Greedy Analysis Strategies

Greedy algorithm stays ahead. Show that after each step of the greedy algorithm, its solution is at least as good as any other algorithm's.

Exchange argument. Gradually transform any solution to the one found by the greedy algorithm without hurting its quality.

Structural. Discover a simple "structural" bound asserting that every possible solution must have a certain value. Then show that your algorithm always achieves this bound.

Q. Which strategy did we use for the problems in this lecture (interval scheduling, interval partitioning, minimizing lateness) ?

Greedy Algorithms

Q. Fill in the following table:

Problem name	Problem description	Greedy algorithm	Idea of proof	Run-time
Interval scheduling				
Interval partitioning				
Minimize lateness				

Greedy Algorithms

Problem name	Problem description	Greedy algorithm	Idea of proof	Run-time
Interval scheduling	Choose as many non-overlapping intervals as possible.	Earliest finishing time first	Greedy algorithm stays ahead (induction)	O(n log n)
Interval partitioning				
Minimize lateness				

Greedy Algorithms

Problem name	Problem description	Greedy algorithm	Idea of proof	Run-time
Interval scheduling	Choose as many non-overlapping intervals as possible.	Earliest finishing time first	Greedy algorithm stays ahead (induction)	O(n log n)
Interval partitioning	Divide intervals over least number of resources.	Earliest starting time first	Structural bound	O(n log n)
Minimize lateness				

Greedy Algorithms

Problem name	Problem description	Greedy algorithm	Idea of proof	Run-time
Interval scheduling	Choose as many non-overlapping intervals as possible.	Earliest finishing time first	Greedy algorithm stays ahead (induction)	O(n log n)
Interval partitioning	Divide intervals over least number of resources.	Earliest starting time first	Structural bound	O(n log n)
Minimize lateness	Schedule to minimize the lateness regarding the deadline.	Earliest deadline first	Exchange argument (contradiction)	O(n log n)

Greedy Algorithms

Problem name	Problem description	Greedy algorithm	Idea of proof	Run-time
Interval scheduling	Choose as many non-overlapping intervals as possible.	Earliest finishing time first	Greedy algorithm stays ahead (induction)	O(n log n)
Interval partitioning	Divide intervals over least number of resources.	Earliest starting time first	Structural bound	O(n log n)
Minimize lateness	Schedule to minimize the lateness regarding the deadline.	Earliest deadline first	Exchange argument (contradiction)	O(n log n)
Shortest path	Find the shortest path from s to t.			