CS F364
Design & Analysis of Algorithms

ALGORITHM DESIGN TECHNIQUES - GREEDY

Greedy Algorithms

- Greedy Choice and Optimal Substructure
- Limitation of Greedy Choice - Example

2/23/2015 Sundar B. CSIS, BITS, Pilani

Optimal Substructure

- A problem exhibits optimal substructure if
  - an optimal solution to the problem contains optimal solutions for sub-problems:
    - can be decomposed into optimal solutions for sub-problems.
- Optimal substructure is necessary for greedy choice:
  - Otherwise local choice may not lead to global optimality
    - i.e. the choice may not preserve optimality
  - Examples:
    - Schedule for tasks T' = T – { j } is part of the schedule for tasks T where j is the earliest starting task
    - Schedule for tasks T' = T – { k } is not necessarily part of the schedule for tasks T if k is not the earliest starting task

Greedy Choice, Optimal Substructure, and Induction

- Design of Greedy Algorithms can be viewed as a special case of Divide-And-Conquer:
  - where the problem of size N is divided into
    i. a sub-problem of size 1 (or size k for some constant k)

    and

    ii. a sub-problem of size N-1 (or N-k as the case ma y be)
      - where the latter is the same as the original problem.
- Greedy Choice
  - refers to making the right choice – locally – for sub-problem (i).
- Optimal substructure property is necessary to ensure that
  - optimal solution to sub-problem (ii) can be used as is if sub-problem (ii) is part of the solution to the problem.

Greedy Choice - Limitation

- Greedy Choice does not always hold when Optimal Substructure Property holds.
- e.g. Consider the 0/1 KnapSack problem:
  - Optimal Substructure holds for 0/1 KnapSack:
    - Consider the most valuable subset of items with weight at most W
    - If we remove item j from this subset, the remaining subset must be the most valuable weighing at most $W - w_j$

Greedy Choice - Limitation

- Greedy Choice Property does not hold for 0/1 KnapSack:
  - Suppose we use *per unit profit* as the greedy choice
  - Consider three items:
    1. (10kg, Rs. 5,000). Value = Rs. 500 / kg
    2. (20kg, Rs. 8,000) Value = Rs. 400 / kg
    3. (40kg, Rs. 9,000) Value = Rs. 225 / kg
  - Let W = 60kg
    - Any solution with item 1 is not optimal!
      - i.e. ordering by unit weight is not useful
- Exercise:
  - Generalize this argument (about remaining capacity) for any input.
  - Similarly argue that other greedy choices (profit or weight) are also not useful.