

Q4

You are given a set of n jobs where each job i has a deadline $d_i \geq 1$ and profit $p_i > 0$. Only one job can be scheduled at a time. Each job takes 1 unit of time to complete. We earn the profit if and only if the job is completed by its deadline. The task is to find the subset of jobs that maximises profit. Your algorithm should run in time $O(n^2)$

Answer:

Greedy Strategy :

- Firstly, sort the jobs in decreasing order of their profit using merge sort which in the worst case will take time $O(n \log n)$.
- First loop : Now we will need to schedule jobs into an array to ensure we get the maximum profit. To do this will iterate through the sorted array for all values i between 1 to n and schedule job $i \rightarrow$ this will be $O(n)$
 - Second loop : To schedule job i we will start at the deadline k for that specific job and iterate backwards.
 - If there is already a job scheduled at k then we will go down to $(k-1)$
 - We will keep going down until we get to the start of the array or an empty spot.
 - The complexity to find the latest possible free slot meeting a jobs deadline will be $O(n)$
- To iterate through both loops the complexity will be $O(n^2)$
 - Thus the total complexity will be $O(n \log n) + O(n^2) = O(n^2)$. Hence we can find the subset which maximises profit in $O(n^2)$