Tutorial-8

Graded

Student

Abhinav Shripad

Total Points

2 / 3 pts

Question 1

1

Resolved 2 / 3 pts

+ 0.6 pts do not know how to approach this problem

✔ **+ 0.25 pts** Sorting in non increasing order (or non decreasing order if ans is calculated for stacking big box on small box) on w_i or l_i or $w_i \times l_i$

✔ **+ 0.25 pts** Correct base case for the recurrence

✔ **+ 1 pt** Correct recurrence relation

✔ **+ 0.25 pts** justification/correctness of the recurrence relation

+ 0.25 pts Mentioning the correct order of filling the DP table -

+ 0.5 pts Outputting the optimal stacking (not just the optimal height)

✔ **+ 0.5 pts** Brief justification of the time complexity

+ 0 pts Incorrect

– 0.25 pts Point adjustment

1 no constraint check on length

2 incorrect base case, it should be dp[i] = hi or if you are taking dp[0] = 0 then box 0 should have width and length (inf, inf).

Regrade Request **Submitted on: Oct 14**

Sir I did not check the constraint on length, because I sorted the boxes in decreasing order of lengths, written in the first line of the solution.
I also wrote the correct order of computation/filling of dp table, it is just above the underlined Answer. It has 0.25 marks allotted for it in the rubric.

It can be the case that two boxes have same length so you still need a check their.
Ok increased.

Reviewed on: Oct 15

COL351: Analysis and Design of Algorithms
Tutorial 8

Name: Abhinav R. Shripad
Entry number: 2022CS11596
Date: Sep 26, 2024
Group: 3

Sort the boxes in decreasing order of length l_i, rename them to be $1, 2, 3 \cdots n^{th}$ box.

dp(i) = maximum height if only boxes $1, 2, 3 \cdots i$ are considered and box 'i' is always taken

(n+1) size table

~~dp(1) =~~ $dp(0) = 0$ # Base Case

$$dp(i) = h_i + \max_{0 \le j \le i-1,\ w_j \ge w_i} (dp(j)) \quad \cdots (1)$$

Recurrence (1) is algorithm.

T.C. = $O(n^2)$, S.C. = $O(n)$

Order of computation of dp → Increasing order of i

Answer = $\max_{0 \le i \le n} (dp(i))$

Justification:- Consider the ~~top~~ longest box (longest l), ~~it can~~ ie l_1
only way it can be in solution is if it is at the bottom. ~~so~~ Similarly for

ith box, it can be at top of the tower only if tower is made from boxes 1,2 .. i (because sorted in ↓ order of l). So if ith is at the top, the lower half must be compatible with weight (as length is taken care of) thus we take max(dp(j)) under the constraint

$0 \le j \le i-1$, $w_j > w_i$.

Answer is $\max_{0 \le i \le n} (dp(i))$ because say box x is at the top in optimal soln → dp(x) will give that.