

Analyse d’algorithmes et validation de programmes – Projet

IATIC 4

2023-2024

Le but de ce projet est de comparer plusieurs algorithmes résolvant le même problème, en utilisant les différents outils vus en cours. Ces différents algorithmes peuvent être des solutions différentes, ou différentes versions d’une même solution (récursive et itérative, par exemple, ou utilisant des structures de données différentes). La comparaison se fera par une analyse théorique, mais aussi en comparant leurs implémentations. Ce travail sera fait en monômes.

1 Choix de l’algorithme

Les algorithmes étudiés doit permettre une implémentation, dans le langage de votre choix. Pour permettre une étude de complexité, il devra bien évidemment contenir au moins une boucle ou un appel récursif. Le problème résolu peut être l’un des problèmes abordé en cours, ou un problème tout autre. Si un algorithme manipule des structures de données, vous prendrez soin de les implémenter de façon séparée, en écrivant les fonctions de manipulation adaptées. Vous devrez choisir le problème résolu et les algorithmes étudiés de façon à pouvoir mener l’étude décrite ci-dessous.

2 Traitements à effectuer

Vous devrez tester à la fois la validité et le temps de calcul de l’implémentation, et vérifier leur conformité avec les résultats de l’étude de l’algorithme. Ce travail doit vous permettre de corriger et/ou d’améliorer votre programme.

2.1 Validité

Dans l’étude de sa validité, il vous est de demander de déterminer un jeu de test permettant de vérifier les points-clefs de votre programme et d’y soumettre votre programme. Vous justifierez soigneusement le choix du jeu de données test. Vous en fournirez les résultats et les commenterez.

Il vous est demandé d'écrire et de tester la spécification du problème. De même, vous identifierez un ou plusieurs invariants de boucle, et testerez leur validité. Vous devez prouver les différents algorithmes que vous présentez.

2.2 Complexité

Vous étudierez également le temps de calcul du programme, que vous confronterez à la complexité de l'algorithme implémenté. Lorsque ces notions ont un sens pour l'algorithme étudié, il sera intéressant de mettre en lumière le temps de calcul en moyenne, ainsi que dans le pire des cas. Lorsque c'est possible, il sera intéressant de pousser le programme aux limites des capacités (en termes de mémoire en particulier) de l'ordinateur sur lequel vous l'exécutez, et d'observer les phénomènes qui se produisent alors. Une étude (par exemple avec Valgrind) des facteurs limitant l'efficacité du programme sera également la bienvenue.

2.3 Structures de données

Il vous est demandé de justifier le choix des structures de données utilisées et de les décrire dans le détail (fonctions de manipulation avec leurs complexités, choix d'implémentation).

3 La notation de ce projet

L'élément essentiel de notation est le rapport. Il devra donc être soigné. Son introduction et sa conclusion devront contextualiser le projet dans le cadre de l'algorithmique et non de votre situation personnelle¹.

Votre rapport ne doit pas être un manuel d'utilisation, et devra au contraire expliquer les choix que vous aurez opérés. Vous devrez expliquer et argumenter les choix que vous avez faits, ainsi que vos partis pris d'implémentation. Votre rapport ne peut comporter d'éléments de code que de façon très limitée, comme support à vos explications.

En particulier, votre rapport devra présenter de façon détaillée et argumentée les structures de données utilisées.

L'utilisation de sources bien référencées est encouragée. Le plagiat (c'est-à-dire, vous attribuer le travail d'un autre, en l'utilisant sans le mentionner et sans le modifier substantiellement) est par contre proscrit, tant en matière de rapport que de code : il vous vaudra un 0, sans préjuger de procédures disciplinaires éventuelles qui s'ensuivraient et pourraient vous coûter jusqu'à une exclusion pendant 5 ans de l'enseignement supérieur.

La fonctionnalité de votre code sera bien sûr prise en compte (voir la section 4), ainsi que votre code, qui devra donc être lisible et structuré.

1. Il est inutile, et sera pénalisé, de rappeler que vous êtes en IATIC4, que c'est dans le cadre du cours d'AAVP, etc. Expliquez pourquoi vous avez choisi ce problème, son intérêt et ou son importance en algorithmique, et dans la mesure du possible, problématisez votre rapport.

Voici, à titre indicatif (c'est-à-dire que je pourrai m'en éloigner si je le juge opportun), un barème :

- Rapport :
 - Introduction (incluant présentation du problème, motivation, problématique, annonce et explication du plan) : 3 points.
 - Conclusion (incluant réponse à la problématique, perspectives) : 1 point.
 - Rédaction : 2 points.
- Code :
 - Exécution : 4 points.
 - Clarté : 2 points.
- Récursivité :
 - Récursivité : 2 points.
 - Dérécursivation : 4 points.
- Preuve :
 - Preuve d'un algorithme récursif : 2 points.
 - Invariant de boucle (formulation et preuve) : 2 points.
 - Spécification du problème : 2 points.
 - Explication (éléments de preuve formalisés seulement partiellement, et rédigés) : 2 points.
- Complexité :
 - Algorithme récursif : 3 points.
 - Algorithme itératif : 3 points.
 - En moyenne : 2 points.
- Validation : 3 points.
- Types de données :
 - Description : 2 points.
 - Complexité : 2 points.

Ce barème somme à bien plus que 20, car il est évident pour moi que vous devez faire des choix, et creuser certaines pistes plutôt que d'autres ; le nombre de points indiqués est le nombre maximal de point que l'on peut avoir sur un sujet, en faisant un travail détaillé et original (c'est-à-dire qui ne soit pas la recopie simple du cours ou d'une autre source, ou une transposition immédiate). Il est conçu, avec la précision précédente, pour que vous ayez intérêt à aborder tous les thèmes du cours.

4 Les conditions du rendu de ce projet

Vous me remettrez un projet par monôme, en réponse à ce mail, le 11 mars au plus tard. Ce mail contiendra deux pièces jointes portant votre nom :

- l'une, au format tgz, contenant un répertoire portant le même nom, le code source et un Makefile ;
- l'autre, au format pdf, lisible par Adobe Reader, contiendra votre rapport.

L'invocation de **make** doit produire un exécutable. Tous les outils doivent m'être

fournis pour me permettre de tester simplement votre implémentation (fichiers d'entrée simplement modifiables, bref README donnant le mode d'utilisation). Chaque fichier source doit également contenir en commentaire le nom de son auteurs. Votre code doit s'exécuter sur un Linux « standard » sans avoir besoin d'installer d'outil supplémentaire.

Un rendu ne satisfaisant pas ces conditions encourt le risque de ne pas être corrigé.