3-1 Dynamic Programming

(Part I: Examples)

Hengfeng Wei

hfwei@nju.edu.cn

September 10, 2018

Printing Neatly (Problem 15-4)

A sequence of n words of lengths $l_1, l_2, \cdots, l_n$

Line width M

$$\textit{extra}[i, j] = M - (j - i) - \sum_{k=i}^{j} l_k$$

Printing Neatly (Problem 15-4)

A sequence of n words of lengths $l_1, l_2, \cdots, l_n$

Line width M

$$\textit{extra}[i,j] = M - (j-i) - \sum_{k=i}^{j} l_k$$

To minimize the sum, over all lines *except the last*, of the cubes of the numbers of extra space characters at the ends of lines.

$$c(n) = \min_{L} \sum_{l_{[i,j]} \in L \,\wedge\, j \neq n} \Big(\textit{extra}[i,j]\Big)^3$$


LaTeX
Beginner's Guide
TEX}

A sequence of n words of lengths $l_1, l_2, \cdots, l_n$

A sequence of n words of lengths $l_1, l_2, \cdots, l_n$

$C(i)$: $\min$ cost of neatly printing the *first* i words

$C(i)$: $\min$ cost of neatly printing the *last* words i through n

$C(i, j)$: $\min$ cost of neatly printing *words* i *through* j

$C(i) : \min$ cost of neatly printing the *last* words i through n

$C(i): \min$ cost of neatly printing the *last* words i through n

$\boxed{\textit{How many words to place on the first line?}}$

$C(i)$: $\min$ cost of neatly printing the *last* words i through n

How many words to place on the first line?

$$C(i) = \min_{\substack{0 < k \le (n-i+1) \\ \textit{extra}[i, i+k-1] \ge 0}} \Big(\textit{extra}[i, i+k-1]\Big)^3 + C(i+k)$$

$C(i) : \min$ cost of neatly printing the *last* words i through n

How many words to place on the first line?

$$C(i) = \min_{\substack{0 < k \leq (n-i+1) \\ \textit{extra}[i, i+k-1] \geq 0}} \Big(\textit{extra}[i, i+k-1]\Big)^3 + C(i+k)$$

$$C(i) = 0, \quad \text{if } \textit{extra}[i, n] \geq 0$$

$C(i) : \min$ cost of neatly printing the *last* words i through n

How many words to place on the first line?

$$C(i) = \min_{\substack{0 < k \le (n-i+1) \\ \textit{extra}[i, i+k-1] \ge 0}} \Big(\textit{extra}[i, i+k-1] \Big)^3 + C(i+k)$$

$$C(i) = 0, \quad \text{if } \textit{extra}[i, n] \ge 0$$

$$O(nW)$$

procedure $\textsc{Printing-Neatly}(n)$

  for $i \leftarrow n \downarrow 1$ **do**

    if $\textit{extra}[i, n] \geq 0$ **then** ▷ put w_i through w_n on a line

      $C[i] \leftarrow 0$

    else

      $$C(i) = \min_{\substack{0 < k \leq (n-i+1) \\ \textit{extra}[i, i+k-1] \geq 0}} \Big(\textit{extra}[i, i+k-1]\Big)^3 + C(i+k)$$

procedure $\textsc{Printing-Neatly}(n)$

  for $i \leftarrow n \downarrow 1$ **do**

    if $\textit{extra}[i, n] \geq 0$ **then** ▷ put w_i through w_n on a line

      $C[i] \leftarrow 0$

      $B[i] \leftarrow n$

    else

      $$C(i) = \min_{\substack{0 < k \leq (n-i+1) \\ \textit{extra}[i, i+k-1] \geq 0}} \Big(\textit{extra}[i, i+k-1]\Big)^3 + C(i+k)$$

      $B[i] = i + k$ ▷ line break

procedure $\textsc{Printing-Neatly}(n)$

  for $i \leftarrow n \downarrow 1$ **do**

    if $\textit{extra}[i, n] \geq 0$ **then** ▷ put w_i through w_n on a line

      $C[i] \leftarrow 0$

      $B[i] \leftarrow n$

    else

      $$C(i) = \min_{\substack{0 < k \leq (n-i+1) \\ \textit{extra}[i, i+k-1] \geq 0}} \Big(\textit{extra}[i, i+k-1]\Big)^3 + C(i+k)$$

      $B[i] = i + k$ ▷ line break

$B[1],$

procedure PRINTING-NEATLY(n)

  for $i \leftarrow n \downarrow 1$ **do**

    if $extra[i, n] \geq 0$ **then** ▷ put w_i through w_n on a line

      $C[i] \leftarrow 0$

      $B[i] \leftarrow n$

    else

      $C(i) = \min\limits_{\substack{0 < k \leq (n-i+1) \\ extra[i, i+k-1] \geq 0}} \Big(extra[i, i+k-1]\Big)^3 + C(i+k)$

      $B[i] = i + k$ ▷ line break

$$B[1], \quad B[B[1]+1], \quad \cdots$$

A $\boxed{\text{sequence}}$ of n words of lengths $l_1, l_2, \cdots, l_n$

A $\boxed{\text{sequence}}$ of n words of lengths $l_1, l_2, \cdots, l_n$

A $\boxed{\text{set}}$ of n words of lengths $l_1, l_2, \cdots, l_n$

A $\boxed{\text{sequence}}$ of n words of lengths $l_1, l_2, \cdots, l_n$

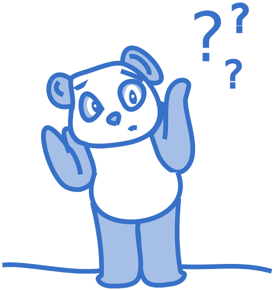
A $\boxed{\text{set}}$ of n words of lengths $l_1, l_2, \cdots, l_n$

Greedy? *DP?* *NP-hard?*

A sequence of n words of lengths $l_1, l_2, \cdots, l_n$

A set of n words of lengths $l_1, l_2, \cdots, l_n$

Greedy? *DP?* *NP-hard?*



Longest Increasing Subsequence (Problem 15.4-5)

$$A[1 \ldots n]$$

$$5,\ 2,\ 8,\ 6,\ 3,\ 6,\ 9,\ 7$$

Find (the length of) a longest increasing (non-decreasing) subseqence.

$$5,\ \color{red}{2},\ 8,\ 6,\ \color{red}{3},\ \color{red}{6},\ \color{red}{9},\ 7$$

$L(i)$: $\boxed{\text{the length of an LIS of } A[1 \ldots i]}$

$$L(n)$$

$L(i)$: $\boxed{\text{the length of an LIS of } A[1 \ldots i]}$

$L(n)$

Is $A[i]$ in this LIS of $A[1 \cdots i]$?

$L(i)$: $\boxed{\text{the length of an LIS of } A[1 \ldots i]}$

$$L(n)$$

Is $A[i]$ in this LIS of $A[1 \cdots i]$?

$$L(i) = \max\Big(\underbrace{L(i-1)}_{\text{NO}}, \qquad\qquad\qquad \Big)$$

$L(i)$: $\boxed{\text{the length of an LIS of } A[1 \ldots i]}$

$$L(n)$$

Is $A[i]$ in this LIS of $A[1 \cdots i]$?

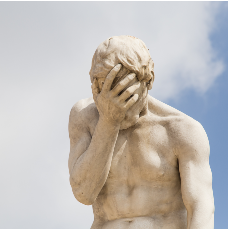
$$L(i) = \max\Big(\underbrace{L(i-1)}_{\text{NO}},\ \underbrace{1 + \max_{j<i \,\land\, A[j] \le A[i]} L(j)}_{\text{YES}}\Big)$$

$L(i)$: $\boxed{\text{the length of an LIS of } A[1 \dots i]}$

$$L(n)$$

Is $A[i]$ in this LIS of $A[1 \cdots i]$?

$$L(i) = \max \Big(\underbrace{L(i-1)}_{\text{NO}}, \underbrace{1 + \max_{j<i \,\wedge\, A[j] \le A[i]} L(j)}_{\text{YES}} \Big)$$



$L(i)$: $\boxed{\text{the length of an LIS } \textit{ending with } A[i]}$

$$\max_i L(i)$$

$L(i)$: the length of an LIS *ending with* $A[i]$

$$\max_i L(i)$$

What is the previous element?

$L(i)$: $\boxed{\text{the length of an LIS } \textit{ending with } A[i]}$

$$\max_i L(i)$$

What is the previous element?

$$L(i) = 1 + \max_{j<i \,\wedge\, A[j] \le A[i]} L(j)$$

$L(i)$: the length of an LIS *ending with* $A[i]$

$$\max_i L(i)$$

What is the previous element?

$$L(i) = 1 + \max_{j<i \,\wedge\, A[j] \le A[i]} L(j)$$

$$L(1) = 1$$

$L(i)$: $\boxed{\text{the length of an LIS } \textit{ending with } A[i]}$

$$\max_i L(i)$$

What is the previous element?

$$L(i) = 1 + \max_{j<i \,\wedge\, A[j] \le A[i]} L(j)$$

$$L(1) = 1$$

$$O(n^2) = \Theta(n) \cdot O(n)$$

$$\mathsf{LIS(A)} = \mathsf{LCS}\Big(A, \textsc{Sort}(A)\Big)$$

$$\mathsf{LIS(A)} = \mathsf{LCS}\Big(A, \textsc{Sort}(A)\Big)$$

$$O(n^2) = O(n \log n) + O(n^2)$$

Longest Increasing Subsequence (Problem 15.4-6)

$$O(n \log n)$$

LIS$[i]$: all increasing subsequences of length $1 \le l \le i$ using $A[1 \cdots i]$

Longest Increasing Subsequence (Problem 15.4-6)

$$O(n \log n)$$

LIS$[i]$: all increasing subsequences of length $1 \leq l \leq i$ using $A[1 \cdots i]$

E : $E[l]$ = all increasing subsequences of length $l, \quad 1 \leq l \leq n$

Longest Increasing Subsequence (Problem 15.4-6)

$$O(n \log n)$$

LIS$[i]$: all increasing subsequences of length $1 \leq l \leq i$ using $A[1 \cdots i]$

E : $E[l] =$ all increasing subsequences of length $l, \quad 1 \leq l \leq n$

$$\max\{k \mid E[k] \neq \emptyset\}$$

$E[l]$ = all increasing subsequences of length l, $\quad 1 \le l \le n$

$E[l]$ = all increasing subsequences of length $l, \quad 1 \le l \le n$

$$\{3 \quad 9 \quad 11 \quad 13\}$$

$$\{4 \quad 6 \quad 10 \quad 15\}$$

$E[l] =$ all increasing subsequences of length $l, \quad 1 \le l \le n$

$$\{3 \quad 9 \quad 11 \quad 13\}$$

$$\{4 \quad 6 \quad 10 \quad 15\}$$

Keep only the *smallest* ending number per distinct length l

$\text{LIS}[i]$:

$E[l] =$ the *smallest ending number* for the increasing subsequence of length l using $A[1 \cdots i]$


THANK YOU