Question4:

Given: List of jobs, List of profit of jobs, List of deadline of jobs.

Aim: Design a algorithm that have O(n^2)

Solution:

Step 1:

Sort the profit list from highest to lowest.

Step 2:

Sort the list of jobs depends on the profit.

Step 3:

Sort the deadline jobs depends on the profit. Create a timeline set that start from 0 and end at latest deadline.

Step 4:

Apply greedy method, Then pick the first element ‘Jn’ of sorted job list put it into the timeline in the section from Dn-1 to Dn, because in greedy method select the highest profit and put them into corresponded deadline will reduce the conflict condition and it is the highest profit which is partial optimal solution.

Step 5:

If the deadline is occupied then put it into the previous section which is from Dn-2 to Dn-1, if the previous is also occupied keep finding the previous sections until there is an empty and put it in. Else ignore this job and find next job in the job list. In Worst case, each jobs from the job list will try to go through each deadline time unit on the time line. So there will be 1+2+3+4+…+n-1+n times compare which made the time complexity become O(n^2).

Prove to be optimal:

In this case, the problem happens on the point that when there are conflict of new job are trying to replace the job on the timeline at a certain time unit, after that the replaced job will smaller than or equal to the job that already on the timeline and have the same time unit with the new job. If we swapped any job which is on the timeline with the new job then at this moment, the total profit will smaller than the original. So that we can assure that the final result is consist of the largest priorities.