

Greedy Approach

classmate

Date _____

Page _____

Basic Idea

- Construct a solution to an optimization problem piece by piece through a sequence of choice that are:
 - feasible
 - locally optimal
 - Irrevocable
- for some problems, they yield an optimal solution
- for most, they don't but can be useful for fast approximations.

Greedy method

- Most straight forward design technique
- Progress in top-down fashion.
- It always make the choice that looks best at the moment, means it make a locally optimal choice in the hope that this choice will lead to a globally optimal solution.
- Solution is constructed through a sequence of steps, each expanding a partially constructed solution obtained so far, until a complete solution to the problem is reached.
- Greedy methods provides optimal solutions.

(AJAY RAWAT)

Feasible and optimal solutions

- Most of the problems have 'n' inputs and require us to obtain a subset that satisfies some constraints.
- Any subset that satisfies these constraints is called a feasible solution.
- Greedy method finds a feasible solution that either maximizes or minimizes a given objective function.
- A feasible solution which meets the objective function is called optimal solution.

Greedy Steps

- 1) Select some solution from input domain.
- 2) Then check the selected solution is feasible or not.
- 3) Select the optimal solution (solution that satisfy or nearly satisfy the objective function from set of feasible solution)

* Greedy method works in stages. At each stage only one input is considered at each time. Based on this input it is decided whether particular input gives the optimal solution or not.

Algorithm

Greedy (D, n)

// In greedy approach D is a domain from which Solution
// is to be obtained of size n

Solution \leftarrow 0

for $i \leftarrow 1$ to n do {

$S \leftarrow \text{Select}(D)$ // Selection of Solution from D

 if (feasible (Solution, S)) then

 Solution \leftarrow Union (Solution, S)

}

Return Solution.

Two Key Ingredients

The following are the two key properties (ingredients) to prove the greedy algorithms are optimal

1. Greedy choice property,
2. Optimal Substructure.

* If we can demonstrate that the problem has these properties then we can develop Greedy Algorithm for it.

Greedy choice property

- Global optimal Solution will be made with local optimal (Greedy) choices
- In a greedy algo, the best choice will be Selected at the moment to solve the subproblem that remains.
- The choice of the Greedy may depend on the choice so far, but it can't depend on the future choices or on the solutions to the subproblems.

Optimal Substructure

- optimal Substructure is a property of Greedy Solution.
- A problem exhibits optimal Substructure if an optimal Solution to the problem contains within it optimal Solutions to the Subproblems.

Procedure to prove the optimal Greedy Algorithm

1. Consider globally-optimal Solution.
2. Show greedy choice at first step reduce problem to the same but smaller problem. Greedy choice must be part of an optimal Solution and made first.
3. Use induction to show greedy choice is best at each step i.e. optimal Substructure.

Applications

1. Optimal Solutions
 - Minimum Spanning Tree (MST)
 - Single Source shortest Path
 - Scheduling problems.
 - Huffman Codes
2. Approximations
 - Traveling Salesman Problem (TSP)
 - Knapsack problem
 - Other Combinatorial optimization problems.