ECOLE POLYTECHNIQUE-ESPCI CONCOURS D'ADMISSION 2024
ENS : LYON, PARIS-SACLAY

Composition d'Informatique B - 2h, Filière PC (XELS)

1 Remarques générales

Dans cette épreuve, 524 copies ont été corrigées. La moyenne est de 9,34/20 et l'écart-type de 3,85. Le sujet était découpé en 4 parties, dont une partie d'énumération récursive, et une partie de programmation dynamique. L'énumération n'avait pas été abordée dans les sujets de ces dernières années et a posé d'énormes difficultés aux candidats. La programmation dynamique, déjà présente dans le sujet de l'an dernier, est toujours aussi obscure pour les candidats.

Les consignes de l'épreuve ont été parfois négligées : les opérations autorisées dans l'épreuve étaient indiquées en début de document, et des primitives de manipulation des solutions étaient proposées (init_sol et copy_sol) qui ont pu être remplacées lors de tentatives souvent erronées. Les questions de complexité, globalement moins présentes que les années précédentes, sont toujours autant négligées par les candidats. La complexité des sous-fonctions est souvent oubliée dans le calcul de la complexité ce qui pose particulièrement problème pour les fonctions récursives. Tout aussi problématique, dans les premières questions, certains candidats ont indiqué une complexité linéaire en O(n), mais sans expliciter la variable n.

L'inattention est également une grande source d'erreurs évitables. Une simple relecture en gommerait probablement la majorité. Par exemple, des algorithmes renvoient une variable intermédiaire au lieu de la valeur calculée correctement, voire même ne renvoient rien. Les candidats pensent à vérifier que la case de la liste qu'ils lisent existe, mais après avoir utilisé sa valeur. Certains noms de variables changent en cours d'algorithme.

Les problèmes de compréhension des boucles de type for et while sont moins présents, mais certaines confusions existent. Par exemple, certains candidats confondent :
- for i in l:,
- for i in range(l): et
- for i in range(len(l)):

2 Partie I

2.1 Question 1

Une première question simple toujours traitée avec très peu d'erreurs, essentiellement dues à une incompréhension de la question ou à de l'inattention. La question de complexité a eu un succès mitigé avec beaucoup de cas de complexité en O(n) sans donner la signification de ce n.

2.2 Question 2

Question largement réussie là aussi. Le sujet ne mentionnait pas explicitement le besoin de renvoyer False lorsque la condition n'était pas remplie, c'était tout de même attendu et a été légèrement pénalisé.

2.3 Question 3

Encore une question très accessible et bien réussie dans l'ensemble. Quelques rares cas d'incompréhension du sujet ou l'oubli de retirer 1 pour prendre en compte le bord.

2.4 Question 4

Cette question était la première difficulté de l'énoncé. La première partie a été globalement réussie, même si certains candidats semblent avoir été déstabilisés et n'ont donné aucun ou un seul exemple. La seconde partie était plus ouverte et a posé plus de problèmes, de nombreux candidats n'ont pas su imaginer l'ensemble des risques et ont essayé de justifier que la fonction pouvait renvoyer False de manière erronée. Les explications demandées pour corriger la fonction étaient souvent trop "haut niveau" alors que la modification exacte tenait en une ligne. Certains candidats ont identifié une partie du problème, mais pas sa globalité et se sont focalisés sur le cas d'une solution ne comportant que des cases blanches.

3 Partie II

3.1 Question 5

Question un peu moins bien réussie que les précédentes, parfois par manque de maîtrise de la division euclidienne et régulièrement parce que le candidat se contentait d'écrire la division euclidienne sans expliciter les valeurs demandées de k et l. Il était demandé d'exprimer k et l en fonction de n et non l'inverse.

3.2 Question 6

La question demandait une génération exhaustive de tous les éléments d'un ensemble, ce qui a été discriminant (moins du quart des points en moyenne). Malgré des consignes précises pour les guider, beaucoup de candidats ont été perdus, sans doute en raison de difficultés à se représenter la dynamique des fonctions récursives. Le calcul des indices de ligne et de colonne a, semble-t-il, posé beaucoup de problèmes, de nombreux candidats ajoutant ou retranchant 1. La complexité a aussi suscité des difficultés, peu ont traité cette question et parmi eux, peu ont vu la complexité exponentielle.

3.3 Question 7

La question dépendait directement de la précédente et, logiquement, très peu de réponses ont mérité des points. Une difficulté était, sans écrire le nouveau code, de donner suffisamment de détails. Les réponses données étaient souvent trop succinctes pour permettre de visualiser l'idée du candidat, par exemple "Si une ligne est déjà complète, on s'arrête." ou "On teste ligne et colonne pour savoir si on doit continuer." Il était attendu de décrire le test, dire où il intervient dans le code et quel est le comportement en fonction du résultat.

4 Partie III

4.1 Question 8

Retour à une question accessible, presque toujours traitée et globalement réussie. Des points ont été perdus soit parce que les tests n'étaient pas placés dans le bon ordre :
if sol_p[i_ligne][c-1]==1 and c>0: au lieu de if c>0 and sol_p[i_ligne][c-1]==1:
soit parce que les tests aux bords n'étaient pas effectués.
La plupart des candidats ont traité la question de complexité, quasiment toujours correctement.
Certains n'ont pas eu les points parce qu'ils ont montré O(nc) bien que ce ne soit pas ce que demandait le sujet.

4.2 Question 9

À partir de cette question, les bonnes réponses ont été rares. Cette question a été souvent traitée, mais avec deux grandes sources d'échec. La première était d'utiliser la fonction conflit de la question précédente, ce qui amenait à une complexité trop grande. La seconde était d'oublier certaines conditions : pour respecter la complexité demandée, il était impossible de tester puis revenir en arrière en cas d'échec, il fallait un parcours linéaire en mettant à jour l'origine et sans oublier de tester à la fin la condition aux bords.
La question de complexité a été peu traitée, mais les rares réponses étaient dans l'ensemble bonnes.

5 Partie IV

5.1 Question 10

Question de programmation dynamique, moins traitée que les précédentes. Il fallait implémenter l'algorithme proposé, donc il était attendu d'être précis et de ne pas oublier de tests. Souvent les candidats ont repris les formules de l'algorithme donné en oubliant de tester les conditions d'application.
La question de complexité a été peu traitée, mais réussie.

5.2 Question 11

Peu de tentatives et peu de bonnes réponses parmi elles. Il fallait reprendre et adapter les arguments précédents au cas particulier considéré. Les candidats qui ont voulu donner une explication sans reprendre le schéma proposé dans le cas général n'ont presque jamais réussi à donner une explication convaincante.

5.3 Question 12

De nouveau, question très peu traitée et très peu réussie (un huitième des points en moyenne).
L'habitude de remonter la matrice de programmation dynamique en partant de la fin n'est clairement pas ancrée.

5.4 Question 13

Toujours moins de réponses et de réussite. La difficulté, ici, était de bien visualiser tout ce qui a été fait auparavant et de comprendre la question.

5.5 Question 14

Dernière question quasiment jamais traitée, quelques rares candidats ont récolté des points. La question était plus compliquée et demandait là aussi d'avoir très bien intégré tout le reste du sujet (les rares points sont allés à des candidats ayant traité une grande partie du sujet correctement).