Python - Algorithm Classes

Algorithms are unambiguous steps which should give us a well-defined output by processing zero or more inputs. This leads to many approaches in designing and writing the algorithms. It has been observed that most of the algorithms can be classified into the following categories.

Greedy Algorithms

Greedy algorithms try to find a localized optimum solution, which may eventually lead to globally optimized solutions. However, generally greedy algorithms do not provide globally optimized solutions.

So greedy algorithms look for a easy solution at that point in time without considering how it impacts the future steps. It is similar to how humans solve problems without going through the complete details of the inputs provided.

Most networking algorithms use the greedy approach. Here is a list of few of them −

- Travelling Salesman Problem
- Prim's Minimal Spanning Tree Algorithm
- Kruskal's Minimal Spanning Tree Algorithm
- Dijkstra's Minimal Spanning Tree Algorithm

Divide and Conquer

This class of algorithms involve dividing the given problem into smaller sub-problems and then solving each of the sub-problem independently. When the problem can not be further sub divided, we start merging the solution to each of the sub-problem to arrive at the solution for the bigger problem.

The important examples of divide and conquer algorithms are −

- Merge Sort
- Quick Sort
- Kruskal's Minimal Spanning Tree Algorithm
- Binary Search

Dynamic Programming

Dynamic programming involves dividing the bigger problem into smaller ones but unlike divide and conquer it does not involve solving each sub-problem independently. Rather the results of smaller sub-problems are remembered and used for similar or overlapping sub-problems.

Mostly, these algorithms are used for optimization. Before solving the in-hand sub-problem, dynamic algorithm will try to examine the results of the previously solved sub-problems.Dynamic algorithms are motivated for an overall optimization of the problem and not the local optimization.

The important examples of Dynamic programming algorithms are −

- Fibonacci number series
- Knapsack problem
- Tower of Hanoi

Print Page