramètres suivant les fonctions.

Pour les mesures de nombres d'opérations, vous pouvez évidemment modifier ces fonctions pour y ajouter des compteurs. Pour les mesures de temps d'exécutions, il faut exécuter plusieurs mesures et les moyenner pour obtenir une estimation plus robuste.

Pour chaque fonction, votre rapport doit contenir les informations suivantes

- L'estimation théorique de la complexité, y compris une justification de cette estimation. (5 lignes au maximum)
- Le pire cas, le meilleur, et le cas moyen quand la complexité dépend du contenu des données. Sinon, indiquez que la complexité en est indépendante (5 lignes au maximum).
- Un tableau reprenant au moins 6 mesures empiriques en fonction du paramètre N ou de la taille des données.
- Un graphique reprenant ces données et les comparant à l'estimation théorique. Veillez à bien choisir l'échelle des axes et les valeurs de N pour que la représentation graphique soit pertinente.

Le rapport fera au maximum 6 pages.

Mesure du temps en C++

```
#include <chrono>
using namespace std::chrono;

//prendre le moment de départ
high_resolution_clock::time_point t1 = high_resolution_clock::now();

//exécuter les opérations à chronométrer ici

//prendre le moment d'arrivée
high_resolution_clock::time_point t2 = high_resolution_clock::now();

//calcul du temps, ici en nanosecondes
double temps = duration_cast<nanoseconds>( t2 - t1 ).count();
```

Contacts

- Prof. Olivier Cuisenaire : olivier.cuisenaire@heig-vd.ch
- Prof. Mireille Goud : mireille.goud@heig-vd.ch
- Prof. Laura Elena Raileanu : laura.raileanu@heig-vd.ch
- Dominique Jollien : dominique.jollien@heig-vd.ch
- Racine Raphaël : raphael.racine@heig-vd.ch