V. Greedy Algorithms
Greedy algorithms – Overview

- Algorithms for solving (optimization) problems typically go through a sequence of steps, with a set of choices at each step.

- A greedy algorithm always makes the choice that looks best at the moment, without regard for future consequence, i.e., “take what you can get now” strategy.

- Greedy algorithms do not always yield optimal solutions,

  Local optimum $\Rightarrow$ Global optimum

- but for many problems they do.
An activity-selection problem

Problem statement:

*Input:* Set $S = \{1, 2, \ldots, n\}$ of $n$ activities

- $s_i = $ start time of activity $i$
- $f_i = $ finish time of activity $i$

*Output:* Maximum-size subset $A \subseteq S$ of compatible activities

Notes:

- Activities $i$ and $j$ are compatible if the intervals $[s_i, f_i)$ and $[s_j, f_j)$ do not overlap.
- Without loss of generality, assume

$$f_1 \leq f_2 \leq \cdots \leq f_n$$
An Activity-selection problem

Greedy algorithm:
- *pick the compatible activity with the earliest finish time.*

Why?
- Intuitively, this choice leaves as much opportunity as possible for the remaining activities to be scheduled
- That is, the greedy choice is the one that maximizes the amount of *unscheduled time* remaining.
An Activity-selection problem

GreedyActivitySelector(s,f)
n = length(s)
A = {1}
j = 1
for i = 2 to n
  if s[i] >= f[j]
    A = A U {i}
    j = i
  end if
end for
return A

Remarks
- Assume the array f already sorted
- Complexity: \( T(n) = O(n) \)
An Activity-selection problem

Question: Does Greedy-Activity-Selector work?
Answer: Yes!

Why? The proof of the greedy algorithm producing the solution of maximum size of compatible activities is based on the following two key properties:

- **The greedy-choice property**
  a globally optimal solution can be arrived at by making a locally optimal (greedy) choice.

- **The optimal substructure property**
  an optimal solution to the problem contains within it optimal solution to subprograms.

Specifically, for the Greedy-Activity-Selectior, these two properties are phased as follows.
An Activity-selection problem

The greedy-choice property:

There exists an optimal solution $A$ such that the greedy choice “1” in $A$.

The proof goes as follows:

- Let’s order the activities in $A$ by finish time such that the first activity in $A$ is “$k_1$”.
- If $k_1 = 1$, then $A$ begins with a greedy choice.
- If $k_1 \neq 1$, then let $A' = (A - \{k_1\}) \cup \{1\}$.
  Then
  1. the sets $A - \{k_1\}$ and $\{1\}$ are disjoint
  2. the activities in $A'$ are compatible
  3. $A'$ is also optimal, since $|A'| = |A|$

Therefore, we conclude that there always exists an optimal solution that begins with a greedy choice.
An Activity-selection problem

**The optimal substructure property:**

*If $A$ is an optimal solution, then $A' = A - \{1\}$ is an optimal solution to $S' = \{i \in S, s[i] \geq f[1]\}$.*

Proof: By contradiction. If there exists $B'$ to $S'$ such that $|B'| > |A'|$, then let

$$B = B' \cup \{1\},$$

we have

$$|B| > |A|,$$

which is contradicting to the optimality of $A$. 
An Activity-selection problem

In summary,

- After each greedy choice is made, we are left with an optimization problem of the same form as the original.

- *By induction* on the number of choices made, making the greedy choice at every step proceduces an optimal solution.

Therefore, *the greedy activity selector works!*