ÉPREUVE PRATIQUE D'ALGORITHMIQUE ET DE PROGRAMMATION DU CONCOURS COMMUN DES ÉCOLES NORMALES SUPÉRIEURES — RAPPORT DE JURY 2015

École concernées : Cachan, Lyon, Paris, Rennes
Membres du jury : Jérémie Detrey, Marc Mezzarobba, Michaël Rao, Olivier Teytaud

1. Organisation de l'épreuve

Comme à l'accoutumée, l'épreuve se composait d'un travail sur machine d'une durée de 3h30, suivi d'une présentation orale d'environ 25 minutes. Lors de l'épreuve pratique, les candidats doivent mettre en œuvre une chaîne complète de résolution d'un problème informatique, du choix d'algorithmes et de structures de données pour répondre à une spécification au calcul effectif de résultats numériques. La présentation orale leur permet de démontrer leur compréhension du sujet et d'aborder avec l'examinateur des questions qu'ils n'auraient pas eu le temps de traiter complètement auparavant.

À la différence des années précédentes, seul le système Debian GNU/Linux était proposé comme environnement de travail. Les langages de programmation disponibles changeaient également cette année, avec notamment l'introduction de Python et le retrait de Maple. Les candidats pouvaient se familiariser à l'avance avec l'environnement informatique grâce à une image .iso disponible sur le site internet de l'épreuve. Par ailleurs, ils disposent d'une période de 10 minutes avant la distribution des sujets pour poser des questions aux surveillants s'ils rencontrent des difficultés d'ordre pratique. Il ne saurait donc y avoir d'excuse pour ne pas maîtriser cet environnement. Le jury rappelle également aux candidats qu'ils sont seuls responsables de la bonne sauvegarde de leur travail.

Les langages les plus choisis sont Caml et Python, suivis de loin par C++. Quelques candidats maîtrisent plusieurs langages et sont capables de faire leur choix en fonction du problème.

Quatre sujets ont été proposés au total, sur les cographes, la résolution du jeu de société Quarto !, des algorithmes de géométrie algorithmique, et la cryptanalyse d'un modèle légèrement simplifié de la machine Enigma. Chaque énoncé comportait comme d'habitude des questions « pratiques » à résoudre sur machine et des questions « théoriques » à développer pendant l'oral. Les premières aboutissent chacune à un résultat à reporter sur la fiche réponse fournie et, sauf situation exceptionnelle, sont notées uniquement en fonction de la correction de ces résultats, sans que le jury n'examine les programmes écrits par les candidats. Les secondes sont à résoudre au tableau lors de la partie orale de l'épreuve, mais il est très fortement recommandé de les préparer lors de la partie sur machine, ce que de nombreux candidats ne font qu'insuffisamment.

Chacune des deux catégories de questions représentait la moitié de la note finale. On observe dans l'ensemble une bonne corrélation entre les résultats obtenus aux deux parties. Le jury attire cependant l'attention des candidats sur une évolution

possible de ce barème dans les années à venir pour attribuer un poids plus important aux questions pratiques.

2. Remarques générales

Les sujets sont longs et difficiles ; il n'y a pas lieu d'être inquiet de ne pas arriver à la fin. Quelques candidats brillants ont néanmoins traité certains sujets de manière quasi-parfaite. À l'opposé, sans doute sous l'effet du stress et dans de rares cas, des candidats essaient de fournir des résultats sans avoir vérifié que leurs codes retrouvaient bien les valeurs indiquées dans l'énoncé pour la graine $\tilde{u}_0$. Cette approche économise bien peu de temps, en regard du fort risque d'erreur dans les résultats obtenus.

Quasiment tous les candidats s'avèrent capables de calculer la suite des u_n utilisée dans la plupart des énoncés. Tous ne pensent pas en revanche à stocker en mémoire les u_n pour éviter des temps de calcul prohibitifs. (Le coût de certains algorithmes par ailleurs de complexité linéaire peut devenir quadratique du seul fait du recalcul dez u_n.)

Plusieurs sujets comportaient des questions demandant d'estimer l'ordre de grandeur d'un temps de calcul, qui ont été diversement traitées. Il convient de savoir qu'un gigahertz représente un milliard de hertz. Un microprocesseur contemporain fonctionne à une fréquence de l'ordre du gigahertz : on ne peut donc traiter, sur un seul cœur, des millions de milliards de cas par seconde. Inversement, des calculs massivement parallèles demandant de l'ordre de 2^{60} opérations sont aujourd'hui considérés comme faisables.

Les résultats classiques sur les algorithmes de tri, utiles à plusieurs reprises, sont généralement bien connus. Certains candidats perdent cependant du temps à implémenter eux-mêmes ces algorithmes au lieu d'utiliser les fonctions disponibles dans la bibliothèque standard du langage de programmation choisi.

3. Cographes

Le sujet demandait de programmer différents algorithmes travaillant sur une classe de graphe assez simple, et bien connue en théorie des graphes : les cographes. Ces graphes ont la particularité de pouvoir être modélisés par des arbres étiquetés. De ce fait, quasiment tous les algorithmes qu'il était demandé d'implémenter travaillaient uniquement sur des arbres. Seules les questions 9 et 10 demandaient à travailler effectivement des graphes en général. La majorité des candidats a traité le sujet jusqu'à la question 5 comprise. La question 6, qui demandait de transformer un arbre pour le mettre sous une forme canonique, a été résolue par 12 candidats sur 40, et a fait souvent la différence entre les notes en-dessous et au-dessus de la moyenne. Les meilleures copies traitent de plus les questions 7 à 9 ou la question 10.

4. Quarto !

Ce sujet demandait de modéliser le jeu de Quarto ! (plateau, condition de gain...), puis d'en déterminer des stratégies gagnantes par l'algorithme « du minimax » avec diverses améliorations (mémorisation des positions connues dans une table de hachage, élagage alpha-bêta...), idéalement jusqu'à résoudre complètement le jeu. Les principales difficultés étaient de programmer sans erreur les règles du jeu et de bien comprendre le principe du retour sur trace (*backtracking*). Globalement, les candidats ont traité le sujet jusqu'à la fin de la partie 3 (question 5), avec une

réussite variable, et abordé par la suite avec l'aide de l'examinateur les questions théoriques 5 à 7.

Certaines questions comportaient malheureusement une ambiguïté due à une erreur dans l'indexation de certaines suites de positions ($\gamma^{(0)} \to \cdots \to \gamma^{(15)}$ au lieu de $\gamma^{(0)} \to \cdots \to \gamma^{(16)}$). Cela ne semble pas avoir posé de problème majeur aux candidats, et le jury a apprécié de voir certains relever explicitement la difficulté.

Les questions pratiques 1 et 2 n'ont guère posé problème. De façon inattendue, en revanche, les questions à développer pendant l'oral 1 et 2 se sont avérées discriminantes. Près de la moitié des candidats a des difficultés avec les dénombrements demandés dans la première, peut-être par excès de précipitation, tandis que moins d'un tiers donne spontanément une preuve convaincante de l'algorithme de construction de la permutation π_n. Il est attendu des candidats qu'ils soient capables de formaliser correctement un raisonnement sur la correction d'un algorithme.

Les questions théoriques 3 et 4, qui demandaient d'analyser des algorithmes simples, ont été bien voire très bien traitées. (Plusieurs candidats ont par exemple observé qu'il était possible, en pratique, de répondre à la question 3 en utilisant des opérations booléennes bit à bit.) Beaucoup de candidats ne donnent en revanche des résultats corrects que pour une partie des instances dans les questions pratiques de la partie 3.

Concernant les parties suivantes, la plupart des candidats a compris le principe du retour sur trace, ainsi que le système de « coupes » (*pruning*) permettant d'accélérer la résolution. En revanche, peu sont arrivés à les mettre en œuvre dans le temps imparti. La première question pratique sur la détermination des stratégies gagnantes (question 6) n'a ainsi été traitée que par un candidat sur trois.

La question à développer pendant l'oral 6 (souvent posée à des candidats qui ne l'avaient pas préparée) a rarement inspiré les candidats, alors qu'il s'agit d'une application assez directe d'un résultat fondamental du cours de mathématiques.

Il n'était pas facile de trouver toutes les symétries demandées à la question théorique 7, mais un candidat y est parvenu, et quelques autres en ont énuméré une bonne partie.

5. Géométrie

La capacité à choisir le langage le plus adéquat s'est avérée utile. Peu de candidats ont su utiliser des astuces simples, comme des méthodes heuristiques pour filtrer les points susceptibles d'entrer dans l'enveloppe convexe. Les candidats avaient besoin de calculer des déterminants, et des enveloppes convexes. La partie concernant le calcul de distance minimale entre deux points d'un ensemble fini a parfois été diversement traitée au niveau théorique. En moyenne, le sujet a été traité jusqu'à Q4 inclus. Une partie des candidats a été capable de cumuler la complexité d'une fonction récursive parmi les différents niveaux des appels.

6. Enigma

Ce sujet, dont l'aspect stimulant a été souligné par plusieurs candidats, portait sur la machine Enigma, utilisée par l'armée allemande pour chiffrer ses communications durant la seconde guerre mondiale, ainsi que sur deux cryptanalyses majeures de ce système. Le sujet était plutôt long, et s'articulait essentiellement en trois parties.

La première partie (questions 4 à 7), consacrée à l'étude de la machine Enigma et à la modélisation de son mécanisme de chiffrement a été globalement bien traitée par

les candidats, bien que les réponses à la question d'oral 2, qui demandait d'effectuer quelques dénombrements, aient été très inégales. Nous notons que cette question a parfois donné lieu à des estimations peu réalistes des capacités de calcul des machines actuelles.

La seconde partie (questions 8 et 9) étudiait la cryptanalyse d'Enigma utilisant la méthode des caractéristiques, mise au point par les analystes polonais. Les questions théoriques (7 à 10) ont été ici les plus discriminantes du sujet, car l'aisance des candidats pour manier les ensembles de permutations et les signatures a été très variable. Par contre, très peu d'entre eux ont su traiter les questions pratiques.

Enfin, la dernière partie, qui présentait la cryptanalyse britannique à laquelle Alan Turing a grandement contribué, n'a malheureusement pu être abordée que par une poignée de candidats.