Taxis Tokyotes

Épreuve pratique d'algorithmique et de programmation
Concours commun des écoles normales supérieures

Durée de l'épreuve: 3 heures 30 minutes

Juillet 2008

ATTENTION !

N'oubliez en aucun cas de recopier votre u_0
à l'emplacement prévu sur votre fiche réponse

Important.

Sur votre table est indiqué un numéro u_0 qui servira d'entrée à vos programmes. Les réponses attendues sont généralement courtes et doivent être données sur la fiche réponse fournie à la fin du sujet. À la fin du sujet, vous trouverez en fait deux fiches réponses. La première est un exemple des réponses attendues pour un $\widetilde{u_0}$ particulier (précisé sur cette même fiche et que nous notons avec un tilde pour éviter toute confusion!). Cette fiche est destinée à vous aider à vérifier le résultat de vos programmes en les testant avec $\widetilde{u_0}$ au lieu de u_0. Vous indiquerez vos réponses (correspondant à votre u_0) sur la seconde et vous la remettrez à l'examinateur à la fin de l'épreuve.

En ce qui concerne la partie orale de l'examen, lorsque la description d'un algorithme est demandée, vous devez présenter son fonctionnement de façon schématique, courte et précise. Vous ne devez en aucun cas recopier le code de vos procédures!
Quand on demande la complexité en temps ou en mémoire d'un algorithme en fonction d'un paramètre n, on demande l'ordre de grandeur en fonction du paramètre, par exemple: $O(n^2)$, $O(n \log n)$,...
Il est recommandé de commencer par lancer vos programmes sur de petites valeurs des paramètres et de ***tester vos programmes sur des petits exemples que vous aurez résolus préalablement à la main ou bien à l'aide de la fiche réponse type fournie en annexe***. Enfin, il est recommandé de lire l'intégralité du sujet avant de commencer afin d'effectuer les bons choix de structures de données dès le début.

1 Introduction

La plus grande ville du monde, Tokyo, se veut toujours à l'avant-garde du progrès. Récemment, elle a décidé d'améliorer considérablement son système de transports, concernant notamment les bus et les taxis, dans l'optique d'une réduction de la pollution et de l'amélioration du service aux usagers par de meilleurs temps d'attente et de desserte.

Pour ce faire, elle compte utiliser le fait que la plupart des Tokyotes disposent de téléphones mobiles équipés de systèmes GPS, ce qui permet d'une part de les localiser précisément, et d'autre part leur permet de signaler facilement par SMS l'endroit où ils désirent se rendre. L'idée est donc de modifier le schéma traditionnel de fonctionnement des bus et des taxis pour utiliser l'information des trajets demandés au mieux, dès qu'elle est disponible.

Pour modéliser le problème, nous considérerons que les bus ont des trajectoires unidimensionnelles : ils évoluent chacun sur leur ligne indépendamment. Leur position est donc modélisée à un instant donné par un entier dans l'intervalle $[0; x]$, et ils transportent un certain nombre de passagers, au maximum 40. Les taxis, quant à eux, ont davantage de liberté et ont la possibilité d'évoluer en 2 dimensions, leur position étant repérée cette fois par une paire d'entiers dans $[0; x] \times [0; x]$, et ils peuvent transporter au maximum 4 passagers.

À chaque instant t, un usager peut signaler qu'il désire effectuer un trajet d'un endroit à un autre.

Le but de l'épreuve est de simuler ces situations afin d'étudier quelques stratégies pour permettre d'optimiser les transports, notamment aux heures de pointe de sortie des bureaux. Le principe est le même pour les bus que pour les taxis. Nous allons commencer par étudier le cas unidimensionnel des bus.

2 Étude du cas des bus (unidimensionnel)

Nous allons modéliser la situation de la manière suivante : l'heure de pointe est un intervalle de temps $[0; h]$, où h est un entier. À chaque pas de temps (unitaire), les évènements suivants se déroulent, dans l'ordre spécifié. Tout d'abord, de nouveaux usagers peuvent déclarer leurs trajets souhaités, puis le bus dépose les usagers arrivés à destination, puis le bus décide de prendre ou non des usagers là où il se trouve (il prend toujours en priorité ceux qui attendent depuis le plus longtemps), puis il se déplace d'une distance 1 dans la direction de son choix (il y arrivera au pas de temps suivant). À partir de l'heure $h + 1$, l'heure de pointe est terminée, aucun nouvel usager ne va à présent se manifester pour de nouveaux trajets, et le bus va pouvoir terminer sa desserte. Enfin, à l'instant 0, le bus est situé à la position 0.

Il y a plusieurs buts à atteindre. D'une part, minimiser l'heure de fin de service, où le bus a terminé de véhiculer son dernier usager. D'autre part, minimiser le maximum et la moyenne du temps de desserte des usagers (la différence entre le moment où ils signalent leur intention de trajet, et le moment où ils y arrivent). Dans la suite du sujet, nous appellerons ces triplets de valeurs les résultats d'une stratégie.

2.1 Suite pseudo-aléatoire de trajets

Considérons la suite d'entiers (u_n) définie pour $n \geq 0$ par :

$$u_n = \begin{cases} \textit{votre } \texttt{u}_0 \textit{ (à reporter sur votre fiche)} & \text{si } n = 0 \\ 15\,091 \times u_{n-1} \mod 64\,007 & \text{si } n \geq 1 \end{cases}$$

Question 1 *Que valent :* ***a)*** u_{10} ***b)*** $u_{1\,000}$ ***c)*** $u_{100\,000}$

On s'assurera de pré-calculer et stocker suffisamment de valeurs de u_n de manière à pouvoir y accéder en temps constant par la suite.

Nous allons supposer qu'à chaque instant t, 1 nouvel usager manifeste son intention d'effectuer le trajet $A_t^{(x)}$, représenté par une paire d'entiers (départ, arrivée), chacun dans l'intervalle $[0; x]$. Nous définissons maintenant la suite $(A_n^{(x)})$ par :

$$A_n^{(x)} = (u_{2n} \mod (x+1), (u_{2n} + 1 + (u_{2n+1} \mod x)) \mod (x+1))$$

Notons que cette définition ne génère pas de trajets ponctuels.

Question 2 *Que valent :* ***a)*** $A_{10}^{(10)}$ ***b)*** $A_{1\,000}^{(100)}$ ***c)*** $A_{100\,000}^{(1000)}$

Nous allons noter $C(H, X)$, la configuration telle que $h = H$ et $x = X$.

2.2 Cas simplifié où tout le monde veut partir en même temps

Pour commencer, nous allons supposer que tous les usagers se manifestent en même temps, au temps 0. Cela permet d'étudier des algorithmes de planification où toutes les données sont connues au départ. De plus, nous allons supposer pour cette question uniquement que le bus a une capacité infinie.
Trouver un algorithme de desserte qui minimise l'heure de fin de service du bus.

Question 3 *Donner l'heure de fin de service pour les configurations suivantes (en supposant donc ici que chacun des passagers déclare son trajet au temps $t = 0$) :* ***a)*** $C(10, 10)$ ***b)*** $C(10, 100)$ ***c)*** $C(100, 1\,000)$

2.3 Cas simplifié du bus monoplace

Dans la suite du sujet, nous nous intéresserons au cas où les usagers se manifestent l'un après l'autre (donc au temps t, un usager manifeste son désir de faire le trajet $A_t^{(x)}$).
Pour cette question uniquement, nous allons supposer que le bus n'a qu'une seule place, et ne peut donc transporter qu'un seul usager à la fois. Sa stratégie va également être simple : il va transporter les passagers dans l'ordre où ils se sont manifestés, l'un après l'autre. Il va bien sûr faire au plus vite selon ces contraintes.

Question 4 *Donner les résultats, sous la forme (heure de fin de service, temps de desserte maximum, temps de desserte moyen), pour les configurations suivantes :* ***a)*** $C(10, 10)$ ***b)*** $C(10, 100)$ ***c)*** $C(100, 1\,000)$

2.4 Servir celui qui est le plus proche

Nous allons maintenant tester une autre stratégie, qui consiste à éviter que le bus ne fasse trop de trajets. S'il est plein, il se dirige systématiquement vers la plus proche destination d'un usager qu'il transporte. S'il n'est pas plein, il prend également en compte les demandes de chargement des usagers. En cas d'égalité de distance (dans des directions opposées), le bus choisira de servir la requête de l'usager ayant fait sa demande en premier.

Question 5 *Donner les résultats pour les configurations suivantes :* ***a)*** $C(10, 10)$ ***b)*** $C(10, 100)$ ***c)*** $C(100, 1\,000)$

Question à développer pendant l'oral : À quelle heure le bus terminera-t-il, dans le cas le pire ? On demande une estimation asymptotique en fonction de x et h.

2.5 La stratégie de l'ascenseur

Les bus mettent du temps à changer de direction, donc ils ont intérêt à minimiser les changements de direction. De plus, les usagers qui sont dans les extrémités risquent d'être pénalisés par la stratégie décrite précédemment. Nous allons donc tester maintenant un algorithme classique, qui tient compte de la direction du bus. Cet algorithme est utilisé dans les ascenseurs de manière similaire.
Le bus va conserver sa direction, tant qu'il a des requêtes à traiter devant lui (déposer ou charger un usager), et il charge les usagers qu'il rencontre, tant qu'il n'est pas plein, à condition qu'ils aillent dans sa direction. Quand le bus est vide et qu'il n'a plus de requêtes devant lui, il peut changer de direction et recommencer. Attention à ne pas oublier de réembarquer d'éventuels passagers après un changement de direction !

Question 6 *Donner les résultats pour les configurations suivantes :* ***a)*** $C(10, 10)$ ***b)*** $C(10, 100)$ ***c)*** $C(100, 1\,000)$

Question à développer pendant l'oral : À quelle heure le bus terminera-t-il, dans le cas le pire ? On demande une estimation asymptotique en fonction de x et h.

3 Le cas bidimensionnel des taxis

Nous allons maintenant étudier le cas des taxis. Les taxis évoluent en 2 dimensions. À chaque pas de temps, ils peuvent se déplacer d'une unité soit en abscisse, soit en ordonnée. Pour se déplacer d'un point A à un point B, un taxi se déplace toujours d'abord selon les abscisses, puis selon les ordonnées.
Ils sont plus maniables que les bus, et n'ont pas de problème de changement de direction, donc l'algorithme de l'ascenseur n'a pas vraiment de sens pour eux. Nous allons chercher d'autres stratégies adaptées.
Nous allons définir une suite pseudo-aléatoire de trajets bidimensionnels $B_n^{(x)}$ de la manière suivante :

$$B_n^{(x)} = \left(A_{2n}^{(x)}, A_{2n+1}^{(x)}\right)$$

C'est-à-dire que le trajet en deux dimensions se décompose en un trajet unidimensionnel (comme défini précédemment) en abscisse qui est considéré comme étant $A_{2n}^{(x)}$, et celui des ordonnées est $A_{2n+1}^{(x)}$.

3.1 Cas d'un seul taxi

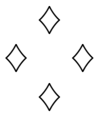
Nous allons tout simplement adapter la stratégie décrite dans la section 2.4, celle qui consiste à servir l'usager le plus proche. Comme précédemment, en cas d'égalité, on choisira systématiquement l'usager ayant fait sa demande en premier.

Question 7 *Donner les résultats pour les configurations suivantes :* ***a)*** $C(10, 10)$ ***b)*** $C(10, 100)$ ***c)*** $C(100, 1\,000)$

3.2 Cas de plusieurs taxis

Les taxis sont évidemment nombreux, et ils peuvent coopérer pour agir efficacement de manière collective. Nous allons considérer 4 taxis, numérotés 1, 2, 3, 4 qui démarrent aux 4 coins de la ville (respectivement $(0, 0)$, $(x, 0)$, (x, x) et $(0, x)$). Dans le cas ou plusieurs taxis se trouvent au même endroit, celui qui a le numéro le plus petit a priorité pour charger les usagers de cet endroit.

Question 8 *Donner les résultats pour les configurations suivantes :* ***a)*** $C(10, 10)$ ***b)*** $C(10, 100)$ ***c)*** $C(100, 1\,000)$



Fiche réponse type: Taxis Tokyotes

$\widetilde{u_0}$: 123

Question 1

a) 13158

b) 1300

c) 510

Question 2

a) (8,3)

b) (88,5)

c) (984,147)

Question 3

a) 20

b) 189

c) 1980

Question 4

a) (93,83,45.45)

b) (659,649,354.63)

c) (69614,69514,34357)

Question 5

a) (29,24,13.636)

b) (195,189,93.36)

c) (4142,4120,1550.29)

Question 6

a) (27,22,13.27)

b) (189,183,101.545)

c) (2614,2545,1184.68)

Question 7

a) (75,70,40.55)

b) (925,922,487.55)

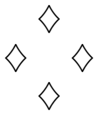
c) (48228,48170,23026)

Question 8

a) (33,29,16.91)

b) (365,362,187.91)

c) (13639,13617,6172.19)



Fiche réponse: Taxis Tokyotes

Nom, prénom, u_0: . . . . . .

Question 1

a)

b)

c)

Question 2

a)

b)

c)

Question 3

a)

b)

c)

Question 4

a)

b)

c)

Question 5

a)

b)

c)

Question 6

a)

b)

c)

Question 7

a)

b)

c)

Question 8

a)

b)

c)