CS 141: Intermediate Data Structures and Algorithms

Discussion - Week 8, Winter 2018
Greedy Algorithms

- General idea
- Examples
- Group activity
- Review assignment #4
General idea

- Like dynamic programming, used to solve optimization problems.
- Problems exhibit optimal substructure (like DP).
- Problems also exhibit the *greedy-choice* property.
  - When we have a choice to make, make the one that looks best at the moment.
  - Make a *locally optimal choice* in hope of getting a *globally optimal solution*. 
Greedy vs Dynamic Programming

What are their differences?
Greedy algorithm’s elements

- **Greedy choice property**: a globally optimal solution can be reached by making a locally optimal choice.
- **Optimal substructure**: optimal solution to the problem consists of optimal solutions to sub-problems.
- **Notice**: you have to show both elements above in every algorithm in your assignment solution.
Example: Activity selection problem
Example: Activity selection problem

- **Greedy choice property:** if \( a_m = [s_m, f_m] \) has the earliest finish time \( f_m \), it must be included in some optimal solution.
- **Optimal substructure:** if \( A \) is optimal to \( S \), then \( A' = A - \{a_m\} \) is optimal to \( S' = \{i \in S : s_i \geq f_m\} \).
Group activity

Given a set $T$ of 11 tasks with \{start, finish\} are:

$[1, 3], [2, 5], [2, 5], [2, 5], [4, 7], [6, 9], [8, 11], [10, 13], [10, 13], [10, 13], [12, 14]$ 

- Apply the algorithm in Problem 4, assignment #4 with this input.
- Discuss the result.
Group activity
Review assignment #4