Design and Analysis of Algorithm (DAA)

Greedy Algorithms

[Module 2]

Dr. Dayal Kumar Behera

School of Computer Engineering
KIIT Deemed to be University, Bhubaneswar, India

Optimization Problems

- For each instance there are (possibly) multiple valid solutions
- Goal is to find an optimal solution

  - Minimization problem:

    associate cost to every solution, find min-cost solution

  - Maximization problem:

    associate profit to every solution, find max-profit solution

Techniques for Optimization

- Solving Optimization problems typically involve making choices and various techniques are
  - Brute-Force: Systematically checking all possible candidates
  - Greedy Algorithms
  - Dynamic Programming
  - Backtracking
  - Branch and Bound

Multistage Optimization

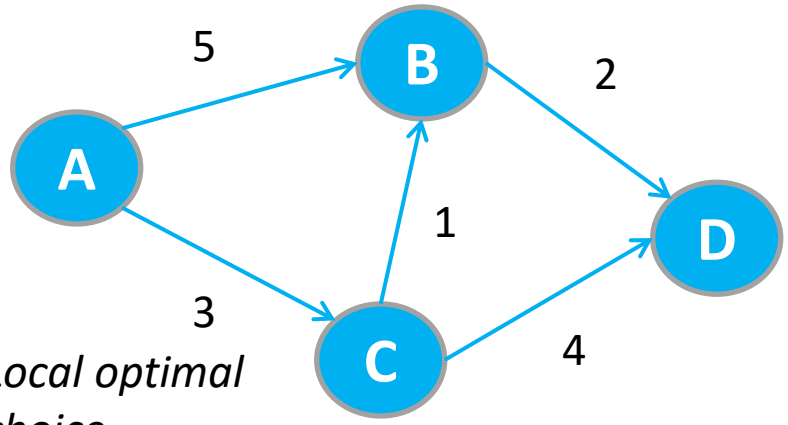
- Each Optimization Problem has an objective function and a set of constraints/ restrictions.
  - Any solution that satisfies those constraints is called a **feasible solution**
  - We need to find a feasible solution that either **maximizes** or **minimizes** a given **objective function**
  - A feasible solution that does this is called an **optimal solution**
- In multistage optimization problem, decisions are taken at multiple stages to obtain a global solution.
- In each stage, optimal choices have to made in order to achieve the optimal global solution.

Greedy Algorithms

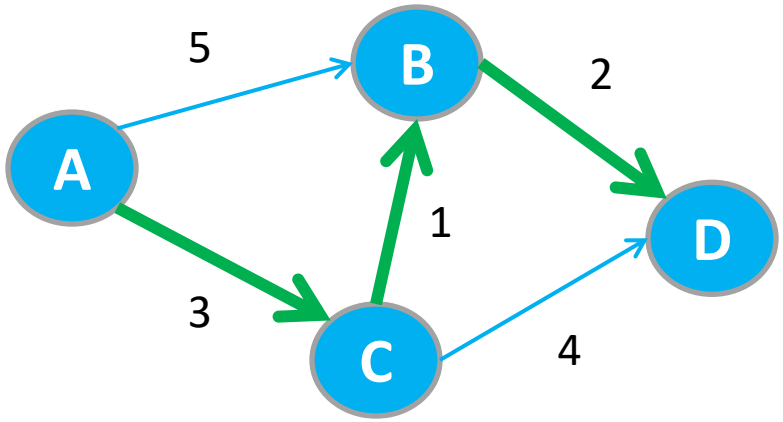
- A greedy algorithms obtain an optimal solution to a problem by making a sequence of choices
- At each decision point algorithm selects the choice that seems best at that moment.
- Make a locally optimal choice (greedy choice) in hope of getting a global optimal solution.
- Greedy algorithms do not always yield an optimal solution, but in many cases they work.

Example

Shortest path from A to D?



Local optimal choice



Global optimal solution: A→ C→ B→ D

Elements of the greedy strategy

Greedy choice property: A global optimal solution can be achieved by making a local optimal (best) choice

- (If you make a choice that seems the best at the moment and solve the remaining sub-problems later, you still reach an optimal solution. You will never have to reconsider your earlier choices.)

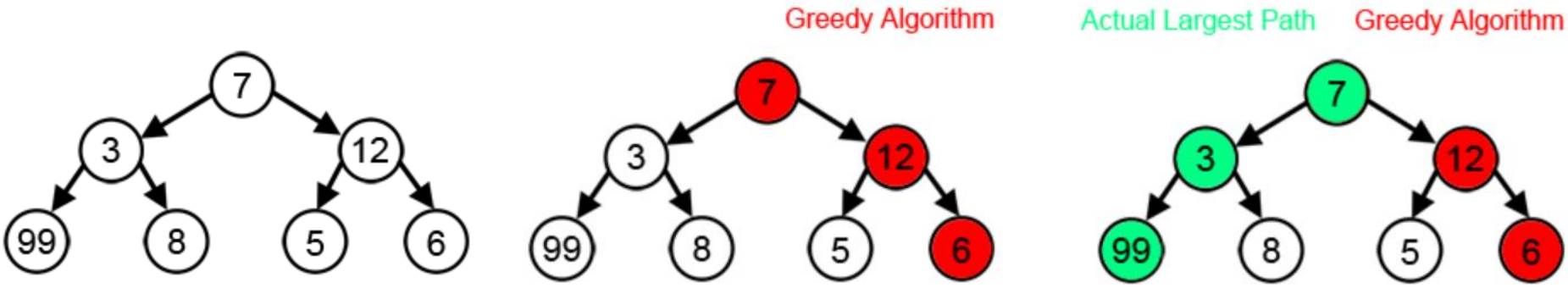
Optimal substructure: A problem has an optimal substructure if an optimal solution to the entire problem contains the optimal solutions to the sub-problems.

When To Apply Greedy Methods?

- Apply **Greedy Methods** when
  - Problems exhibit **optimal substructure**
  - Problems also exhibit the **greedy-choice property**
    - When we have a choice to make, make the one that looks best right now
    - Make a locally optimal choice in hope of getting a globally optimal solution
- The greedy approach is also used in the context of hard (difficult to solve) problems in order to generate an approximate solution

Limitations of Greedy

Problem statement: find the path with the largest sum from root to leaf.



With a goal of reaching the largest sum, at each step, the greedy algorithm will choose what appears to be the optimal immediate choice, so it will choose 12 instead of 3 at the second step and will not reach the best solution, which contains 99.

Greedy method may not always lead to the globally optimal solution.
Analyzing the correctness and efficiency of greedy algorithms can be challenging.


Each of your
actions will
have an
impact on your
future.


Once you know
who is walking
with you on your path.
you will never
be afraid.

Thank you