HOMEWORK 3 – Q4

MINGLANG XIE

z5228006

4. 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 $O(n^2)$.

Solution:

Ignore the deadline and sort the jobs in decreasing order of profit. We need to find out the latest deadline in jobs, and create a time slot, which have length equal to the latest deadline in jobs (e.g. the latest deadline is 8, then the time slot has length 8). We now go through the sorted jobs, and for each job we want to find a slot that can fit that job's deadline, which means take a job and put it in the place of time slot where as late as possible but before its deadline.

For example,

	Profit	Deadline
job_1	200	5
job_2	300	1
job_3	900	5

We first sort it by profit,

	Profit	Deadline
job_3	900	5
job_2	300	1
job_1	200	5

We have a time slot:

\	\	\	\	\

Let insert first job,

\	\	\	\	job_3

And second job

job_2	\	\	\	job_3

Finally third job which also have deadline 5, so we insert it as late as possible

job_2	\	\	job_1	job_3

Therefore, [job_2, job_1, job_3] is the subset of jobs that maximises profit.

This strategy is clearly optimal, as we begin with the set of all jobs that sorted in decreasing order of profit, and always choose a job has higher profit until the time slot is full.

Time complexity, sorting the jobs in decreasing order of profit take at most $O(nlog(n))$. Jobs run in linear time, and checking if there is a slot for the job, take at most n time, so the time complexity is $O(nlog(n) + n^2) = O(n^2)$.

```
Merge-sort(jobs) by profit
max-length = max-deadline(jobs)
T = [-1] * max-length

for i in jobs:
    k = i
    while k >0:
        if T[k-1] == -1:
            T[k-1] = i
            break
        else:
            k -= 1

    if k < 0:
        break
```