Sous-suite croissante de longueur maximum

Thibaut Cantaluppi

October 14, 2024

Sous-suite croissante

Soit L une liste.

Définition

Une **sous-suite croissante** de L est une suite $L[i_1] \leq L[i_2] \leq ... \leq L[i_k]$ avec $i_1 \leq i_2 \leq ... \leq i_k$.

Sous-suite croissante

Soit L une liste.

Définition

Une **sous-suite croissante** de L est une suite $L[i_1] \leq L[i_2] \leq ... \leq L[i_k]$ avec $i_1 \leq i_2 \leq ... \leq i_k$.

Problème

Trouver la longueur maximum d'une sous-suite croissante de L.

Sous-suite croissante

Soit L une liste.

Définition

Une **sous-suite croissante** de L est une suite $L[i_1] \leq L[i_2] \leq ... \leq L[i_k]$ avec $i_1 \leq i_2 \leq ... \leq i_k$.

Problème

Trouver la longueur maximum d'une sous-suite croissante de L.

Exemple :

$$L = [8, 1, 3, 7, 5, 6, 4]$$

Sous-suite croissante

Soit L une liste.

Définition

Une **sous-suite croissante** de L correspond à des éléments $L[i_1] \leq L[i_2] \leq ... \leq L[i_k]$ avec $i_1 \leq i_2 \leq ... \leq i_k$.

Problème

Trouver la longueur maximum d'une sous-suite croissante de L.

Exemple :

$$L = [8, \textcolor{red}{1}, \textcolor{red}{3}, 7, \textcolor{red}{5}, \textcolor{red}{6}, 4]$$

Longueur maximum : 4.

Sous-suite croissante

Soit L une liste.
Soit $dp[k]$ la longueur d'une plus longue sous-suite croissante (**LIS** en anglais, pour Longest Increasing Subsequence) terminant en $L[k]$ (c'est à dire de la forme $L[i_1] \leq L[i_2] \leq ... \leq L[i_p] = L[k]$).

Sous-suite croissante

Soit L une liste.
Soit $dp[k]$ la longueur d'une plus longue sous-suite croissante (**LIS** en anglais, pour Longest Increasing Subsequence) terminant en $L[k]$ (c'est à dire de la forme $L[i_1] \leq L[i_2] \leq ... \leq L[i_p] = L[k]$).

<u>Exemple</u> :

$$L = [8, 1, 3, 7, 5, 6, 4]$$

LIS terminant en $L[6]$ $(= 4)$:

Sous-suite croissante

Soit L une liste.
Soit $dp[k]$ la longueur d'une plus longue sous-suite croissante (**LIS** en anglais, pour Longest Increasing Subsequence) terminant en $L[k]$ (c'est à dire de la forme $L[i_1] \leq L[i_2] \leq ... \leq L[i_p] = L[k]$).

Exemple :

$$L = [8, 1, 3, 7, 5, 6, 4]$$

LIS terminant en $L[6]$ $(= 4)$:

$$L = [8, \textcolor{red}{1}, \textcolor{red}{3}, 7, 5, 6, \textcolor{red}{4}]$$

Donc $dp[6] = 3$.

Sous-suite croissante

Soit $L[i_1] \leq ... \leq L[i_{p-1}] \leq L[i_p] = L[k]$ une LIS terminant en $L[k]$.

Sous-suite croissante

Soit $L[i_1] \leq ... \leq L[i_{p-1}] \leq L[i_p] = L[k]$ une LIS terminant en $L[k]$.

Alors $L[i_1] \leq ... \leq L[i_{p-1}]$ est une LIS terminant en $L[i_{p-1}]$

Sous-suite croissante

Soit $L[i_1] \leq ... \leq L[i_{p-1}] \leq L[i_p] = L[k]$ une LIS terminant en $L[k]$.

Alors $L[i_1] \leq ... \leq L[i_{p-1}]$ est une LIS terminant en $L[i_{p-1}]$ (s'il y avait une LIS plus grande on pourrait l'utiliser dans la LIS initiale pour contredire sa maximalité).

Sous-suite croissante

Soit $L[i_1] \leq ... \leq L[i_{p-1}] \leq L[i_p] = L[k]$ une LIS terminant en $L[k]$.

Alors $L[i_1] \leq ... \leq L[i_{p-1}]$ est une LIS terminant en $L[i_{p-1}]$ (s'il y avait une LIS plus grande on pourrait l'utiliser dans la LIS initiale pour contredire sa maximalité).
Donc :

$$dp[k] = 1 + dp[i_{p-1}]$$

Sous-suite croissante

Soit $L[i_1] \leq ... \leq L[i_{p-1}] \leq L[i_p] = L[k]$ une LIS terminant en $L[k]$.

Alors $L[i_1] \leq ... \leq L[i_{p-1}]$ est une LIS terminant en $L[i_{p-1}]$ (s'il y avait une LIS plus grande on pourrait l'utiliser dans la LIS initiale pour contredire sa maximalité).
Donc :

$$dp[k] = 1 + dp[i_{p-1}]$$

Comme on ne connaît pas i_{p-1}, on peut essayer toutes les possibilités et conserver le maximum :

$$\boxed{dp[k] = 1 + \max_{\substack{i<k \\ L[i]<L[k]}} dp[i]}$$

Sous-suite croissante : Devoir Maison 1

A faire pour la rentrée et à envoyer sous forme d'un unique fichier .py (ou .ipynb pour les fans de jupyter) à thibaut.cantaluppi@ac-lyon.fr.

La complexité de vos fonctions devra systématiquement être justifiée en commentaire.

Sous-suite croissante : Devoir Maison 1

Question 1

Écrire une fonction `lis_rec(L)` renvoyant la plus longue sous-suite croissante de `L`. Cette fonction devra être "naivement" recursive.

Question 2

Écrire une fonction `lis_dyn(L)` renvoyant la plus longue sous-suite croissante de `L`. Cette fonction devra être implémentée en programmation dynamique.

Question 3

Écrire une fonction `lis_mem(L)` renvoyant la plus longue sous-suite croissante de `L`. Cette fonction devra utiliser la mémoïsation (pas de @cache !!).

Bonus - Théorème d'Erdős-Szekeres

Lemme

Supposons que $L[k]$ contienne p fois la même valeur. Montrer que L possède une sous-suite décroissante de longueur p.

Bonus - Théorème d'Erdős-Szekeres

Lemme

Supposons que $L[k]$ contienne p fois la même valeur. Montrer que L possède une sous-suite décroissante de longueur p.

Théorème d'Erdős-Szekeres

Si n est la taille de L, montrer que L contient soit une sous-suite croissante de longueur $\lfloor \sqrt{n} \rfloor$, soit une sous-suite décroissante de longueur $\lfloor \sqrt{n} \rfloor$