

TD : Plus longue séquence croissante

Dominique Michelucci, Université de Dijon

5 novembre 2012

Un tableau non trié d'entiers $E[0], \dots, E[n-1]$ est donné. Le problème est de calculer la longueur de la séquence croissante la plus longue. Note : dans cette séquence, tout élément (sauf le dernier) est inférieur ou égal à son élément suivant. En première approximation, vous pouvez supposer pour simplifier que tous les éléments sont différents. Par exemple, si $E = [0; 300; 100; 200; 1000; 400; 500; 1100; 900; 800; 600; 700; -100]$, alors les séquences croissantes les plus longues ont 7 éléments. L'une d'elles est $[0; 100; 200; 400; 500; 600; 700]$.

Proposez une méthode en temps polynomial ($O(n^2)$). Par exemple, définir récursivement $LT[i]$, comme étant la longueur de la séquence croissante la plus longue qui se termine (et utilise) E_i . $LT[0] = 1$. Définissez $LT[i]$ en fonction de $LT[0], \dots, LT[i-1]$. Exemple :

i	0	1	2	3	4	5	6	7	8	9	10	11	12
E_i	0	300	100	200	1000	400	500	1100	900	800	600	700	-100
LT_i	1	2	2	3	4	4	5	6	6	6	6	7	1

Cette méthode est en temps $O(n^2)$. Donnez une méthode en $O(n \log n)$. Piste : stockez dans un tableau $V[l]$ la dernière valeur de la séquence de longueur l . Quand vous cherchez quelle est la plus longue séquence croissante que peut prolonger E_i , vous pouvez procéder par dichotomie dans le tableau V . Il faut aussi gérer L , la plus grande longueur courante des séquences croissantes. N'oubliez pas de mettre à jour le tableau V . Exemple :

i	0	1	2	3	4	5	6	7	8	9	10	11	12
E_i	0	300	100	200	1000	400	500	1100	900	800	600	700	-100
LT_i	1	2	2	3	4	4	5	6	6	6	6	7	1
V_i	-	-100	100	200	400	500	600	700	-	-	-	-	-

Programmez ceci en TP, en deux temps, d'abord la méthode en $O(n^2)$, puis la méthode en $O(n \log n)$. Vérifiez que les 2 programmes rendent les mêmes résultats, et que la méthode avec dichotomie est nettement plus rapide que la méthode en temps quadratique pour n assez élevé (quelques dizaines ou centaines de milliers).