Greedy Algorithms

謝仁偉 副教授
jenwei@mail.ntust.edu.tw
國立台灣科技大學 資訊工程系
2017 Spring

Greedy Algorithms

- For many optimization problems, using dynamic programming to determine the best choices is overkill; simpler, more efficient algorithms will do.
- A greedy algorithm always makes the choice that looks best at the moment.
  - It makes a locally optimal choice in the hope that this choice will lead to a globally optimal solution.
  - Greedy algorithms do not always yield optimal solutions, but for many problems they do.

Outline

- ***An Activity-Selection Problem***
- Elements of the Greedy Strategy
- Huffman Codes

Activity-Selection Problem (1/2)

- Suppose we have a set $S = \{a_1, a_2, \ldots, a_n\}$ of n proposed activities that wish to use a resource, such as a lecture hall, which can serve only one activity at a time.
- Each activity a_i has a start time s_i and a finish time f_i, where $0 \le s_i < f_i < \infty$.
  - If selected, activity a_i takes place during the half-open time interval $[s_i, f_i)$.
- Activities a_i and a_j are compatible if the intervals $[s_i, f_i)$ and $[s_j, f_j)$ do not overlap.
  - That is, a_i and a_j are compatible if $s_i \ge f_j$ or $s_j \ge f_i$.

Activity-Selection Problem (2/2)

- In the activity-selection problem, we wish to select a maximum-size subset of mutually compatible activities.
- We assume that the activities are sorted in monotonically increasing order of finish time: $f_1 \le f_2 \le f_3 \le \ldots \le f_{n-1} \le f_n$.

Example

i	1	2	3	4	5	6	7	8	9	10	11
s_i	1	3	0	5	3	5	6	8	8	2	12
f_i	4	5	6	7	9	9	10	11	12	14	16

- The subset $\{a_3, a_9, a_{11}\}$ consists of mutually compatible activities.
- Could you find a largest subset of mutually compatible activities?

➡ We will first consider a dynamic-programming approach and then show that we can always make greedy choices to arrive at an optimal solution.

The Optimal Substructure of the Activity-Selection Problem (1/3)

- Let us denote by $\boldsymbol{S}_{ij}$ the set of activities that start after activity a_i finishes and that finish before activity a_j starts.
- Suppose that we wish to find a maximum set of mutually compatible activities in $\boldsymbol{S}_{ij}$, and suppose further that such a maximum set is $\boldsymbol{A}_{ij}$, which includes some activity a_k.
- By including a_k in an optimal solution, we are left with two subproblems: finding mutually compatible activities in the set $\boldsymbol{S}_{ik}$ and finding mutually compatible activities in the set $\boldsymbol{S}_{kj}$.

The Optimal Substructure of the Activity-Selection Problem (2/3)

- Let $\boldsymbol{A}_{ik} = \boldsymbol{A}_{ij} \cap \boldsymbol{S}_{ik}$ and $\boldsymbol{A}_{kj} = \boldsymbol{A}_{ij} \cap \boldsymbol{S}_{kj}$, so that $\boldsymbol{A}_{ik}$ contains the activities in $\boldsymbol{A}_{ij}$ that finish before a_k starts and $\boldsymbol{A}_{kj}$ contains the activities in $\boldsymbol{A}_{ij}$ that start after a_k finishes.

➡ We have $A_{ij} = A_{ik} \cup \{a_k\} \cup A_{kj}$

➡ The maximum-size set $\boldsymbol{A}_{ij}$ of mutually compatible activities in $\boldsymbol{S}_{ij}$ consists of $|\boldsymbol{A}_{ij}| = |\boldsymbol{A}_{ik}| + |\boldsymbol{A}_{kj}| + 1$ activities.