
ÉPREUVE ORALE PRATIQUE D'ALGORITHMIQUE ET PROGRAMMATION
ENS : PARIS LYON CACHAN
COEFFICIENTS : PARIS 4 LYON 4 CACHAN 6
MEMBRES DE JURYS : Benoît Caillaud, Arnaud Legrand, Sylvain Pion

Organisation de l'épreuve

Généralités et statistiques

Comme pour les années passées, cette épreuve demandait aux candidats de mettre en œuvre une 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 102 candidats, sur 5 sessions organisées comme les années précédentes en « pipeline » : les candidats d'une même session 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 XP, avec CAML Light, Maple, Java ou Pascal (Delphi).
- PC sous Debian/Linux Gnome, avec CAML Light, Objective CAML, C, C++, Pascal, ou Java.

Il était demandé aux candidats de choisir le langage et l'environnement à l'avance. Encore une fois, CAML Light sous Windows a été très majoritairement choisi : 56,8 % des étudiants ont opté pour cet environnement. 27,5 % des étudiants ont choisi CAML Light ou Ocaml sous Linux, 6 % ont choisi Pascal sous Windows, 4,5 % C/C++ sous Linux. Trois candidats ont choisi Maple cette année et deux d'entre eux ont d'ailleurs brillamment réussi l'épreuve.

Nouveautés

Cette année, une légère modification a été apportée au format des épreuves. Les années précédentes, chaque candidat disposait d'un numéro u_0 propre servant d'entrée à son programme et devait donner les réponses sur une fiche. Cette année, chaque sujet était accompagné d'une fiche réponse "type" correspondant à un $\tilde{u}_0 \neq u_0$. Ainsi, chaque candidat pouvait comparer les réponses produites par son programme avec \tilde{u}_0 avec celles de la fiche réponse "type", ce qui, d'après un certain nombre de candidats, leur a été d'une grande aide. Un des effets visibles de cette aide est qu'à la différence

des années précédentes, la majorité des réponses données par les candidats étaient correctes. Aussi surprenant que cela puisse paraître, certains ont avoué ne pas avoir comparé leurs résultats avec ceux de la fiche réponse. Ils ont d'ailleurs tous eu tout faux dès la seconde question. . . Notons également que trouver les mêmes réponses pour \tilde{u}_0 ne garantit pas pour autant que l'algorithme soit correct et que la réponse sera bonne pour le u_0 des candidats. Quoi qu'il en soit, cela a permis dans l'ensemble de mieux évaluer le niveau des candidats et de distinguer ceux ayant des réels problèmes avec la programmation de ceux qui ont pu commettre de petites erreurs (problème d'indice, légère incompréhension de la question posée, . . .) et ont pu se rattraper grâce à la fiche réponse.

Les sujets ont également été plus « balisés » que l'an dernier. Par exemple, l'an dernier un certain nombre de candidats ne stockaient pas les valeurs de la suite récurrente pseudo-aléatoire de la première question. Du coup, la construction de structures pseudo-aléatoires plus élaborées était de complexité quadratique au lieu de linéaire, ce qui avait gêné un certain nombre de candidats. Beaucoup n'arrivaient même pas à générer des instances de « grande » taille¹. Même s'il est surprenant que les candidats ne réalisent pas d'eux-mêmes ce genre de problèmes (la complexité des algorithmes semblant alors complètement décorrélée du temps d'exécution observé de leur programmes), notre objectif n'est pas de les départager sur ce genre de « détails ». Nous avons donc fortement suggéré dans les sujets que cette suite soit stockée afin que dans la suite les candidats puissent y accéder en temps constant. Certains irréductibles ont néanmoins ignoré ce conseil et persévéré dans leur erreur, ce qui les a empêché de générer les instances de grande taille.

Commentaires généraux sur les candidats

Plusieurs sujets proposaient plusieurs tailles d'instances de problèmes 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). 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. Pour trouver les erreurs d'un programme défectueux, certains candidats n'ont rien fait d'autre que de le lire et le relire encore et encore sans même penser à rajouter quelques commandes affichant le contenu de leurs variables et à observer ainsi le déroulement de leur programme !

Quelques remarques générales :

- Les candidats (surtout sous CAML) utilisent beaucoup la structure de liste chaînée (ou, plus généralement, des types de données récursifs). Si celles-ci peuvent être très utiles, elles peuvent aussi se révéler handicapantes et lourdes à gérer. Dès lors qu'ils ne sont pas adaptés et que le nombre maximum d'éléments est connu à

¹Se référer au rapport de l'an dernier pour plus de détails

- l'avance, il vaut mieux préférer l'utilisation d'un tableau ou d'une matrice.
- 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.
 - Bien des candidats lisent mal les sujets et n'ont pas confronté leurs réponses avec celles de la fiche réponse type. Certains, par exemple, avaient faux, dès la seconde question à cause d'erreurs dans les indices (par exemple $i.(n.j + i - 1)$ au lieu de $n.i + j - 1!!!$).
 - Un certain nombre de candidats perdent du temps par manque de familiarité avec les environnements de programmation proposés. À part Maple, ils sont pourtant tous gratuits et librement téléchargeables sur Internet. Il y a peu de différences entre ces environnements gratuits et ceux que les candidats ont utilisé au cours de l'année, mais elles suffisent à leur faire perdre un temps précieux le jour de l'épreuve. Nous ne pouvons donc que vivement encourager les préparateurs à utiliser les environnements offerts aux candidats passant cette épreuve.
 - Les candidats ne préparent pas suffisamment la partie orale de l'épreuve. Ils disposent pourtant de 3h30 pour se familiariser avec l'ensemble du sujet. Passer une demi-heure à préparer l'oral semble être un bon compromis.

Commentaires spécifiques aux différents sujets

Les taxis tokyotes Hormis la question 3 qui présentait quelques finesses algorithmiques, l'objectif de ce sujet était d'évaluer la capacité des candidats à programmer des algorithmes simples et complètement spécifiés dans le sujet mais dont la mise en œuvre devait être soignée. La question 4 en particulier était très simple et a été correctement traitée par 75% des candidats. La question 5 était un peu plus délicate même si elle ne présentait absolument aucune difficulté algorithmique et n'a été correctement traitée que par 2 candidats ! De même, la question 6 n'a été correctement traitée que par un candidat. Certains candidats ont traité les questions 5 et 6 à la main pour la plus petite instance. Aucun candidat n'a abordé les questions 7 et 8.

La distinction entre les différents candidats s'est donc faite sur leur compréhension de la fameuse question 3. À peu près les deux tiers des candidats sont passés complètement à côté de la difficulté de cette question et ne l'ont réalisée qu'à l'oral. Ils ont du coup perdu beaucoup moins de temps sur cette question durant la partie pratique que ceux qui avaient réalisé qu'elle était délicate, le plus souvent sans pouvoir en tirer parti.

Vestiges troglodytiques Ce sujet portait sur l'exploration d'un graphe valué avec une autonomie limitée et sur une extension d'un problème d'empaquetage. À part les questions 4, 7, 10 et 11 qui nécessitaient effectivement un parcours dans un arbre,

toutes les autres questions ne posait aucune difficulté et revenaient à un simple parcours de tableau, éventuellement trié auparavant). La question 4 demandait un parcours dans un arbre et a été bien traitée dans l'ensemble. Les deux tiers des candidats ont donc répondu correctement jusqu'à la question 6 mais la plupart des candidats se sont arrêtés à la question 7 qui constituait la difficulté algorithmique principale de ce sujet. Aucun n'a abouti même si certains candidats ont eu quelques idées pour y répondre. En particulier, le terme de programmation dynamique a été évoqué par un certain nombre de candidats alors que cette technique n'était absolument pas nécessaire ici. Elle pouvait être utilisée mais avait alors une complexité dépendant de la longueur des arêtes alors qu'on pouvait obtenir une bien meilleure complexité ($O(n^2)$ au total) en fusionnant récursivement les supports des fonctions λ . Certains candidats l'ont résolue à la main pour les petites instances. Seul un petit nombre de candidats a tenté sa chance et abouti pour les questions 8 et 9 alors qu'elles étaient particulièrement faciles.

Dioclétien contre les barbares en Pannonie Ce sujet portait sur le calcul de la couverture d'un graphe non orienté, d'abord de façon exacte, puis de façon approchée. La principale difficulté était l'énumération des parties de taille k de l'intervalle $\{1, \dots, n\}$ selon un ordre prédéfini. Beaucoup de candidats généraient toutes ces parties dans une liste pour ensuite les examiner et éventuellement construire les parties de taille $k + 1$. Soixante pourcent des candidats ont réussi à écrire un algorithme permettant d'énumérer ces parties, mais pour les deux tiers d'entre eux, la complexité en espace de leur algorithme ne leur permettait pas de résoudre toutes les instances. Notons que les u_0 donnés aux candidats avaient spécialement été choisis de façon à ce que le temps d'exécution total pour résoudre l'intégralité du sujet n'excède pas dix secondes. Près de 40% des candidats ont réussi à calculer les composantes connexes de la question 5. Enfin, 2 candidats ont abordé avec succès la partie 6 et seulement 7 candidats sont venus à bout de la partie 5 qui était pourtant relativement facile, puisqu'il s'agissait du calcul d'une approximation.

Bonnie & Clyde Ce sujet proposait une illustration du théorème de graphe « max flow = min cut ». Il n'était absolument pas nécessaire de démontrer ni de comprendre ce résultat pour le programmer puisque tous les algorithmes étaient détaillés. Du point de vue de la programmation, tout se faisait à l'aide de matrices et l'ensemble du sujet ne présentait aucune difficulté particulière. Les candidats de cette journée ont été pourtant moins performants que pour les autres jours, la moyenne de cette journée étant de 3 points inférieure à celle de la moyenne globale. Seul 4 candidats ont dépassé la question 2 !

Jeux combinatoires Ce sujet portait sur le calcul de stratégies optimales pour des jeux déterminés. Il comportait deux parties indépendantes : la première portait sur l'étude d'un jeu asymétrique sur des arbres, et la seconde était un jeu sur des nombres entiers. La difficulté de la première partie était d'arriver à déterminer la stratégie optimale. 28% des candidats ont trouvé cette stratégie et l'ont mise en œuvre. La partie 2 était beaucoup plus facile. La seule subtilité était qu'il ne

fallait surtout pas utiliser un algorithme récursif ne mémorisant pas les résultats intermédiaires sous peine d'avoir une complexité en temps de l'ordre de $(\sqrt{n})^n$ au lieu de $n\sqrt{n}$.

On peut trouver les 5 sujets proposés cette année sur le site Web de l'épreuve :

<http://perso.bretagne.ens-cachan.fr/~dit/concoursEP.php>