

Homework 5: Greedy Algorithms

You may work in groups, but you must write solutions yourself. List your collaborators at the top of your submission.

Instructions for submission: Submit your homework as a single PDF file to the course Gradescope page. Either scan your *legible* handwritten solutions or type them up in \LaTeX .

Setting: Suppose you are consulting for a high-performance computing center. They have a single supercomputer that can run one job at a time and have n jobs to run. Each job i has a running time of $t_i > 0$ and a strict deadline d_i .

Problem 1. The center's management suggests a greedy strategy to maximize the total number of jobs completed by their deadlines: "Sort the jobs by their processing time t_i in ascending order, and schedule them one by one as long as they finish before their deadline." Provide a small counterexample showing that this greedy strategy does not always yield the maximum number of completed jobs.

Problem 2. Now, suppose the objective changes. All jobs must be processed, and job i has a weight $w_i > 0$ representing its priority. The goal is to minimize the total weighted completion time, defined as $\sum_{i=1}^n w_i C_i$, where C_i is the completion time of job i . Prove that scheduling jobs in order of decreasing w_i/t_i ratio always yields an optimal solution. **Hint:** Use an exchange argument to show that any schedule that does not follow this order can be improved by swapping two jobs.

Problem 3. The computing center is finalizing the network design for its seven primary racks: Gateway (A), Storage (B), Compute (C), Backup (D), Security (E), Monitoring (F), and Management (G). The cost of connecting each pair of racks is given in the table below.

Rack Pair	Connection Cost
A-B	6
A-C	3
A-G	7
B-C	2
B-D	5
B-G	4
C-D	4
C-E	5
D-E	2
D-F	3
E-F	6

- (a) What is the output of running Dijkstra's algorithm starting from rack A? You should draw/annotate the graph with the order in which nodes are visited and the shortest path distances from A to each node at the end of the algorithm.
- (b) What is the output of Prim's algorithm starting from rack A? Show the final spanning tree edges selected by the algorithm and the order in which they were added.
- (c) What is the output of Kruskal's algorithm starting from rack A? Show the final spanning tree edges selected by the algorithm and the order in which they were considered.
- (d) Notice that Prim's and Kruskal's algorithms may yield different spanning trees. What is the total cost of the spanning tree produced by Prim's algorithm? What about Kruskal's algorithm? Are they the same or different? Explain why this happens.