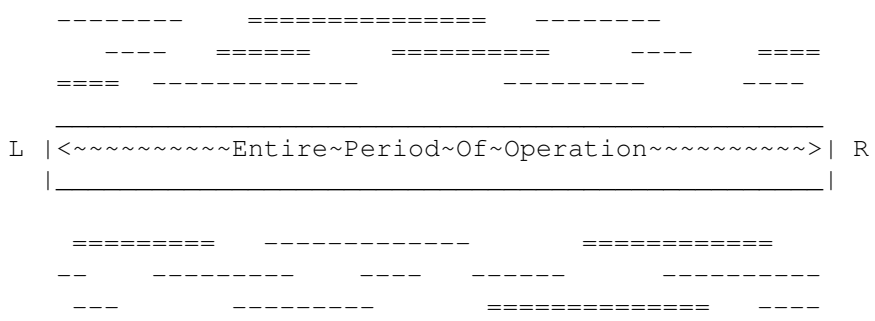
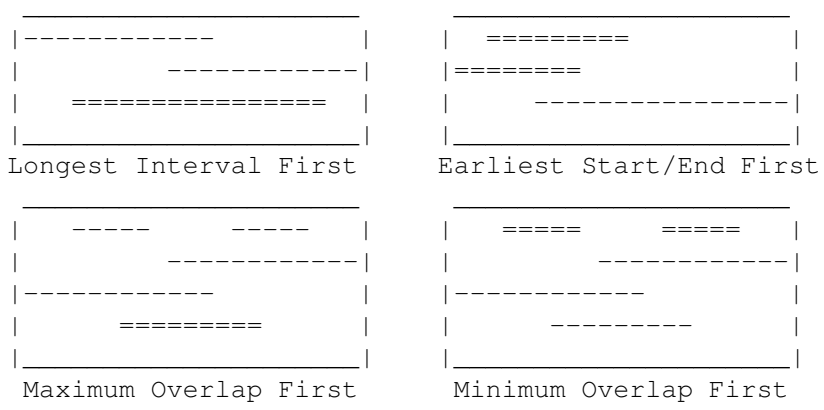
CS21003: Algorithms-I (Theory)
Tutorial – 3 (Greedy Algorithms)
Date: 06-February-2020

1. A set of n UPS (Uninterrupted Power Supply) systems serves as the backup-power devices for the entire period of opertion of a super-computing infrastructure that is planned to be installed at IIT Kharagpur. Each such UPS system, u_i $(1 \leq i \leq n)$, can serve power within a fixed (pre-set) interval period, $[a_i, b_i]$ $(a_i < b_i)$, activated automatically when main-power fails. Upon further inspection, the project manager finds that there are too many of these UPS systems being unnecessarily deployed. So, he plans to remove as many UPS systems as possible without sacrificing the requirement that – during the entire period of the operation, at least one UPS must be there (stand-by) to give support as a backup device (in case of a power failure). The following figure provides an example of the situation, the thick lines (=) indicate the necessary UPS systems and the thin lines (−) are to be removed.



The overall period of supercomputing operation stretches from $L = \texttt{min}_i(a_i)$ to $R = \texttt{max}_i(b_i)$. Your task is to find a minimum set of intervals whose union is the entire interval $[L, R]$.

(i) Propose an efficient algorithm to solve this problem. [*Hint*: Some (greedy) strategies that does not work are:



]

(ii) Prove that your algorithm is optimal, that is, it correctly computes the minimum set of UPS systems required.

(iii) Deduce a good (tight) bound on the running time of your algorithm in the Big-O notation.

2. Suppose the Prof-in-charge, responsible to build the course time-table in this semester, wants to allocate a total n courses in optimal number of rooms in Nalanda Complex. She has been given with the start time s_i and the finish time f_i for every course c_i $(1 \leq i \leq n)$. Propose an efficient algorithm to produce the allocation of all courses in respective rooms, so that the total number of rooms required are minimized. What is the running time of your proposed algorithm? Can you prove that your algorithm always finds the optimal solution?

3. Prof. Hijibijbij wants to drive his car through the entire costal belt of India. He can drive N kilometers with a full-tank of fuel in his car. He follows a travel-map which also indicates him the distances (in kilometers) beween two fuel-pumps in his entire travel route. Prof. Hijibijbij wants to make as few stops as possible for re-fueling his car in the entire route. Give an efficient algorithm by which Professor can determine at which fuel-pumps he should stop, so that his number of stops get minimized. What is the running time of your algorithm? Prove that your algorithm also guarantees optimality in this scenario.