1/ PRÉSENTATION DE L'ÉPREUVE

L'épreuve d'informatique portait cette année sur l'optimisation du réglage d'un correcteur PID via un algorithme génétique. Le sujet était décomposé en 6 parties et recouvrait des notions de première et deuxième année du programme d'informatique.

La première partie avait surtout pour objectif d'amener les candidats à lire la documentation de librairie scipy.signal afin d'être à même de l'utiliser pour tracer la réponse temporelle à un échelon d'une fonction de transfert.
La deuxième partie devait permettre aux candidats de définir les différentes fonctions permettant l'étude des fonctions de transfert : en particulier, les opérations de sommes et produits de fonctions de transfert étaient étudiées.

Références au programme :

- *Manipulation des structures de listes ;*
- *Algorithmique et programmation : inversion d'une liste,*
- *Expliquer le fonctionnement d'un algorithme,*
- *Évaluer, contrôler, valider des algorithmes et des programmes.*

La troisième partie consistait à définir les critères d'optimisation de la fonction de transfert. Il s'agissait de retrouver les principaux critères de qualité d'un système asservi (stabilité, précision, rapidité, dépassement et intégrale de l'erreur absolue).

Références au programme :

- *Écriture d'une fonction à partir d'une documentation ;*
- *Analyse d'une fonction et réécriture ;*
- *Méthodes des rectangles et des trapèzes pour le calcul approché d'une intégrale sur un segment.*
- *Écrire des instructions conditionnelles avec alternatives, éventuellement imbriquées.*

La quatrième partie avait pour but de réaliser l'algorithme génétique permettant de trouver une meilleure combinaison de paramètres pour le réglage du correcteur PID. Dans un premier temps, on s'intéressait à l'initialisation des candidats et de leurs gènes. Puis dans un second temps, intervenaient l'évaluation et le tri des candidats via la fonction-coût développée en partie 3. Finalement, les meilleurs candidats étaient croisés afin de créer la génération suivante.

Références au programme :

- *Introduction : Représentation des nombres entiers*
- *Algorithmique et programmation I : Instructions conditionnelles et itératives, Fonctions, Manipulation de quelques structures de données*
- *Algorithmique et programmation II : Tris*

La cinquième partie portait sur l'analyse de l'historique des candidats issus de l'algorithme génétique. Le profils des meilleurs candidats étant stockés au fil des générations, il s'agissait alors ici d'évaluer les avantages et les inconvénients de l'algorithme génétique.

Référence au programme :

- *Initiation aux bases de données : Projection, Sélection, Fonctions d'agrégation.*

La sixième partie permettait aux candidats de faire l'analyse critique de l'ensemble de la démarche mise en place dans le sujet.

2/ REMARQUES GÉNÉRALES

Les correcteurs ont constaté une nette progression sur la qualité et le volume des réponses données. De nombreux candidats sont ainsi parvenus à traiter l'intégralité du sujet.

Les correcteurs recommandent aux candidats d'être bien attentif aux énoncés des questions. Les réponses proposées ne traitent souvent que partiellement la question posée et les candidats perdent ainsi une partie des points associés à la question. Nous rappelons également que le soin apporté à la copie est évalué, ainsi la qualité de la rédaction et de l'écriture participent à la note finale. Nous recommandons aux candidats, lorsqu'ils se trompent, de barrer proprement l'erreur et de proposer leur nouvelle réponse à côté ou en fin de document réponse, dans l'espace dédié. L'abus de ratures ou de correcteur blanc a été systématiquement sanctionné.

La maîtrise du langage Python s'améliore dans l'ensemble, la structure du code et les indentations sont globalement respectées. Cependant, de trop nombreux candidats ne sont pas capables de réaliser une boucle proprement. La fonction « range », notamment, est souvent mal utilisée voire tout simplement oubliée. Cet outil est pourtant indispensable à la programmation en Python, et son utilisation en sens croissant/décroissant (avec les bonnes bornes) doit être maitrisée.

3/ CONSEILS AUX CANDIDATS

Les correcteurs conseillent aux candidats d'arriver avec une meilleure maîtrise des outils de base de la programmation en langage Python :

- déclaration et appel (partiel) de liste, fonctions usuelles associées aux listes (.append, .pop…) ;
- utilisation de la fonction « range » pour itérer les boucles « for » ;
- manipulation des chaines de caractères.

Les correcteurs regrettent que de nombreux candidats fassent appel à des fonctions exotiques issues des bibliothèques Python. Dans la majorité des cas, elles ne sont de toutes façons pas maîtrisées. Dans l'ensemble, nous recommandons d'aller au plus simple et de ne pas chercher les complications inutiles.

4/ REMARQUES SPÉCIFIQUES

Q1 : cette question élémentaire ne demandait que de lire la documentation.

Q2 : cette question, également simple, a été traitée par la majorité des candidats. Étonnamment, un certain nombre n'a pas su écrire correctement le numérateur et le dénominateur du correcteur.

Q3 : peu de candidats ont compris la fonction des trois boucles imbriquées et ont proposé une analyse correcte.

Q4 : question bien réussie, toutes les solutions correctes étaient acceptées.

Q5 : les candidats ont généralement bien réussi cette question et ont compris qu'il suffisait de reprendre les deux fonctions précédemment définies.

Q6 : cette question a posé des problèmes à une majorité de candidats. Peu ont répondu correctement à cause d'une mauvaise définition des dimensions des listes, des indices de boucles, de l'imbrication des boucles. Quelques candidats ont cru qu'il suffisait de reproduire les boucles de la question Q3 pour résoudre le problème !

Q7 : une majorité de candidats a bien compris l'utilisation des deux fonctions de la librairie scipy.signal -roots() et real()- et a proposé une fonction vérifiant la stabilité. Certains n'ont pas compris que les deux fonctions renvoyaient une liste dont il fallait vérifier tous les termes, d'autres ont seulement testé les racines négatives en sortant à la première rencontrée.

Q8 : question en général bien réussie.

Q9 : plus de la moitié des candidats n'a pas répondu à cette question relativement complexe. Il ne suffisait pas ici de reprendre la condition de la boucle précédente !

Q10 : question a priori assez simple (recherche d'un maximum), toutes les réponses correctes étaient admises, de l'utilisation de la fonction max() au tri à bulles pour un candidat.

Q11 : cette question classique n'a pas été traitée par une grande majorité de candidats qui devaient proposer une méthode d'intégration et la mettre en œuvre.

Q12 : une question simple qui demandait seulement de lire et comprendre une formule mathématique, question qui a été la moins traitée du sujet.

Q13 : hormis les candidats qui écrivent les formules mathématiques sans respecter la syntaxe Python, la première partie de la question a été bien traitée. En revanche, le nombre de combinaisons est souvent faux.

Q14 : question en général bien traitée.

Q15 : les candidats qui ont correctement répondu à la question Q13 ont en général bien abordé cette question.

Q16 : question plutôt bien traitée, néanmoins beaucoup de candidats oublient de répondre à l'ensemble des items de la question (rôle de la fonction, type de renvoi et valeur de n).

Q17 : le « range » nécessaire à la boucle « for » est souvent omis, très peu de candidats ont pensé à créer une liste de 3 gènes.

Q18 : question abordée par la majorité des candidats, mais la plupart n'ont pas pris en compte l'inversion de l'ordre des bits dans « b2 ».

Q19 : cette question a permis de mettre en avant les candidats capables de réaliser un algorithme simple reposant sur une boucle. Il ne fallait pas oublier de convertir les caractères de « b2 » en entier.

Q20 : il fallait ici identifier un tri par insertion qui semble mal connu. De nombreux candidats ont proposé le tri bulle qui n'est pourtant pas au programme d'IPT. La complexité était demandée, dans le meilleur et le pire des cas : les candidats proposant une complexité sans préciser le cas de figure ont été pénalisés.

Q21 : cette question était simple, elle s'est pourtant révélée complexe pour les candidats ne maîtrisant pas l'appel partiel des éléments d'une liste.

Q22 : les candidats qui ont abordé la question ont, dans l'ensemble, compris la fonction mais ont oublié que la numérotation des listes en Python commençait à l'indice 0.

Q23 : moins de la moitié des candidats a obtenu des points à cette question, qui consistait à recopier et à réadapter 4 lignes d'instruction.

Q24 : cette question, plus ouverte, a permis de mettre en avant les candidats capables de proposer un algorithme basé sur une double boucle avec condition « if ».

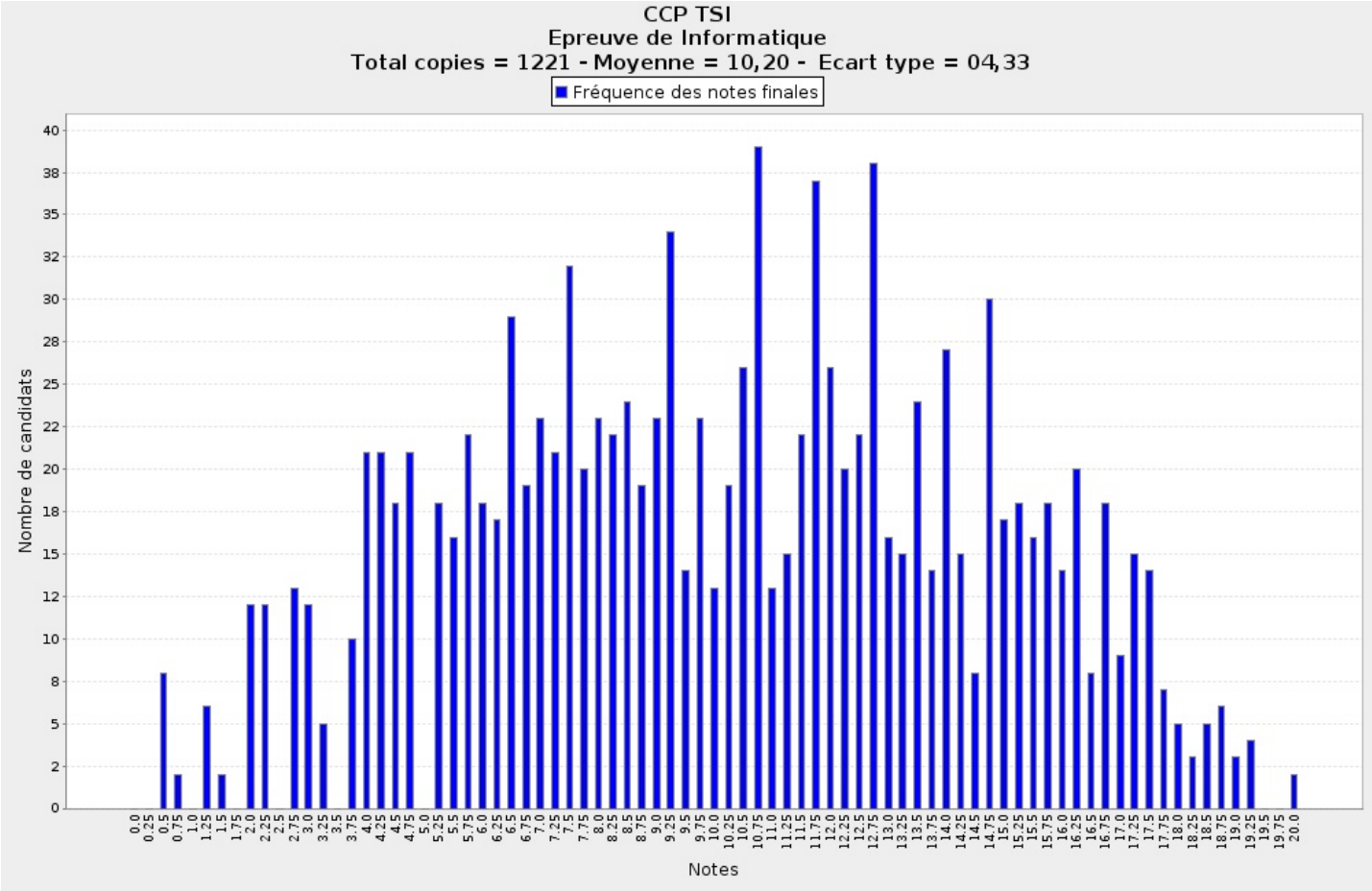
Q25 : première question sur les bases de données, largement abordée par les candidats. Elle a été, dans l'ensemble, correctement traitée.

Q26 : cette deuxième question a posé beaucoup plus de difficultés, de nombreux candidats ne l'ont pas compris et les autres ont eu du mal à réaliser cette requête avec requête imbriquée.

Q27 : question traitée par la majorité des candidats. Beaucoup ont compris la requête à l'envers et ont conclu qu'il n'y avait aucun renvoi en justifiant qu'aucun candidat ne pouvait disparaître avant d'apparaitre.

Q28 : cette question a été dans l'ensemble correctement traitée par les candidats ayant compris la question 27.

Q29 : cette dernière question demandait aux candidats de prendre un peu de recul sur la démarche mise en place dans le sujet. Il était demandé de conclure sur la pertinence de l'emploi d'un algorithme génétique afin d'optimiser le réglage d'un correcteur PID.


CCP TSI
Epreuve de Informatique
Total copies = 1221 - Moyenne = 10,20 - Ecart type = 04,33
Fréquence des notes finales
Nombre de candidats
Notes