Partioning Intervals

There is a set of n jobs $R = \{1, 2, ..., n\}$, i^{th} job have starting time of s_i and finishing time of f_i. We want to find a minimum d such that we can partition R into d subsets of $R_1, R_2, ..., R_d$ in a way that no two jobs in a subset will overlap.

1. Give an efficient greedy algorithm that finds an optimal solution.
2. Explain why your algorithm returns an optimal solution.
3. Is the solution returned by the algorithm the only possible solution for all possible inputs? Explain your answer.
4. Prove that your algorithm returns an/the optimal solution.
5. Prove a tight asymptotic bound on the running time of your algorithm.

Scheduling to Minimize Lateness (4.2 of Textbook)

Consider a situation in which we have a single resource and a set of n requests to use the resource for an interval of time. Assume that the resource is available starting at time s. However, each request is flexible. Instead of a start time and finish time, the request i has a deadline d_i , and it requires a contiguous time interval of length t_i, but it is willing to be scheduled at any time before the deadline. Each accepted request must be assigned an interval of time of length t_i, and different requests must be assigned nonoverlapping intervals.

We say that a request i is late if it misses the deadline, that is, if $f(i) > d_i$. The lateness of such a request i is defined to be $l_i = f(i)d_i$. We will say that $l_i = 0$ if request i is not late. The goal in our new optimization problem will be to schedule all requests, using nonoverlapping intervals, so as to minimize the maximum lateness, $L = max_i\ l_i$.

1. Give a counter example for greedy approach which chooses a job with minimum t_i.
2. Give a counter example for greedy approach which chooses a job with minimum $d_i - t_i$.
3. Give an efficient greedy algorithm that finds an optimal solution.
4. Explain why your algorithm returns an optimal solution.
5. Is the solution returned by the algorithm the only possible solution for all possible inputs? Explain your answer.
6. Prove that your algorithm returns an/the optimal solution.
7. Prove a tight asymptotic bound on the running time of your algorithm.

Products in Fridge

We bought n products from grocery store, i_{th} product can only be in a fridge if the temprature of that fridge is between $c_i \ \ to \ \ h_i$. Each fridge can only be in a fixed temprature. How many fridge we need to buy in order to be able to keep all n products in fridges.

1. Give an efficient greedy algorithm that finds an optimal solution.
2. Explain why your algorithm returns an optimal solution.
3. Is the solution returned by the algorithm the only possible solution for all possible inputs? Explain your answer.
4. Prove that your algorithm returns an/the optimal solution.
5. Prove a tight asymptotic bound on the running time of your algorithm.