Algorithm Design Techniques

The following is a list of several popular design approaches:

1. Divide and Conquer Approach: It is a top-down approach. The algorithms which follow the divide & conquer techniques involve three steps:

- Divide the original problem into a set of subproblems.
- Solve every subproblem individually, recursively.
- Combine the solution of the subproblems (top level) into a solution of the whole original problem.

2. Greedy Technique: Greedy method is used to solve the optimization problem. An optimization problem is one in which we are given a set of input values, which are required either to be maximized or minimized (known as objective), i.e. some constraints or conditions.

⇧

- Greedy Algorithm always makes the choice (greedy criteria) looks best at the moment, to optimize a given objective.
- The greedy algorithm doesn't always guarantee the optimal solution however it generally produces a solution that is very close in value to the optimal.

3. Dynamic Programming: Dynamic Programming is a bottom-up approach we solve all possible small problems and then combine them to obtain solutions for bigger problems.

This is particularly helpful when the number of copying subproblems is exponentially large. Dynamic Programming is frequently related to **Optimization Problems**.

4. Branch and Bound: In Branch & Bound algorithm a given subproblem, which cannot be bounded, has to be divided into at least two new restricted subproblems. Branch and Bound algorithm are methods for global optimization in non-convex problems. Branch and Bound algorithms can be slow, however in the worst case they require effort that grows exponentially with problem size, but in some cases we are lucky, and the method coverage with much less effort.

5. Randomized Algorithms: A randomized algorithm is defined as an algorithm that is allowed to access a source of independent, unbiased random bits, and it is then allowed to use these random bits to influence its computation.

6. Backtracking Algorithm: Backtracking Algorithm tries each possibility until they find the right one. It is a depth-first search of the set of possible solution. During the search, if an alternative doesn't work, then backtrack to the choice point, the place which presented different alternatives, and tries the next alternative.

7. Randomized Algorithm: A randomized algorithm uses a random number at least once during the computation make a decision.

Example 1: In Quick Sort, using a random number to choose a pivot.

Example 2: Trying to factor a large number by choosing a random number as possible divisors.

Loop invariants

This is a justification technique. We use loop invariant that helps us to understand why an algorithm is correct. To prove statement S about a loop is correct, define S concerning series of smaller statement S_0 S_1....S_k where,

- The initial claim so is true before the loop begins.
- If S_{i-1} is true before iteration i begin, then one can show that S_i will be true after iteration i is over.
- The final statement S_k implies the statement S that we wish to justify as being true.

← prev next →

Please Share

Join Javatpoint Test Series

- Placement Papers
- TCS
- HCL
- Infosys
- IBM
- Accenture
- AMCAT
- eLitmas
- Java
- Python
- C Programming
- Networking
- Bank PO/Clerk
- UPSSSC
- Government Exams
- SSC
- Civil Services
- SBI
- GATE
- NEET
- CAT
- Railway
- CTET
- IIT JEE

⇧

Learn Latest Tutorials

NGINX

SSRS

Blockchain

ETL

Bugzilla

Hive

Jenkins

Pytorch

QA

Agile

JIRA

Tableau

Preparation

Aptitude

Reasoning

Verbal A.

Interview

Company

Trending Technologies

AI

AWS

Selenium

IoT

Cloud

Hadoop

ReactJS

React Native

Node.js

D. Science

Angular 7

B.Tech / MCA

DBMS

DS

DAA

OS

C. Network

Compiler D.

COA

D. Math.

E. Hacking

C. Graphics

Software E.

Web Tech.

Cyber Sec.

Automata

C

C++

Java

.Net

Python

Programs

Control S.