PSO 2 — Week 6: Problems

CS381, Fall 2026

Problem 1. Scheduling with weights

You are given n jobs; job i occupies the closed interval $[s_i, f_i]$ and carries a weight $w_i > 0$. A *schedule* is a set of pairwise disjoint jobs, and the goal is one of maximum total weight.

(a) When all weights are 1, the greedy that repeatedly takes the compatible job with the smallest finish time is optimal. Show that with weights it is unboundedly bad: for every $c > 1$ give an instance on which it returns less than $1/c$ of the optimum. Do the same for the greedy that repeatedly takes the heaviest compatible job. Then say which step of the unweighted optimality proof stops working for the first greedy, and why the obvious exchange argument never gets started for the second.

(b) Give an $O(n \log n)$ algorithm. State your convention for two jobs that touch at a single coordinate and say which line of the algorithm encodes it (compare Week 4, Problem 1(a)); prove correctness; and show how to output an optimal schedule, not just its weight.

(c) *(harder)* Now you may select at most k jobs. Give an algorithm running in $O(n \log n + nk)$ time and prove it correct. Then rule out the natural shortcut — run (b) and discard the lightest selected jobs until k remain — with a concrete instance for $k = 1$, and say what that shortcut is assuming about how the optima for different budgets relate to one another.