

Informatique

Présentation du sujet

Cette épreuve traite de la sécurité aérienne, en particulier du référencement des vols et de leur niveaux (altitude de croisière), de l'attribution de ces niveaux, et du traitement en temps réel des trajectoires pour éviter les collisions entre appareils.

Le sujet aborde les bases de données et requêtes SQL et l'algorithmique autour d'un graphe et d'autres listes ou tableaux de données. Des questions qualitatives, ou mettant en jeu des calculs simples, permettent de discuter l'applicabilité des méthodes établies aux contraintes réelles de la sécurité aérienne.

Analyse globale des résultats

La structure de base d'une requête SQL est maîtrisée par la majorité des candidats. La syntaxe du langage Python est remarquablement mieux acquise que lors de l'épreuve du concours 2015. La présentation des copies, en particulier des algorithmes, est bonne. Cependant, de trop nombreux candidats ne font pas encore l'effort de s'approprier vraiment le sujet, afin d'en maîtriser les structures des données à manipuler. Enfin, les candidats ont pratiquement tous abordés les questions non algorithmiques autour du stockage ou de l'évaluation des paramètres en fin du sujet.

Commentaires sur les réponses apportées et conseils aux futurs candidats

Partie I

La fonction d'agrégation `COUNT` doit être connue. De plus, il faut avoir conscience que seuls des extraits de tableaux sont donnés, ainsi pour la question **I.B** il faut obligatoirement réaliser une jointure entre les tables `vol` et `aeroport` : CDG et ORY ne sont à priori pas les seuls aéroports parisiens. La requête demandée en **I.D** n'a pratiquement jamais été correctement écrite.

Partie II

Le jury tient à rappeler que la fonction `print` permet d'afficher une valeur, et ne sert à rien d'autre. Cette fonction apparaît dans de nombreuses copies alors que le sujet ne demande jamais de procéder à un affichage. Il y a bien souvent confusion entre `print` et l'instruction `return`.

Le parcours des éléments d'un tableau, ou liste de liste, grâce à une double boucle imbriquée, est globalement maîtrisé. Cependant, la question **II.B.2**, pour laquelle certains éléments du tableau seulement sont consultés, a souvent posé problème. Il est impératif pour ce type de question de s'approprier les structures des données à manipuler : il paraît essentiel de schématiser au brouillon les structures, ou de réfléchir sur de petits exemples, pour mener à bien la gestion des indices de parcours du tableau.

Une minorité des candidats a su estimer le nombre de régulations possibles pour n vol. Le manque de recherche au brouillon, ne serait-ce que par le calcul des quelques premiers éléments de la récurrence, est ici aussi probablement en cause.

La question **II.C.1**, de calcul du cout d'un sommet, a été source d'erreur : on pouvait certes vérifier que le sommet dont on calcule le cout n'était pas supprimé, mais il était surtout primordial de consulter l'état de chaque sommet adjacent avant d'ajouter le conflit correspondant.

La question **II.D** sur l'algorithme de recuit simulé a été parfois abordée ; elle a été traitée de manière excellente par quelques candidats.

Partie III

Le calcul de l'instant t_c du *closest point of approach* (CPA) n'a pratiquement jamais été correctement réalisé. Pourtant, l'écriture d'un schéma simple peut permettre d'établir une relation entre t_c et le projeté orthogonal du vecteur position sur le vecteur vitesse. Néanmoins, il est intéressant de voir que de nombreux candidats n'ayant pas déterminé l'expression de t_c ont quand même traité le cas de l'absence de collision possible dans la fonction `calculer_CPA`.

Les questions en fin de partie III autour de l'évaluation des paramètres du système de détection anti-collision ont été traitées plutôt correctement par la majorité des candidats.

Conclusion

Le jury encourage les candidats à redoubler d'efforts sur les questions difficiles de ce type de problème, notamment celles mettant en jeu plusieurs données de structures différentes (tableaux, listes) devant être parcourues concomitamment. De même, il faut s'entraîner à écrire des requêtes SQL dont la complexité va au-delà de la requête de base. Ces compétences ne peuvent évidemment qu'être le fruit d'un travail régulier et assidu au cours de la préparation du concours, et les efforts évoqués deviennent de plus en plus faciles, voire ludiques, à force de pratique.