

## Greedy algorithms

- LIS
- Weighted flow
- Gone fishing.

◦ LIS: Longest Increasing Subsequence

input:  $\underline{2}$  4  $\underline{3}$   $\underline{5}$  1  $\underline{6}$  - I

Subsequence: 2 4 5 ✓ 3 1 6 ✓  $\underline{2 3 4}$  X

Increasing subseq: ✓ X X

Output: length of the LIS

Trial 1    ① Sort  $I \rightarrow I'$   
                   ②  $LCS(I, I')$

proof: LIS = 0 0 0  
 LIS is subsequence  
 of  $I$

①  $O(n \log n)$   
 ②  $O(n^2)$     }  $O(n^2)$

LIS is subseq of  $I'$

LIS is CS of  $Z$  &  $I'$

$LIS \subseteq LCS(I, I')$

$LCS(Z, I')$  is IS of  $I$

$LCS(I, I') \subseteq LIS$

Trial 2    Solitaire algorithm <sup>distinct</sup>

$c_1, c_2, \dots, c_n = I$

1	3		
2	4	5	6
<u>      </u>	<u>      </u>	<u>      </u>	<u>      </u>
$s_1$	$s_2$	$s_3$	... stacks

for  $i = 1, 2, \dots, n$

- find the first stack  $s_j$   
 s.t. the top elt on  $s_j > c_i$

- put  $c_i$  on  $s_j$

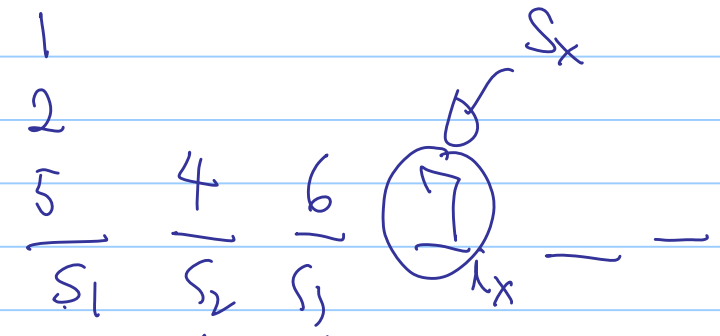
~~2~~ ~~4~~ 5 6

LIS = # of non empty stack

5 2 1 4 6 7

Proof:

Claim 1 We can find an LIS that takes one int from each stack



$\Rightarrow$  LIS  $\geq$  # of non empty stack

- ① take any int from  $S_x : i_x$  ✓
- ② the top int  $i_{x-1}$  in  $S_{x-1} < i_x$  : ①  $i_{x-1} < i_x$  ②  $i_{x-1}$  occurs before  $i_x$   
 top int in  $S_{x-1}$  when we process  $i_x$
- ③ the top int  $i_{x-2}$  in  $S_{x-2} < i_{x-1}$

$\Rightarrow$  take  $i_x, i_{x-1} \dots i_2, i_1 : X$  int #

Claim 2 LIS is put on different stacks  $\Rightarrow$  # of stacks  $\geq$  LIS

$\downarrow$   
 $j_1 j_2 \dots j_x$

$\underbrace{j_1 \quad j_2 \quad j_3}_{S_y}$

Alg correct  $\checkmark$

Time  
-

Obs: At any time,  $\text{top}(S_1) < \text{top}(S_2) < \dots < \text{top}(S_x)$

MI: After  $C_1$   $\checkmark$

Assume: After  $C_k$   $\checkmark$   $\text{top}(S_1) < \text{top}(S_2) \dots < \text{top}(S_x)$

$C_{k+1}$ , Assume  $C_{k+1}$  is part of  $S_y$

$$\text{top}(S_1) < \text{top}(S_2) < \dots < \text{top}(S_{y-1}) < \text{top}(S_y) < \text{top}(S_{y+1}) < \dots < \text{top}(S_x)$$

$\downarrow$   $C_{k+1}$

$\Rightarrow$  to find the first  $S_y$  with  $\text{top}(S_y) > C_i$

Can be done by binary search

lower\_bound

poj 3670:

input: 1 2 1 3 1 2 1 3  
          1      1      2 3

Target:

change some outputs re. 1 1 2 ... 3 ...  
                                  3 3 ... 2 2 ... 1 1

$$I \Rightarrow 1^* 2^* 3^* \dots 3^* 2^* 1^*$$

Believe:  $n$ -LIS or  $n$ -LDS is  $n$ .

Assume OPT changes:  $P_{i_1} P_{i_2} \dots P_{i_k}$

Claim  $I / \{P_{i_1} P_{i_2} \dots P_{i_k}\} = \text{LIS or LDS}$

Assume  $I / \{P_{i_1} P_{i_2} \dots P_{i_k}\}$  is not LIS or LDS

$I$     0   ~~0~~   0   /0   0   ~~0~~   /0

LIS    0   0   0   X   0   X   X   0    $\rightarrow$  longer LIS

