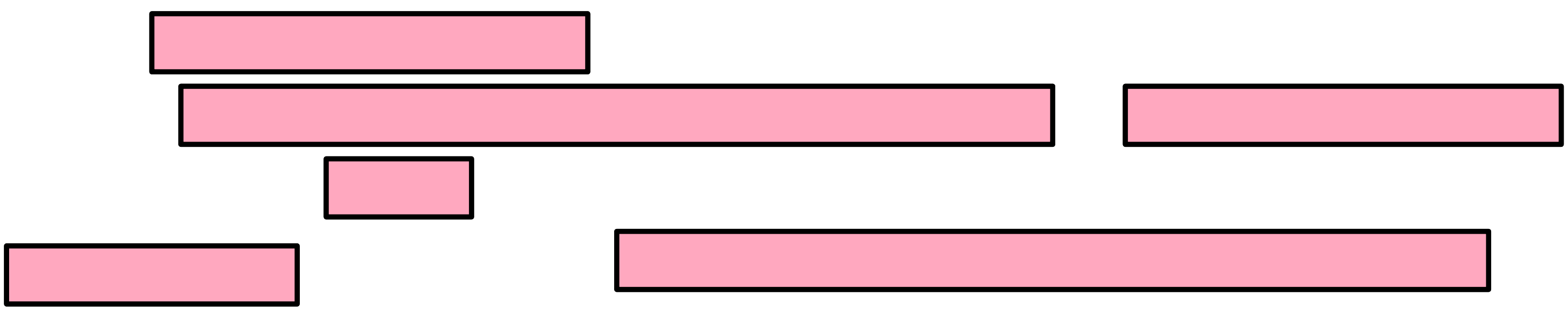
Greedy Algorithms

Consider the *Activity-selection problem* from the book:

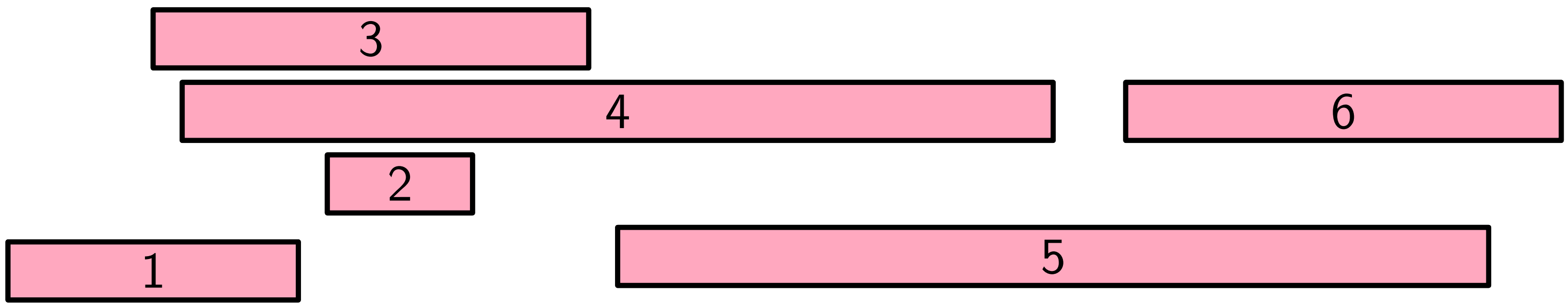
- Given set S of n activities.
  - ▷ $s_i =$ start time of activity i.
  - ▷ $f_i =$ finish time of activity i.

- Find max-size subset A of compatible activities.



Activity-selection problem (cont'd)

- Assume $f_1 \leq f_2 \leq \cdots \leq f_n$.



- Greedy approach: select the first activity to finish that is compatible with all previously selected activities $\Rightarrow \{1, 2, 5\}$.

- The greedy approach is optimal because the problem has
  1. Greedy-choice property.
  2. Optimal substructure.

Greedy choice-property

- Let $A \subseteq S$ be an optimal solution.
- If $f_1 \leq f_2 \leq \cdots$ then activity 1 is part of A.
- If the first activity in A is $k \neq 1$ (not greedy), then there is another optimal solution B that begins with 1.
- $B = A - \{k\} \cup \{1\}$.
- Because $f_1 \leq f_k$, activity 1 is still compatible with A.
- $|B| = |A|$, so B is also optimal.

Optimal substructure

- Once we make the first greedy choice the problem is

$$S' = \{i \in S : s_i \geq f_1\}$$

  with optimal solution $A' = A - \{1\}$.

- Solution A' must be optimal.

- If B' solves S' with more activities than A', adding activity 1 to B' makes it bigger than A which is a contradiction.

- Therefore, a greedy choice yields an optimal solution.