

6. Informatique

6.1. Introduction

Le sujet s'attelle à résoudre informatiquement le jeu Rush Hour. Ce jeu consiste à essayer de faire sortir une voiture d'un parking en déplaçant les différents véhicules à l'intérieur.

La partie A cherche à résoudre concrètement ce problème. Elle est découpée en plusieurs sous-parties. La sous-partie I teste les compétences en C des étudiants, leur maîtrise des tableaux, structures et pointeurs. À la fin de cette partie, un calcul de complexité amortie est demandé. Les sous-parties II et III traitent des structures de données (files et tables de hachage). Ces sous-parties demandent à la fois de programmer ces structures et d'être capable de raisonner dessus. Une connaissance assez fine des enjeux des tables de hachage est demandée à la question 16. La sous-partie IV fait le bilan de la partie A et demande, à l'aide de tout ce qui précède, d'implémenter une solution du problème Rush Hour.

La partie B, elle, demande de démontrer la NP-complétude du problème résolu dans la partie A. La sous-partie I se limite à un cas particulier polynomial. La résolution de ce cas particulier demande de maîtriser une autre structure de données, les graphes. En outre, cette sous-partie teste les compétences en programmation en OCaml.

La sous-partie II traite le cas général. Il est demandé de réduire PLANAR 3 SAT au problème du Rush Hour, puis d'en déduire la NP-complétude de Rush Hour.

6.2. Analyse globale des résultats

Les candidats ont bien avancé dans le sujet, traitant en moyenne les deux tiers du sujet (25 questions sur 38).

Sur l'aspect pratique, les candidats ont montré qu'ils savent bien programmer, bien manipuler des structures de données complexes. Les questions de programmation ont été bien réussies. L'allocation dynamique de mémoire et la manipulation de pointeurs sont globalement bien comprises. De manière surprenante, manipuler une structure contenant un point vers un tableau à deux dimensions (II et III de la partie A) s'est révélé plus facile pour beaucoup de candidats que manipuler une structure contenant un tableau statique à deux dimensions (I de la partie A).

Les questions en C ont plus été traitées que celles en OCaml, mais il est difficile de savoir s'il s'agit d'une préférence des candidats pour le C ou si c'est simplement dû au fait que les questions de OCaml sont à la fin du sujet.

Sur l'aspect théorique, les candidats ont montré qu'ils sont capables de raisonner et de démontrer. Les questions théoriques ont été plus souvent et mieux traitées que l'année précédente. En particulier, un plus grand pourcentage de candidats que l'année dernière savent raisonner sur le concept de NP-complétude et comprennent l'idée de la réduction d'un problème à un autre.

Le jury ne peut que se réjouir de cette montée en compétence des candidats.

6.3. Commentaires sur les réponses apportées et conseils aux candidats

Le jury souhaite attirer l'attention des candidats sur certaines difficultés du sujet.

Q1 le fait que la structure contienne le tableau et pas simplement un pointeur vers le tableau a perturbé de nombreux candidats.

Q10 et Q11 78 % des candidats ont traité **Q11** de manière au moins partiellement juste et 38 % ont traité cette question de manière parfaitement juste. Il est satisfaisant de constater que les candidats comprennent la notion de complexité amortie. Cependant, certains candidats ont eu du mal dans les calculs menant à estimer cette complexité : **Q10** a été bien moins réussie que **Q11**.

Q14 L'objectif de la question est bien évidemment d'utiliser les avantages de la structure de table de hachage décrits juste avant cette question.

Le jury est heureux que la majorité des candidats aient bien compris les enjeux de cette structure de données et pensent à utiliser la fonction de hachage pour débiter la recherche. Cependant une minorité de candidats a préféré parcourir systématiquement la table de hachage à partir du début, ce qui a bien évidemment été fortement pénalisé.

Q16 c'est une question nécessitant du recul sur la notion de table de hachage. Elle teste la compréhension du concept et des enjeux des collisions. C'est une question qui est censée être traitée rapidement.

Q24 Une erreur s'est glissée dans cette question. Le problème vient de la définition de « chemin augmentant ». Un chemin augmentant, par définition, est un chemin alternant. Or, dans la définition de « chemin alternant », il est demandé qu'un chemin alternant soit « élémentaire ». Cette hypothèse qu'un chemin doit être élémentaire pour être augmentant a été oubliée dans l'énoncé de la question.

La plupart des candidats ne s'en sont pas aperçus. Bien évidemment, nul ne reprochera à un étudiant de ne pas avoir remarqué un problème qui a échappé à tous les relecteurs du sujet. Certains candidats ont mentionné l'erreur, ce qui montre leur compétence et leur recul par rapport au problème posé.

6.4. Conclusion

Les candidats ont montré tant leurs compétences pratiques, techniques que théoriques. Ils savent aussi bien programmer que raisonner sur des structures de données complexes.

Le niveau des candidats est en progression, montrant la montée en puissance de la filière MPI. Tout cela présage le meilleur pour la poursuite, par les candidats, d'études exigeantes sur les plans scientifiques et techniques.