4.1-Informatique pour tous

PRÉSENTATION DU SUJET

L'épreuve d'informatique commune portait sur le traitement informatique de la modélisation d'une propagation d'épidémie. L'épreuve faisait appel à des notions variées du programme des deux années de classe préparatoires : tris, invariants de boucle, requêtes en langage SQL, algorithmique sur des tableaux, résolution numérique d'un système différentiel. Malgré un temps d'épreuve assez court, cette épreuve nous a paru de nature à garantir un classement efficace des candidats.

Quelques excellentes copies témoignent d'un travail approfondi sur le programme, doublé d'une grande rapidité et d'une capacité à écrire des programmes clairs, concis, et syntaxiquement irréprochables. A l'inverse, quelques copies témoignent d'une méconnaissance flagrante des règles de bases de cette discipline, et parfois aussi d'un niveau de raisonnement scientifique remarquablement faible à ce niveau de formation.

Nous souhaitons rappeler aux candidats quelques conditions nécessaires pour espérer réussir cette épreuve :

- La maîtrise de la syntaxe de base des langages python et SQL est absolument indispensable. Le respect de l'indentation est capital, et la lisibilité de l'écriture est appréciée : il est rappelé que le doute ne bénéficie pas souvent au candidat.

- Les candidats sont invités à réfléchir sur les spécificités de chaque type informatique : une liste (ou une liste de listes) n'a pas les même propriétés qu'un tableau numpy, par exemple.

- Les questions qualitatives, où un court paragraphe est attendu (par exemple la question 12) doivent être traitées avec soin : la concision nécessaire n'excuse pas l'imprécision parfois très gênante des termes utilisés, ou du raisonnement.

Voici quelques commentaires question par question.

Q1 : Bien traitée par une grande majorité de candidats.

Q2 : La notion d'invariant de boucle a été souvent maltraitée. Un certain nombre de candidats montre son ignorance presque complète de cette notion par des raisonnements tels que «si la liste est triée au début, alors elle est triée à la fin et donc c'est un invariant». Parmi les candidats qui ont compris l'intérêt du raisonnement par récurrence, certain oublient l'importance de l'initialisation, ou bien d'utiliser l'invariant pour prouver que la liste est effectivement triée à la fin de l'exécution de la fonction.

Q3 : Cette question a donné lieu à un florilège de résultats particulièrement exotiques. Le cours sur les tri semble être passé de manière approximative chez beaucoup de candidats. Les complexités

Q4 : Si de nombreux candidats ont bien pensé à adapter le programme fourni par l'énoncé, il est dommage que certains n'aient réalisé le tri que sur le second élément de la liste en oubliant le premier !

Q5 : La notion de clef primaire est très mal maîtrisée. On a ainsi pu lire des phrases très étonnantes, comme « l'attribut iso peut servir de clef primaire, et de même le couple iso/année peut servir de clef primaire car elle renvoie à une seule ligne du tableau ». Un travail supplémentaire sur cette notion importante semble nécessaire.

Q6 : Une requête commençant par **FROM palu IMPORT...** laisse dubitatif sur la qualité du travail de préparation des candidats sur le langage SQL. La majorité des candidats a cependant traité correctement cette question. Il est aussi rappelé à certains candidats créatifs que l'épreuve d'informatique n'est pas une épreuve d'anglais, et que bricoler une instruction vague avec des pseudo-mots d'anglais n'est pas suffisant pour écrire une requête en SQL.

Q7 : Cette requête comportait plusieurs difficultés. Tous les candidats n'ont pas perçu la nécessité d'une jointure symétrique, qui fut par ailleurs souvent mal écrite. Il fallait noter que la condition de jointure faisait intervenir deux attributs pour chaque table.

Q8 : Question plus difficile, que des candidats malins ont traité astucieusement avec LIMIT et OFFSET, profitant d'une certaines ambiguïté du programme sur ce qui était exigible. On pouvait aussi s'en sortir avec des sous-requêtes.

Q9 : Question souvent bien traitée.

Q10 : Nous avons constaté sur cette question un grand nombre de réponses comme $f(X)=S(t) + I(t) + R(t) + D(t)$, qui témoignent d'une incompréhension du fonctionnement des systèmes différentiels. Certains candidats ont tenu à exprimer f sous forme d'une matrice carrée, ce qui était faux ici, le système différentiel n'étant pas linéaire.

Q11 : Question bien traitée par ceux qui ont réussi la question précédente, mais il ne fallait pas oublier de renvoyer un tableau numpy et pas une liste.

Q12 : Cette question qualitative pourtant simple a été assez mal traitée. On passe sur les candidats qui après un raisonnement parfois très torturé, arrivent à la conclusion que la simulation avec N=7 nécessite plus de temps... Le rôle du pas de discrétisation pour la précision de la méthode d'Euler était attendu, et n'est pas apparu clairement dans la majorité des copies. A l'inverse, on a pu lire des horreurs comme «la simulation est discrète pour N=7 alors que pour N=250 elle est continue».

Q13 et **Q14** : Questions plus difficiles qui nous ont souvent permis de repérer les meilleures copies.

Q15 : Question simple souvent bien traitée.

Q16 : Tous les candidats ne maîtrisent pas le double indice pour accéder à un élément d'un tableau (**G[i][j]**) ou le confondent avec la syntaxe adaptée aux tableau numpy (**G[i,j]**). Par ailleurs, quelques

Q20 : Une erreur fréquente des candidats est de n'avoir pas perçu la nécessité de créer une nouvelle grille dès le début de la fonction, et de ne pas modifier **G** avant la fin de celle-ci: en effet, les valeurs de **G** sont nécessaires à la détermination de l'évolution, et on ne peut pas à la fois effectuer des tests utilisant les valeurs de **G** et modifier **G**.

Q21 : Le caractère aléatoire du résultat renvoyé par la fonction suivant rendait impossible la syntaxe :

```
while G!=suivant(G):
    G=suivant(G)
```

Pour le comptage, on a parfois assisté à des appels de fonction très maladroits, comme **n0=compte(G)[0]**, **n1=compte(G)[1]**, etc. Les candidats sont invités à réfléchir à l'efficacité algorithmique de leurs programmes pour éviter des appels multiples inutiles.

Q23 : Cette question nécessitait une maîtrise satisfaisante de l'algorithme de recherche par dichotomie. Elle a été peu traitée, par manque de temps.

Q24/Q25 : Certains candidats ayant perçu que ces questions étaient faciles mais manquant de temps ont proposé des réponses non justifiées (« Non c'est impossible » par exemple). Inutile de préciser que ces réponses n'ont valu aucun point à leurs auteurs.

Nous terminons ce rapport par quelques perles que nous avons choisi de distinguer particulièrement cette année:

- Le prix de la complexité la plus remarquable :

« Dans le pire des cas, la complexité est en $O(1/n^n)$ »

- Le prix de l'écologie :

« Un tri plus efficace dans le pire des cas est le tri sélectif »

- Le prix de la maîtrise de programme :

« pour trier une liste, il est plus efficace d'utiliser le pivot de Gauss »

- Le prix de la piété informatique :

« Cette fonction renvoie n^2 individus saints »

Nous souhaitons une bonne préparation et bon courage aux futurs candidats !