Algorithmique avancée

Exercices de révisions (partiel) n° 2
(Correction)

Exercice 1 : Rendez-vous on the line

Supposons qu'on ait 2 robots sur une ligne à une distance normalisée de deux unités entre eux. Chaque robot peut avancer dans un sens ou dans l'autre d'une distance unitaire par unité de temps (temps discret). Sachant que les robots ne savent pas dans quelle direction est placé son vis-à-vis et qu'ils n'ont pas de capteur visuel, ils appliquent la stratégie dite du "rendez-vous 1F2B" : de manière indépendante chaque robot choisit une direction avec probabilité 1/2 et avance dans cette direction d'une unité ("1F" pour "1 step forward") et revient de deux unités dans l'autre ("2B" pour "2 steps backward").

1. Dans quelle positions (à quelles distances) peuvent être les robots après 3 unités de temps ?

▷ *4 possibilités : GG, DG ou GD et DD*
Avec GG et DD, on a rien gagné. Avec DG au bout d'une unité de temps on se rencontre et avec GD au bout de trois unités de temps on se rencontre.

2. Quel est le temps moyen de rendez-vous pour cette stratégie ?

▷
$$m = \frac{2}{4}m + \frac{1}{4}.1 + \frac{1}{4}.3$$

Donc $m = 2$.

3. Si les robots partent d'une distance de δ unités, donner une récurrence vérifiée par le temps moyen en fonction de δ pour la même stratégie "1F2B" ?

▷
$$m_\delta = \underbrace{\frac{1}{2}m_\delta}_{\text{essuie-glace}} + \underbrace{\frac{1}{4}m_{\delta+2}}_{\text{on s'approche puis on s'éloigne 2 fois plus}} + \underbrace{\frac{1}{4}m_{\delta-2}}_{\text{le contraire}} .$$

4. Donner une version "offline" du problème de rendez-vous.

▷ *On voit les données. Coût* $\delta/2$ *unités de temps (un robot va vers l'autre).*

5. Définir un ratio de compétivité de n'importe quelle stratégie S par rapport au problème courant

▷
$$\rho(S) = \max_{\delta>0} \frac{2}{\delta} \times \textit{Espérance de } S(\delta)\,.$$

6. Donner alors un critère d'optimalité sur toutes les stratégies.

▷
$$\min_S \max_{\delta>0} \frac{2}{\delta} \times \textit{Espérance de } S(\delta)\,.$$

Exercice 2 : Algorithme randomizé de recherche

Etant donné un ensemble $A = \{a_1, \cdots, a_n\}$ de n entiers strictement positifs et 2 à 2 distincts, le but est de trouver le k-ième plus petit élément de A. On trie et on a fini en $\Theta(n \log n)$.
Cependant, on peut faire l'algorithme suivant qu'on appelera Algo(A, k) :

- Si $n = 1$ alors retourner a_1 sinon tirer un chiffre i uniformément parmi $\{1, \cdots, n\}$.
- Calculer $A_{\{<\}} = \{x \in A,\, x < a_i\}$ et $A_{\{>\}} = A - A_{\{<\}}$.
- Si Card($A_{\{<\}}$) $> k$ alors on fait un appel récursif de l'algorithme présent sur $A_{\{<\}}$, i.e. Algo($A_{\{<\}}$, k).
- Si Card($A_{\{<\}}$) $= k - 1$ alors retourner a_1. Sinon appel récursif Algo($A_{\{>\}}, k - |A_{\{<\}} - 1)$.

1. Montrer sur un petit exemple le fonctionnement de l'algorithme Algo(A, k).

2. Quel est le gain par rapport à l'algorithme déterministe basé sur le tri ?

▷ *ici on est avec un temps moyen de* $O(n)$.