Esiee-Paris - cours d’algorithmique - TP 1 - début février 2015

R. Natowicz, I. Alamé, A. Çela, D. Courivaud

Intersection de deux ensembles.

Les ensembles E_1 et E_2 sont des ensembles d’entiers, de cardinaux n_1 et n_2, $n_1 \leq n_2$, représentés dans deux tableaux strictement croissants $E_1[0 : n_1]$ et $E_2[0 : n_2]$. Le tableau E est de longueur n_1. On demande de calculer $E_1 \cap E_2$ dans E. La sortie $\mathcal{S}$ du programme est $E[0 : i] = E_1[0 : n_1] \cap E_2[0 : n_2]$.

Exemple : avec $E_1 = [0, 3, 6, 10, 13, 16, 20, 23, 26, 30]$ ($n_1 = 10$) et $E_2 = [0, 3, 6, 9, 12, 15, 18, 21, 24, 27, 30, 33, 36, 39, 42]$ on aura en sortie du programme $E = [0, 3, 6, 30, ...]$ et $i = 4$ (les valeurs $E[4 : 10]$ sont quelconques.)

q. 1 Soit $T[0 : n]$ un tableau d’entiers strictement croissants. Écrire une fonction booléenne $\mathrm{membre(x, T, n)}$ qui retourne en temps $\Theta(\log n)$ la vérité de la proposition “$x \in T[0 : n]$.” Cette fonction est construite sur le résultat de la recherche dichotomique de x dans T. La fonction $\mathrm{membre(x, T, n)}$ ne comportant aucune itération, il n’y a aucune propriété invariante à chercher.

Puis, ayant écrit la fonction $\mathrm{membre(x, T, n)}$, écrire un programme calculant la sortie $\mathcal{S}$ en temps en $\Theta(n_1 \times \log n_2)$.

q. 2 On se donne la propriété $\mathrm{I(i, j, k)}$: $E[0 : i] \cup (E_1[j : n_1] \cap E_2[k : n_2]) = E_1[0 : n_1] \cap E_2[0 : n_2]$. Écrire un programme calculant la même sortie $\mathcal{S}$, en temps $\Theta(n_1 + n_2)$.

q. 3 Les deux ensembles E_1 et E_2 sont de cardinaux $n_1 = n_2 = 10^6$. Lequel de vos deux programmes utiliserez-vous ? Même question lorsque $n_1 = 10$ et $n_2 = 10^6$. Justifiez vos réponses.

Recherche arrière.

Le tableau $T[0..n-1]$ d’entiers positifs ou nuls est strictement croissant. On veut calculer dans une variable C le nombre de couples de T dont la somme est dans T.
Exemple: avec $T = [0, 1, 2]$ on a $0 = 0 + 0,\ 1 = 0 + 1 = 1 + 0,\ 2 = 1 + 1 = 0 + 2 = 2 + 0,$ d’où : $C = 6$.

Un premier algorithme est : pour chaque couple d’indices (i, j), $(i, j) \in [0 : n] \times [0 : n]$, chercher la valeur $T[i] + T[j]$ dans T. Le tableau T étant strictement croissant, la recherche sera dichotomique (plutôt que séquentielle.) Cet algorithme est en $\Theta(n^2 \times \log n)$.

Donner un algorithme en $\Theta(n^2)$.

Pour chacun des programmes on donnera la propriété sur laquelle le programme est construit, l’initialisation, la condition d’arrêt, la progression, et le programme en langage Java commenté par l’invariant. On donnera également des exemples d’exécution.

Les TP sont à faire en binôme (un trinôme ou un monôme par groupe si le nombre d’élèves est impair.)

Les rapports sont à rendre au début de la prochaine séance de TD (11 ou 12 mars.)