

ÉPREUVE DE TIPE D'INFORMATIQUE 2013

ÉNS : CACHAN - LYON - PARIS

	ENS Cachan	ENS Lyon	ENS Paris
Coefficients :	Concours MP : 2	1,5	8
	Concours Info : 3	1,5	1

MEMBRES DU JURY : A. PECHER, N. SABOURET, O. SERRE

Le jury a évalué cette année 92 candidats présentant un dossier TIPE d'informatique (contre 88 candidats en 2012, 83 candidats en 2011, 80 candidats en 2010, 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, *Invariance*, *Similitude*). 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, son esprit critique et sa capacité à s'impliquer dans un projet qui le motive.

Le jury tient à souligner que le niveau des candidats auditionnés s'est significativement amélioré depuis 2007. De très 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 que, cette année encore, une proportion non négligeable de candidats (une demi-douzaine) 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). De manière générale, le choix de présenter le TIPE en informatique ne peut pas se baser uniquement sur le fait qu'il y a eu un travail de programmation : nous attendons des candidats un réel questionnement sur les algorithmes et les structures de données utilisés.

Nous constatons que de trop nombreux rapports divergent significativement des recommandations en dépassant largement les 6 pages. **Ceci ne peut être toléré** car cela induit non seulement un biais entre les candidats mais pose également au jury un problème de temps de lecture du rapport. Nous demandons aux candidats de **ne pas faire figurer le code développé**

à l'intérieur du rapport lui-même, mais de le mettre systématiquement en annexe. Dans la plupart des cas, ces longs rapports sont issus de collaboration avec d'autres candidats. Nous recommandons aux candidats de se focaliser sur la partie qu'ils ont développée eux-même, dans le rapport comme à l'oral. Notons également qu'en général, les candidats qui présentaient de longs rapports se sont retrouvés piégés en ne réussissant pas à terminer leur présentation ni à présenter la partie la plus avancée de leur travail.

Nous rappelons également que le candidat doit être capable de refaire tout ce qui figure dans son rapport : de nombreux candidats se sont retrouvés en difficulté car ils avaient survolé des points essentiels dans les parties *préliminaires* de leur travail ; embourbés dans une maîtrise approximative de ces points préliminaires, ils n'ont pas pu présenter le cœur de leurs travaux.

L'épreuve orale se déroule ainsi : il est demandé au candidat de faire une brève présentation synthétique du sujet choisi (de préférence à l'aide de transparents), 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 fonctionnement¹ ; le candidat utilise une structure de données 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 (nous recommandons vivement que ce code source soit joint en annexe à la version électronique du rapport). 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. Les candidats repartent avec leurs transparents à la fin de l'oral. Il n'est pas possible d'utiliser des supports numériques. **Si, faute de temps, le candidat ne peut présenter l'intégralité de ses travaux, il est préférable qu'il se focalise sur ses contributions personnelles qu'il estime les plus pertinentes.**

Nous avons apprécié l'originalité de certains sujets traités ou la démarche créative de certains travaux. Il n'est pas acceptable 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 : les connaissances minimales attendues d'un sujet sont au moins celles qui figurent dans une encyclopédie (comme par exemple, les pages wikipedia consacrées à ce sujet).

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. Ainsi un TIPE ne peut se résumer à de la programmation uniquement. 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 un problème en utilisant des méthodes à base de méta-heuristiques (algorithmes génétiques, colonies de fourmis...) ou des techniques d'apprentissage automatique (réseaux de neurones...)

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).

ne doit pas se limiter à la mise en œuvre de la méthode². Les algorithmes génétiques, par exemple, sont 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, en relation avec le problème à résoudre. **Il est également important que le candidat s'interroge préalablement sur l'existence d'algorithmes classiques efficaces (et exacts) pour résoudre le problème considéré** (c'est par exemple le cas pour la recherche de chemin dans un graphe...) et soit capable de comparer les performances obtenues par les approches heuristiques (par exemple des algorithmes de colonies de fourmis) avec celles des autres paradigmes plus classiques. Le jury attend une analyse fine des différentes options et de leurs impacts sur les performances et non la simple écriture d'un simulateur inadapté au problème étudié. De même, lors de la présentation d'un algorithme d'exploration de type α - β , le jury attend du candidat une compréhension précise du sens de ces valeurs, de leur rôle et de leur utilisation.

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 $O(n \log n)$. Nous recommandons aux candidats de faire également des estimations du temps de calcul de leur algorithme en partant sur une base de $\sim 1.000.000.000$ d'opérations par secondes (correspondant grosso-modo au microprocesseur actuel de fréquence 1GHz), cela leur permettra de mieux comprendre les parties de leur programme qui pourraient être à l'origine des mauvaises performances.

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 en informe explicitement le jury lors de son audition et 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. De manière générale, le recul du candidat sur son sujet et sa culture générale en informatique sont des éléments appréciés par le jury et qu'il cherchera systématiquement à évaluer à travers les questions sortant du cadre strict de l'étude présentée.

Comme chaque année, nous encourageons les enseignants à diriger les candidates vers des sujets d'informatique. En effet, cette année encore, la proportion de candidates (sept cette année) présentant un TIPE d'informatique est anormalement faible au regard de celle des chercheuses en informatique (supérieure à 25%).

2. Des propos tenus par plusieurs candidats durant ces quatre dernières années tendent à confondre Intelligence Artificielle et l'« ordinateur qui réfléchirait à la place de l'humain ».