

**RAPPORT DE L'ÉPREUVE ÉCRITE ALGORITHMIQUE ET INFORMATIQUE**

Cette année le sujet portait sur la recherche de sous-chaines communes de longueurs maximales dans des couples de chaînes de caractères. Ce problème algorithmique présente un intérêt pratique en phylogénétique, pour réaliser un indicateur de proximité génétique entre deux génotypes. Ce point était indiqué dans le préambule et toutes les chaînes données en exemple étaient constituées des seuls caractères A, T, C, G.

L'objectif *in fine* était ambitieux, une réalisation efficace étant implantée par une approche par programmation dynamique, cependant le sujet a été conçu pour être accessible et très progressif, débutant par des questions faciles : compréhension sur des exemples simples, écriture de code élémentaire, et QCMs. Les questions plus techniques de complétion du code de programmation dynamique ne survenaient que dans la dernière partie ; le sujet s'achevait par des questions sur l'évaluation de la complexité de l'algorithme et ses possibles adaptations. Le barème de correction a été établi, de façon à ce que ces dernières questions fassent office de bonus et soient évaluées comme des questions ouvertes (un exposé rigoureux de la notion de complexité n'étant pas un attendu du programme officiel).

L'essentiel des candidats a retiré des points sur les questions les plus faciles portant sur la compréhension du problème et l'écriture de code élémentaire. Les meilleurs candidats ont été départagés sur les questions les plus difficiles en fin de sujet. Les candidats les plus faibles (quartile inférieur) n'ont pas bien compris la notion de sous-chaines communes de longueurs maximales, malgré une définition rigoureuse et les exemples donnés en illustration dans le sujet. Ils ont de plus montré des lacunes en programmation.

Cette année, aucun candidat n'est parvenu à traiter parfaitement et dans sa totalité le sujet. Cependant la plupart des candidats ont pu traiter toutes les questions dans le temps imparti.

Les notes s'étalent de 0,7 à 17,8 sur 20 avec une moyenne proche de 10 et un écart-type de 3,6.

En résumé, le sujet s'est avéré adapté à l'épreuve et a permis de correctement départager les candidats.

Les meilleurs candidats sont parvenus en 45 minutes à analyser une problématique complexe et à en approcher une résolution efficace en étant guidés par le sujet.

|                          |           |
|--------------------------|-----------|
| Nombre de copies         | 150 / 150 |
| Copies non traitées      | 0 / 150   |
| En dessous de la moyenne | 79        |
| Au dessus de la moyenne  | 71        |

|                   |                   |
|-------------------|-------------------|
| <b>Minimum</b>    | <b>0,70 / 20</b>  |
| <b>Moyenne</b>    | <b>9,83 / 20</b>  |
| <b>Maximum</b>    | <b>17,80 / 20</b> |
| <b>Ecart type</b> | <b>3,62</b>       |

|                     |            |
|---------------------|------------|
| Quartile inférieur  | 7,40 / 20  |
| Médiane             | 9,60 / 20  |
| Quartile supérieur  | 12,60 / 20 |
| Ecart interquartile | 5,20       |

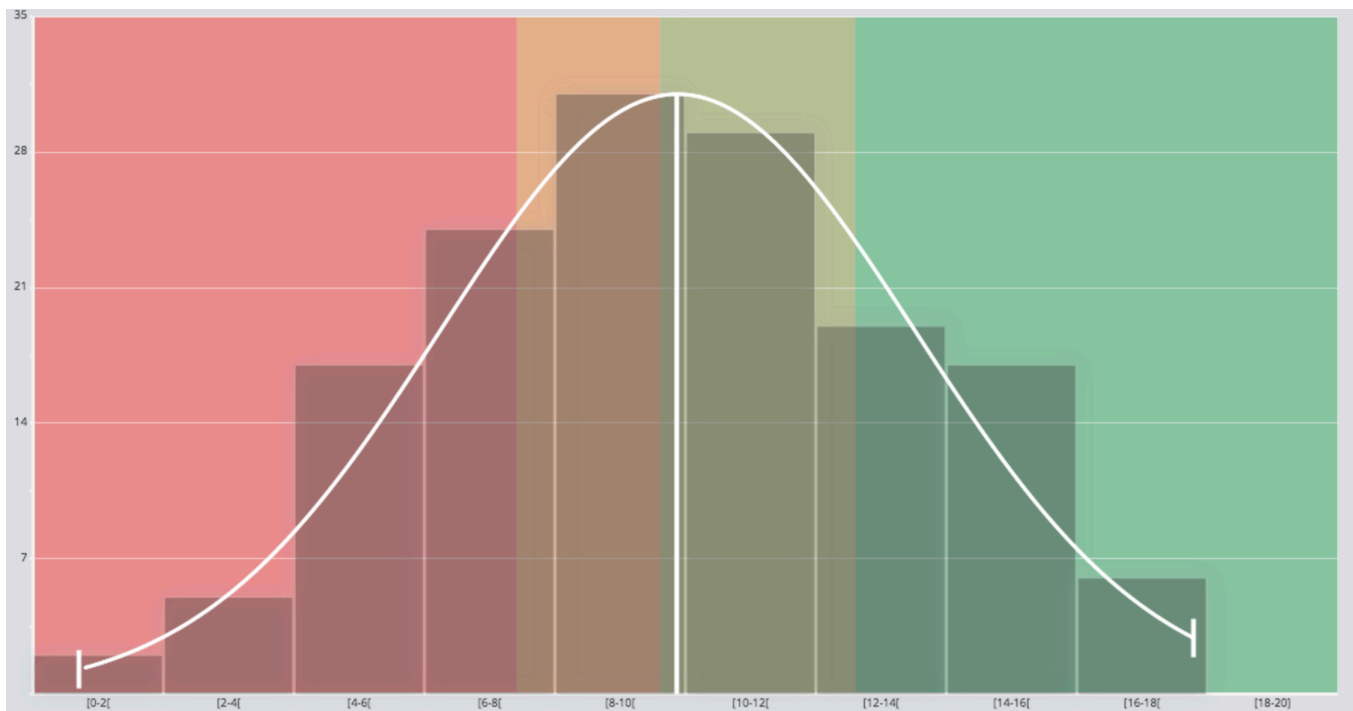


FIGURE 1. Histogramme des notes finales

## Dans le détail :

### Partie 1.

Elle consistait en 2 questions d'écriture de code élémentaire : renvoi du maximum de deux nombres, et construction d'une matrice nulle (sous forme de liste) dont les nombres de lignes et de colonnes était passés en paramètres.

La première question, très facile a été bien réussie ; elle ne consistait qu'en l'usage d'un test.

La réussite à la deuxième question a été beaucoup plus disparate ; l'usage de deux boucles for imbriquées et les déclaration et remplissage d'une liste posant problème à nombre de candidats. Notons que certains candidats ont fait usage de fonctions du module numpy ; cependant un tableau numpy n'est pas une liste, et en l'absence de conversion de type, ils n'ont pas obtenu la totalité des points à la question.

### Partie 2.

Les questions a), b), c), d) portaient sur la compréhension de la notion de plus longue sous-chaîne commune. Très accessibles, seuls les moins bons candidats (dernier quartile) n'ont pas répondu correctement, au moins partiellement, à ces questions.

La question e), question à choix multiple, a été relativement peu réussie. Rares sont les candidats ayant apporté les deux bonnes réponses. Notons que pour dévaluer des réponses données au hasard, les deux bonnes réponses (sur 4) rapportaient 3 points, tandis qu'une seule bonne réponse sur deux ne rapportait qu'1 seul point.

La question f) d'écriture de code sur les chaînes de caractère a été relativement mal réussie ; elle se voulait pourtant accessible.

**Partie 3.**

Elle portait sur la résolution par programmation dynamique. La solution était donnée sous forme d'une relation de récurrence, et il s'agissait essentiellement de montrer qu'elle avait été comprise et de compléter le code l'implantant.

La question a) qui demandait la taille de la matrice nécessaire au stockage a été mal réussie ; rappelons que de 0 à l'entier  $n$  inclus ( $0 \leq n$ ) il y a  $(n + 1)$  entiers consécutifs.

La question b), de complétion de code à proprement parler, n'a été réussie que par les meilleurs candidats. C'était l'une des questions les plus difficiles (mais aussi des plus payantes) ; la notation a valorisé une compréhension même partielle de l'algorithme.

La question c) d'illustration sur un exemple a inspiré nombre de candidats ; seules les meilleurs ont répondu correctement à la question.

La question d) d'approfondissement était une question ouverte difficile. Très peu de candidats y ont retiré des points, et aucun en totalité.

De même la question e) portant sur la complexité n'a rapporté des points qu'à très peu de candidats, beaucoup ne l'ayant pas comprise.

**En conclusion.**

On pourra donner comme conseils aux futurs 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 un usage régulier et investi durant les 2 ou 3 années de préparation.