

z5242692

Chenqu Zhao

COMP9101 (T2-2020)

Homework 3 – Q4

Write job in (d, p) format, where d is its deadline and p is its profit. Put all jobs in a list J and use merge sort to sort them in decreasing order of profit p_i . Notice that if there are several jobs with the same profit, sort them in decreasing order of deadline d_i . The cost of merge sort is $O(n \log n)$. Initialise time T as n and the candidates' subset S as an empty list.

Go through the list J from the start. Find the first job i satisfies that $d_i \geq T$ as a candidate. If there exists such candidate i , remove it from J and append it to S . Deduct T by one.

Do the above operations iteratively until $T == 0$. Currently, the **reverse** of the result of list S is the subset of jobs we want.

There are n iterations corresponding to n time, in each iteration, the search of candidates cost up to n lookups. So, it costs $O(n^2)$ in this stage.

In total, this algorithm costs $O(n \log n) + O(n^2) = O(n^2)$.