EPREUVE : ECRIT MATHS INFO

ENS : PARIS – LYON - CACHAN

Durée : 4 heures ***Coefficients*** **: PARIS 4 – LYON 4 – CACHAN 5**

MEMBRES DE JURYS : J. GOUBAULT-LARREC, J. MAIRESSE

Cette épreuve de « mathématiques-informatique » en est à sa deuxième édition (première pour Cachan). Rappellons que quelques éléments généraux sur l'épreuve figurent dans le rapport de l'an dernier.

L'épreuve de cette année portait sur la combinatoire des groupes symétriques, l'un des objectifs principaux étant la démonstration de la correspondance de Robinson-Schensted. La première partie introduit la notion de plus longue sous-suite d'une permutation et propose une version allégée de l'algorithme de Robinson-Schensted sous forme de réussite (jeu de cartes à un joueur). La seconde partie introduit les tableaux de Young et l'algorithme proprement dit. La troisième partie s'intéresse aux représentations linéaires du groupe symétrique. La dernière question permet d'unifier le sujet en explorant quelques aspects élémentaires des liens entre ces représentations et les tableaux de Young.

Cette épreuve s'est avérée très sélective. Traiter tout le sujet dans le temps imparti n'était certainement pas réalisable. Par contre, le sujet permettait plusieurs parcours, avec notamment une troisième partie indépendante à l'exception de la dernière question. Les tous meilleurs arrivent à traiter plus des 2/3 du sujet et obtiennent 20. À l'inverse, beaucoup sont incapables de s'abstraire du formalisme et ne dépassent pas la question 1.3. Environ 1/3 des candidats comprennent une partie du sujet et répondent à au moins une question difficile, obtenant 8 et plus. La moyenne est obtenue par 15% des candidats.

Remarques générales et faits saillants

Comme toujours, prime a été donnée aux candidats sachant répondre aux questions difficiles au détriment de ceux qui se contentent de répondre à des bouts de questions élémentaires éparpillés tout au long du sujet. (Voir aussi les commentaires des questions 1.3 et 2.3.) En guise d'illustration, un candidat répondant parfaitement aux questions 1.3, 1.4 et 1.5 obtenait la moyenne.

Deux questions exigeaient l'écriture d'un algorithme (1.5 et 2.1). Contrairement à il y a 3/4 ans, la plupart des candidats tentent de répondre à ces questions, ce dont on ne peut que se féliciter. Par contre, une tendance fâcheuse et persistante est de croire qu'un programme se suffise à lui-même : de nombreux candidats fournissent des programmes bruts de fonderie, non commentés, non justifiés, et donc essentiellement illisibles. À l'opposé, certains croient qu'il suffit d'expliquer le principe de l'algorithme, sans écrire de pseudocode. Ceci n'est pas non plus acceptable, car le but de ces questions est de tester si le candidat sait décrire un algorithme en utilisant les constructions fondamentales d'affectation, de boucle, etc. présentes dans les langages de programmation usuels. (Il était de plus expressément précisé dans le premier paragraphe de l'énoncé quel style de programmes était attendu.)

La plus grosse surprise a été l'incapacité de la majorité des candidats à écrire la récurrence permettant d'énumérer le nombre de partitions d'un entier (question 2.1). La question était posée sous la forme de la demande d'un algorithme. Est-ce cela qui a induit les candidats à se dispenser de démontrer les formules avancées ? S'ils avaient sérieusement essayé de les justifier, on imagine que de nombreux candidats auraient réalisé qu'elles étaient erronées !

Terminons avec un extrait du rapport de l'an dernier : *Les « arnaques » sont presque irrémédiablement décelées et ont le don de faire perdre au correcteur son a priori de bienveillance.*

Commentaires sur chaque partie

Les trois parties du sujet étaient complémentaires mais de natures assez différentes.

Partie 1. Pour être bien traitée, celle-ci supposait une certaine familiarité avec les groupes mais surtout la capacité à formaliser mathématiquement et algorithmiquement des intuitions, parfois subtiles, de nature "discrète" ou "graphique". On avait volontairement séparé les questions mathématiques (1.3, 1.4, 1.6) et les questions algorithmiques (1.5, 1.7). De fait, les questions 1.4-1.5 et 1.6-1.7 fonctionnaient de façon couplée. En 1.4 (resp. 1.6), il s'agissait de démontrer une propriété et en 1.5 (resp. 1.7) d'écrire (resp. d'analyser) l'algorithme correspondant. Cette approche s'est avérée très instructive et sélective. Ainsi de nombreux candidats ont su répondre à 1.4 mais pas à 1.5 et réciproquement. Seuls les très bons candidats se sont montrés également à l'aise avec les deux types de question.

Partie 2. À l'opposé, les questions de cette partie étaient de nature *à la fois* mathématique et algorithmique. Le point culminant était la question 2.5 ou pour démontrer une bijection entre deux ensembles, il fallait calculer l'application inverse sous forme d'un algorithme. Formellement, cette partie était peut-être plus facile que la précédente mais elle demandait incontestablement un peu d'initiative et d'inventivité de la part des candidats. Répondre aux questions nécessitait des allers-retours entre des arguments abstraits et formels, une intuition "géométrique" et une compréhension algorithmique de l'objet étudié. Peu de copies abordent le cœur de cette partie, néanmoins certains la traitent entièrement et remarquablement.

Partie 3. Cette dernière partie était de nature beaucoup plus traditionnelle, les candidats retrouvant le langage rassurant de l'algèbre linéaire ainsi que ses modes de raisonnement. De très nombreux candidats abordent cette partie. Cependant, fatigue accumulée ou relative difficulté des questions, très peu s'en sortent honorablement. Le dernière question, plus combinatoire, permettait de faire le lien avec les parties précédentes (pour le plus grand plaisir des concepteurs, à défaut de celui des candidats, la question n'étant traitée dans aucune copie).

Questions détaillées, erreurs typiques

Questions 1.1. et 1.2. Questions faciles et bien traitées. L'objectif est de familiariser le candidat avec les objets à étudier tout en le mettant en confiance. On observe toutefois quelques réponses fantaisistes. Pour 1.1, un candidat invoque le théorème d'inertie de Sylvester. Pour 1.2, un certain nombre de candidats "démontrent" $U((i,j)) = |j-i|$.

Questions 1.3. Première question plus difficile. La plupart des candidats démontrent $U(\sigma) + L(\sigma) \geq n$ mais ne trouvent pas l'autre inégalité. Il est bien compréhensible qu'un candidat qui a passé un certain temps à prouver tout ce qui était balisé (ici, $L(\sigma) + U(\sigma) \geq n$, pour lequel une indication était donnée), croie qu'il a répondu à une grande partie de la question. Or il est dans l'esprit des épreuves des ENS que ce sont les parties non balisées, nécessitant davantage d'invention, qui seront naturellement dotées du plus grand nombre de points. À noter qu'ici la partie non balisée, $L(\sigma) + U(\sigma) \leq n$, n'était en fait pas spécialement difficile, et que la réponse demandée n'était pas spécialement longue. Autre point : dans la démonstration de $|L(\sigma c_{ij}) - L(\sigma)| \leq 1$, beaucoup s'embrouillent dans les indices et proposent une page ou plus d'arguments confus. Ceux qui s'en sortent bien sont ceux qui utilisent explicitement l'intuition : *l'effet de c_{ij} est de déplacer une carte de la position i à la position j.*

Question 1.4. Il fallait montrer séparément $L \geq C$ et $L \leq C$ où L était la longueur de la plus longue sous-suite croissante et C le nombre de colonnes de la réussite. La partie difficile était de montrer la deuxième inégalité. Une moitié des candidats a trouvé, sous une forme plus ou moins claire et rigoureuse, une des idées nécessaires. Les candidats ont été notés aussi bien sur leur aptitude à trouver ces idées que sur leur capacité à en déduire une démonstration correcte. Certains candidats ont insisté pour prouver que la ligne du bas formait une sous-suite croissante maximale, alors que l'énoncé les prévenait que ça n'était pas le cas pour la permutation donnée en exemple !

Question 1.5. La majorité des candidats ont fourni des programmes, allant du plus approximatif au plus détaillé. Remarquons qu'aucune forme d'optimisation du programme n'était requise. Il était parfaitement acceptable de proposer un programme construisant *toutes* les sous-suites croissantes pour déterminer la plus longue. Cela permettait en particulier de répondre à cette question sans avoir traité la précédente. Seules 3 ou 4 copies pensent à utiliser cette possibilité.

Question 1.6. La bonne approche consiste à démontrer séparément et avec des arguments différents les inégalités $K \geq \text{Inv}$ et $\text{Inv} \geq K$. Beaucoup de candidats tentent une démonstration directe et par récurrence de l'égalité, sans succès. On observe un florilège de raisonnements faux où les candidats font semblant de croire que le groupe symétrique est commutatif (du style : si $\sigma = g_1 \cdots g_k$ et qu'il existe $g_i = g_j$ alors les deux transpositions s'annulent et il existe une décomposition de longueur $k-2$). Très peu de candidats pensent à s'inspirer de la question 1.3 en étudiant $\text{Inv}(\sigma(i, i+1)) - \text{Inv}(\sigma)$.

Question 1.7. Beaucoup de candidats se sont contentés de démonstrations trop approximatives du fait que $\mathcal{K}$-MÉLANGE calculait bien $K(\sigma)$. Un point important est que la question 1.6 était cruciale : il est facile de voir que $\mathcal{K}$-MÉLANGE calcule $\text{Inv}(\sigma)$ et donc $K(\sigma)$ par la question 1.6. Démontrer directement que $\mathcal{K}$-MÉLANGE calcule $K(\sigma)$ semble irréaliste.

De nombreux candidats ont aussi traité à la légère la question de savoir si l'algorithme modifié calculait bien $U(\sigma)$. Une bonne moitié de ceux qui abordent ce point répondent d'ailleurs "oui". D'autres répondent "non" sans donner de contre-exemple explicite, typiquement en arguant du fait qu'il n'y a pas de moyen de démontrer que l'algorithme modifié calcule $U(\sigma)$; ceci n'est évidemment pas un argument acceptable. Une fois encore, l'exemple $\sigma = 7, 2, 3, 6, 1, 5, 4$ de l'énoncé fournissait un contre-exemple ; et ce n'était évidemment pas le contre-exemple le plus simple.

Plus généralement, cette permutation σ était essentiellement un contre-exemple à toutes les idées fausses que l'on pouvait avoir dans les parties 1 et 2. Bien peu de candidats ont su l'exploiter.

Question 2.1. Parmi les copies abordant la question, seules 1/3 donnent une réponse correcte, à notre grande stupéfaction. Il s'agissait dans notre esprit d'une question facile pour introduire la partie. On a observé une incroyable diversité de récurrences fausses, dont certaines affirmées péremptoirement sans aucune justification. De nombreux candidats ont cherché à définir un algorithme récursif directement, sans passer par un algorithme à deux arguments. C'était sans espoir. Quelques-uns n'écrivent même pas de récurrence et assènent que la solution est 2^n ou n !... Deux copies commencent par décomposer n en facteurs premiers (confusion entre $\sum \lambda_i = n$ et $\prod \lambda_i = n$?).

Question 2.2. Un nombre étonamment élevé de candidats n'ont pas vu qu'au niveau de la première ligne, le mécanisme était exactement celui de la réussite de la Partie 1. De fait, la raison d'être de cette réussite et de cette Partie 1 était d'introduire de façon progressive les difficultés inhérentes à l'algorithme de Robinson-Schensted ! La plupart des candidats ont cherché à raisonner directement sur le tableau de Young. Ceci menait à une argumentation complexe et confuse. Un seul candidat a réussi dans cette voie, et a reproduit l'argument des pointeurs de la question 1.4 en l'adaptant à cette question.

Question 2.3. La grande majorité des candidats n'a pas traité l'égalité $P(\overline{\sigma}) = P(\sigma)^t$, sans doute pressée par le temps et cherchant à obtenir des points sur la deuxième partie, facile, de la question. En l'occurrence, la deuxième partie ne comptait que pour un huitième des points de la question. Observons que la première partie demandait un peu d'astuce et de réflexion mais se rédigeait extrêmement rapidement une fois trouvée la bonne idée.

Question 2.4. Peu de candidats ont réalisé que répondre à la question revenait à montrer qu'on ne peut pas loger n^2 cases dans un rectangle dont les deux côtés sont de longueur strictement inférieure à n. La deuxième partie de la question est traitée par une proportion raisonnable des candidats.

Question 2.5. Très peu de candidats ont vu que l'on pouvait définir l'algorithme RS^{-1}, mais ceux qui l'ont vu ont en général démontré correctement qu'il était inverse de l'algorithme RS. La deuxième partie, facile, rapportait au mieux un dixième des points de la question.

Question 2.6. Une bonne surprise a été de constater que, parmi les candidats qui se sont attaqués à la démonstration de l'égalité de droite, la plupart ont effectué un raisonnement correct.

Question 3.1. Pratiquement tous les candidats abordant la question ont vu que l'application constante 1 et la signature étaient deux représentations possibles. Beaucoup de candidats n'ont pas réussi à montrer que $\Psi((i,j))$ était le même pour tous les (i,j) et donc qu'elles étaient les deux seules représentations ; c'était évidemment l'essentiel de la question.

Question 3.2. De nombreux candidats ont prouvé la bilinéarité au lieu de la sesquilinéarité, ou la symétrie au lieu de $\langle v,u\rangle = \overline{\langle u,v\rangle}$. Bizarrement, la plupart des candidats n'ont pas vu l'évidence : si l'on demandait de montrer que $\langle\cdot|\cdot\rangle$ était un produit scalaire, c'était pour fabriquer l'orthogonal de E_1 dans la suite de la question.

Question 3.3. L'irréductibilité de S_H était très probablement la question la plus complexe de toute l'épreuve. Elle n'a été traitée par aucun candidat. Le cas $x_i \neq x_j$ pour tous $i \neq j$, plus simple, a été traité dans deux copies.

Question 3.4. Seuls deux candidats ont attaqué (très partiellement) cette dernière question.