

Agrégation d'informatique 2023

Résultats et analyse

La SIF

Ce document résume le rapport du jury de la deuxième édition de l'agrégation externe d'informatique, qui s'est déroulée en 2023, sous la présidence de Sylvie Boldo¹. La rédaction de ce rapport est très modulaire, ce qui a permis de rédiger le présent résumé en extrayant du document complet – avec l'accord des auteurs – des passages entiers très peu modifiés plutôt qu'en rédigeant un texte spécifique, qui aurait risqué de déformer le propos originel. Le rapport complet est disponible sur le site de l'agrégation². La première section du présent résumé précise le déroulement du concours des statistiques sur les candidats, la deuxième section est dédiée aux épreuves écrites, et la troisième section aux les épreuves orales.

Déroulement et statistiques

Le rapport a plusieurs objectifs : c'est d'une part une analyse et un bilan de cette édition (statistiques sur les candidats, réussite aux épreuves, erreurs fréquentes, conseils...) et d'autre part un document à l'attention des futurs candidats et préparateurs (attentes du jury, écueils à éviter...).

Déroulement du concours

L'agrégation externe section informatique est régie par l'arrêté MENH2112-666A du 17 mai 2021, qui en définit le programme et les épreuves.

1. Directrice de recherche, Inria.

2. <https://agreg-info.org>.

Le programme des épreuves d'admissibilité se compose :

- des programmes d'enseignement de la spécialité « numérique et sciences informatiques » (NSI) de la première et de la terminale de la voie générale du lycée ;
- des programmes des classes préparatoires scientifiques aux grandes écoles « mathématiques, physique, ingénierie et informatique » (MP2I) et « mathématiques, physique, informatique » (MPI) ;
- d'un programme complémentaire élaboré par le jury.

Il a pour but de permettre au jury d'évaluer le recul des candidats sur les notions enseignées et est conçu par « agrégation » des programmes sus-cités. Pour tout le programme, il est attendu des candidats un recul correspondant au niveau master.

Les épreuves écrites se sont déroulées du 6 au 8 mars dans les académies, avec 5 heures pour la composition d'informatique, 6 heures pour l'étude d'un problème informatique et 6 heures pour l'épreuve spécifique. Les épreuves orales se sont déroulées du 17 au 22 juin 2023 au lycée Paul Valéry à Paris, organisées par le jury. Chaque admissible a été convoqué pour les trois épreuves orales sur trois jours consécutifs pour la leçon d'informatique (4 heures de préparation, 1 heure d'épreuve), les travaux pratiques de programmation (5 heures de préparation, 1 heure d'épreuve), la modélisation (4 heures de préparation, 1 heure d'épreuve).

Statistiques

Comme en 2022, le concours a été attractif, avec un très grand nombre de candidats. De plus, le niveau final des agrégés est excellent. Le jury a pourvu sans hésiter tous les postes, avec une liste complémentaire de 4 noms en plus des 22 places initiales, pour 48 admissibles. Les agrégés ont montré de très belles qualités, autant disciplinaires que pédagogiques. Le jury les a volontairement testés sur des domaines différents de l'informatique et des langages différents avec succès et a pu sélectionner des informaticiens complets, avec une vision large de la discipline.

Pour 2023, les épreuves écrites ont été communes avec l'agrégation d'informatique du Maroc (215 présents pour 511 inscrits au Maroc, et 174 présents pour 440 inscrits en France). Les candidats des deux concours ont passé les épreuves écrites en même temps. Leurs copies ont ensuite été corrigées par les mêmes correcteurs, anonymement et sans connaissance de la nationalité. Le chiffre des inscrits et des présents est en baisse par rapport à 2023, entre 20 et 25 %. L'agrégation d'informatique reste cependant l'un des concours les plus sélectifs de l'agrégation externe, avec environ 8,4 candidats pour un poste.

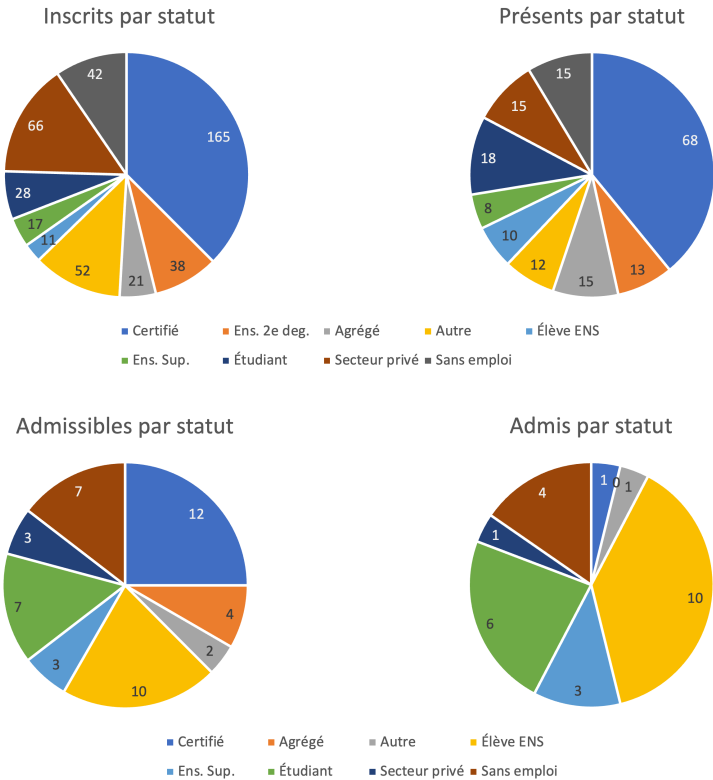
La moyenne des candidats présents aux trois épreuves écrites est de 4,68³ et celle des candidats admissibles de 12,35, avec une barre d'admissibilité à 8,76.

3. Toutes les notes données sont sur 20 points.

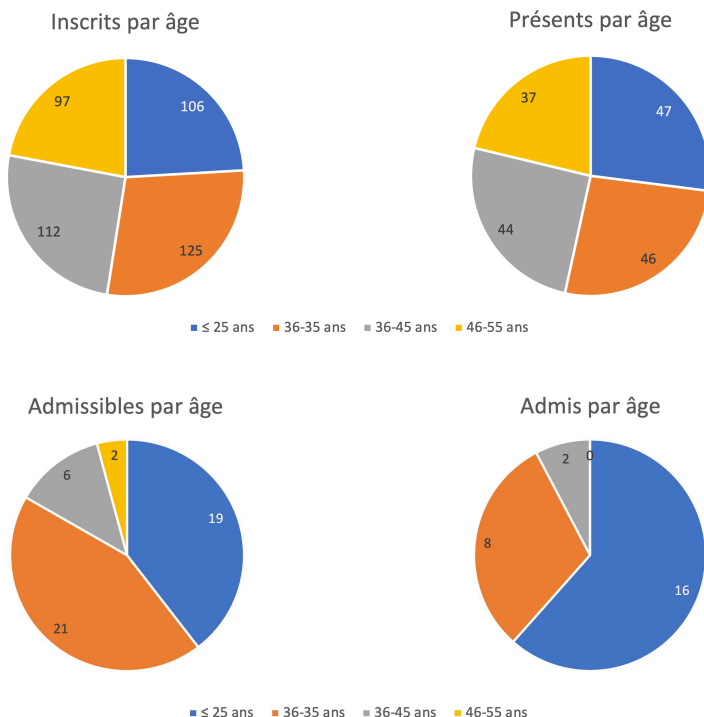
La moyenne des candidats présents aux trois épreuves orales est de 11,06, et la moyenne des candidats admis en liste principale de 13,32. Sur l'ensemble des épreuves, la barre d'admission est de 12,17 pour la liste principale et de 11,44 pour la liste complémentaire.

STATISTIQUES PAR STATUT

Les statuts sont donnés par les candidats de manière purement déclarative. Les profils des candidats sont extrêmement divers, même si cette diversité tend à se réduire tout au long du concours, la population des candidats bénéficiant de l'environnement d'une préparation devenant prépondérante à mesure que le concours se déroulait. Le profil des admis est un peu plus varié que l'an passé, même si les élèves des écoles normales supérieures constituent encore un bon tiers des lauréats.



STATISTIQUES PAR ÂGE



STATISTIQUES PAR GENRE

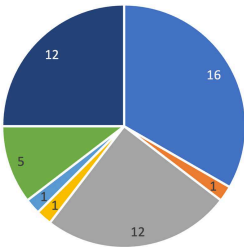
Le genre, déclaratif, ne permet à l'inscription que les deux choix M. ou Mme. La représentation des femmes en sciences et en informatique en particulier reste très faible. Le taux de femmes inscrites est faible (moins de 19%), les femmes se présentent en moindre proportion, et elles ont plutôt moins bien réussi les épreuves écrites (étant entendu que les copies sont anonymes). Pour les épreuves orales, il est délicat de faire des analyses sur un échantillon aussi réduit, probablement corrélé avec d'autres facteurs statistiques mais ce point reste une préoccupation du jury, surtout au vu de la diminution du taux de femmes admises entre les sessions 2022 et 2023.

Genre	Candidats	Présents	Admissibles	Admis
M.	359	148	45	25
Mme	81	26	3	1

STATISTIQUES PAR ACADÉMIE

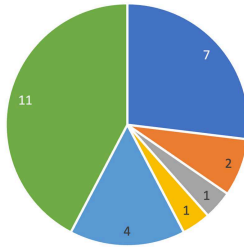
Les seules académies à proposer une préparation à l'agrégation externe d'informatique sont les académies de Créteil-Paris-Versailles, Lyon et Rennes, même si la première préparation peut se suivre partiellement à distance.

Admissibles par académie



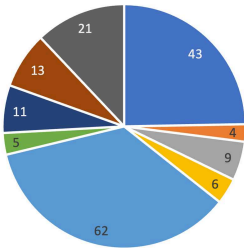
■ IDF ■ Strasbourg ■ Autres ■ Nantes ■ Lille ■ Lyon ■ Rennes

Admis par académie



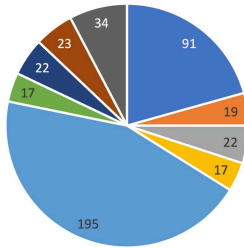
■ IDF ■ Autres ■ Nantes ■ Lille ■ Lyon ■ Rennes

Présents par académie



■ IDF ■ Poitiers ■ Montpellier ■ Strasbourg ■ Autres ■ Nantes ■ Lille ■ Lyon ■ Rennes

Inscrits par académie



Épreuves écrites

L'architecture générale de l'écrit de l'agrégation externe d'informatique est définie par l'arrêté du 17 mai 2021. Dans les limites de la maquette définie par cet arrêté, le jury affirme sa volonté d'utiliser l'ensemble des épreuves à sa disposition pour recruter des informaticiennes complètes,⁴ capables à la fois de réflexes sains dans un ensemble de domaines assez larges de l'informatique (épreuve 1), et dotées d'une solide maîtrise de la programmation et de l'algorithmique (épreuve 2). L'épreuve 3 est l'occasion de démontrer des connaissances approfondies dans un domaine au choix des candidates : l'ingénierie logicielle ou les fondements de l'informatique.

Certaines remarques s'appliquent à toutes les épreuves. En particulier, on attend d'une candidate aux fonctions de professeur qu'elle sache rédiger une copie. En conséquence, un passage rayé ou hachuré n'est jamais lu. De plus, quand plusieurs versions sont proposées par la candidate, la correctrice ne prend pas la meilleure. C'est à la candidate de choisir ce qu'elle présente à l'évaluation du jury.

Une préoccupation du jury est de valoriser les candidates qui sont précises sur les parties élémentaires du sujet plus que celles qui bâclent le début pour avancer plus vite vers les parties plus ambitieuses. Le jury recherche des candidates solides et précises sur les bases de la discipline, afin de pouvoir transmettre ces dernières. Il est attendu, pour l'écrit et pour l'oral, que les candidates maîtrisent le vocabulaire des mathématiques nécessaire à l'étude des éléments au programme de l'agrégation d'informatique, dont les algorithmes probabilistes et l'analyse de complexité.

Le jury tient compte dans la notation des épreuves de la maîtrise écrite et orale de la langue française (vocabulaire, grammaire, conjugaison, ponctuation, orthographe). Le jury attend donc une certaine tenue dans l'écriture, la présentation et l'orthographe. Ces attentes sont modérées, mais réelles. Le jury fait la part des choses entre ce qui est explicable par le contexte « concours » et ce qui risque d'handicaper une future agrégée dans sa pratique professionnelle et la transmission des savoirs. Les travers relevant de la seconde catégorie sont pris en compte, aux côtés de la maîtrise disciplinaire, dans l'évaluation d'une copie. Le jury a apprécié de lire des réponses construites, et rappelle qu'une phrase commence par une majuscule et se termine par un point, même quand elle est très courte et qu'elle constitue à elle seule une réponse à une question.

4. Pour alléger le texte, ce chapitre est uniquement écrit au féminin, mais cela doit être pris dans le sens du neutre : l'ensemble de ce texte, sans exception, concerne uniformément candidates et candidats.

Épreuve 1

185 candidates ; meilleure note : 18,09 ; moyenne : 6,51 ; écart-type : 3,89

Cette épreuve était composée de trois parties, balayant largement les bases de données, les graphes et l'architecture des ordinateurs.

Le jury a été déçu par les réponses à cette épreuve, car de nombreuses candidates ont peu abordé certaines parties. Hormis la sous-partie de la première partie, sur les requêtes SQL, les correctrices ont estimé que trop de questions ont été mal ou pas traitées dans cette épreuve. Pour cette épreuve en particulier, composée de plusieurs parties indépendantes, les correctrices souhaitent que les candidates notent le numéro complet de la question traitée et qu'elles ne mélangent pas les réponses aux différentes parties.

PARTIE I – PROVENANCE EN BASES DE DONNÉES

Cette partie a pour but de tester les connaissances en bases de données, notamment sur les requêtes SQL. Le sujet se poursuit par un point de vue plus logique permettant de parler de la provenance des résultats d'une requête.

Cette partie a été beaucoup traitée, les requêtes SQL étant souvent correctes. Le jury regrette la syntaxe SQL parfois approximative et le fait que vraiment peu de copies traitent la partie sur la provenance. Pour les requêtes SQL, les candidates doivent veiller à l'indentation pour aider à la lisibilité et à ne pas faire de jointure inutile. Il est aussi conseillé de bien lire les questions de façon à construire la requête demandée. Enfin, certaines candidates ont utilisé des doubles négations dans les requêtes, rendant le code peu compréhensible et souvent faux. En ce qui concerne l'algèbre relationnelle — seulement au programme complémentaire —, le jury a toléré certaine souplesse dans la syntaxe utilisée.

PARTIE II – PONTS D'UN GRAPHE

L'épreuve portait sur l'implémentation, en langage OCaml, d'une structure de données permettant de déterminer efficacement les blocs d'un graphe non orienté (c'est-à-dire les composantes connexes restant après suppression des ponts, définis comme les arêtes dont la suppression fait croître le nombre de composantes connexes) dans le cas où le graphe est construit de façon incrémentale en ajoutant les arêtes une à une. Cette approche se base sur une variante de la structure classique de *d'Union-Find* (« Unir et trouver »), présente au programme de MPI.

Globalement, le jury regrette que cette partie de la première épreuve ait été si peu et si mal traitée, tant au niveau de la maîtrise du langage OCaml que du sujet lui-même, à savoir l'étude d'une structure de données. Ainsi, il est manifeste que les bases d'OCaml ne sont pas acquises pour une proportion non négligeable de candidates, alors que ce langage est explicitement au programme.

Dans certaines copies, on a pu lire du pseudo-code ressemblant à du Python, voire du code Python de manière totalement assumée. Cela n'est pas acceptable. Au début de l'énoncé, il était rappelé un ensemble de notations concernant en particulier la manipulation de tableaux. Il est donc regrettable de trouver dans des copies des notations incorrectes et une mauvaise utilisation des fonctions de base pour manipuler les tableaux.

Le jury a aussi eu l'impression que de trop nombreuses candidates pensent que programmer en OCaml revient à programmer exclusivement récursivement, et que toutes les fonctions doivent commencer par un *match ... with*. Cette situation, naturelle lorsque l'on manipule des paramètres d'un type somme, ne correspondait pas à un grand nombre de questions de l'énoncé. D'ailleurs, un usage excessif et mal maîtrisé du filtrage peut être vu comme maladroit, voire créer d'inutiles complications.

Notons enfin que certaines questions attendaient de façon naturelle une solution itérative et que si l'utilisation de fonctions auxiliaires est admise voire recommandée, il est indispensable d'en expliciter clairement le rôle.

PARTIE III – ARCHITECTURE DES ORDINATEURS

Cette partie vise à évaluer les candidates sur l'architecture des ordinateurs, avec une première section sur l'algèbre de Boole, une deuxième sur les bascules et une troisième sur un automate de Moore.

La première section a été plutôt bien réussie et abordée par une majorité de candidates. La deuxième section concerne le fonctionnement des bascules RS et D et a été plus décevante. Les correctrices ont pu constater qu'une part importante des candidates n'a pas compris précisément le fonctionnement d'une bascule D. La troisième section, probablement la plus facile de la partie, a été complètement ratée. Il s'agissait uniquement de donner les tables d'un automate de Moore donné, pour ensuite le synthétiser à l'aide de portes NOR. La notion d'automate de Moore, au programme de l'agrégation, ne semble pas assimilée.

Épreuve 2

179 candidates ; meilleure note : 20 ; moyenne : 5,80 ; écart-type : 4,76

L'objet de cette épreuve est d'étudier la décidabilité du problème d'équivalence entre deux expressions rationnelles. Elle est composée d'une partie I qui contient des préliminaires algorithmiques, d'une partie II qui étudie le passage d'une expression rationnelle à un automate non déterministe équivalent, et d'une partie III qui porte sur l'équivalence de deux automates déterministes. Le but principal de l'épreuve est, en plus de vérifier les connaissances des candidates sur ces parties du programme de l'agrégation, de tester leurs capacités à raisonner sur des programmes informatiques.

Cette épreuve part de concepts et de résultats explicitement au programme de l'agrégation, pour aller jusqu'à des résultats classiques et des extensions tirées d'articles de recherche. Le parti pris concernant les résultats non explicitement au programme fut de les introduire et les analyser à travers le prisme de la programmation. Le sujet comportait régulièrement des questions de déroulement d'algorithmes sur des exemples simples qui devaient permettre aux candidates de se familiariser progressivement avec les notions introduites.

La partie I permettait aux candidates de prouver leur maîtrise de base de la programmation Python et de l'analyse de la complexité temporelle de fonctions simples, à travers l'étude de la complexité amortie pour les listes Python et le tri lexicographique.

La partie II commençait par une section visant à transformer une expression rationnelle en arbre. Il s'agissait ensuite d'appliquer l'algorithme de Berry-Sethi pour obtenir l'automate de Glushkov, et de prouver la correction totale du code fourni et sa complexité temporelle. Enfin l'énoncé s'inspirait de l'article « Passage d'une expression rationnelle à un automate fini non-déterministe » de D. Ziadi, J.-L. Ponty et J.-M. Champarnaud (Bull. Belg. Math. Soc. 4 (1997), p. 177-203) pour modifier la fonction de l'énoncé dans le but d'améliorer sa complexité. Cette partie permettait de tester les capacités des candidates à faire des raisonnements par récurrence et par induction.

La partie III portait sur la minimisation d'automates déterministes, pour tester l'équivalence de deux automates déterministes, en étudiant l'équivalence de Nérode, l'algorithme de minimisation de Moore, et en terminant par un algorithme de type d'*Union-Find*. Cette partie reposait sur les capacités des candidates à analyser des fonctions pour prouver des résultats sur les objets manipulés par ces fonctions. L'algorithme de type d'*Union-Find* est inspiré de l'article « A linear algorithm for testing equivalence of finite automata. » de J. E. Hopcroft and R. M. Karp (Technical Report 114, Cornell University, December 1971). Cette partie permettait aux candidates de montrer leur capacité à faire le lien entre les algorithmes et structures de données au programme et la théorie des automates.

PRÉSENTATION

Comme l'an passé, le jury invite les candidates à soigner la qualité scripturale, grammaticale et rédactionnelle de leur copie. Il y est doublement sensible : une copie convenablement écrite est plus facile à évaluer et, par ailleurs, il paraît inconcevable de recruter des enseignantes ne montrant pas une maîtrise raisonnable de la langue française. Il convient également d'utiliser un vocabulaire approprié : on ne dit pas d'une propriété qu'elle est « forcément » vraie, et il n'est pas davantage adéquat d'écrire « On est sur une complexité $O(n)$ ». De la même façon, il faut respecter la syntaxe et la graphie lorsqu'on écrit du code informatique : la casse n'est pas anodine en Python. Le jury n'apprécie pas le code

rempli de ratures ou le code sur deux colonnes. Il faut éviter les renvois d'une page à une autre avec des flèches, parce que cela pose des problèmes de propreté de la présentation. Enfin, il n'est pas utile de recopier la question : en rappeler le numéro suffit.

PROGRAMMATION

Le jury est surpris que certaines candidates ne sachent pas programmer en Python, langage pourtant au programme de l'agrégation. Dans les questions de programmation, le jury n'attend pas que le programme vérifie que les paramètres respectent les préconditions. Il n'est pas non plus utile de traduire du code en français.

FORMALISME ET RAISONNEMENT MATHÉMATIQUES

Comme l'an passé, le jury constate une très grande variabilité dans la qualité des raisonnements, qu'il s'agisse de leur présentation ou du fond. Certaines copies pourraient servir de modèle de corrigé. D'autres semblent rédigées par des candidates qui n'ont pas compris qu'elles passent un concours scientifique. Ainsi, un exemple ne peut servir à prouver la correction d'une fonction. Les symboles mathématiques doivent être employés à bon escient et ne se substituent pas à un verbe : on ne peut utiliser $\forall$ dans une phrase comme si c'était une abréviation. Enfin, il convient d'éviter de commencer une phrase par une formule, un mot-clé du langage de programmation ou un nombre.

Plus profondément, le jury déplore que certaines candidates emploient le symbole d'implication $\Rightarrow$ en lieu et place d'une déduction. On ne peut, en tant que future enseignante, se permettre cet abus d'écriture qui peut aussi entraîner des erreurs de logique graves. Il convient d'employer le mot « donc » ou un synonyme pour affirmer chacune des propriétés en les déduisant les unes des autres. De même, la conclusion d'une réponse ne doit pas être introduite par une flèche, et moins encore par la flèche d'implication. L'absence de maîtrise de ce point fait désordre pour toute professeure scientifique et révèle un manque de recul sur les chapitres de logique du programme de l'agrégation.

La manipulation de la notation $O(\cdot)$ doit être raisonnablement rigoureuse. Une définition précise des notations O , Ω , Θ , doit être maîtrisée par les candidates à l'agrégation. Il est important que ces trois symboles soient bien différenciés les uns des autres dans l'écriture. Enfin, il convient d'être rigoureux lorsqu'on veut prouver une majoration de complexité par induction structurelle. On ne peut pas supposer que, pour une sous-expression E , la complexité est en $O(|E|)$. Cela n'a pas de sens. Il faut passer par une majoration avec des constantes explicites.

Épreuve 3

Cette épreuve permet à la candidate de choisir (à l'inscription) entre étude de cas informatique (107 candidates ; meilleure note : 17 ; moyenne : 7,30 ; écart-

type : 3,60) ou fondements de l'informatique (68 candidates ; meilleure note : 20 ; moyenne : 8,64 ; écart-type : 5,38).

Il y a eu d'excellentes copies dans les deux épreuves. Par contre, il apparaît que le vivier des candidates se présentant à chacune des épreuves n'est pas le même. Les deux épreuves ont donné lieu à des moyennes assez différentes. Cet écart se reflète dans les autres épreuves : les candidates ayant choisi l'étude de cas informatique ont eu une moyenne inférieure de plusieurs points à celle des candidates ayant choisi les fondements de l'informatique, et ceci sur les deux autres épreuves.

ÉTUDE DE CAS $(u_n)_{n \in N}$ ET $(\alpha_n)_{n \in N}$ INFORMATIQUE

Le sujet s'inspire d'un projet concernant des livraisons par des véhicules. La partie 1 présente la problématique et s'attache à des considérations algorithmiques et de programmation objet en Python. L'idée de la partie 2 était d'explorer un protocole applicatif de communication multisaut entre véhicules. Un grand nombre de réponses n'ont pas su faire la différence entre un protocole applicatif et le protocole TCP. La partie 3 explore la concurrence et la partie 4 concerne la gestion de bases de données. La partie 5 s'intéresse à la programmation Web.

Il est attendu que tout code intègre des commentaires faisant le lien avec le sujet, par exemple pour fusionnable et ses trois critères. On s'étonne que certaines candidates ne sachent pas écrire de code Python, JavaScript, HTML ou SQL, alors que tous ces langages sont au programme. Au-delà de la maîtrise syntaxique d'un langage de programmation, il est attendu que tout code soit lisible et intelligible. À cette fin, le recours à des fonctions et attributs, intermédiaires ou privés, avec un libellé explicite peut être une aide utile à la compréhension d'un code.

FONDEMENTS DE L'INFORMATIQUE

Le sujet de cette épreuve consistait à étudier différents modèles de réseaux de neurones formels, et à déterminer leur puissance en tant que modèles de calculs. Il visait à relier ce modèle à différents modèles plus classiques de l'informatique fondamentale : circuits booléens, automates finis, machines à compteurs, machines de Turing, et à voir que chacun de ces derniers modèles correspond à une certaine classe de réseaux de neurones formels.

Dans la partie I, on se focalisait sur le cas d'un unique neurone formel dont la fonction d'activation est la fonction de Heaviside (neurone à seuil). On devait dans un premier temps étudier comment calculer certaines fonctions booléennes, et montrer que des fonctions comme la fonction parité ne peuvent pas se représenter de cette façon. On discutait alors la représentation de ces fonctions, en utilisant des coefficients entiers. On étudiait ensuite la question de l'apprentissage des coefficients d'un neurone à seuil, en étudiant l'algorithme

classique du Perceptron proposé en 1957 par Frank Rosenblatt, en démontrant sa convergence selon les arguments proposés par Novikoff en 1962.

Dans la partie II, on commençait à étudier des réseaux de neurones, avec des questions de programmation en OCaml, et l'observation que ces réseaux permettent de coder les fonctions booléennes, et que l'on peut toujours supposer que les réseaux sont organisés en couches. Puis, on cherchait à établir qu'il y a des fonctions comme la fonction parité qui se calculent avec un réseau de neurones de façon beaucoup plus succincte qu'avec des circuits booléens.

Dans la partie III, on montrait la NP-complétude du problème de l'apprentissage des coefficients d'un réseau de neurones à partir d'un ensemble d'exemples.

Dans la partie IV, on reliait les réseaux de neurones récurrents aux automates finis, en jouant avec différentes façons de donner les entrées à un calcul.

Dans la partie V, on reliait les réseaux de neurones dont la fonction d'activation est la fonction ReLU (unité linéaire rectifiée) aux machines à compteurs, et les réseaux de neurones dont la fonction d'activation est la fonction linéaire saturée aux machines de Turing.

La plupart des copies dénotent une très bonne compréhension du sujet et montrent un traitement convenable des parties I et II. La partie III y a été manifestement moins inspirante, en dehors des questions les plus élémentaires. La partie IV a été traitée par un nombre plus grand de candidates, cependant trop souvent sans rentrer suffisamment dans les détails. En particulier, les candidates ne distinguent pas avec suffisamment de clarté le raisonnement pour les différentes variantes possibles d'automates finis correspondants, alors que c'était parfois précisément l'objet de plusieurs des questions. Les parties IV et V ont été traitées globalement par très peu de candidates, en dehors des questions de programmation par les machines à compteur.

Globalement, trop de questions nécessitant du code OCaml sont esquivées, ou traitées dans une syntaxe plus qu'approximative. La nature inclusive des bornes des boucles OCaml ne semble pas connue de nombreuses candidates. Trop d'inégalités strictes se transforment en inégalités larges au gré de la rédaction, et la manipulation des inégalités est trop souvent erratique, en particulier, en ce qui concerne les inversions d'inégalités lors de multiplications par des quantités négatives.

Épreuves orales

Le jury est conscient du stress et de la fatigue des candidats⁵ avec des temps de préparation longs et des horaires parfois décalés. Son objectif n'est pas de piéger les candidats mais de les mettre dans un contexte leur permettant de montrer le meilleur d'eux-mêmes. Le jury a jugé l'oral de l'agrégation d'un très solide

5. Pour alléger le texte, ce chapitre est uniquement écrit au masculin, mais cela doit être pris dans le sens du neutre : l'ensemble de ce texte, sans exception, concerne uniformément candidates et candidats.

niveau global, à la fois en matière de connaissances disciplinaires et de compétences pédagogiques.

Le jury rappelle que les livres sont disponibles pour les préparations de toutes les épreuves. La liste des livres fournis par le jury est disponible sur le site de l'agrégation⁶ et sera mise à jour pour la session 2024. Les préparations à l'agrégation ont elles aussi mis à disposition des livres, également accessibles à tous les candidats. L'environnement informatique mis à disposition des candidats est celui qui est téléchargeable sur le site du jury et les fichiers des candidats sont sauvegardés sur un serveur. Il est à noter que les responsables informatiques ne sont pas là pour aider les candidats à debugger leur code, mais peuvent intervenir en cas de problème avec l'environnement à disposition. Consulter la documentation disponible dans cet environnement doit être le premier réflexe.

De nombreux conseils et remarques s'appliquent à l'ensemble des épreuves orales et sont donnés dans le rapport. Les trois épreuves sont distinctes et permettent au jury d'évaluer des points différents. Il est donc requis des candidats qu'ils connaissent la description et les attendus des trois épreuves. Des temps maxima sont donnés pour les trois épreuves et rappelés au début de chaque oral. Il est important d'essayer de se rapprocher le plus possible de ces temps sans toutefois les dépasser : même s'ils sont indiqués comme des temps à ne pas dépasser, ils sont en fait des temps-cibles.

Le jury d'une épreuve d'oral n'a pas connaissance des résultats de l'écrit du candidat et s'impose une discipline stricte de ne pas discuter des prestations d'un candidat devant les autres membres du jury avant que celui-ci ait passé l'ensemble de ses épreuves. Lorsqu'un candidat se présente à une épreuve, il est ainsi assuré d'être examiné sans aucun préjugé. Il est à noter qu'en 2023, l'ensemble des interrogations étaient ouvertes aux visiteurs.

Leçon

45 présents ; meilleure note : 19,25 ; moyenne : 10,44 ; écart-type : 4,62

Le candidat se voit proposer (aléatoirement) deux sujets au choix ; il doit choisir de traiter l'un des deux, mais n'a pas à justifier son choix sur ce point. Il indique son choix au début de l'interrogation proprement dite et peut donc changer d'avis en cours de préparation (même si ce n'est guère conseillé). Le jury n'interroge pas sur le sujet qui n'a pas été choisi.

Si la leçon d'agrégation permet d'évaluer les compétences didactiques des candidats, elle est un oral scientifique avant tout. Cela a des conséquences sur les trois moments de l'oral : le plan doit comporter du contenu scientifique, et pas seulement l'annoncer ; le développement doit bien sûr montrer des qualités d'exposition, mais appliquées à un contenu scientifique clair et précis ; les questions posées par le jury peuvent porter sur la façon dont on expliquerait telle

6. <https://agreg-info.org>.

chose à tel type d'élève, mais elles portent également fréquemment sur le fond des notions abordées, au niveau de recul M2 attendu des candidats.

Sur chaque point, la maîtrise et le recul didactique sur le contenu scientifique sont appréciés et valorisés, mais il s'agit de la cerise sur le gâteau : il faut d'abord qu'il y ait un gâteau. Par ailleurs, l'agrégation est un concours de recrutement d'enseignants : une expertise sur des domaines hors programme ne se substitue pas à un recul pédagogique. Le document complet donne de nombreux conseils sur le plan et le développement de la leçon.

QUESTIONS

De nombreuses questions simples visent à vérifier que le candidat comprend et maîtrise ce qu'il écrit ; elles sont posées systématiquement et le candidat ne doit pas s'en trouver destabilisé. Certaines questions peuvent porter sur l'enchaînement des éléments du plan ou sur sa cohérence interne. Le jury s'attend à des notations homogènes de la part du candidat, même s'il a utilisé plusieurs sources adoptant des conventions de notation ou des définitions légèrement différentes. Si le travail d'homogénéisation n'est pas fait dans le plan, le candidat sera interrogé et invité à le faire pendant l'échange avec le jury. Le candidat doit également faire attention au développement logique de son cours : il peut être maladroit d'utiliser des notions essentiellement introduites postérieurement dans le plan. Le jury ne manquera pas de questionner ces choix.

Les candidats doivent bien écouter les questions qui leur sont posées, et bien différencier, par exemple :

- comment expliqueriez-vous l'algorithme de Rabin-Karp à un bon élève de terminale NSI ? Ici, de nombreuses réponses sont possibles ; on peut envisager de dérouler l'algorithme à la main sur un exemple bien choisi ;
- pouvez-vous expliquer l'algorithme de Rabin-Karp Ici, on attend une réponse scientifique de même nature que pour un oral de master.

S'ils présentent une « méthode générale » dans leur plan, les candidats peuvent s'attendre à ce que le jury leur fournisse un exemple ou un cas d'usage à dérouler. Par exemple, dans une leçon sur la programmation dynamique où le candidat expose une manière générale d'obtenir une équation de récurrence entre sous-problèmes, le jury peut proposer sous forme d'un petit exercice un exemple de problème où il faut d'abord trouver cette équation et les bons sous-problèmes.

Travaux pratiques

44 présents ; meilleure note : 20 ; moyenne : 12,38 ; écart-type : 4,33

Cette épreuve orale a pour objectif principal l'évaluation de compétences pédagogiques des candidats.

L'épreuve se décompose en deux parties : les travaux pratiques proprement dits et la revue de code. Dans la première partie, le candidat doit produire et présenter du code, et justifier sa correction (à l'aide de tests, de preuves, etc.). Dans la seconde, le candidat est confronté à du code de mauvaise qualité et défectueux, comme pourrait en écrire un élève durant une séance de travaux pratiques, et doit proposer des corrections ou améliorations, comme le ferait un enseignant durant une telle séance.

Dans les deux cas, l'accent est mis sur la pédagogie et non sur la compétence disciplinaire ou la virtuosité, les codes à écrire ou à commenter étant conçus pour ne pas poser de difficultés techniques insurmontables aux candidats.

Comme pour les deux autres épreuves orales, on attend une conduite claire et organisée de l'exposé, avec mise en contexte (qui doit être présente sans être excessive) et présentation structurée des résultats. L'usage du tableau doit être au service de la compréhension du discours du candidat, et celui d'une vidéo-projection doit se concentrer autour de la projection de code, de l'exécution de tests et éventuellement de la présentation (éventuellement sous forme graphique) de résultats d'exécution. Une présentation basée sur des diapositives n'est pas souhaitée.

Le candidat doit veiller à respecter l'équilibre entre les deux parties de l'épreuve et, en particulier, ne pas passer trop de temps sur la revue de code. Le rapport fourmille de conseils pour la partie travaux pratiques et la partie revue de code.

Modélisation

45 présents ; meilleure note : 17,50 ; moyenne : 10,66 ; écart-type : 4,03

FORME ET STRUCTURE DE L'ORAL

L'oral se décompose en deux parties. La première partie consiste en un exposé de 35 minutes, qui prend la forme d'un cours d'ouverture à destination d'élèves de niveau L1, L2 ou CPGE. Le cadre d'énonciation repose sur une triple fiction : le jury n'a pas connaissance du texte proposé au candidat ; le candidat a trouvé ce texte tout seul lors de recherches pour préparer son exposé ; le candidat a formulé lui-même les pistes de réflexion qui sont proposées à la fin du texte.

On ne demande pas un exposé sur le texte, mais sur le sujet traité par le texte. Il n'y a donc pas lieu de citer le texte et encore moins de s'y référer. L'exposé doit être structuré et il est fortement conseillé de présenter son plan en début d'oral après une courte introduction. L'exposé doit aussi inclure la présentation argumentée d'illustrations sur ordinateur, et la discussion d'une dimension éthique, sociétale, environnementale, économique ou juridique telle que mentionnée par l'arrêté. La deuxième partie de l'oral consiste en un échange avec le jury.

CONTENU DE L'EXPOSÉ

Il est attendu des candidats d'une part une présentation et une discussion du problème et de sa formalisation, d'autre part une capacité à développer, compléter, voire améliorer les ébauches de solution esquissées. Par exemple, lorsque l'on fait des études de complexité asymptotique d'algorithmes présentés pour être appliqués à des cas concrets, le jury attend que le candidat soit capable de raisonner en ordres de grandeur.

Outre la capacité à mobiliser ses connaissances dans un contexte concret, le jury cherche également à évaluer la communication scientifique et pédagogique, c'est-à-dire la capacité de tenir un discours structuré, cohérent et pédagogique sur un sujet non trivial. Pour cela, le candidat doit construire un exposé fondé sur le texte fourni. Il ne s'agit pas de faire une lecture commentée du texte, mais bien de construire un exposé autonome. En particulier, les candidats ne doivent supposer du jury aucune connaissance préalable du texte. Le jury n'attend pas un traitement exhaustif de tout ce qu'abordent le texte et les pistes, mais l'esprit du document doit être respecté.

L'évaluation de l'utilisation du tableau est partie intégrante de l'évaluation des compétences didactiques et pédagogiques des candidats. Le jury attend que le candidat montre sa capacité à utiliser le tableau pour transmettre des notions et expliquer des exemples bien choisis.

Les illustrations doivent s'intégrer à l'exposé de manière cohérente et fluide. L'exercice porte sur la communication ; on attend donc une présentation efficace d'illustrations qui éclairent un aspect du texte. Le jury est plus sensible à la pertinence de l'illustration qu'à la complexité de l'implémentation, même s'il reste attentif à la qualité du code présenté. Le candidat peut insister plus ou moins sur la programmation mais le reste de l'exposé doit rester équilibré et contenir une présentation détaillée et précise d'une solution esquissée par le texte. En cas de programmation, le jury attend que le candidat exécute ses programmes pour vérifier leur fonctionnalité.

Le jury est attaché à la capacité des candidats à prendre du recul et envisager l'informatique comme une discipline et non comme un patchwork de domaines ; la capacité à s'appuyer sur le texte pour faire apparaître des liens dans d'autres sous-domaines que celui étudié par le texte constitue une vraie preuve de maturité scientifique. Le discours doit toutefois rester dans le cadre du texte, et ce dernier ne doit pas servir de prétexte à une digression dans un domaine où le candidat se sent à l'aise.

Les candidats doivent intégrer une discussion synthétique d'une dimension éthique, sociétale, environnementale, économique ou juridique. Cette ouverture ne suit pas nécessairement une piste proposée par le jury, toute initiative personnelle pertinente ayant d'ailleurs vocation à être valorisée.

L'épreuve de modélisation nécessite une réflexion préalable des candidats sur ses différents aspects : construction d'un exposé, utilisation du tableau,

illustration informatique, conclusion du propos (voir les exemples de sujets donnés aux sessions 2022 et 2023 publiés sur le site de l'agrégation⁷).

QUESTIONS

L'interrogation se poursuit par les questions du jury, qui peuvent porter sur toute la présentation du candidat, sur tous les sujets du programme se rattachant à celle-ci, ou sur les dimensions éthiques, sociétales, environnementales, économiques ou juridiques en lien avec le texte. C'est l'occasion pour le jury d'affiner sa perception de la compréhension du texte par les candidats, de faire préciser ou corriger des points, de fournir des indications pour que le candidat développe certains aspects qui lui ont résisté, ou encore de proposer des pistes de réflexion, des prolongements, ou des variantes du problème étudié.

Comme pour les séances de questions des autres épreuves, il est primordial que le candidat adopte une posture facilitant les échanges. Il doit écouter les questions jusqu'au bout et répondre de façon concise et la plus précise possible. Bien que cela soit parfois nécessaire, il est difficile et désagréable pour le jury de devoir couper la parole d'un candidat qui s'est lancé dans une réponse très longue.

REMARQUES GÉNÉRALES

L'épreuve de modélisation de cette session reposait sur deux sujets portant sur des thèmes différents : l'analyse de complexité en prenant en compte la prédiction de branchement et l'impact de la réduction du nombre de couleurs sur la taille des images stockées au format *png*. Les candidats ont plutôt bien compris le format de l'épreuve. Il reste tout de même quelques exceptions qui poussent le jury à rappeler des conseils d'ordre général (le rapport contient des remarques spécifiques à chaque sujet).

- Il est important de présenter un plan de son exposé dans son introduction.
- L'épreuve de modélisation n'est pas une seconde épreuve de TP, les illustrations informatiques ne doivent pas représenter l'essentiel de la présentation. En revanche, elles doivent servir le propos, mettre en lumière des phénomènes et ne pas être considérées comme de simples exercices de programmation.
- L'épreuve de modélisation est une épreuve d'informatique, même si les interactions avec d'autres disciplines scientifiques sont très intéressantes, ce sont bien les dimensions informatiques qui doivent être au cœur de l'exposé. En particulier, il n'est pas judicieux de passer trop de temps sur les aspects très mathématiques de certains sujets.
- La gestion du tableau est un enjeu important de l'épreuve. Il ne faut pas utiliser le tableau comme une feuille de brouillon, mais bien comme un support pédagogique.

7. <https://agreg-info.org>.

- La gestion du temps est également un enjeu important de l'épreuve. Il est par conséquent déconseillé de passer beaucoup de temps à recopier du code fourni dans le texte.

Le jury a été satisfait du niveau global des candidats. Malgré tout, certaines maladresses et lacunes sont apparues à plusieurs reprises.

- Le jury attend une posture d'enseignant de la part des candidats. Ils doivent en particulier parler et écrire dans un français correct, en utilisant un vocabulaire précis et rigoureux. Par exemple, les termes « aléatoire » et « optimal » ont été utilisés de façon très maladroite. En particulier, l'optimalité n'est pas une échelle de valeur. On n'est pas plus ou moins optimal.
- Certains candidats ont montré des fragilités concernant l'analyse de complexité et les ordres de grandeur. Le logarithme en base 2 et la racine carrée ne doivent pas être confondus.