

Informatique

Présentation du sujet

Cette année, ce sujet d'informatique propose d'étudier des aspects informatiques d'un robot effectuant une mission sur Mars.

Une première partie nous amène à créer une exploration pour le robot et gérer des points d'intérêt.

La deuxième partie présente un algorithme permettant de planifier une exploration.

Enfin, une troisième partie utilise un algorithme génétique pour proposer une autre solution au problème de l'exploration.

Analyse globale des résultats

Le sujet est de longueur satisfaisante et a permis un étalement des notes convenable avec une moyenne brute voisine de 10 et une répartition des notes adaptée à une épreuve de concours. Certains candidats ont traité quasi-correctement l'intégralité du sujet. À contrario, on peut noter quelques copies presque vides.

Le langage Python est désormais globalement plutôt bien maîtrisé par les candidats et le code écrit est plutôt bien présenté même si vraiment peu de commentaires sont présents et si quelques problèmes techniques subsistent.

Commentaires sur les réponses apportées et conseils aux futurs candidats

Les parties sont indépendantes et ont été plus ou moins toutes abordées.

Partie I

I.A.1) Le caractère distinct des points choisis est souvent mal traité.

I.A.2) Quelques difficultés à gérer la position initiale du robot, surtout au niveau des indices. On a aussi pu voir des calculs de distance farfelus.

I.B.1) La lecture du programme ligne par ligne ne permet pas au candidat d'avoir une vision globale du programme, souvent le bilan qui en résulte est inexact.

I.B.2) Parfois, la taille de la photo a posé problème avec une mauvaise utilisation de fonction.

I.C – L'écriture en SQL est dans l'ensemble plutôt réussie pour les requêtes simple (**I.C.1**) et **I.C.2**) mais bien plus confuse pour les autres.

Partie II

II.A.1) Oubli récurrent de la position initiale du robot et gestion erronée des indices.

II.A.2) Pas de problèmes particuliers, l'énoncé décrit la marche à suivre.

II.B.1) Le nombre de chemins possibles n'est souvent pas justifié.

II.B.2) Bien peu de candidats justifient convenablement (référence au nombre d'opérations par seconde pour un processeur) le fait que l'algorithme n'est pas utilisable dans ce cas.

II.C.1) Question difficile rarement bien traitée même si souvent abordée. Des difficultés fréquentes lors de l'initialisation.

II.C.2) La notion de complexité est souvent mal traitée par les candidats.

Partie III

La partie III a été un peu moins souvent abordée que les deux précédentes, mais elle a été plutôt bien réussie par les candidats qui sont parvenus jusque-là.

III.A – Oubli fréquent de la position initiale du robot dans l'initialisation

III.B – Une majorité de candidats a pensé à trier la liste mais la suppression des éléments a été moins bien traitée (suppression des termes les plus petits, mauvaise utilisation de `del`)

III.C.1) On a souvent vu un oubli d'avoir 2 points distincts.

III.C.2) Plutôt bien traitée quand elle est abordée.

III.D.1-2) Pas de problème particulier.

III.E.1) L'algorithme génétique : ceux qui l'ont traité ont bien réussi en général.

III.E.2) Souvent bien traitée quand abordée.

III.E.3) Question plus difficile avec parfois des réponses tout à fait pertinentes.

Remarques générales

Les noms des variables doivent être choisis de manière judicieuse et pas seulement par ordre alphabétique.

Certaines variables utilisées ne sont pas définies.

La gestion des indices dans les listes n'est pas totalement maîtrisée. Il n'est pas rare de lire `L(len(L))` dans des copies.

Le nombre d'itérations d'une boucle doit être bien réfléchi, en notant que l'instruction `range(n)` permet de réaliser `n` itérations indicées de 0 à `n-1`.

Il faut lire attentivement le sujet pour utiliser correctement les fonctions décrites et fournies par l'énoncé.

les requêtes SQL, dès qu'elles sont un peu évoluées ne sont pas bien traitées : tout candidat devrait pouvoir utiliser convenablement une jointure et une fonction d'agrégation.

L'instruction `del` est très souvent utilisée pour un résultat régulièrement faux (surtout dans une boucle).

Les calculs de complexités ne sont clairement pas assimilés par beaucoup de candidats. On a par exemple appris que $O(n^2) + O(n^2) = O(n^4)$.

Une conséquence du point précédent est que beaucoup de fonctions sont écrites sans se soucier de leur efficacité propre, ce qui peut être gênant pour la maîtrise réelle de l'outil informatique.

Dans l'écriture de fonctions, il est naturel d'attendre quelques explications, mêmes succinctes.

Conclusion

Le sujet proposé cette année est de longueur raisonnable et a permis une bonne répartition des notes. Globalement, le niveau des candidats est convenable même si de nettes différences entre candidats peuvent être remarquées. Le jury rappelle qu'un investissement constant sur toutes les années de préparation est essentiel et peut permettre facilement de se démarquer. Il serait aussi souhaitable de se poser la question de l'efficacité dans l'écriture de programmes.