Note Title 2011-03-31

Greedy algorithms

- LIS
- Weighted flow
- Gone fishing.

- LIS: Longest Increasing Subsequence

input: 2 4 3 5 1 6 — I

Subsequence: 2 4 5 ✓ 3 1 6 ✓ 2 3 4 ✗

Increasing subseq: ✓ ✗ ✗

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

LIS is subseq of I'

LIS is CS of I & I'

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

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

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

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

$\Big\} \; O(n^2)$

Trial 2 Solitaire algorithm

distinct → $c_1 \; c_2 \; .. \; c_n = I$

for $i = 1, 2 .. n$
- find the first stack S_j s.t the top elmt on $S_j > c_i$
- put c_i on S_j

1	3			
2	4	5	6	
S_1	S_2	S_3	..	stacks

LIS = # of non empty stack

~~2~~ ~~4~~ ~~3~~ ~~5~~ 1 6

5 2 1 4 6 7

Proof:

1			S_x		
2					
5	4	6	7		
S_1	S_2	S_3	i_x		

<u>Claim 1</u> We can find an IS 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 $S_{x-2} < i_{x-1}$

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

Claim 2 LIS is put on different stacks ⇒ # of stacks ≥ LIS

$j_1 \; j_2 \dots j_x$

Alg correct ✓

Time

$<j_i' \quad <j_1' \quad j_1 \quad j_2 \quad j_3$

S_y

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

MI: After C_1 ✓

Assume: After C_k ✓ $top(S_1) < top(S_2) \dots < top(S_x)$

C_{k+1}, Assume C_{k+1} is ~~put~~ on S_y

$$\mathrm{top}(S_1) < \mathrm{top}(S_2) < \mathrm{top}(S_{y-1}) \overset{\checkmark}{<} \overset{C_{k+1}}{\cancel{\mathrm{top}(S_y)}} < \mathrm{top}(S_{y+1}) < \mathrm{top}(S_x)$$

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

can be done by $\boxed{\text{binary search}}$

lower_bound

POJ 3670:

input: $1 \; \overset{}{\cancel{2}}_{1} \; 1 \; \cancel{3}_{1} \; 1 \; 2 \; \cancel{1}_{2} \; \cancel{1}_{3} \; 3$

Target:

change some integers s.t. 1 1 1. 2... 3..

3 3 3 .. 2 2 .. 1 1 1

$I \Rightarrow \quad 1^* \quad 2^* \quad 3^* \quad n \quad 3^* \quad 2^* \quad 1^*$

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

Assume OPT changes: $p_{i_1} \quad p_{i_2} \; . \; p_{i_k}$

Claim $I / \{p_{i_1} \, p_{i_2} . \, p_{i_k}\} = $ LIS or LDS

Assume $I / \{p_{i_1} \, p_{i_2} . \, p_{i_k}\}$ is not LIS or LDS

$I \quad o \quad \cancel{o} \quad o \quad \cancel{o} \quad o \quad \cancel{o} \quad \cancel{o} \quad o$

LIS $\quad o \quad o \quad o \quad X \quad o \quad X \quad X \quad o \quad \Rightarrow$ longer IS or D/