Informatique

Présentation du sujet

Le sujet porte sur la prévention des collisions aériennes, à travers l'étude des stratégies réellement mises en œuvre dans la gestion d'un trafic en expansion constante. Il est composé de trois parties. La première introduit la problématique, au moyen de la présentation d'une base de données sur les vols en Europe. La seconde précise l'organisation des routes aériennes autour de la notion de couloir de circulation. Enfin la dernière étudie le système effectif (*Traffic Collision Avoidance System*) utilisé pour assurer la sécurité de la phase de croisière.

Chaque partie évalue des compétences complémentaires : la première évalue la faculté à interroger une base de données au moyen de requêtes SQL. La seconde est d'essence algorithmique, avec de nombreuses évaluations de complexité ; elle valide également l'acquisition des fondamentaux du langage. La troisième laisse plus d'initiative aux candidats, en leur demandant à travers l'écriture de fonctions, de concevoir un algorithme répondant à un problème précisément posé.

Analyse globale des résultats

L'épreuve d'informatique est une épreuve simple : les meilleurs candidats ont traité de façon satisfaisante plus de 95% du problème. Elle vise à valider un socle minimum de compétences informatiques que doit posséder le futur ingénieur, mises en œuvre dans un contexte réel. Les candidats en ont bien compris l'enjeu : la moyenne brute de l'épreuve est voisine de 10/20. Les notes sont bien étalées, gage d'une évaluation de qualité.

Au delà de la correction d'un algorithme ou d'une fonction, le jury attache une grande importance à la qualité du code : respect de l'indentation et de la syntaxe, utilisation de l'expressivité du langage pour obtenir des programmes concis, introduction de variables pertinentes, choix intelligent des noms de variables, décomposition d'un algorithme en plusieurs parties naturelles, etc. sont autant de savoir-faire qui rende le code facile à lire et à comprendre. Le barème valorise ces qualités tout au long du problème.

Commentaires sur les réponses apportées et conseils aux futurs candidats

Partie I

L'écriture de requêtes sur les bases de données est une compétence relativement bien acquise. Le jury déplore que 10% des candidats fassent l'impasse sur cette partie pourtant bien pondérée. La notion d'auto-jointure, nécessaire dans la dernière question, est celle qui a posé plus de difficultés aux candidats.

Partie II

II.A, B – Ces questions demandaient d'écrire quelques fonctions itératives simples, avec une ou plusieurs boucles, et ont été bien traitées par la majorité des candidats. Le sujet demandait de donner la complexité des fonctions demandées. Si la notion est bien comprise, son expression est trop souvent maladroite (« la complexité est $9n^2 + 3n$ »), ou lourdement justifié par un décompte exhaustif des opération élémentaires qui s'étale parfois sur plusieurs pages, là où une analyse

de la structure des boucles est tout aussi convaincante et efficace. De nombreux candidats en oublient qu'une fonction de complexité linéaire dans un boucle conduit en principe à une complexité quadratique.

II.C – Les fonctions demandées étaient un peu plus longues et nécessitaient un peu plus de réflexion sur la signification des données manipulées. Les candidats ayant une lecture trop superficielle de la problématique du sujet s'y sont égarés.

II.D – La question, plus libre, exposait la mise en œuvre de la technique du recuit simulé. Il s'agit de mettre en œuvre une succession d'étapes élémentaires, répétées, qui conduisent à diminuer une fonction de cout. Tous les candidats (60%) qui se sont donné la peine de s'approprier la méthode ont fourni des solutions globalement valides et ont été récompensés. Le jury incite les futurs candidats à ne pas renoncer devant une apparente difficulté. Trop peu de candidats ont mentionné que la solution proposée n'est pas optimale.

Partie III

Cette partie appelait à décrire les procédures mises en œuvre réellement pour éviter les collisions. Elles faisait appel aux qualités de compréhension et de bon sens des candidats. Dans ces questions moins guidées, le jury juge autant la structure et la lisibilité du code que son aspect purement fonctionnel.

Pour illustrer ce propos, la question **III.B.4)** demandait de travailler sur les coordonnées et la vitesse d'un avion. 40% des candidats qui ont abordé la question ont posé :

```
id, x, y, z, vx, vy , vz, t0 = intrus
```

Il va de soi que la suite du code est claire et concise, et qu'elle évite des expressions telles que :

```
if intrus[1]*intrus[4]+intrus[2]*intrus[5]+intrus[3]*intrus[6]>0:
```

plus difficile à lire, comprendre, vérifier et même écrire. Le barème bonifie les réponses des candidats ayant pris cette initiative.

Autre illustration : la question **III.C.2)** demandait de trier une liste où une seule valeur n'était pas à sa place. Certains proposent une solution quadratique, ce qui est visiblement maladroit. D'autres une solution linéaire, ce qui est mieux. D'autres enfin s'attachent, dans l'esprit de la question, à minimiser le nombre d'opérations élémentaires. Il va de soi que le jury pondère ces différentes approches, même si elles répondent toutes à la question.

Conclusion

L'impression globale de sérieux de la préparation perçue l'an passé perdure pour cette deuxième session. Pour poursuivre dans cette voie, le jury invite les étudiants et leurs formateurs à insister sur la lisibilité du code, en profitant en particulier des traits du langage qui favorisent celle-ci.