CONCOURS A-TB - 2022

Rapport de l'épreuve écrite d'Algorithmique et informatique

Cette année le sujet portait sur la recherche de sous-chaîne dans une chaîne de caractère. Les applications d'un tel algorithme sont très nombreuses ; citons par exemple les moteurs de recherche internet, plus généralement la recherche de mots-clés dans un document numérique, ou la recherche de gênes dans une séquence ADN.

Le sujet revenait d'abord sur l'algorithme naïf figurant au programme, qui consiste à rechercher la sous-chaîne à toutes les positions, en comparant caractère après caractère et qui s'implémente grâce à deux boucles imbriquées.

Il présentait ensuite un algorithme plus efficace, celui de Knuth-Morris-Pratt (KMP), que le sujet proposait de programmer. Cet algorithme astucieux suit la même démarche que le précédent, mais en réduisant considérablement (le plus souvent) le nombre de positions où rechercher la sous-chaîne : il n'avance plus, lorsque la recherche aura échoué à l'indice i, la recherche suivante à l'indice i+1, mais à l'indice i+k, où k est déterminé grâce à la notion de bord des préfixes de la sous-chaîne à rechercher.

Pour illustrer l'idée, la recherche de ATCATG dans la chaine ATCATCATG aura échouée en position i=0, à cause de la discordance des 6e caractères G et C, mais l'existence des préfixes et suffixes communs AT de longueur 2 dans les 5 premiers caractères non discordants ATCAT de la sous-chaîne, permet de décaler la recherche suivante à la position i+5-2=i+3. Sur cet exemple k = 3 ; et il se calcule en soustrayant à la longueur du plus long préfixe concordant, la longueur de son bord, c'est-à-dire du plus long de ses préfixes qui en soit aussi un suffixe. À cette position, sur cet exemple, la recherche aboutira.

La recherche des longueurs des bords de tous les préfixes du mot à chercher, sera faite initialement, dans un algorithme dit « de prétraitement » du mot.

Remarquons que David Knuth, co-auteur de l'algorithme (1970), est un pionnier majeur de l'Informatique théorique et de l'Algorithmique de la deuxième moitié du XXe siècle.

Le sens du sujet consistait, *in fine*, à présenter cet algorithme, à l'illustrer sur des exemples, et à la programmer.

Le sujet faisait appel à des compétences de base du programme officiel, essentiellement articulées cette année autour des chaînes de caractères. Bien qu'ambitieux, il a été conçu pour être progressif et accessible en 45 minutes. Il comportait des questions de compréhension, d'écriture de code plus ou moins élémentaire, de QCM, et des questions de complétion de code.

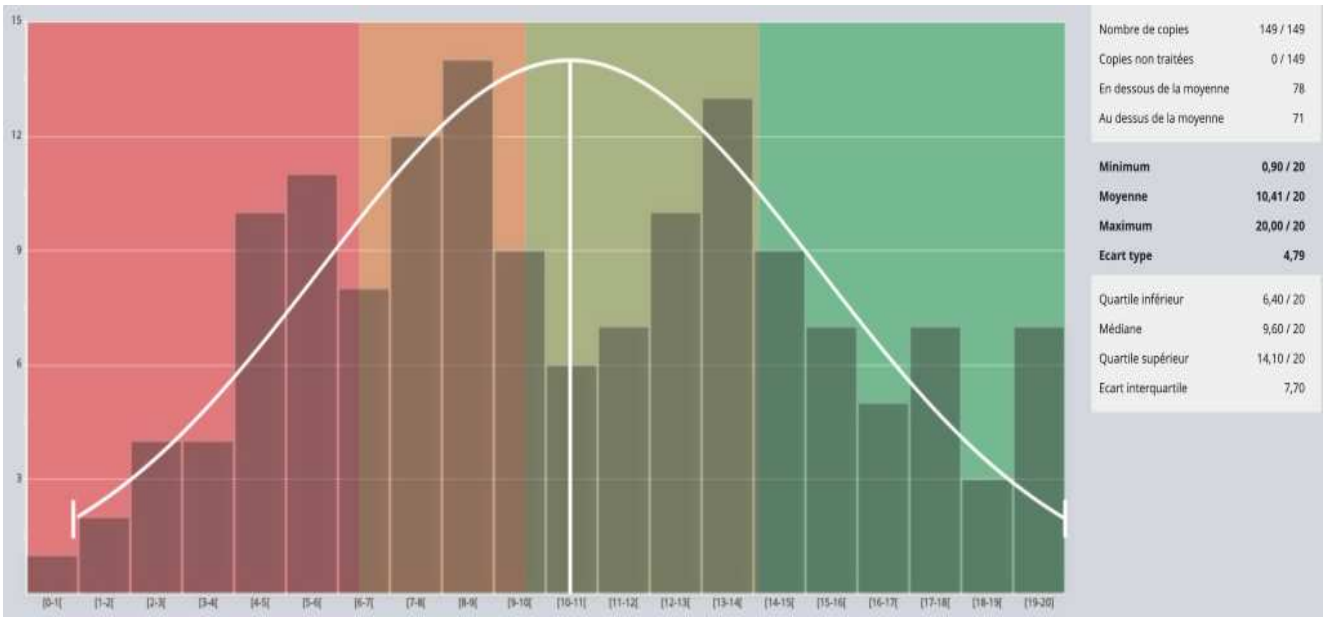
Sa difficulté ne semblait pas moindre que celle de l'année précédente, qui pour mémoire n'avait pas été parfaitement bien réussi. Sa réussite fût cependant meilleure, confirmant que l'écueil précédent semble relever essentiellement de la crise sanitaire et du manque de pratique des candidats qui en a découlé.

Cette année encore la plupart des candidats sont parvenus à terminer le sujet, montrant que sa longueur était bien adaptée à une épreuve de 45 minutes.

CONCOURS A-TB - 2022

Rapport de l'épreuve écrite d'Algorithmique et informatique

On a constaté une hétérogénéité marquée du niveau des candidats, une part de niveau relativement faible, celle des candidats ne sachant pas écrire un code répondant à une problématique simple, mais une part importante aussi de bons à très bons candidats qui ont su en un temps très limité comprendre un algorithmique complexe et l'implanter. S'approprier en 45 minutes un algorithme hautement non-trivial et l'implanter correctement constitue une belle performance ; c'est à souligner.



Les notes s'étalent de 0,9 à 20 sur 20, avec une moyenne d'environ 10,4, et un écart-type important de 4,8 ; la médiane est à 9,6 et les quartiles inférieurs et supérieurs à 6,4 et 14,1.

La distribution observée dénote une part importante de candidats de bon à très bon niveaux mais aussi de candidats de niveau nettement inférieur aux attentes.

En résumé le déroulement de l'épreuve de cette année est satisfaisant. Candidats et préparateurs sont encouragés à maintenir un bon 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 :

- L'indexation correcte de la fonction *range* au sein d'une boucle *for*. Elle pose problème à l'immense majorité des candidats. On peut parfois mettre à profit la propriété des listes, chaînes de caractères et autres séquences, d'être itérables pour implanter une boucle for sans utiliser la fonction *range*, et le jury persiste à encourager cette approche, lorsqu'elle est possible.

- 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 seul profit de *L.append(a)*.

- Parcours de liste ou de chaînes de caractère, au sein d'une boucle *for*, somme de termes consécutifs d'une liste numérique.
- Usage convenable de l'extraction de liste ou chaînes : *C[i :j]* extrait la sous-séquence de *C* allant de l'indice *i* inclus à l'indice *j* exclu.
- 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*.

Le jury insiste sur le fait que ce qui discrimine les bonnes copies des moins bonnes, c'est l'écriture de code algorithmiquement et syntaxiquement correct, même sur des algorithmes très basiques ! C'est d'évidence le point à travailler en priorité : coder. Une réussite convenable à l'épreuve ne peut se passer d'une pratique régulière, investie et rigoureuse de la programmation par les candidats durant leurs 2 à 3 années de préparation.

Les candidats peinent aussi largement à identifier qu'un code écrit ou donné dans une question intermédiaire est en général appelé à être utilisé dans les questions suivantes ! C'est un écueil sur lequel nous conseillons aux préparateurs d'insister lourdement.

Dans le détail.

PARTIE 1. Recherche de sous-chaîne par force brute.

On revenait ici sur l'algorithme de recherche de sous-chaîne figurant au programme.

1. Question facile sur la compréhension de la notion d'indice. Hormis quelques décalages, question bien réussie.

2. QCM : Question relativement bien réussie.

3. Écriture de code : Question assez mal réussie ; la distribution est très (trop) hétérogène (répartition des notes uniforme) sur un algorithme très accessible en l'état, et figurant au programme officiel.

4. Écriture de code : Même remarque. L'écueil provient du fait que les candidats peinent à comprendre que les codes donnés ou écrits précédemment sont là pour être utilisés ; et ce malgré même l'usage dans la question de la locution « En déduire »…

CONCOURS A-TB - 2022

Rapport de l'épreuve écrite d'Algorithmique et informatique

PARTIE 2. Bord d'une chaîne de caractère.

Il s'agissait de comprendre la notion de bord, et d'implémenter l'algorithme de pré-traitement de la chaîne à chercher.

1. Un peu plus de la moitié des candidats ont réussi la question, montrant leur compréhension de la notion de bord qui était exposée.

2. Complétion de code ; elle portait essentiellement sur la notion d'indice, d'extraction de chaînes, et l'algorithme de recherche d'un maximum. Sa réussite fût très corrélée à celle de la question précédente, c'est-à-dire à la compréhension de la notion de bord.

3. Écriture de l'algorithme de prétraitement. Là encore il suffisait d'appeler adroitement la fonction écrite dans la question précédente. Réussite très hétérogène et assez décevante.

4. On demandait le résultat renvoyé par la fonction précédente sur un exemple. Question assez bien réussie.

PARTIE 3. Recherche de sous-chaîne par l'algorithme de Knuth-Morris-Pratt.

On exposait ici l'algorithme, on l'illustrait sur un exemple avant de demander au candidat de compléter son code.

1. La distribution des notes est encore ici très hétérogène, avec une distribution des notes quasiment uniforme. C'est, sur cette question, peu surprenant, puisque clôturant le sujet, c'était probablement la plus difficile à réussir.