
EPREUVE ORALE PRATIQUE D'ALGORITHMIQUE ET PROGRAMMATION

ENS : PARIS LYON CACHAN

Coefficients : PARIS 4 LYON 4 CACHAN 6

MEMBRES DE JURYS : Yves Robert, Benoît Caillaud, Damien Massé

Comme l'année dernière, cette épreuve demandait aux candidats de mettre en œuvre la chaîne *complète* de résolution d'un problème informatique: analyse des spécifications, choix des structures de données, mise en forme et évaluation de l'algorithme et de sa complexité, programmation sur machine et test des programmes. En plus de la partie programmation, la présentation orale permettait d'évaluer les candidats sur leur capacité à expliquer leurs approches et solutions.

Le jury a examiné cette année 114 candidats¹ (dont 2 de filière PC), sur 5 sessions organisées comme l'année précédente en « pipeline »: les candidats d'une session commune arrivent par groupe de 3 toutes les demi-heures, de sorte que les derniers arrivés rentrent avant que les premiers n'aient fini leur présentation orale. Une fois encore, le jury a fait un barème répartissant les points à égalité entre la partie « résultats pratiques » et la présentation des algorithmes à l'oral.

Comme l'année dernière, plusieurs langages et environnements de programmation étaient proposés :

- PC sous Windows 2000, avec CamlLight, Objective Caml, Maple, Java ou Pascal (Delphi).
- PC sous Debian/Linux KDE, avec Caml Light, Objective Caml, C, MuPad et Java.

Il était demandé aux candidats de choisir le langage et l'environnement à l'avance. Plus encore qu'en 2005, CamlLight sous Windows a été très majoritairement choisi: 71 % des étudiants ont opté pour cet environnement, 13 % des étudiants ont choisi CamlLight ou Ocaml sous Linux, 10 % ont choisi Pascal sous Windows, 4 % C sous Linux, et 2 % ont choisi Maple sous Windows. Encore une fois, ni Java ni MuPad n'ont été choisis.

Plusieurs sujets proposaient plusieurs tailles pour le problème à résoudre. Le but était à la fois de permettre l'utilisation d'algorithmes « naïfs » peu efficaces pour résoudre la taille moyenne, et d'inciter les candidats à tester leur programme sur des petites valeurs solubles à la main (ce que beaucoup de candidats ont fait). Dans le cas où le sujet ne proposait pas plusieurs tailles, les candidats devaient tester leurs programmes de leur côté. Le jury rappelle une fois de plus l'importance pour les candidats d'apprendre à tester et corriger leur programme, en utilisant éventuellement des commandes d'affichage pour tracer l'exécution de l'algorithme.

Quelques remarques générales :

- Les candidats (surtout sous Caml) utilisent beaucoup les structures de liste chaînée (ou, plus généralement, des types de données récurifs). Si celles-ci peuvent être très utiles, elles peuvent aussi se révéler handicapantes et lourdes à gérer. Dès lors qu'ils ne sont pas inadaptés et que le nombre maximum d'éléments est connu, préférer l'utilisation d'un tableau ou d'une matrice, même pour représenter une structure complexe comme un arbre (comme pour le sujet Apprentissage de Langages Réguliers).

¹Dont 7 candidates, ce qui représente une très légère augmentation par rapport à 2005.

- De la même façon, l'utilisation de la récursivité est parfois très utile et de nombreux candidats maîtrisent bien cette technique de programmation, mais tous les algorithmes ne s'y ramènent pas. De nombreux problèmes peuvent se résoudre plus facilement à l'aide de boucles, et le jury encourage les candidats à programmer aussi des algorithmes itératifs en Caml.

Les sujets ont probablement été plus longs que l'an passé, et à l'exception du sujet Objets Géométriques Planaires Iso-orientés, n'ont été achevés par aucun candidat. Néanmoins, tous les sujets disposaient de questions de difficultés croissantes permettant de tester les capacités des différents candidats. Voici maintenant des commentaires pour chacune des épreuves:

- Objets géométriques plans iso-orientés: Le but de ce sujet était de montrer comment des structures de données bien conçues permettaient d'optimiser certains algorithmes. Tous les problèmes pouvaient ainsi être résolus avec une complexité linéaire (en le nombre d'abscisses plus le nombre de rectangles). La majeure partie des candidats a bien traité la partie 3 (sans forcément suivre la structure proposée dans l'énoncé) mais n'a pas réussi la partie 4. Un candidat (au moins) a traité l'ensemble du sujet.
- Etoiles: Ce sujet a été donné pour une session avec peu de candidats, et demandait un peu de réflexion géométrique. Les meilleurs candidats ont atteint la question 5. Du point de vue algorithmique, la recherche dichotomique doit être connue des candidats.
- Arbres phylogénétiques: Un certain nombre de candidats ont cherché à résoudre les questions sans construire l'arbre, ce qui devenait rapidement très difficile. L'arbre pouvait être représenté à l'aide d'un tableau, et construit de façon itérative. Traiter parfaitement les quatre premières questions permettait d'avoir la moyenne.
- Apprentissage de langages réguliers: Encore une fois, l'utilisation de tableaux pour la construction de l'automate était plus simple que l'utilisation d'une structure récursive. Il suffisait de constater que le nombre d'états de l'automate était inférieur à k (ce que peu d'étudiants ont vu).
- Classifier avec des arbres: La question 4 a été la plus discutée à l'oral, à la fois pour sa difficulté algorithmique, et l'étude de sa complexité. Un tableau d'arbres était une solution satisfaisante, et de nombreux étudiants ont trouvé un algorithme raisonnable (sans forcément réussir à le programmer). Peu d'étudiants ont trouvé directement une complexité cubique. Les meilleurs étudiants sont allés jusqu'à la question 6.

On peut trouver les cinq sujets proposés cette année sur le site web de l'épreuve :

<http://www.ens-lyon.fr/LIP/ConcoursInfo>