
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-Michel Couvreur, Louis Granboulan, 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 et pratique, cette épreuve est donc transversale et permet de tester la maîtrise du candidat sur la résolution concrète d'un problème informatique.

Le jury a examiné cette année 113 candidats¹ répartis en 5 sessions de 4 heures chacune. 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 : 72% CamlLight sous Windows, 13% CamlLight sous Linux, 7% Pascal sous Windows, 3% OCaml sous Linux, 3% C sous Linux, et un candidat a choisi Maple sous Windows. Aucun candidat n'a choisi les nouveaux langages disponibles cette année : Java (pourtant très répandu) et MuPad (alternative gratuite à Maple).

Tous les candidats ont concouru sur des machines « neuves » et strictement identiques : à la fin de chaque épreuve, les machines étaient systématiquement restaurées (reformatées et réinstallées) dans leur état initial. **Nous tenons à rappeler qu'il est strictement interdit d'importer données, fichiers, ou configurations, sous quelque forme que ce soit (disquette, clé USB, ...), dans les ordinateurs.** Au début de chaque épreuve, 15 minutes ont été laissées aux candidats afin de se familiariser avec la plate-forme. Durant ces 15 minutes, les candidats étaient autorisés à poser toutes les questions qu'ils souhaitaient.

Les sujets ont été corrigés avec les mêmes exigences que l'an passé :

- l'épreuve est évaluée *uniquement* à partir des copies rendues par les candidats (réponses aux questions et résultats numériques des calculs). Il n'a jamais été tenu compte ni de la qualité de la rédaction du programme, ni de l'adéquation entre l'algorithme décrit dans la copie et celui effectivement programmé, ni du langage ou du système choisi, ni de la possibilité de rejouer l'exécution des programmes après l'épreuve.

¹ 3 candidates pour 110 candidats – une proportion anormalement faible, par rapport à la proportion actuelle de 15 % à 20% des chercheuses en informatique en France (typiquement au CNRS).

- afin de maintenir l'équité entre les différents choix de langage de programmation, les données pouvaient toujours être codées par des tableaux ou des matrices (qui peuvent être à leur tour codées dans un tableau unidimensionnel comme cela était proposé dans les sujets).
- Les questions sont calibrées pour que l'exécution du programme dure toujours moins de 20 secondes avec l'algorithme attendu.

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 à tester leur programme.

Contrairement à l'an passé, le niveau des candidats était assez homogène, et les candidats étaient mieux préparés. En particulier, la plupart des résultats résolubles à la main étaient corrects. Nous voulons de nouveau insister sur la nécessité d'apprendre aux candidats à tester et surtout à déboguer leurs programmes (par exemple, en ajoutant des commandes d'affichage, ne serait-ce que pour vérifier que l'exécution du programme avance).

Les sujets sont volontairement longs cette année encore afin de donner une large base de travail aux professeurs. Les examinateurs ne s'attendent pas à leur complète résolution dans le temps imparti. Voici quelques commentaires sur chacune des épreuves :

- Frogger : Dans ce sujet la moitié des points portait sur la conception des algorithmes, et l'autre moitié sur la programmation. Un candidat ayant tout fait correctement jusqu'à la question 7 avait 9,5. Un candidat ayant tout fait juste jusqu'à la question 11 avait 20 (un candidat y est presque arrivé). Plusieurs candidats ont eu des difficultés pour calculer le plus court chemin entre deux points dans $\mathbb{Z}/n\mathbb{Z}$.
- Halma : aucun candidat n'a abordé la question 14, deux ont abordé les questions 12 et 13. . Un candidat ayant tout fait correctement jusqu'à la question 10 inclusive aurait eu 22. Un candidat ayant correctement répondu aux 7 premières questions aurait eu 11 points. Un seul candidat a trouvé la meilleure réponse à la question 2.
- Les ascenseurs : aucun candidat n'a abordé les questions 11 et 12, deux ont abordé la question 9 et deux autres la question 10. Un candidat ayant tout fait correctement jusqu'à la question 7 inclusive aurait eu 20. Le candidat ayant correctement répondu aux 4 premières questions a ainsi gagné 11 points.
- Équations diophantiennes : à l'exception d'un candidat ayant traité la question 13, nous avons établi notre barème sur 20 à partir des huit premières questions. Pour obtenir la moyenne, il suffisait de répondre correctement aux cinq premières questions et partiellement à la sixième. Une bonne compréhension des algorithmes de transformation des automates finis (minimisation, union,...) permettait de traiter les deux premières parties du sujet.
- Parcours de graphes : aucun candidat n'a été au-delà de la question 10. Des réponses correctes aux quatre premières questions permettaient d'atteindre la moyenne. Pour avoir une note maximale, il suffisait de répondre parfaitement aux sept premières questions. Cependant, le barème a été établi sur 30 pour tenir compte des rares candidats ayant répondu aux questions 8, 9 et 10. Nous noterons qu'une bonne maîtrise des algorithmes de parcours suffisait pour atteindre une note maximale.

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