

CSCI – B505 Applied Algorithms

Assignment – 3

Submitted by – Vishal Singh (singhvis)

Problem 2

Method-

We use **bottom-up approach** and create a table(**dynamic programming (DP) table**) which stores the maximum sum of weights for a given set of jobs with a maximum finish time such that the jobs do not overlap, since the jobs are specified in increasing order of finishing time table[i] will store the value of the maximum sum of weights up till job i.

Now for a given value of i in the DP table, we find the maximum sum of weights for jobs which do not overlap with the job[i], we find all such jobs using binary search and then select the jobs which give the maximum weight.

For e.g.

For input:

Start time	End time	Weight
1	2	50
3	5	30
6	19	100
2	100	200

If we create a DP table:

I = 1 (up till job 1)

50			
----	--	--	--

This can be considered as the **base case** as in case of one job, the DP table can be filled in just one way

Since there is only 1 job

I = 2 (up till job 2)

50	80		
----	----	--	--

Since the 1st job doesn't overlap with 2nd job, the binary search will return the index of the job with max finish time (1st job) which does not overlap with 2nd job

Similarly

$I = 3$

50	80	180	
----	----	-----	--

Now for the final job ($I = 4$)

The binary search will return the highest index of the job which does not overlap with the 4th job, in this case $I = 1$

Now, we will compare the value of weight of Job 1 + job 4 against the ones calculated for Job 3 and store the value which is greater

In this case $\text{job 1} + \text{job 4} > \text{job 3}(\text{job 3} + \text{job 2} + \text{job 1})$

There the DP table will look like

50	80	180	250
----	----	-----	-----

This method uses a binary search to return values for compatible jobs and fill the DP table which is of length n

Therefore, time complexity $\rightarrow O(n * \log n)$

References:

1. <https://www.geeksforgeeks.org/weighted-job-scheduling-log-n-time/>
2. <https://www.youtube.com/watch?v=cr6lp0J9izc>

Instructions for running the code:

1. The program dp2.py takes parameter -f of the input file
2. The code prints of the maximum weight that can be obtained from the given input