TP : Plus longue séquence croissante

Votre programme et sa classe doivent porter votre nom (et prénom en cas d'homonymie). Appelez l'encadrant chaque fois que vous avez terminé une question.

Un tableau non trié d'entiers naturels $E[0], \ldots E[n-1]$ est donné. Il est généré par la méthode `exemple`. Le problème est de calculer la longueur de la séquence croissante (au sens large) la plus longue. Dans cette séquence, tout élément (sauf le dernier) est *inférieur ou égal* à son élément suivant. Par exemple, ci dessous, pour

$$E = [61; 44; \mathbf{15}; \mathbf{28}; \mathbf{31}; 20; \mathbf{57}; 4; 10; 28]$$

la séquence croissante la plus longue a 4 éléments. C'est :

$$E[2] = 15; E[3] = 28; E[4] = 31; E[6] = 57$$

Question 1. 6 points.

Calculez $LT[i]$, la longueur de la séquence croissante la plus longue qui se termine en E_i (et donc utilise E_i). Clairement $LT[0] = 1$; ensuite exprimez, pour l'indice i croissant de 1 à $n-1$, $LT[i]$ en fonction de $LT[0], \ldots LT[i-1]$; avant de tenter de le programmer, faites le "à la main" sur l'exemple ci-dessus, obtenu avec `exemple( 10)`. Vous devez obtenir cette figure (à lire de gauche à droite) :

i	0	1	2	3	4	5	6	7	8	9
E_i	61	44	15	28	31	20	57	4	10	28
LT_i	1	1	1	2	3	2	4	1	2	3

Question 2. 4 points.

Modifiez la méthode d'affichage qui est fournie pour que la séquence soit écrite dans le bon sens (croissant). Par exemple, au lieu d'écrire sur le terminal les termes de la séquence, vous les copiez dans un tableau auxiliaire, puis vous parcourez ce tableau auxiliaire en ordre inverse et affichez ces éléments sur votre terminal.

Question 3. 4 points.

Au lieu de chercher dans $E_0, \ldots E_{i-1}$ quelle est la plus longue séquence croissante que peut prolonger E_i, cherchez la dans un tableau $V_1, V_2, \ldots V_L$; $L = \max_{k=0}^{i-1} LT[k]$ est la longueur maximum des plus longues séquences connues quand E_i est traité, et V_l (avec $1 \leq l \leq L$) est la valeur du dernier élément (dans E) de la plus longue séquence croissante de longueur l. Si un tel l n'existe pas (quand E_i est plus petit que tous ses précédents, ou que i est nul), alors votre recherche séquentielle dans $V_1, \ldots V_L$ rendra 0. Refaites l'exemple précédent à la main. Vous devez obtenir cette figure (à lire de haut en bas) :

i	E_i	LT_i	V_1	V_2	V_3	V_4	L
0	61	1	**61**				1
1	44	1	**44**				1
2	15	1	**15**				1
3	28	2	15	**28**			2
4	31	3	15	28	**31**		3
5	20	2	15	**20**	31		3
6	57	4	15	20	31	**57**	4
7	4	1	**4**	20	31	57	4
8	10	2	4	**10**	31	57	4
9	28	3	4	10	**28**	57	4

Question 4. 6 points.

Cherchez l par dichotomie dans le tableau V. Il faut trouver dans V l'indice l (l est une longueur entre 1 et L) le plus grand tel que V_l soit inférieure ou égale à E_i. Or le tableau $V_1, \ldots V_L$ est toujours trié par ordre croissant, comme sur l'exemple précédent : $V_1 \leq \ldots \leq V_L$. Vérifiez que toutes les méthodes rendent les mêmes tableaux LT, pour des E identiques, et que la seconde méthode avec une recherche dichotomique est nettement plus rapide pour des tailles de cent mille ou un million d'éléments dans E.

Bonus, 2 points.

(pour rattraper des points sur votre partiel, si vous avez les 20 points). La méthode d'affichage est en $O(n)$, et on préférerait une méthode qui prenne un temps proportionnel à la longueur de la séquence. Dans la méthode naïve, stockez dans un tableau P_i l'indice k de l'élément E_k précédant immédiatement E_i dans la plus longue séquence qui utilise E_i (-1 s'il n'y en a pas de tel k). Ecrivez une méthode d'affichage (en ordre inverse, si c'est le plus simple) qui utilise P. Remarque : P_i est similaire au prédécesseur du sommet i dans les méthodes de plus courts chemins.

Bonus, 2 points.

Faites de même dans la méthode rapide avec recherche dichotomique.