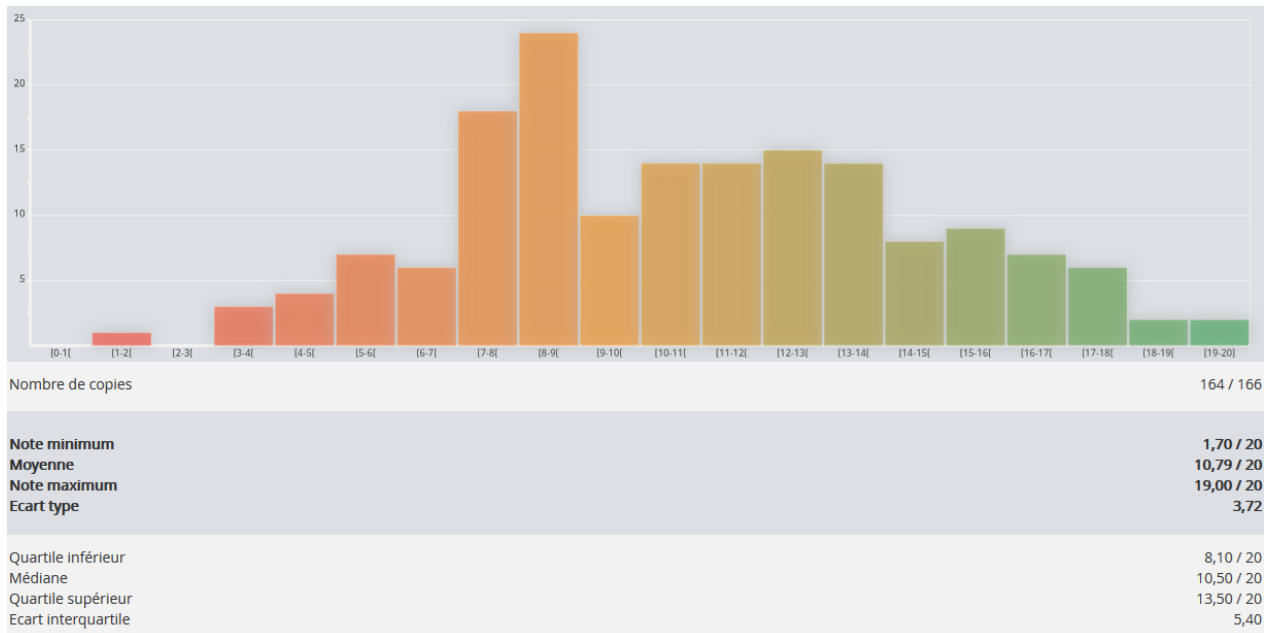


Statistiques de l'épreuve



Généralités

Cette année le sujet portait sur la recherche de chemin de coût minimal ; il implantait une approche par programmation dynamique.

Le sujet a été conçu pour être progressif et accessible, alternant des questions de compréhension de l'algorithme, d'écriture de code élémentaire, de QCM, et des questions plus techniques de complétion de code.

Cependant une réussite parfaite du sujet en 45 minutes était un exercice difficile témoignant d'un bon niveau en algorithmique et programmation ; certains candidats y sont presque parvenus.

La plupart des candidats sont parvenus à terminer le sujet ; les notes s'étalent de 1,7 à 19/20, avec une moyenne d'environ 10,8, une médiane à 10,5, un écart-type de 3,7 et un écart interquartile de 5,4.

Le sujet était bien adapté à la durée de l'épreuve ainsi qu'au niveau des candidats, il portait sur des compétences soulignées dans les rapports des années précédentes (utilisation des listes, implantation de matrices à l'aide de liste, parcours de listes, implantation de relations de récurrence).

Les candidats semblent avoir été préparés de manière sérieuse à l'épreuve, en s'entraînant sur les sujets posés les années précédentes. Ils ont globalement démontré leur capacité à s'appropriier en un temps très limité une problématique complexe, et à retranscrire sa résolution en langage de programmation.

Cependant bon nombre de candidats éprouvent des difficultés à écrire un code syntaxiquement correct répondant à une problématique simple ; les erreurs de code sont nombreuses, et c'est essentiellement ce point qui discrimine les mauvaises copies des meilleures. A contrario, les meilleurs candidats (premier quartile) y parviennent, quand les tous meilleurs candidats (premier

décile) font preuve d'un très bon niveau en programmation.

En résumé le bilan à tirer du déroulement de l'épreuve de cette année est globalement satisfaisant. Candidats et préparateurs sont encouragés à maintenir leur niveau d'exigence dans la préparation de l'épreuve. Les domaines perfectibles concernent l'écriture de code syntaxiquement et logiquement correct. Notamment sur les points suivants qui posent problème à de nombreux candidats :

- L'indexation correcte de la fonction range au sein d'une boucle for. Elle pose problème à l'immense majorité des candidats. Par contre certains candidats mettent à profit la propriété des listes d'être itérables pour implanter une boucle for sans utiliser la fonction range, et le jury ne peut qu'encourager cette approche qu'il avait conseillée dans un rapport précédent.
- L'ajout d'un élément en fin de liste. Rappelons à ce sujet qu'il faut privilégier l'usage de la méthode append. Les instructions $L += [a]$ et $L = L + [a]$ sont moins efficaces que $L.append(a)$, et en outre une majorité de candidats y oublient les crochets.
- Parcours de liste au sein d'une boucle for, d'une liste de listes au sein de 2 boucles for imbriquées.
- Le bon usage des connecteurs logiques and, or, not.

Par parties

Partie 1.

- a) Question portant sur l'accès aux éléments d'une liste de listes. Elle a été plutôt largement bien réussie. Certaines confusions (avec range ?) : $len(A)$ renvoie le nombre N d'éléments, de A , il ne renvoie pas $N-1$.
- b) QCM (fonction renvoyant une matrice nulle). Elle n'a pas toujours été très bien réussie.
- c) Ecriture de code : inversion d'une liste. Les erreurs de code sont pléthoriques ; problème d'indexation dans la plupart des réponses. Remarquons qu'il était préférable que la fonction crée **une nouvelle liste en ne détruisant pas la liste passée en argument.**

Partie 2.

- a) Question portant sur l'implantation des matrices. Elle a été très bien réussie par l'immense majorité des candidats.
- b) Implantation « à la main » de l'algorithme. Réussie par les meilleurs candidats.
- c) Complétion du code de l'algorithme. Probablement la question la plus difficile, et celle rapportant le plus de points. Elle a été réussie seulement par les meilleurs candidats, bien que la notation échelonnée ait permis au plus grand nombre d'y retirer des points.

Partie 3.

- QCM : recherche du chemin solution. Réussite variable. Beaucoup de candidats n'ont fait que tenter leur chance à ce dernier QCM.



CONCOURS A TB - 2020

RAPPORT DE L'ÉPREUVE ÉCRITE ALGORITHMIQUE ET INFORMATIQUE

Conclusion

Comme les années précédentes, nous conseillons aux candidats, de bien lire l'énoncé pour assimiler correctement le problème. Il n'y a pas de "question piège", et les questions les plus accessibles figurent généralement en début de sujet. Tenter de glaner des points en s'aventurant jusqu'en fin du sujet sans aborder les premières questions s'avère très improductif. Une aptitude correcte en algorithmique et programmation est nécessaire et ne peut s'acquérir que par une écriture de code régulière et investie durant les 2 ou 3 années de préparation.