Matthew Soicher
z5160737

Question 4

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

So,

$$p_1 \geq \; p_2 \geq p_3 \geq \cdots \geq p_n$$

We earn this profit p_i, given that its deadline $d_i \geq 1$ is less than the current time, t. Each job takes one unit of t 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, p_i in order (starting from p_1) at the latest possible timeslot available (or the next best), given the condition that $d_i \leq t$.

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 $O(n)$ and each job assignment is similarly in $O(n)$, so the algorithm is run in $O(n^2)$ time.