3-1 Dynamic Programming

(Part I: Examples)

Hengfeng Wei

hfwei@nju.edu.cn

September 10, 2018

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.

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

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

$L(n)$

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

$$L(n)$$

$\boxed{\textit{Is } A[i] \textit{ in this LIS of } A[1 \cdots i]\textit{?}}$

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

$$L(n)$$

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

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

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

$$L(n)$$

$\boxed{\textit{Is } A[i] \textit{ in this LIS of } A[1 \cdots i]\textit{?}}$

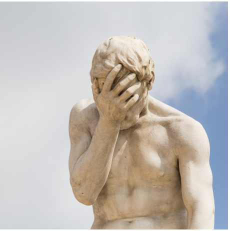
$$L(i) = \max \Big(\underbrace{L(i-1)}_{\text{NO}}, \underbrace{1 + \max_{j < i \,\wedge\, A[j] \le A[i]} L(j)}_{\text{YES}} \Big)$$

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

$$L(n)$$

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

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



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

$$\max_i L(i)$$

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

$$\max_i L(i)$$

What is the previous element?

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

$$\max_i L(i)$$

What is the previous element?

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

$L(i)$: 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)

$$O(n \log n)$$

Longest Increasing Subsequence (Problem 15.4-6)

$$O(n \log n)$$

Longest Increasing Subsequence (Problem 15.4-6)

$$O(n \log n)$$

Answer by Eugene Yarovoi @ Quora

$$E[l] = \{\textit{all increasing subsequences of length } l\}$$

$$E[l] = \{\textit{all increasing subsequences of length } l\}$$

```
procedure LIS(n)
    E[l] ← ∅,    ∀1 ≤ i ≤ n
    for i ← 1 ↑ n do
        ∀1 ≤ l ≤ i : E[l] ←
            {all inc. subseq. of length l}
    return
```

$$E[l] = \{\textit{all increasing subsequences of length } l\}$$

procedure *LIS*(n)

  $E[l] \leftarrow \emptyset, \quad \forall 1 \le i \le n$

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

    $\forall 1 \le l \le i : E[l] \leftarrow$

      $\{\textit{all inc. subseq. of length } l\}$

  return $\max \{l \mid E[l] \neq \emptyset\}$

$$E[l] = \{\textit{all increasing subsequences of length } l\}$$

procedure *LIS*(n)
  $E[l] \leftarrow \emptyset, \quad \forall 1 \le i \le n$
  for $i \leftarrow 1 \uparrow n$ **do**
    $\forall 1 \le l \le i : E[l] \leftarrow$ ▷ *By extending each shorter subseq.*
      $\{\textit{all inc. subseq. of length } l\}$
  return $\max \{l \mid E[l] \ne \emptyset\}$

$$E[l] = \{\textit{all increasing subsequences of length } l\}$$

$$E[l] = \{\textit{all increasing subsequences of length } l\}$$

$$\langle 3 \quad 9 \quad 11 \quad 13 \rangle$$

$$\langle 4 \quad 6 \quad 10 \quad 15 \rangle$$

$$\langle 2 \quad 5 \quad 12 \quad 18 \rangle$$

$$E[l] = \{\textit{all increasing subsequences of length } l\}$$

$$\langle 3 \quad 9 \quad 11 \quad 13 \rangle$$

$$\langle 4 \quad 6 \quad 10 \quad 15 \rangle$$

$$\langle 2 \quad 5 \quad 12 \quad 18 \rangle$$

$\Big|E[l]\Big| = 1$: *the one with the smallest ending number*

procedure *LIS*(n)

  $E[l] \leftarrow \emptyset, \quad \forall 1 \leq i \leq n$

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

    $\forall 1 \leq l \leq i : E[l] \leftarrow$ ▷ *By extending each shorter subseq.*

      the inc. subseq. of length l with the smallest ending number

  return $\max \{l \mid E[l] \neq \emptyset\}$

$$E[l] = \{\textit{all increasing subsequences of length } l\}$$

$$\Big|E[l]\Big| = 1 : \textit{the one with the smallest ending number}$$

$$E[l] = \{\textit{all increasing subsequences of length } l\}$$

$\Big|E[l]\Big| = 1$: *the one with the smallest ending number*

$$\langle 2 \quad 8 \quad 15 \rangle$$

$$\langle 4 \quad 6 \quad 11 \quad 13 \rangle$$

$$\langle 3 \quad 9 \quad 11 \quad 12 \quad 13 \rangle$$

$E[l] = \{$*all increasing subsequences of length* $l\}$

$\Big|E[l]\Big| = 1$: *the one with the smallest ending number*

$$\langle 2 \quad 8 \quad 15 \rangle$$

$$\langle 4 \quad 6 \quad 11 \quad 13 \rangle$$

$$\langle 3 \quad 9 \quad 11 \quad 12 \quad 13 \rangle$$

$\forall i < j$: *the ending number of* $E[i] <$ *the ending number of* $E[j]$

procedure *LIS*(n)

$E[l] \leftarrow \emptyset, \quad \forall 1 \le i \le n$

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

$\forall 1 \le l \le i : E[l] \leftarrow$ ▷ *By extending each shorter subseq.*

the inc. subseq. of length l *with the smallest ending number*

return $\max \{l \mid E[l] \neq \emptyset\}$

procedure *LIS*(n)

$E[l] \leftarrow \emptyset, \quad \forall 1 \leq i \leq n$

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

$\forall 1 \leq l \leq i : E[l] \leftarrow$ ▷ *Extending only one subseq using binary search*

the inc. subseq. of length l *with the smallest ending number*

return $\max \{l \mid E[l] \neq \emptyset\}$

Extending

Extending

by looking at only the ending number of the inc. subseq

Extending

by looking at only the ending number of the inc. subseq

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


THANK YOU