Algorithm Design Techniques

1. Brute-force or Exhaustive search
2. Divide and Conquer
3. Greedy algorithm
4. Dynamic Programming
5. Randomized Algorithm
6. Backtracking
7. Branch and Bound

Brute-Force Algorithm

It is an approach to find all the possible solutions of a problem until a satisfactory solution is not found . this is also called an “exhaustive search algorithm”.

Divide and Conquer

It involves dividing the problem into sub-program recursively solving them and then recombining them for final answer .

It is a top-down approach . This algorithm follow divide and conquer techniques involves 3 steps:-

- ➔ Divide the original problem into set of sub problems .
- ➔ Solve every sub-problem.
- ➔ Combine solution of sub-problems into a solution of the whole original problem.

E.g., = quick sort, merge sort, Matrix inversion, matrix multiplication.

Greedy Algorithm

At each step , a decision is made to choice the local optimum , without thinking about the future consequences.

It is use to solve optimization problem in which we are given a set of input values, which are given a set of input values , which are required either to be maximized or minimized i.e., some conditions.

Greedy algorithm always makes the choice looks best at the moment , to optimize given objective.

It doesn’t always guarantee the optimal solution however it generally produces a solution that is very close in value to the optimal.

E.g., = fractional knapsack, Huffman .

Dynamic Programming

The approach is similar to divide and conquer .the difference is that whenever we have recursive function , instead of calling them again we try to store the result in data structure in the form of table and retrieve the results from the table . Thus, the overall time complexity is reduced.

“Dynamic” means it is dynamically decided, whether to call a function or retrieve values from table.

It is bottom-up approach we solve all possible small problems and then combine them to obtain solutions for bigger problems. This is helpful when number of copying sub-problems is exponentially large. It is related to Optimization Problem.

Randomized Algorithm

This algorithm makes use of random numbers to decide the next step in the algorithm . It involves a certain degree of randomness in its logic . The randomness is involved to reduce to complexity of algorithm over other standard algorithm .

E.g., = random quick sort algorithm

Backtracking

It involves searching every possible combination for searching a computational problem and removing the sub-solutions that don’t satisfy the constraints of problem.

It tries each possibility until they find the right one . It is a depth-first search of the set of possible solution.

During search, if an alternative doesn’t work , then backtrack to the choice point, the place which presented different alternative, and tries next alternative .

e.g.,=N-queen problem, maze problem.

Branch and Bound

It is used for solving the optimization problem. In this approach ,the root problem is further divided into subproblems which helps in ruling out a large number of alternatives.

In tis algorithm a given subproblem, which cannot be bounded has to be divided into at least 2 new restricted subproblems.

These are the methods for global optimization in non-complex problem . It can be slow.

e.g., = Job sequencing, Travelling salesman problem .