

Banque MP Inter-ENS - Session 2021

Rapport sur l'épreuve de TIPE d'informatique

ENS (Paris) - ENS Lyon - ENS Paris-Saclay – ENS Rennes

Coefficients (en pourcentage du total d'admission) :

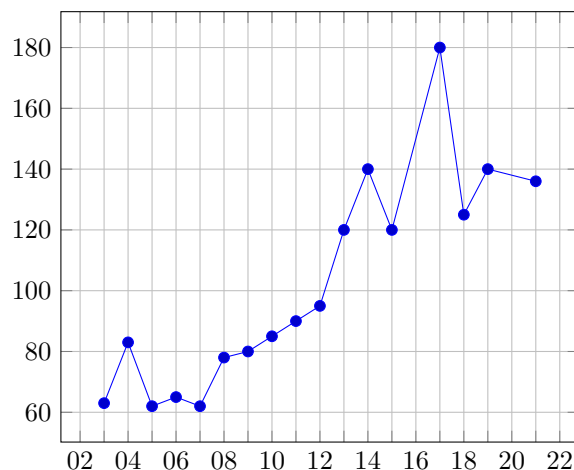
	ENS Lyon	ENS (Paris)	ENS Paris-Saclay	ENS Rennes
Concours MP :	5.4%	7.4%	3.8%	3.8%
Concours Info :	5.6%	3.3%	7.9%	11.4%

Membres du jury : P. BOUYER-DECITRE, J-L. DESSALES, É. FABRE, M. RAO

Le jury a évalué cette année 136 candidats présentant un dossier TIPE d'informatique.

Le nombre de candidates (9 cette année) présentant un TIPE d'informatique reste anormalement faible au regard de la proportion des chercheuses en informatique (supérieure à 25%). Nous encourageons vivement les enseignants à diriger les candidates vers des sujets d'informatique.

Figure 1 – Nombre de candidats en TIPE d'informatique, par année.



Comme les années précédentes, le jury a apprécié la qualité d'un nombre important de 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 du sujet choisi. Le jury tient à souligner que le niveau des candidats auditionnés s'est significativement amélioré depuis une dizaine d'années. De très nombreux oraux étaient de niveau très satisfaisant et plusieurs de niveau excellent ; de nombreux candidats se sont révélés

lors des questions. Le jury a particulièrement apprécié l'enthousiasme, la créativité et la curiosité de certains candidats, et les risques pris en explorant des thématiques nouvelles en lien avec les évolutions récentes de l'informatique. Toute la palette des notes a été utilisée, et le jury n'a pas hésité à mettre plusieurs fois la note maximale.

Pour illustration, et sans que cela soit une recommandation pour les éditions futures du concours¹, le jury a particulièrement apprécié cette année les sujets suivants

- l'écriture de *smart contracts* dans une *blockchain*
- la vérification de tautologies dans le calcul des séquents
- l'étude d'algorithmes pour les ordinateurs quantiques
- les schémas d'attaque contre des fonctions de hashage
- l'étude de la résilience de réseaux de distribution d'électricité
- des structures de données astucieuses pour la programmation compétitive
- les schémas d'attaque pour la cryptographie par courbes elliptiques
- les inter-réductions entre problèmes de Karp distants
- en synthèse d'image, la production d'effets d'atmosphères autour d'une planète, inspirés de la physique du rayonnement
- le calcul en précision arbitraire sur les flottants
- la construction d'un interpréteur Prolog
- ...

L'épreuve orale, d'une durée de 40 minutes se déroule de la façon suivante : le candidat a tout d'abord l'opportunité de présenter de manière synthétique son travail. Pour ce faire, le candidat peut, s'il le souhaite, s'aider d'un support visuel par ordinateur (diaporama en format PDF). L'idée est de se baser sur une présentation d'environ 10 minutes; même si les questions du jury viendront systématiquement interrompre l'exposé, il est recommandé de le préparer et de le répéter sur cette base. Bien qu'optionnelle, le jury a constaté que cette présentation permettait aux candidats de mieux valoriser leur travail et les aider à structurer leurs réponses aux questions en servant de point de départ pour la discussion avec le jury. En effet, pour cette épreuve, le candidat est amené à répondre aux multiples questions des membres du jury, qui portent soit directement sur le travail présenté (pour éclaircir des points techniques, par exemple), soit sur des sujets connexes (par exemple, des approches alternatives ou des notions de base en informatique). De manière générale, le recul du candidat sur son sujet et sa culture générale en informatique sont des éléments importants que le jury cherche systématiquement à évaluer à travers des questions sortant du cadre strict de l'étude présentée. Le candidat peut, à tout moment, consulter ses notes pour s'aider dans ses réponses aux questions du jury. Dans ses interactions le jury s'adapte au niveau du candidat, ce qui implique que si ce dernier souhaite utiliser des concepts qui sont hors du programme (par exemple la NP-complétude ou autre), le jury vérifiera systématiquement les connaissances du candidat sur ces points particuliers pour lesquels il sera attendu plus qu'une connaissance superficielle.

¹Pour rappel, au-moins 30% des membres du jury sont renouvelés chaque année.

Voici une liste de situations typiques : le candidat utilise une notion complexe et le jury lui propose de la définir précisément ; le candidat décrit un algorithme et le jury lui demande d'en évaluer la complexité en temps ou bien de prouver son bon fonctionnement² ; 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. Notons que le jury attend en termes d'implémentation plus que de simples appels à des bibliothèques préexistantes (solveur de programmation linéaire, simulateur d'automates cellulaires,...).

Le jury s'autorise à poser des questions sortant du cadre strict du travail réalisé. Par exemple, il pourra demander au candidat de réfléchir en direct à des variantes ou extensions du problème étudié. La réactivité et la capacité de proposition du candidat seront alors évaluées. L'épreuve des TIPE doit être vue comme un **oral à part entière portant sur un sujet choisi et spécialement préparé par le candidat**. Enfin si des expériences sont les bienvenues elles doivent impérativement s'inscrire dans une démarche scientifique construite : les impressions subjectives du candidat ne peuvent constituer une justification suffisante, et le jury attend une analyse critique des résultats obtenus comme du cadre expérimental.

Nous encourageons les candidats à approfondir l'analyse de la complexité des algorithmes présentés. Trop souvent, la complexité annoncée ne correspond pas à celle effectivement implémentée du fait d'un mauvais choix des structures de données, ou d'une vision naïve du fonctionnement du langage de haut niveau utilisé. Dans les problèmes où des nombres flottants sont utilisés, nous nous attendons également à une réflexion sur les problèmes de précision liés à l'implantation de ces flottants. Nous regrettons également que les candidats ne pensent pas systématiquement à l'encodage par des entiers (peut-être même à des fonctions de hachage) pour les tests d'égalité répétitifs d'objets structurés.

Recommandations sur le choix du sujet et la préparation du rapport. Le choix du sujet est 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. L'originalité et la prise de risque sont souvent valorisées par le jury, surtout quand elles traduisent une implication particulière des candidats.

Chaque TIPE doit impérativement répondre à un questionnement informatique, avec une formalisation explicite du problème étudié. Un exemple de mauvais sujet est la simulation informatique d'un phénomène (physique, économique, social, ...) au comportement sophistiqué, mais où la part informatique est réduite à l'écriture d'un programme qui ne fait pas appel à des structures de données ou des algorithmes évolués. Il en va de même si le cœur du problème traité est uniquement d'ordre mathématique (par exemple, la résolution approchée d'équations différentielles), sans aucun questionnement d'ordre informatique. Ces sujets placent immédiatement le candidat dans une situation délicate : la part informatique ne présente aucune difficulté et le jury est peu compétent pour juger de la qualité scientifique de la part physique ou mathématique traitée. Une autre classe de sujets problématiques sont ceux purement informatiques pour lesquels le candidat se limite à l'implantation de techniques connues et pour lesquelles son apport est minimal. Par exemple, un TIPE ne peut se limiter à implanter l'algorithme de rétropropagation du gradient dans un réseau de neurones et ensuite à tester celui-ci sur quelques exemples tout en modifiant quelques paramètres "au doigt mouillé". Ce type de sujet est délicat en ce qu'il permet difficilement de mesurer le recul du candidat vis-à-vis de techniques connues dont il n'aurait fourni qu'une implantation particulière. Certains sujets classiques d'informatique (les problèmes de

²La preuve du bon fonctionnement de tout algorithme présenté doit pouvoir être donnée par le candidat (ou au moins esquissée dans les grandes lignes si celle-ci est difficile ou fait appel à des notions hors programme).

flots, de plus courts chemins, les codes correcteurs,...) peuvent se révéler trompeurs, en conduisant le candidat à se contenter d'un cours présentant des résultats anciens, plutôt qu'à approfondir un angle particulier (un algorithme efficace, une structure de données, un domaine d'application nouveau). Faire un cours sur un algorithme basique dont on trouve de nombreuses implémentations sur le web ne peut suffire. Aussi, 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 doit pas se baser uniquement sur le fait qu'il y a eu un travail de programmation : le jury attend des candidats un réel questionnement sur les algorithmes et les structures de données utilisés. Réciproquement, un TIPE en informatique n'est pas complet sans une part d'implémentation.

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. De même, un TIPE ne peut pas se résumer à la présentation d'un TP ou d'un devoir maison.

Le jury observe aussi depuis quelques années un investissement a minima dans le travail de TIPE, consistant à présenter un article de recherche récent en implémentant un des algorithmes présentés dans cet article. Cela ne peut suffire et le jury attend une exploration plus large de la littérature et une analyse critique des questions posées et des résultats existants, qui témoignent d'un réel investissement du candidat dans son 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 requière 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. Plus généralement, un TIPE ne peut se résumer uniquement à de la programmation. 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...) ne doit pas se limiter à la mise en œuvre de la méthode. Dans le cas d'un algorithme approché pour un problème d'optimisation, il est important de fournir des garanties en terme de qualité de la solution (facteur d'approximation) et de performance (complexité en ressources). Les algorithmes génétiques, par exemple, sont un paradigme de programmation comme un autre, où le vrai programme réside dans le choix du codage sous forme de génome et 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. Avant d'appliquer une approche "boîte noire" (méta-heuristique, réseaux de neurones, ou autre) à un problème, il est 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 pour un certain nombre de ses variantes) et soit capable de comparer les performances obtenues par les approches heuristiques avec celles d'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 ce type, le jury attend du candidat une compréhension précise du sens de ces valeurs, de leur rôle et de leur utilisation.

Enfin, signalons qu'il est important que les candidats *fassent valider par leur encadrant*, à la fois la *modélisation proposée* du problème considéré, et les *directions de développement*. Le jury a constaté des options particulièrement naïves, voire des erreurs de modélisation, qui auraient pu être évitées.

Présentation des rapports. Le rapport doit se limiter à 6 pages (hors références bibliographiques, annexes

et figures), il doit inclure toute la bibliographie utilisée (avec les références exactes permettant de retrouver les documents), ainsi que l'intégralité du code développé, en annexe. Il doit être construit autour d'un fil rouge clair (un seul sujet, bien traité), éviter le style télégraphique, éviter la présentation textuelle d'algorithmes pour préférer le pseudo-code, s'appuyer sur des exemples simples et parlants, et ne pas renvoyer aux annexes les éléments importants pour la compréhension (exemples, figures, preuves clés). Cette année encore, certains rapports n'ont pas respecté la limite de 6 pages, et l'on observe une utilisation abusive des annexes conduisant à des rapports très longs. 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.

Trois points sont particulièrement à améliorer.

Premièrement, le rapport doit être synthétique, lisible et *citer ses sources*. Le jury peut poser des questions sur toute affirmation écrite ou source citée dans le rapport; *a fortiori* il est problématique, et cela se produit trop souvent, que des affirmations ne soient pas sourcées. De même, la paraphrase des sources utilisées n'est pas admissible, surtout si elle est mal comprise. Le rapport doit démontrer que le candidat s'est approprié le sujet traité.

Deuxièmement, il est important que tous les candidats incluent en annexe l'intégralité de leur code. Il n'est pas nécessaire de venir avec une version imprimée du rapport car les membres du jury ont la version électronique; mais il est important que le jury puisse consulter ce code avant la présentation, afin de préparer d'éventuelles questions sur un point relevant du travail de programmation.

Troisièmement, il est demandé aux candidats d'indiquer systématiquement lorsque le travail de TIPE est issu d'une collaboration avec d'autres étudiants (ce qui est parfaitement acceptable). Dans ce cas, le rapport tout comme les éventuels transparents doivent être un travail personnel, indiquant clairement quelle est la contribution personnelle du candidat. Celui-ci doit en effet être capable de discuter 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 (peut-être faites par un binôme) ; embourbés dans une maîtrise approximative de ces points préliminaires, ils n'ont pas pu présenter le cœur de leurs travaux.

Importance de la bibliographie. Le jury tient à rappeler l'importance du travail bibliographique dans le cadre du TIPE : établir une bibliographie pertinente vis-à-vis du sujet du TIPE est une partie incontournable d'un travail maîtrisé. L'absence de recherche bibliographique a conduit certains candidats à un travail d'une grande naïveté : il est nécessaire de connaître les approches efficaces du problème considéré, les connaissances minimales attendues étant celles qui figurent dans une encyclopédie (comme par exemple, les pages *wikipedia* consacrées à ce sujet). Le jury a pu voir certains candidats assez brillants buter sur des difficultés qui n'en étaient pas car des solutions connues existaient. De manière générale le jury encourage fortement les candidats à faire preuve de curiosité vis-à-vis de leur sujet de TIPE, notamment à se demander si d'autres problèmes similaires ont déjà été traités dans la littérature, et à situer leur travail par rapport à l'existant.

Comme pour la modélisation, il est fortement conseillé aux candidats de faire valider leur bibliographie par leur référent.