ISM MP, Informatique
Année 2024/2025 09/11/2024

Commentaires - Devoir surveillé n°1

Remarques générales : Quasiment aucun code n'a été commenté. Ceci est réellement problématique sur des codes non triviaux. Il faut également penser à matérialiser l'indentation par un trait vertical. Il faut éviter de couper un code donc si l'espace restant sur une page est trop juste, on n'hésite pas à changer de page. Certains commencent par le SQL ce qui fait sens et permet de garantir des points plutôt faciles à prendre. Les confusions entre `=` et `==`, les erreurs de bornes dans `range` et les oublis de crochets dans les sous-listes sont des erreurs qui ne devraient plus figurer dans vos copies. Une procédure, conformément aux consignes du sujet, est une fonction python qui ne renvoie rien et il faut donc respecter cette exigence. Si une question consiste à tester la présence d'un élément dans une liste, il est selon toute vraisemblance attendu qu'un parcours de liste soit programmé et non que l'on recourt à un test d'appartenance en `in`, ceci étant encore plus évident si un calcul de complexité est demandé.

I Réseaux Sociaux

1. OK pour presque tous. Des erreurs d'inattention chez certains, c'est bien dommage car c'est très simple.

2. OK pour presque tous. Les réponses inutilement lourdes (à grand renfort de `append`) sont à éviter.

3. OK pour presque tous. Le sujet n'interdit pas *a priori* l'amitié entre `i` et lui-même et il faut donc réellement tester la présence de `[i,j]` ou `[j,i]` dans les liens.

4. Le calcul de complexité mérite une explication.

5. Bien réussie par un grand nombre. Ici encore, la complexité mérite un détail.

6. Réussite mitigée : plusieurs ont écrit des codes peu efficaces de complexité moins bonne que $O(m)$ ce qui était pourtant facilement atteignable en ne parcourant que les liens d'amitiés.

II Partitions

7. OK.

8. Quelques un n'ont pas compris la structure attendue alors que la question est très simples. À reprendre pour ceux qui sont concernés.

9. OK pour une majorité.

10. OK pour une majorité.

11. Plusieurs confusions sur les exemples proposés qui n'illustrent pas la situation dégradée attendue.

12. Bien traitée par ceux qui ont abordé la question.

13. Question délicate, moyennement réussie. L'idéal consistait à utiliser des dictionnaires pour bénéficier des performances d'implémentation de ces structures.

III Algorithme randomisé pour la coupe minimum

14. Question difficile bien que très guidée. Il ne faut pas court-circuiter les étapes exigées : si on doit fusionner conditionnellement des groupes, on ne doit pas faire l'économie du test sous peine d'alourdir la complexité. Une fonction auxiliaire d'échange de deux termes dans une liste était tout-à-fait bienvenue. La gestion d'un compteur pour le nombre de groupes plutôt qu'un appel systématique à `listeDesGRoupes` était clairement souhaitable.

15. Très peu abordée. Question moins difficile que la précédente.

IV SQL

16. OK pour une majorité. Certains écrivent des choses très compliquées avec jointure alors qu'une requête élémentaire fait le travail.

17. On pouvait procéder par jointure ou sélection et sous-requête mais la jointure est préférable car plus performante relativement à l'implémentation des bases de données.

18. Question délicate : on pouvait faire par auto-jointure ou avec sous-requête.