**Question 4**

First, we order the jobs based off its profit in descending order.

So,

We earn this profit , given that its deadline is less than the current time, . Each job takes one unit of to complete. We must find the subset of jobs that maximises profit.

The solution to this problem is to use a **greedy algorithm**.

So, our approach is that we schedule each job, in order (starting from ) at the latest possible timeslot available (or the next best), given the condition that

If such a timeslot is unavailable, then we do not schedule the job.

For example, if we have 4 jobs, already sorted with profits of $20, $15, $12 and $10, with deadlines t = 4, 2, 2 and 1. Each job takes 1 unit of time.

We schedule the slots as shown.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **T** | 1 | 2 | 3 | 4 |
| **P** | $12 | $15 | n/a | $20 |

We can see here, that despite the $12 having a deadline of t = 2, we placed it into the next best position in order to attain its profit.

Unfortunately, there is no space for the job that provides a profit of $10 as there is not a suitable slot that fits its deadline. Nevertheless, this is the most optimal solution.

Since we are placing the jobs at its latest possible deadline, we iterate through in and each job assignment is similarly in , so the algorithm is run in time.