

IFT2015 Hiver 2011 — Devoir 3

Miklós Csűrös

6 mars 2011

À remettre avant 20 :15 le 14 mars. Remettez un rapport écrit par courriel (à csuros@iro...) en format PDF.

3.1 Paires (5 points)

On a une fonction monotone $f: \mathbb{R}^2 \mapsto \mathbb{R}$:

$$f(a, b) \leq f(a', b') \quad \text{si } a \leq a' \text{ et } b \leq b'.$$

On veut un algorithme qui vérifie si un tableau $A[0..n-1]$ trié (en ordre croissant) contient au moins un paire $A[i], A[j]$ avec $f(A[i], A[j]) = c$, pour un c donné. Donnez un tel algorithme (entrée : $A[]$ et c , sortie : valeur booléenne). L'algorithme doit s'exécuter en temps $O(n)$ pour tout A et c . Justifiez pourquoi votre algorithme prend un temps linéaire.

Indice : trouvez d'abord un algorithme pour un f simple comme $f(a, b) = a + b$ et généralisez l'approche.

3.2 Somme (5 points)

On a un tableau $A[0..n-1]$ d'entiers, dans un ordre quelconque. Tout élément $A[k]$ peut être négatif, positif, ou 0. On veut trouver un sous-tableau $A[i..j-1]$ qui maximise $S(i, j) = \sum_{k=i}^{j-1} A[k]$ (on définit aussi $S(i, i) = 0$). Donnez un algorithme *récurif* qui trouve le meilleur sous-tableau en temps $O(n \log n)$. Justifiez pourquoi votre algorithme prend $O(n \log n)$.

Indice : coupez le tableau en deux au centre — le meilleur sous-tableau est soit entièrement à la gauche, soit entièrement à la droite, soit il traverse le centre.

3.3 ... en temps linéaire (2.015 points boni)

Donnez un algorithme qui trouve le meilleur sous-tableau du Problème 3.2 en $O(n)$.