ÉPREUVE ORALE PRATIQUE D'ALGORITHMIQUE ET DE PROGRAMMATION

ENS : PARIS LYON CACHAN

Durée : 4 heures ***Coefficients* : PARIS 4 LYON 4 CACHAN 6**

MEMBRES DE JURYS :
Jean Goubault-Larrecq, Laurent Mauborgne, et Nicolas Schabanel

L'objectif de cette épreuve est d'évaluer les capacités des candidats à mettre en œuvre la chaîne *complète* de résolution d'un problème informatique : analyse des spécifications abstraites, conception d'algorithme et choix des structures de données, évaluation de sa complexité (coût en temps et en mémoire), programmation sur machine dans l'un des langages proposés, test des programmes sur des petites valeurs des paramètres, et exécution sur des valeurs précises des paramètres. À la fois algorithmique, pratique et orale, cette épreuve est donc transversale et permet de tester la maîtrise du candidat sur la résolution concrète d'un problème informatique et de sa capacité à en présenter la solution.

L'épreuve a été largement modifiée cette année comme annoncé sur le site de l'épreuve début 2005. Le jury a examiné cette année 108 candidats[1] répartis en 5 sessions de 3 heures 30 de préparation sur machine suivies de 20 minutes de présentation orale des résultats obtenus. Chaque session était organisée en « pipeline » : les candidats d'une même session commençaient l'épreuve au rythme d'un trinôme toutes les demi-heures, pour un total d'au plus 9 trinômes par session de sorte à ce que les derniers entrent avant que les premiers aient terminé. L'introduction d'un oral a été particulièrement appréciée tant par le jury que par les candidats : les membres du jury ont ainsi pu mieux évaluer les capacités algorithmiques réelles des candidats, ceux-ci ayant encore, par manque de pratique, beaucoup de difficultés à rédiger un algorithme par écrit de façon concise. Les barèmes ont répartis les points à égalité (avec de petites variations d'un sujet à l'autre) entre la partie « pratique » (évaluée par les résultats numériques reportés sur la fiche rendue au début de l'oral) et la partie « algorithmique » évaluée à l'oral lors des réponses aux questions de l'examinateur.

Comme les années précédentes, les candidats avaient le choix entre plusieurs plates-formes de programmation :

- PC sous Windows 98 SE, avec CamlLight 0.74, Maple 7, MuPadLight 2.53, Java ou Pascal (Borland Delphi 6.0).
- PC sous Debian Linux/KDE, avec CamlLight 0.74, Ocaml 3.04 (sous Xemacs/Tuareg ou Glimmer), C (sous Xemacs ou Kdevelop 2.0, compilateur gcc), MuPadLight 2.52, Java et Pascal (XEmacs, compilateur gpc).

À l'exception de Maple 7 et Windows 98, tous ces programmes sont disponibles librement et gratuitement sur Internet, et en particulier sur le site web de l'épreuve donné à la fin du rapport. Ceci est le gage de la meilleure préparation des candidats, qui peuvent ainsi travailler chez eux sans contrainte. Comme l'an passé, il a été demandé aux candidats de choisir l'un des deux systèmes d'exploitation et l'un des langages par avance. Les candidats ont fait les choix suivant : 58,3% CamlLight sous Windows, 22,2% CamlLight ou Ocaml sous Linux, 12,0% Pascal sous Windows, 4,7% C sous Linux, et trois candidats ont choisi Maple sous Windows. Pour la deuxième année consécutive, aucun candidat n'a choisi les langages Java ou MuPad (alternative gratuite à Maple). Au début de chaque épreuve, 10 minutes ont été laissées aux candidats pour se familiariser avec la plate-forme et poser toutes les questions qu'ils souhaitaient.

Quelques remarques d'ordre général :

[1] 4 candidates pour 108 candidats – encore une fois, une proportion anormalement faible, par rapport à la proportion actuelle de 15 % à 20% des chercheuses en informatique en France (typiquement au CNRS).

- Les questions sont calibrées pour qu'elles nécessitent entre un et dix millions d'itérations pour les plus grandes valeurs des paramètres, soit moins d'une seconde d'exécution avec l'algorithme attendu.
- Préférer la structure de tableau à toute autre structure de données lorsque le nombre d'entrées est connu, et n'utiliser les listes chaînées ou les arbres que lorsque les tableaux sont inadaptés (typiquement pour le sujet Généalogie). En effet, la structure de données de tableaux est bien plus facile à maintenir et à contrôler, donc à débugger.

La plupart des questions proposaient plusieurs tailles pour le problème à résoudre. Elles admettaient quasiment toujours un algorithme naïf qui permettait de résoudre les tailles moyennes (par exemple en $O(n^4)$). En revanche seuls les algorithmes efficaces (par exemple en $O(n^2)$) permettaient de résoudre les tailles les plus grandes. Les tailles les plus petites étaient généralement solubles à la main et étaient posées pour permettre (inciter) aux candidats de tester leur programme.

Nous constatons que les candidats sont mieux préparés d'une année sur l'autre, ce qui est le signe que l'épreuve est entrée dans les mœurs. En particulier, la plupart des résultats résolubles à la main étaient corrects. Nous rappelons la nécessité d'apprendre aux candidats à tester et surtout à débugger leurs programmes (par exemple, en ajoutant des commandes d'affichage, ne serait-ce que pour vérifier que l'exécution du programme avance).

Contrairement aux années passées, la longueur des sujets a été largement revue à la baisse pour être maintenant traitable dans une large partie par les meilleurs candidats dans le temps imparti, les examinateurs considérant que les candidats disposent maintenant d'un stock suffisant de sujets pour leur préparation. Voici quelques commentaires sur chacune des épreuves :

- Code à trois adresses : Un sujet qui imposait l'utilisation de structures de données compactes pour trouver les résultats sur les plus grandes valeurs. Toutes les questions ont été traitées par un candidat (au moins).
- Généalogie : Beaucoup des questions de ce sujet se prêtaient mieux à une approche récursive. L'épreuve a d'ailleurs montré que la plupart des candidats maîtrisent bien cette façon de programmer. Les meilleurs ont tout fait jusqu'à la question 13.
- Labyrinthes : Ce sujet avait pour originalité que les données n'étaient pas engendrées aléatoirement mais chargées à partir d'un fichier. Cette opération s'est déroulée sans difficulté et ouvre donc des perspectives de tests des programmes des candidats à partir d'exemples adhoc et non générés à partir de leur suite u0. Certains candidats ont pu répondre correctement aux questions jusqu'à la question 13. La résolution correcte jusqu'à la question 6 permettait d'obtenir la moyenne.
- Jeux et coalitions : Ce sujet reposait essentiellement sur la manipulation correcte d'une structure d'ensemble. Une grande rigueur était requise sur le traitement des cas particuliers (p. ex. « il existe x dans X, tel que P(x) » lorsque X est l'ensemble vide).
- Graphes d'intervalles : Ce sujet offrait aux candidats une grande latitude au niveau algorithmique. Les meilleurs candidats ont traité l'ensemble du sujet.

Pour plus d'informations, le site web de l'épreuve : http://www.ens-lyon.fr/LIP/ConcoursInfo/