

Devoir surveillé n° 06

Précisions et indications

Les parties I, II et III sont complètement indépendantes. Les sous-parties de la partie III sont relativement indépendantes. Le sujet ne me semble pas réellement progressif au sens où le niveau de difficulté est très variable tout au long des questions du sujet, avec des questions très faciles un peu partout. Bon courage!

1 Logique

- Q 2) Même si la formulation du problème semble s'incrimer dans une logique temporelle, il faut entièrement ignorer cela et simplement traduire les phrases en logique propositionnelle classique.
- Q 4) La question est maladroite, ce n'est pas la formule F qui permet de répondre au problème posé mais la formulation de ce problème sous la forme d'un problème de logique faisant intervenir une formule F . On demande ici quel est ce problème logique et quelle est la formule F .
- Q 5) On peut montrer que la valuation obtenue est l'unique modèle de F mais on ne demande pas de le montrer.

2 Bases de données

- Q 6) Il y a plusieurs choix à faire et ceux-ci peuvent impacter la difficulté des requêtes ensuite. De plus, si vous n'arrivez pas à traduire ce modèle entité-association en modèle relationnel, vous ne pouvez pas traiter la suite. Je vous donne donc le modèle ci-dessous à respecter impérativement pour les questions en SQL.
- **Ville** : Code_Postal, Nom_Ville
 - **Cinéma** : Code_Cine, Nom, Adresse, Code_Postal, Nom_Ville
 - **Salle** : Code_Salle, Capacité, 3D, Code_Cine
 - **Projet** : Code_Film, Code_Salle, nb_Entrées
 - **Film** : Code_Film, Titre, Durée, Code_Réal
 - **Réalisateur** : Code_Réal, Nom_Réal, Prénom_Réal, Âge

Commenter et justifier en quelques lignes les choix proposés pour le passage du modèle entité-association au modèle relationnel ci-dessus. Préciser, comme dans la question initiale, les clés primaires et les clés étrangères.

3 Algorithmique — Problèmes de dominos

3.1 Structures de données

Il faut bien sûr lire $D_k = (i_k, j_k)$ et pas $D_i = (i_k, j_k)$.

- Q 9) Dans ce sujet on représente une liste chaînée par un pointeur vers son première élément et le pointeur `NULL` pour la liste vide.
- Q 11) Il y a une première subtilité. On ne peut pas utiliser l'opérateur `==` pour comparer deux structures pour de sombres histoires d'alignement hors programme : il faut implémenter une fonction qui compare deux dominos. On ne suppose pas que l'élément est présent dans la liste, mais on suppose qu'un élément ne peut être présent qu'au plus une fois.

3.2 Existence d'une chaîne de taille n

- Q 13) Cette question est incompréhensible. On croit comprendre qu'il faut aller chercher les dominos i et j dans un sac qui n'est pas passé comme argument, ce qui serait de toute façon d'une réelle inefficacité. Je vous propose plutôt d'écrire une fonction qui prend en paramètres deux dominos D_i et D_j et qui vérifie s'il est possible de placer le domino D_j à droite (et non à gauche) du domino D_i .
- Q 15) La spécification de cette fonction n'est pas donnée. Je vous propose de supposer qu'elle renvoie `true` tout en effectuant la rotation si et seulement si on peut placer D_j à droite de D_i après rotation et renvoie `false` sans effectuer la rotation sinon. Attention à la syntaxe en C dans cette question, certains objets sont des structures, d'autres des pointeurs.
- Q 18) Cette question est extrêmement difficile pour ne pas dire infaisable correctement sans machine et en temps limité. Il suffit bien de suivre l'idée de la question précédente mais il y a de nombreux points très délicats dans l'implémentation, surtout avec les structures de données suggérées par l'énoncé. Si vous vous y essayez, je serai très tolérant sur la qualité du code proposé (mais il faut que cela reste lisible et judicieusement commenté).
- Q 19) Il n'y a pas besoin d'avoir implémenté l'algorithme en C pour répondre à cette question.

3.3 Nombre de chaînes de taille n

Dans ce sujet un cycle est un chemin fermé simple : il ne peut passer qu'une seule fois par chaque arête mais peut passer plusieurs fois par un même sommet.

On cherche à compter le nombre de chaînes de taille n formés *exactement* des n dominos du sac S_N .

Q 23) Il faut savoir faire cette question.

Q 24) Je vous suggère de montrer la contraposé qui est la propriété qui nous intéresse d'ailleurs : si G est un graphe comportant un chemin eulérien, alors au plus deux de ses sommets sont de degré impair.

Il n'est pas difficile d'adapter la réponse de la question précédente pour montrer que si un graphe G comporte un cycle eulérien alors tous ses sommets sont de degré pair. On pourra utiliser ce résultat par la suite.

L'énoncé ici est difficilement compréhensible. On cherche à montrer deux choses :

- (a) Un graphe connexe G dont tous les sommets sont de degré pair comporte un cycle eulérien.
- (b) Un graphe connexe G dont exactement deux sommets sont de degré impair comporte un chemin eulérien.

On montre tout d'abord que (a) \Rightarrow (b). On suppose donc (a) et on cherche à montrer (b). Pour cela, on considère un graphe G connexe dont exactement deux sommets s_1 et s_2 sont de degré impair.

Q 25) L'indication de l'énoncé n'est possible que si $\{s_1, s_2\}$ n'est pas déjà une arête de G . Montrer déjà le résultat sous cette hypothèse. Proposer ensuite une preuve pour le cas où $\{s_1, s_2\}$ est une arête de G .

On cherche maintenant à montrer (a). On considère pour cela un chemin *simple* C de longueur maximale dans G . Il faut bien sûr ici prendre un chemin qui ne repasse pas deux fois par une même arête. Si ce chemin emprunte toutes les arêtes, on montre sans difficulté qu'il s'agit d'un cycle eulérien. Sinon, on distingue les deux cas proposés selon que C est un chemin ouvert ou fermé. L'énoncé n'est pas très clair avec son arête a . Dans les deux cas il s'agit de montrer que l'on peut prolonger le chemin C en un chemin simple strictement plus long et aboutir à une contradiction.

Le sujet commence à parler de circuits sans définir de quoi il s'agit ni en quoi c'est éventuellement différent d'un cycle. Habituellement, on utilise le terme circuit pour un chemin fermé dans un graphe orienté et le terme cycle pour un chemin simple (et parfois élémentaire) dans un graphe non orienté. Ici, il n'est pas du tout clair si cycle et circuit désignent exactement la même chose ou non. Pour que la suite de

l'énoncé ait un sens, j'ai l'impression qu'il faut considérer qu'un circuit est un cycle mais sans que le sommet de départ et d'arrivée n'ait d'importance, donc formellement une classe d'équivalence de chemins. Par exemple, $0 - 1 - 2 - 3 - 0$ et $1 - 2 - 3 - 0 - 1$ correspondent au même circuit même si ce sont deux chemins différents. En revanche, $1 - 0 - 3 - 2 - 1$ n'est pas le même circuit puisqu'il est parcouru dans l'autre sens. C'est déjà souvent ce que l'on entend implicitement par cycle.

Q 29) Question indépendante qui n'utilise que la question 20, le paragraphe qui la suit ainsi que la question 22. Attention, dans cette question on prend à nouveau en compte les $N + 1$ doubles. La formule proposée n'est vraie que pour $N \geq 2$, ce que l'on supposera.

3.4 Recherche de la plus longue sous-séquence

Cela n'est pas du tout clair dans l'énoncé mais on considère ici qu'une *sous-séquence* d'une séquence de dominos est une suite de dominos *consécutifs* (ce n'est pas le sens de sous-séquence habituellement).

On suppose que le sac S comporte les dominos D_1, D_2, \dots, D_n avec n le nombre de dominos du sac (qui n'a rien à voir avec le n de la section précédente).

Q 30) $l(0)$ n'est pas bien défini. On suppose plutôt que $l : \llbracket 1, n \rrbracket \rightarrow \llbracket 1, n \rrbracket$.

3.5 Recherche du nombre de sous-séquence

Pas d'indications ni de précisions pour cette dernière partie qui contient pourtant de nombreuses erreurs et imprécisions problématiques. Je vous laisse faire au mieux comme ce sera le cas pour vous au concours.