Algorithm Design Methods

Greedy Algorithm/ Methods

- In this method, the optimal solution is constructed in stages.
- In this,
  - at each stage, a decision is made that appears to be the best.
  - A decision made at one stage is not changed in a later stage.
  - **Applications of greedy algorithms:**

Greedy algorithms

- Kruskal's algorithm, Prim's algorithm, Sollin's algorithm

Greedy algorithms -

- Single-Source Shortest Paths
  - The single-source shortest paths problem finds the shortest paths from a given vertex (the source vertex) to all the other vertices.
  - Dijkstra's algorithm can be used if there are no negative edges.
    - It uses greedy algorithm that generates the shortest paths in stages. In each stage a shortest path to a new destination vertex is generated.
  - The Bellman-Ford algorithm is used if there are negative edges.

Divide and Conquer

- Similar to modularization approach to software design.
- To solve a large problem,
  - Divide the problem into two or more smaller instances,
  - Solve each of these smaller problems and
  - Combine the solutions of these smaller problems to obtain solutions to the original problem
- Applications

Divide and Conquer

- Matrix multiplication:
- Merge sort: Eg
- Quick sort :

Dynamic Programming

- In dynamic program the solution to a problem is obtained as a result of a sequence of decisions unlike in the greedy method , where decisions are made one at a time.
- Applications of Dynamic Programming

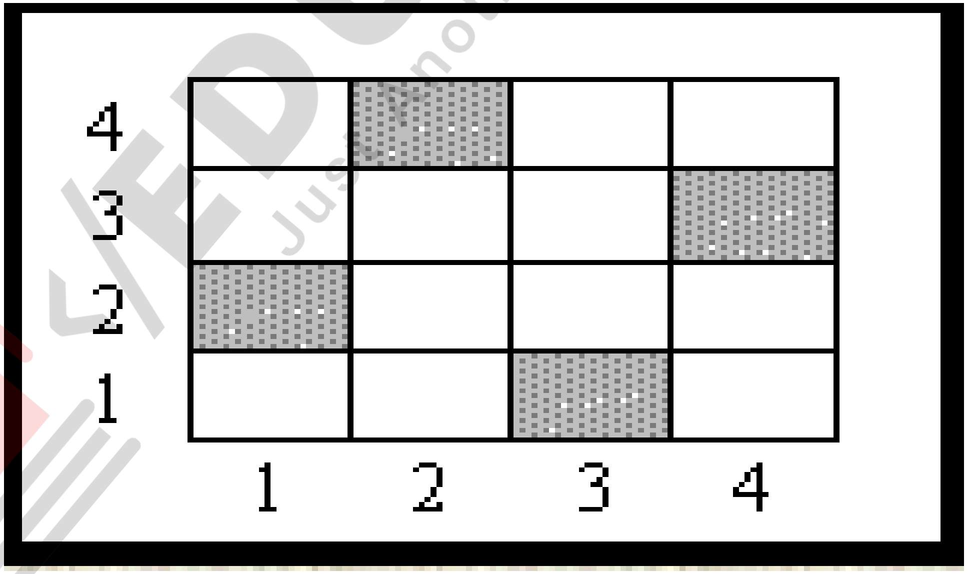
- All- pair shortest paths :
  - If G is the directed graph in which each edge is assigned a weight, then

    cost of any directed path = sum (costs of the edges on this path)
  - For any pair of vertices(i,j), there may be several paths. A path from i to j that has minimum cost among all i to j paths is called shortest path from i to j.
  - Eg:**Floyd-Warshall's algorithm** is used for all-pairs shortest paths if there are no negative weights.

Backtracking

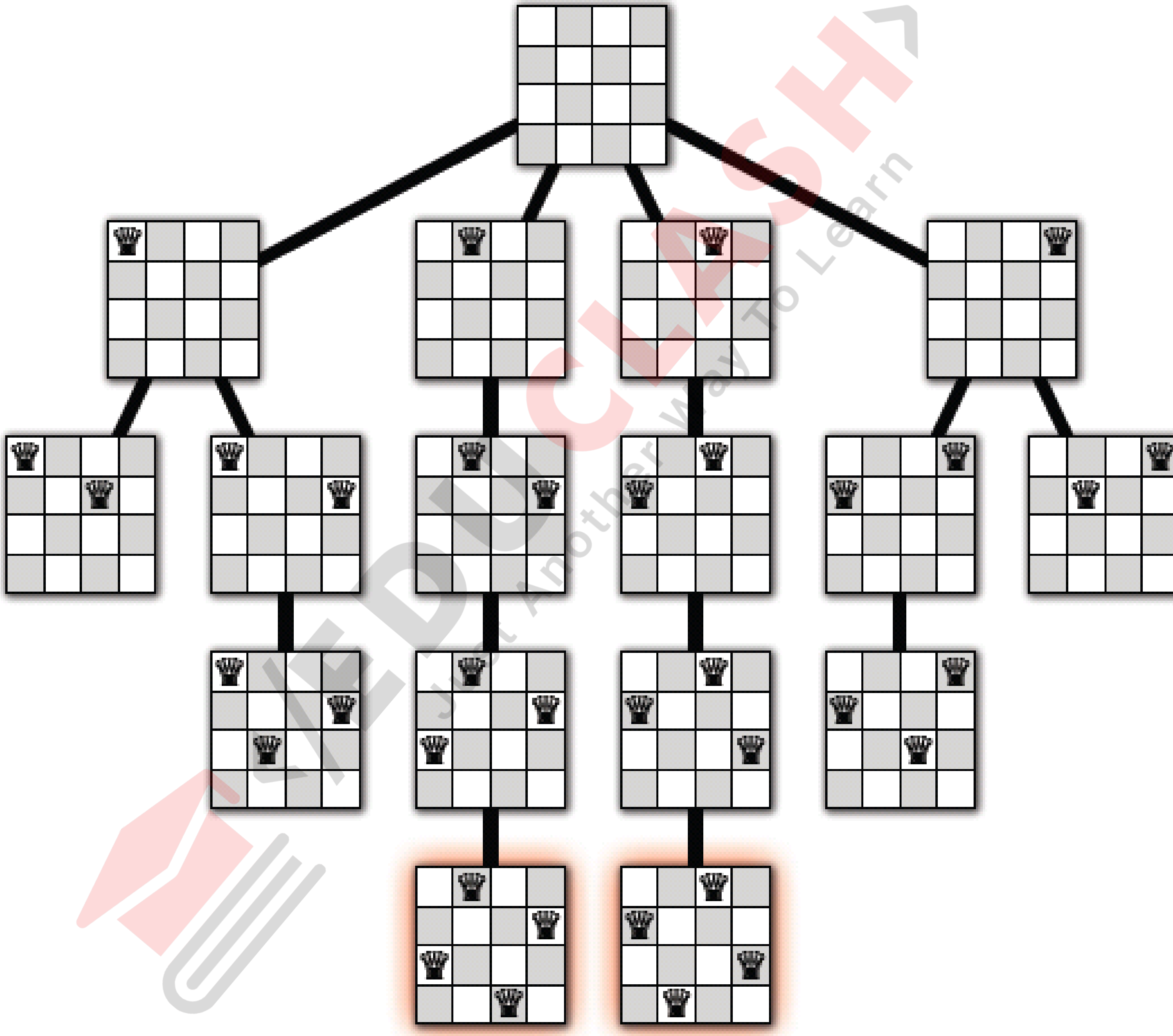
- Is way of solving a problem by trial and error. In this care is taken not to try the same thing twice and also the problem is finite.
- Eg:
  - The n-queens problem
    - It consists in placing n non-attacking queens on an n-by-n chess board. A queen can attack another queen vertically, horizontally, or diagonally. Hence care should be taken to place the queens without attacking each other.

- A queen placed in any of the n^2 squares controls all the squares that are on its row,its column and the 45^0 diagonals.
- Problem :
  - how to put n queens on the chessboard,so that the square of every queen is not controlled by any other queen.
  - Note:
    - For n=2 ,there is no solution,
    - For n=4 a valid solution is given by the drawing below.

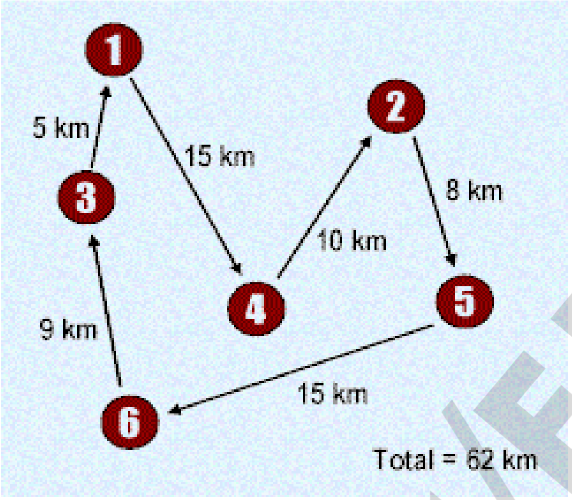


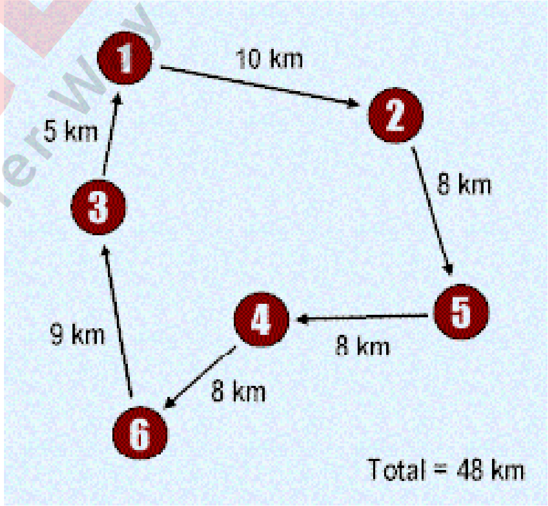
Example :The execution of a backtracking algorithm can be illustrated using a ***recursion tree***.

- The root of the recursion tree corresponds to the original invocation of the algorithm;
- Edges in the tree correspond to recursive calls. A path from the root down to any node shows the partial solution to the *n*-Queens problem, as queens are added to successive rows.
- The leaves correspond to partial solutions cannot be extended because there is already a queen on every row.
- The backtracking algorithm simply performs a traversal of this tree.



- Application of Backtracking
  - Travelling Salesman Problem
    - Given a complete undirected graph *G*=(*V*, *E*) that has nonnegative integer cost *c*(*u*, *v*) associated with each edge (*u*, *v*) in E, the problem is to find a hamiltonian cycle (tour) of *G* with minimum cost.





A salespersons starts from the city 1 and has to visit six cities (1 through 6) and must come back to the starting city i.e., 1.
The first route (left side) 1→ 4 → 2 → 5 → 6 → 3 → 1 with the total length of 62 km, is a relevant selection but is not the best solution. The second route (right side) 1 → 2 → 5 → 4 → 6 → 3 → 1 represents the must better solution as the total distance, 48 km, is less than for the first route.