

ÉPREUVE DE TIPE D'INFORMATIQUE 2010

ÉNS : CACHAN - LYON - PARIS

MEMBRES DU JURY : S. LOMBARDY, N. SCHABANEL, P. SENS

Le jury a évalué cette année 80 candidats présentant un dossier TIPE d'informatique (contre 78 candidats en 2009, 63 en 2008, 70 en 2007, 67 en 2006, 84 en 2005 et 67 en 2004). De même que les années précédentes, de nombreux candidats ont obtenu de bonnes notes en présentant de bons travaux éloignés du thème (cette année, les surfaces). Le jury perçoit ce thème annuel comme une source bienvenue de renouvellement des TIPE. Les candidats ont été évalués sur leur maîtrise des concepts informatiques généraux, sur la qualité scientifique de leur travail, et surtout sur leur maîtrise et leur compréhension de leur sujet. Compte tenu de la durée de l'épreuve (40 minutes), le jury peut évaluer précisément ce dernier point. Pour en juger, **le jury s'autorise à poser des questions sortant du cadre strict du travail réalisé**. Il pourra par exemple être demandé au candidat de réfléchir en direct à des variantes ou extensions du problème étudié. La réactivité et la capacité de propositions du candidat seront alors évaluées. Lorsque le candidat utilise des notions hors du programme des CPGE, il doit s'attendre à des questions sur celles-ci. L'épreuve des TIPE doit être vue comme **un oral à part entière portant sur un sujet choisi et spécialement préparé par le candidat**. Le choix du sujet est ainsi particulièrement important et doit permettre au candidat de mettre en valeur ses capacités créatives, la rigueur de son approche et son esprit critique.

Le jury tient à souligner que le niveau des candidats auditionnés s'est significativement amélioré depuis 2007. De nombreux oraux étaient de niveau très satisfaisant et plusieurs de niveau excellent. Soulignons également que de nombreux candidats se sont révélés lors des questions.

Nous constatons qu'une proportion non négligeable de candidats (une dizaine encore cette année), sans doute à leur corps défendant, se sont trompés de discipline pour leur TIPE. **Chaque TIPE doit impérativement répondre à un questionnement informatique**. Voici quelques exemples qui pourront aider les candidats et leurs encadrants à faire le bon choix. Un TIPE étudiant un phénomène physique au comportement sophistiqué, mais où la part informatique est réduite à l'écriture d'un programme consistant en une ou deux boucles `for` simulant les équations différentielles décrivant ce phénomène physique, place immédiatement le candidat dans une situation délicate : la part informatique ne présente aucune difficulté et le jury est incompetent pour juger de la qualité scientifique de son étude du phénomène physique. En revanche, si par exemple, la simulation requiert l'utilisation de structures de données sophistiquées et essentielles pour l'efficacité du programme, c'est cette partie (plutôt que l'étude du phénomène) qu'il convient de développer. De même, la part informatique ne peut se réduire à une optimisation *ad hoc* d'un code sur un processeur donné, à moins que les méthodes proposées ne soient présentées dans une problématique générale (celle de la compilation dans cet exemple).

L'épreuve orale se déroulait ainsi : il est demandé au candidat de faire une brève présentation synthétique du sujet choisi (au tableau, ou à l'aide de transparents si le candidat le souhaitait), au cours de laquelle le candidat est amené à répondre aux questions des membres du jury. Voici une liste de situations typiques : le candidat utilise une notion complexe et le jury lui propose de la définir ; le candidat décrit un algorithme et le jury lui demande d'en évaluer la complexité en temps ou bien de prouver son bon fonctionnement ;¹ le candidat utilise une structure de données

1. La preuve du bon fonctionnement de tout algorithme présenté doit pouvoir être donnée par le candidat (au moins dans les grandes lignes si celle-ci est difficile ou fait appel à des notions hors programme).

classique et le jury l'interroge sur les algorithmes classiques sur cette structure de données ; un candidat expose sa solution au problème qu'il s'est posé, et le jury le guide vers des structures de données plus performantes pour résoudre son problème. Chaque fois que le sujet s'y prête, un travail expérimental et une réalisation logicielle sont attendus ; le candidat est alors amené à commenter ses programmes et à présenter une copie du code source au jury. Notons que le jury attend en terme d'implémentation plus que de simples appels à des bibliothèques préexistantes (solveur de programmation linéaire, simulateur d'automates cellulaires,...).

Nous encourageons vivement chaque candidat à préparer un jeu de transparents (prévu pour une durée de 5 à 10 minutes maximum) qui lui servira de guide tout au long de l'oral.²

Nous avons apprécié l'originalité de certains sujets traités ou la démarche créative de certains travaux. Nous avons été surpris que certains candidats se contentent de restituer des connaissances acquises dans un livre ou sur Internet. **Une réflexion personnelle doit s'élaborer autour de ces connaissances, avec un esprit critique.** À l'inverse, l'absence de documentation a conduit certains candidats à un travail d'une grande naïveté.

Certains sujets semblent maintenant trop balisés pour permettre des développements suffisants, en voici une courte liste. L'écriture d'un simple algorithme de *backtracking* (pour résoudre un problème d'optimisation) ne peut être l'unique objet d'un TIPE, à moins bien sûr que celui-ci requiert la mise en place de structures de données sophistiquées, ou une induction compliquée, ou une preuve de terminaison originale, ou bien encore une analyse intéressante de sa complexité en temps. On pourra aussi s'intéresser à en améliorer les performances par des méthodes de type *branch-and-bound* où le choix des bornes serait un des objets d'étude du TIPE. Un candidat qui fait le choix de résoudre d'un problème par algorithmes génétiques, ne doit pas tomber dans le piège de l'« ordinateur qui réfléchirait à la place de l'humain ». ³ Les algorithmes génétiques sont en effet un paradigme de programmation comme un autre, où le vrai programme est codé dans le choix des fonctions de *fitness*, de mutation et de croisement. C'est évidemment sur le choix de ces trois fonctions que le débat doit porter. Il convient également de comparer les performances obtenues par ce type d'approche avec celles des autres paradigmes plus classiques. Nous regrettons également que les candidats ne pensent pas systématiquement à l'encodage par des entiers (peut-être même à des fonctions de *hash*) pour les tests d'égalité répétitifs d'objets structurés. Nous encourageons les candidats à approfondir l'analyse de la complexité des algorithmes présentés. Assez souvent, la complexité annoncée ne correspond pas à celle effectivement implémentée du fait d'un mauvais choix des structures de données : par exemple, aucun des algorithmes présentés pour la triangulation de Delaunay n'avait la complexité annoncée $O(n \log n)$.

Notons que le jury n'a aucune objection à ce que le TIPE soit préparé à plusieurs comme cela se pratique couramment. En revanche, il convient que le candidat maîtrise parfaitement l'ensemble de ce qu'il présente, quitte à ne présenter que la partie du TIPE à laquelle il a réellement contribué. Ceci ne pose aucun problème tant que cette partie est suffisamment consistante.

Nous déplorons enfin qu'aussi peu de candidates présentent un TIPE d'informatique (une seule cette année), une proportion anormalement faible par rapport à celle des chercheuses en informatique (supérieure à 25%).

2. Les candidats repartent avec leurs transparents à la fin de l'oral.

3. Des propos tenus par plusieurs candidats durant ces deux dernières années (également à propos des algorithmes de *backtracking* α - β d'ailleurs).