Informatique

Présentation du sujet

Le sujet porte sur la mesure de la raideur d'un brin d'ADN ainsi que la simulation numérique du comportement.

Après une première partie préliminaire sur des algorithmes classiques itératifs et récursifs, la seconde partie vise à réaliser un traitement d'image permettant de mesurer les fluctuations de positions et l'allongement du brin en fonction de la tension exercée. Cette partie mobilise des compétences en algorithmique.

La troisième partie permet d'identifier les paramètres du modèle du ver à partir des données expérimentales par minimisation des écarts. Cette partie s'appuie sur les compétences en simulation numérique au programme.

La quatrième partie propose une modélisation simple du brin sous forme d'une succession de segments, dont le comportement statistique permet de retrouver par simulation un comportement proche de celui du brin d'ADN. Bien qu'il s'agisse d'une simulation numérique, les compétences évaluées relèvent pour l'essentiel de l'algorithmique.

Analyse globale des résultats

Si le sujet est assez dense pour le niveau des étudiants, certains étudiants ont pu traiter l'ensemble des questions.

Le jury se réjouit du niveau des copies qui progresse. Le langage Python est bien mieux maitrisé. Quelques rares candidats ont visiblement négligé la formation en informatique et se contentent de répondre aux questions ne relevant pas immédiatement d'informatique.

Les petites erreurs syntaxiques n'ont pas été retenues par le jury comme un élément discriminatoire, dans la mesure où elles ne cachent pas des erreurs de fond. Les réponses pertinentes d'un point de vue algorithmique sont valorisées. Néanmoins, une accumulation significative de ces erreurs dans certaines copies a pu conduire à une dépréciation.

Certaines copies proposent des programmes particulièrement élégants et concis, et reflètent une réelle prise de recul sur les différentes stratégies de programmation. Ces copies ont été valorisées.

Commentaires sur les réponses apportées et conseils aux futurs candidats

Au regard des copies évaluées, le jury propose aux futurs candidats de prêter attention aux remarques suivantes.

L'indentation en Python délimite les blocs d'instructions et doit apparaitre clairement dans la rédaction. Toute rédaction claire est bienvenue ; bien souvent, un trait vertical marquant l'alignement du bloc d'instruction est suffisant.

L'initialisation d'une variable dans une boucle ou hors de la boucle n'a pas les mêmes conséquences pour l'algorithme.

Le nombre d'itérations d'une boucle doit être bien réfléchi pour s'assurer que les indices des éléments d'une liste appelée dans la boucle sont bien définis. L'instruction `range(n)` parcourt `n` itérations indicées de `0` à `n-1`. L'ordre des conditions après un `while` peut avoir de l'importance.

Lorsque le sujet précise explicitement les arguments des fonctions et les valeurs retournées, ainsi que leur type, il convient de veiller à les respecter.

Les listes ou les tableaux fournis en argument d'une fonction peuvent subir des modifications par effet de bord s'ils ne sont pas copiés, ce qui n'est pas toujours souhaitable. L'instruction `A = B` ne duplique pas l'objet `B`, si celui-ci est mutable (par exemple une liste ou un tableau), les modifications de `A` affecteront également `B`.

Beaucoup de questions sont indépendantes et généralement le prototype d'une fonction, donné dans une question, permet de l'utiliser dans les questions suivantes.

Les opérateurs classiques (+, *, etc.) n'ont pas toujours le même sens selon les types des opérandes (en particulier pour les listes et les tableaux.

La concision et l'élégance des programmes sont appréciées dans l'évaluation. Les candidats qui réinvestissent les fonctions déjà codées sont valorisés par rapport à ceux qui recopient les lignes de code équivalentes.

Des listes de conditions en cascade nuisent à la lisibilité de l'algorithme. Une condition booléenne bien choisie distingue les candidats dont la pensée est claire.

Des noms de variables explicites aident à la lecture du code. De trop nombreux candidats utilisent des noms de variables quelconques (`a`, `b`, `c`...) ce qui nuit à la compréhension du programme. La clarté du programme (en particulier le choix des noms de variables) ainsi que la présence de commentaires opportuns sont prises en compte dans l'évaluation.

Un long paragraphe expliquant le principe d'un algorithme est souvent moins clair qu'un code bien formulé, utilisant des noms de variables évoquant leur contenu.

Lors d'un calcul de complexité, une justification minimale est attendue.

L'ordre des questions importe. Prendre soin de rédiger les réponses aux questions en respectant leur ordre dans le sujet.

La qualité d'expression (l'orthographe notamment) et la qualité visuelle de présentation relèvent des compétences de communication indispensables à un candidat à une école d'ingénieurs. Le correcteur n'attribue les points qu'aux éléments de réponse qu'il parvient à lire et à comprendre. Les copies obscures et difficiles à comprendre sont pénalisées.

Les variables utilisées dans une fonction doivent être définies dans cette fonction ou être explicitement définies comme variables globales (soit par le sujet, soit par le candidat).

Les candidats sont invités à lire attentivement l'annexe contenant certaines fonctions utiles pour traiter le sujet.

I Fonctions utilitaires

La première partie comporte trois questions préliminaires, participant à la progressivité du sujet. Les deux premières questions sur la moyenne et la variance sont réussies par la majorité des candidats. Certains candidats utilisent la fonction moyenne dans les itérations du calcul de la variance. La question **Q3** se traite naturellement par un appel récursif de la fonction `somme`. Cette question a été peu réussie par les candidats.

II Mesures expérimentales

Les questions **Q4** et **Q6** sont très bien réussies. Lorsque la question demande de renvoyer un nouveau tableau, il convient de ne pas modifier le tableau fourni en argument. Les questions **Q5** et **Q7** sont moins bien abordées. La question **Q7** réutilise naturellement les fonctions antérieures.

Les questions **Q8** et **Q9**, plus difficiles, furent discriminantes. Les candidats avaient toute latitude pour proposer plusieurs fonctions permettant de déterminer un profil radial de l'image de diffraction. La discrétisation des anneaux, la répartition des pixels dans chaque anneau et le calcul du profil lui-même nécessitaient une approche structurée pour réussir. Bien que les candidats parvenus à une solution fonctionnelle soit rares, toutes les propositions répondant à certains points clés de l'algorithme ont été valorisées. Le calcul de complexité est rarement traité. Une réponse de complexité sans avoir fourni d'algorithme à la question **Q8** n'a jamais été pris en compte car il ne peut s'appuyer sur rien.

III Modèle du ver

La question **Q11**, visant à mettre en œuvre une fonction de bibliothèque, a posé des difficultés à la plupart des candidats. Il s'agissait de préparer les données en dissociant les abscisses et les ordonnées, préparer la fonction en créant une nouvelle fonction sans l'argument `T` et renvoyer la partie utile des résultats.

Les questions **Q12** et **Q13** ont montré un recul très relatif de certains candidats sur les nombres flottants utilisés en calcul numérique. Combien d'entre eux proposent 52 chiffres significatifs décimaux ou encore 10^{15} chiffres significatifs ! Beaucoup indiquent aussi que 10^{-16} est trop petit pour être représenté en mémoire. La représentation des données en mémoire est pourtant bien au programme.

La question **Q14** est assez bien réussie mais la question **Q15** a posé des problèmes. La fonction dérivée est souvent mal utilisée pour le calcul de la dérivé seconde. Le principe de Newton (question **Q16**) n'est pas maitrisé, cette question est la plupart du temps éludées ou mal traitée.

La question **Q17**, ne faisant pas appel à la programmation, est traitée de manière assez peu rigoureuse. La mise en œuvre au travers des programmes des questions **Q18** et **Q19** est rarement réussie.

IV Modèle de la chaine librement jointe

Les questions **Q20** et **Q21** sont plutôt bien abordées, mais obtenir un nombre aléatoire entre $-\pi$ et π semble poser des difficultés à bon nombre de candidats. La fonction `random.randrange` n'était pas adaptée pour cette question. La question **Q22** vise à créer une nouvelle conformation (et non pas modifier la conformation donnée) en changeant `k` angles successifs. Le tirage aléatoire du début de la section modifiée en s'assurant de ne pas dépasser la taille du tableau a conduit à de nombreuses erreurs.

La question **Q23** nécessitait de sélectionner une des conformations selon une probabilité donnée, ce que peu de candidats ont su faire.

La question **Q24** proposait de mettre en œuvre une simulation de type Monte Carlo. Très peu de candidats ont réussi cette question. La méthode était pourtant détaillée dans le sujet.

Conclusion

Le sujet aborde une large partie du programme d'informatique commune, dont le programme de deuxième année. Le choix d'un sujet s'appuyant la caractérisation d'un phénomène physique par une approche numérique, impliquant une part d'algorithmique, assure une cohérence avec la formation d'ingénieurs. Cette approche sera reconduite sur des problématiques de simulation ou d'algorithmique en informatique, à partir du programme des trois semestres d'informatique.

Les résultats à cette épreuve montrent que les étudiants, soutenus par leurs professeurs, ont acquis des compétences en informatique mais que leurs mises en œuvre restent difficiles. Le jury encourage les futurs candidats à travailler l'informatique en alliant réflexion sur feuille de papier et mise en œuvre des algorithmes sur ordinateur.