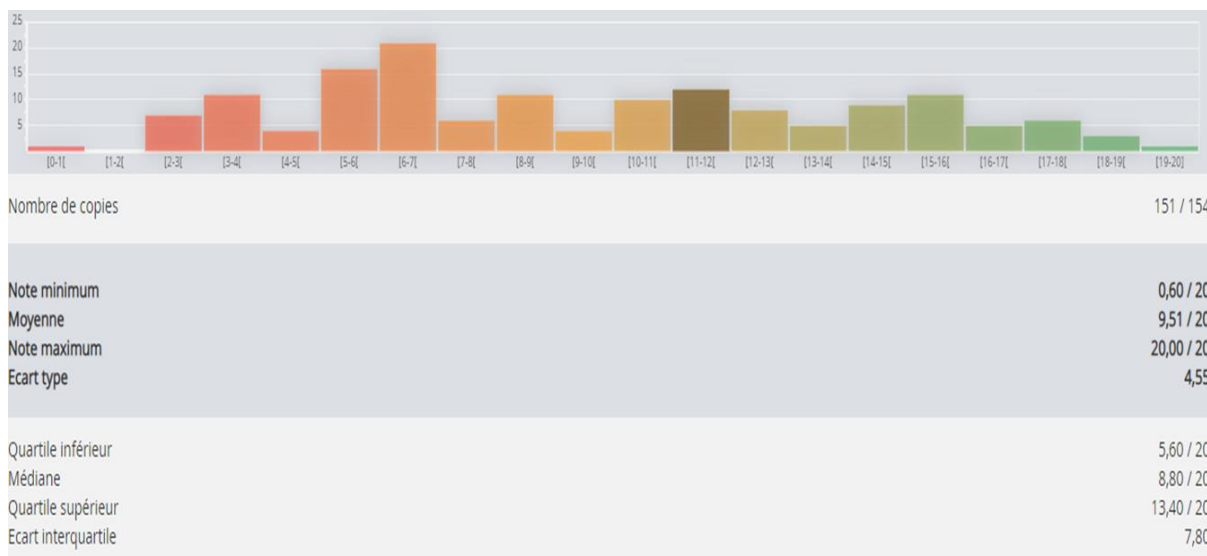


## CONCOURS A TB - 2021

### Rapport de l'épreuve écrite d'algorithmique et informatique

#### Statistiques de l'épreuve



Cette année le sujet concernait la recherche de coupe de somme minimale, c'est à dire d'une sous-liste contiguë d'une liste numérique dont la somme des éléments soit minimale ; il se proposait d'implanter trois approches, une naïve, une intermédiaire, et une efficace par programmation dynamique.

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

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, ce qui démontre que sa longueur était bien adaptée ; les notes s'étalent de 0,6/20 à 20/20, avec une moyenne d'environ 9,5 ; une médiane à 8,8 ; un écart-type relativement important de 4,5 ainsi qu'un écart interquartile de 7,8.

Le sujet portait sur des compétences soulignées dans les rapports des années précédentes (utilisation des listes, parcours de liste, extraction de sous-liste, calcul de la somme des termes d'une sous-liste, implantation de relations de récurrence). Il ressemblait d'ailleurs au sujet de l'année précédente puisqu'assez voisin d'une version unidimensionnelle du problème posée alors, qui consistait à chercher des chemins de coût minimal dans une matrice.

Cette année, relativement à l'an dernier le niveau des copies était globalement assez décevant, et plus hétérogène. On pourra l'interpréter comme une conséquence de la crise sanitaire. Durant les divers confinements l'enseignement des TP d'Informatique, plus encore que celui des autres matières, a été grandement compliqué, voire même restreint.

Les meilleurs candidats ont globalement démontré leur capacité à s'approprier en un temps très limité une problématique complexe, et à retranscrire sa résolution en langage de programmation. Les candidats moins

## CONCOURS A TB - 2021

### Rapport de l'épreuve écrite d'algorithmique et informatique

bons, quant à eux, semblent ne s'être pas tous suffisamment entraînés sur les sujets posés les années précédentes.

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. Cette année de nombreux candidats n'ont pas correctement saisi la problématique exposée.

En résumé le déroulement de l'épreuve de cette année est satisfaisant, mais l'impact de la crise sanitaire paraît perceptible sur le niveau global des candidats, que le jury a jugé assez moindre par rapport à l'année précédente ; vu le caractère exceptionnel du contexte, il est difficile de tirer un bilan définitif. Candidats et préparateurs sont encouragés à maintenir un bon niveau d'exigence dans la préparation de l'épreuve. Cette année de nombreux candidats semblaient manquer de préparation. Pour les autres, les domaines perfectibles concernent l'écriture de code syntaxiquement et logiquement correct ; notamment sur les points suivants :

- 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 lors de rapports précédents.
- L'ajout d'un élément en fin de liste. Rappelons à ce sujet qu'il faut privilégier l'usage de la méthode *append*. L'instruction  $L = L + [a]$  est à proscrire (car beaucoup moins efficace) et l'instruction  $L += [a]$  à déconseiller (une majorité de candidats y oubliant les crochets) au profit de  $L.append(a)$ .
- Parcours de liste au sein d'une boucle *for*, somme de termes consécutifs d'une liste numérique.
- Recherche de minimum, et ses variantes, comme la recherche d'indices où il est atteint.
- Portée des variables : variables locales ou globales.
- Le bon usage des connecteurs logiques *and*, *or*, *not*.

#### Dans le détail.

#### PARTIE 1.

- a) Question portant sur la compréhension du contexte. Seulement la moitié des candidats y ont obtenu tous les points. Beaucoup de candidats n'ont pas compris que pour qu'une coupe soit de somme minimale il pouvait être nécessaire de passer par des éléments positifs afin d'atteindre ensuite des valeurs négatives. Une réponse cohérente, même erronée, a rapporté tout de même la moitié des points.
- b) Ecriture de code : calcul de la somme des éléments d'une sous-liste d'une liste numérique. Cette question a été mal réussie, moins de la moitié des candidats y ayant obtenu tous les points, alors qu'il s'agit d'une compétence élémentaire. Le fait que la somme se fasse de l'indice  $i$  à l'indice  $j$  a posé beaucoup de difficulté aux candidats ; souvent les candidats ont maladroitement itéré deux boucles *for*.

## CONCOURS A TB - 2021

### Rapport de l'épreuve écrite d'algorithmique et informatique

---

#### **PARTIE 2. Résolution par force brute.**

- a) QCM ; un peu plus de la moitié des candidats a fourni la bonne réponse.
- b) Code d'une fonction renvoyant les indices de début et de fin d'une coupe de somme minimale. Il suffisait d'adapter le code de la question précédente pour qu'il ne renvoie non plus la valeur minimale mais les indices où elle est atteinte. La question a été réussie seulement par 20 % des candidats. La plupart ont vu le lien avec la fonction précédente, mais n'ont pas su l'adapter ; beaucoup d'erreurs ici ont découlé d'une méconnaissance de la portée des variables (*i.e.* locales/globales).

#### **PARTIE 3. Résolution par les suffixes.**

- a) Ecriture de code : restriction du problème aux coupes débutant à l'indice  $i$  : la question a été très mal comprise puisqu'un tiers des candidats n'y ont pris aucun point, quand moins de 10 % ont réussi la question. Il s'agissait encore simplement d'une recherche de somme minimale sur le modèle de la question 2.a), il fallait juste l'exécuter au sein d'une seule boucle *for*, en débutant à l'indice  $i$ , et effectuer le calcul de la somme simultanément puisqu'on interdisait de recourir à la fonction somme.
- b) Complétion de code : c'est la question la mieux réussie. Presque deux tiers des candidats y ont pris les points.

#### **PARTIE 4. Résolution par programmation dynamique.**

- a) Question de compréhension. Il s'agissait d'initialiser en déterminant les valeurs de  $x_l$  et  $y_l$ . La question a été réussie par seulement 20% des candidats.
- b) Complétion de code : implantation de la programmation dynamique. Il s'agissait d'implanter l'initialisation et les relations de récurrence des deux suites  $(x_i)$  et  $(y_j)$ . Sans surprise la question, a été très mal réussie (c'était probablement une des plus difficiles). Plus d'un tiers des candidats n'y ont pris aucun point, et seuls 3 (des meilleurs) candidats ont réussi presque parfaitement la question.

Comme les années précédentes, nous conseillons aux candidats, de bien lire l'énoncé pour assimiler correctement le problème. La difficulté de l'épreuve consiste d'abord en la compréhension en un temps très limité d'une problématique complexe et d'algorithmes de résolution, ensuite en leur implantation informatique. Un candidat réussira mieux l'épreuve en comprenant bien le problème et en se restreignant à des premières questions bien traitées, qu'en tentant de glaner des points tout au fil du sujet (ce qui s'avère toujours très improductif !). Il n'y a pas de "question piège", et les questions les plus accessibles figurent généralement en début de sujet. 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. Une bonne préparation devrait passer par un entraînement sur les sujets posés les années précédentes.