UNIT IV ALGORITHM DESIGN TECHNIQUES

Dynamic Programming: Matrix-Chain Multiplication – Elements of Dynamic Programming – Longest Common Subsequence- Greedy Algorithms: – Elements of the Greedy Strategy- An Activity-Selection Problem - Huffman Coding.

GREEDY ALGORITHMS

The greedy method is one of the strategies like Divide and conquer used to solve the problems. This method is used for solving optimization problems. An optimization problem is a problem that demands either maximum or minimum results. Let's understand through some terms.

The Greedy method is the simplest and straightforward approach. It is not an algorithm, but it is a technique. The main function of this approach is that the decision is taken on the basis of the currently available information. Whatever the current information is present, the decision is made without worrying about the effect of the current decision in future.

This technique is basically used to determine the feasible solution that may or may not be optimal. The feasible solution is a subset that satisfies the given criteria. The optimal solution is the solution which is the best and the most favorable solution in the subset. In the case of feasible, if more than one solution satisfies the given criteria then those solutions will be considered as feasible, whereas the optimal solution is the best solution among all the solutions.

The general structure of a greedy algorithm can be summarized in the following steps:

1. Identify the problem as an optimization problem where we need to find the best solution among a set of possible solutions.
2. Determine the set of feasible solutions for the problem.
3. Identify the optimal substructure of the problem, meaning that the optimal solution to the problem can be constructed from the optimal solutions of its subproblems.
4. Develop a greedy strategy to construct a feasible solution step by step, making the locally optimal choice at each step.
5. Prove the correctness of the algorithm by showing that the locally optimal choices at each step lead to a globally optimal solution.

Characteristics of Greedy method

The following are the characteristics of a greedy method:

- To construct the solution in an optimal way, this algorithm creates two sets where one set contains all the chosen items, and another set contains the rejected items.
- A Greedy algorithm makes good local choices in the hope that the solution should be either feasible or optimal.

Components of Greedy Algorithm

The components that can be used in the greedy algorithm are:

- **The feasible solution**: A subset of given inputs that satisfies all specified constraints of a problem is known as a "feasible solution".
- **Optimal solution:** The feasible solution that achieves the desired extremum is called an "optimal solution". In other words, the feasible solution that either minimizes or maximizes the objective function specified in a problem is known as an "optimal solution".
- **Feasibility check:** It investigates whether the selected input fulfils all constraints mentioned in a problem or not. If it fulfils all the constraints then it is added to a set of feasible solutions; otherwise, it is rejected.
- **Optimality check:** It investigates whether a selected input produces either a minimum or maximum value of the objective function by fulfilling all the specified constraints. If an element in a solution set produces the desired extremum, then it is added to a selected of optimal solutions.
- **Optimal substructure property:** The globally optimal solution to a problem includes the optimal sub solutions within it.
- **Greedy choice property:** The globally optimal solution is assembled by selecting locally optimal choices. The greedy approach applies some locally optimal criteria to obtain a partial solution that seems to be the best at that moment and then find out the solution for the remaining sub-problem.

The local decisions (or choices) must possess three characteristics as mentioned below:

- **Feasibility:** The selected choice must fulfill local constraints.
- **Optimality:** The selected choice must be the best at that stage (locally optimal choice).
- **Irrevocability:** The selected choice cannot be changed once it is made.

Greedy choice property:

This property says that the globally optimal solution can be obtained by making a locally optimal solution (Greedy). The choice made by a Greedy algorithm may depend on earlier choices but not on the future. It iteratively makes one Greedy choice after another and reduces the given problem to a smaller one.

Optimal substructure:

A problem exhibits optimal substructure if an optimal solution to the problem contains optimal solutions to the subproblems. That means we can solve subproblems and build up the solutions to solve larger problems.

Applications of Greedy Algorithm

- It is used in finding the shortest path.
- It is used to find the minimum spanning tree using the prim's algorithm or the Kruskal's algorithm.
- It is used in a job sequencing with a deadline.
- This algorithm is also used to solve the fractional knapsack problem.

Pseudo code of Greedy Algorithm

```
Algorithm Greedy (a, n)
{
        Solution : = 0;
        for i = 0 to n do
        {
                x: = select(a);
                if feasible(solution, x)
                {
                        Solution: = union(solution , x)
                }
                return solution;
        }
}
```

Advantages of the Greedy Approach:

- The greedy approach is easy to implement.
- Typically have less time complexity.
- Greedy algorithms can be used for optimization purposes or finding close to optimization in case of Hard problems.
- Greedy algorithms can produce efficient solutions in many cases, especially when the problem has a substructure that exhibits the greedy choice property.
- Greedy algorithms are often faster than other optimization algorithms, such as dynamic programming or branch and bound, because they require less computation and memory.
- The greedy approach is often used as a heuristic or approximation algorithm when an exact solution is not feasible or when finding an exact solution would be too time-consuming.
- The greedy approach can be applied to a wide range of problems, including problems in computer science, operations research, economics, and other fields.

- The greedy approach can be used to solve problems in real-time, such as scheduling problems or resource allocation problems, because it does not require the solution to be computed in advance.
- Greedy algorithms are often used as a first step in solving optimization problems, because they provide a good starting point for more complex optimization algorithms.
- Greedy algorithms can be used in conjunction with other optimization algorithms, such as local search or simulated annealing, to improve the quality of the solution.

Disadvantages of the Greedy Approach:

- The local optimal solution may not always be globally optimal.
- Greedy algorithms do not always guarantee to find the optimal solution, and may produce suboptimal solutions in some cases.
- The greedy approach relies heavily on the problem structure and the choice of criteria used to make the local optimal choice. If the criteria are not chosen carefully, the solution produced may be far from optimal.
- Greedy algorithms may require a lot of preprocessing to transform the problem into a form that can be solved by the greedy approach.
- Greedy algorithms may not be applicable to problems where the optimal solution depends on the order in which the inputs are processed.
- Greedy algorithms may not be suitable for problems where the optimal solution depends on the size or composition of the input, such as the bin packing problem.
- Greedy algorithms may not be able to handle constraints on the solution space, such as constraints on the total weight or capacity of the solution.
- Greedy algorithms may be sensitive to small changes in the input, which can result in large changes in the output. This can make the algorithm unstable and unpredictable in some cases.