Backtracking

22CS420 Design and Analysis of Algorithms

Backtracking – An Intuition

- Three sealed Boxes of Gold
- One of them contains Gold
- Problem: To find which box contains gold

- Solutions:
  - Dynamic Programming Approach (Optimized Solution)

  - Greedy Approach (Makes greedy choices)
  - Backtracking (Brute force each and every choice to arrive at the solution(s))

Backtracking

- N-Queens Problem
- Hamiltonian Circuit
- Subset Sum Problem
- Graph Coloring Problem

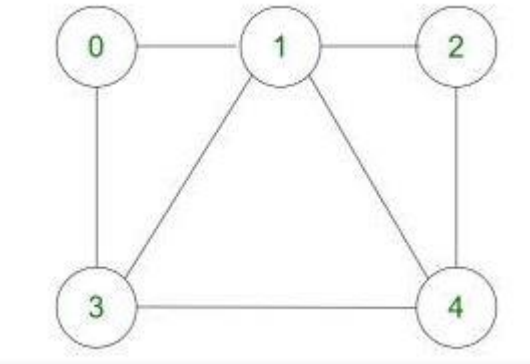
Hamiltonian Circuit

- Hamiltonian Cycle or Circuit in a graph G is a cycle that visits every vertex of G exactly once and returns to the starting vertex.
- Hamiltonian Path in a graph G is a path that visits every vertex of G exactly once and Hamiltonian Path doesn’t have to return to the starting vertex. It’s an open path.

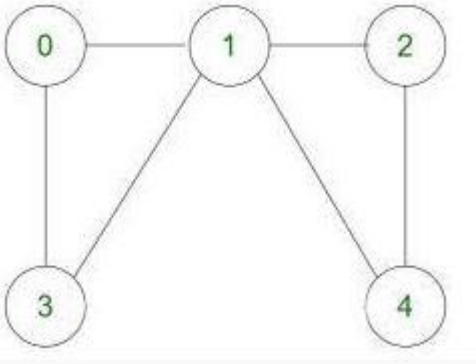
Hamiltonian Circuit Problem

- **Problems Statement:** Given an undirected graph, the task is to determine whether the graph contains a Hamiltonian cycle or not. If it contains, then prints the path.
- NP-Complete Problem
- Applications: logistics, network design, in computer science: particularly in the field of optimization and graph theory.

Example



Graph G

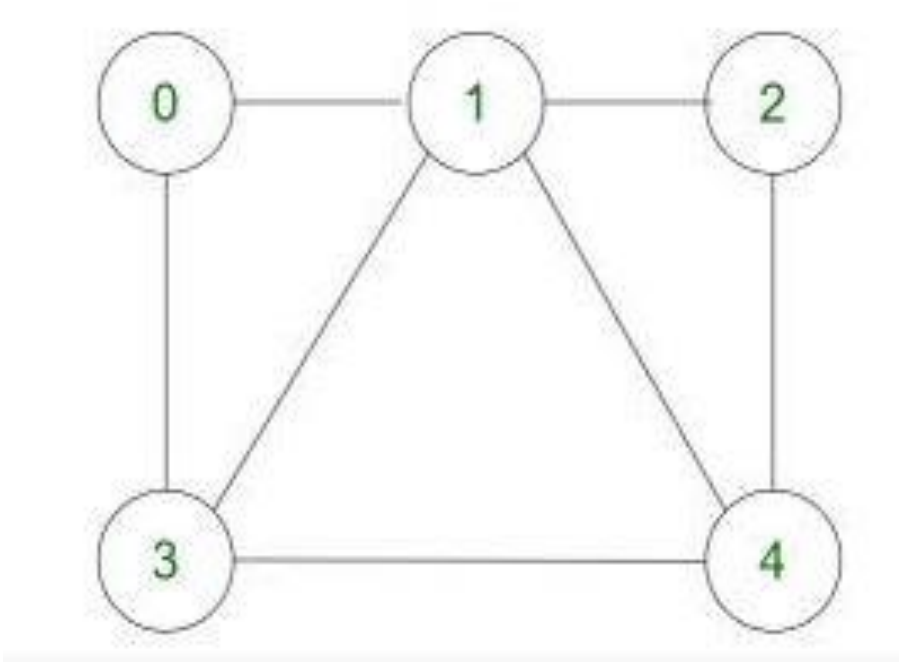


Graph H

Hamiltonian Cycle using Backtracking

- **Start with an empty path** and push the starting vertex into it.
- **Add vertices to the path** one by one, ensuring that each added vertex is **adjacent** to the previously added vertex and **not already in the path.**
- If the path contains all vertices and the **last added vertex is adjacent to the starting vertex**, a Hamiltonian cycle is found.
- If the path cannot be extended further**, backtrack** by removing the last added vertex and trying the next adjacent vertex.

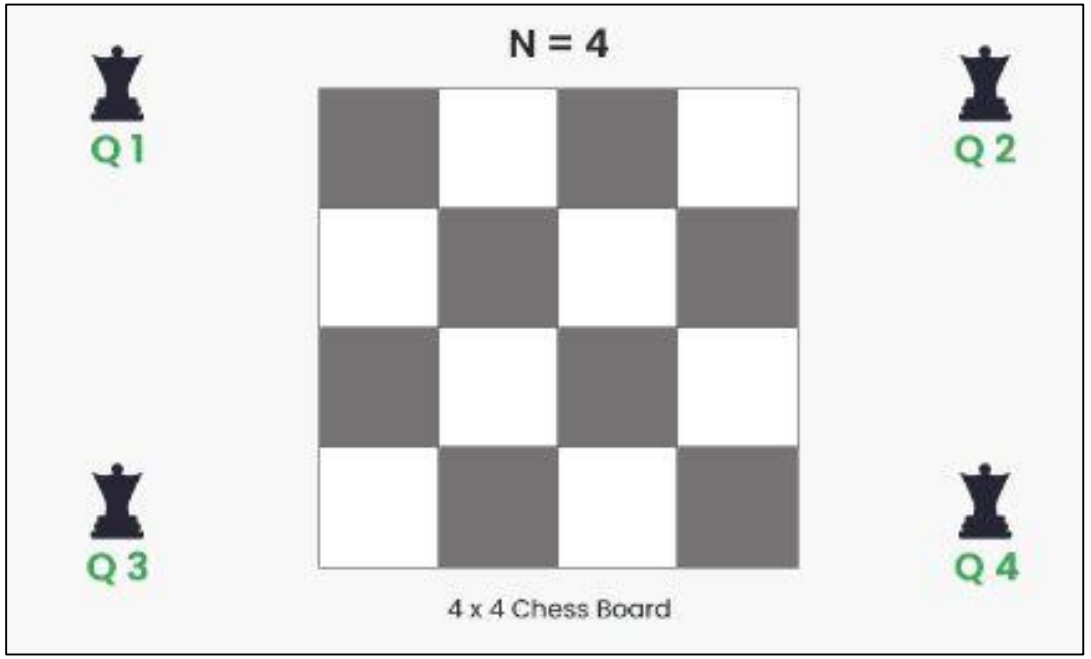
Lets Solve for this example…



Graph G

n-Queens Problem

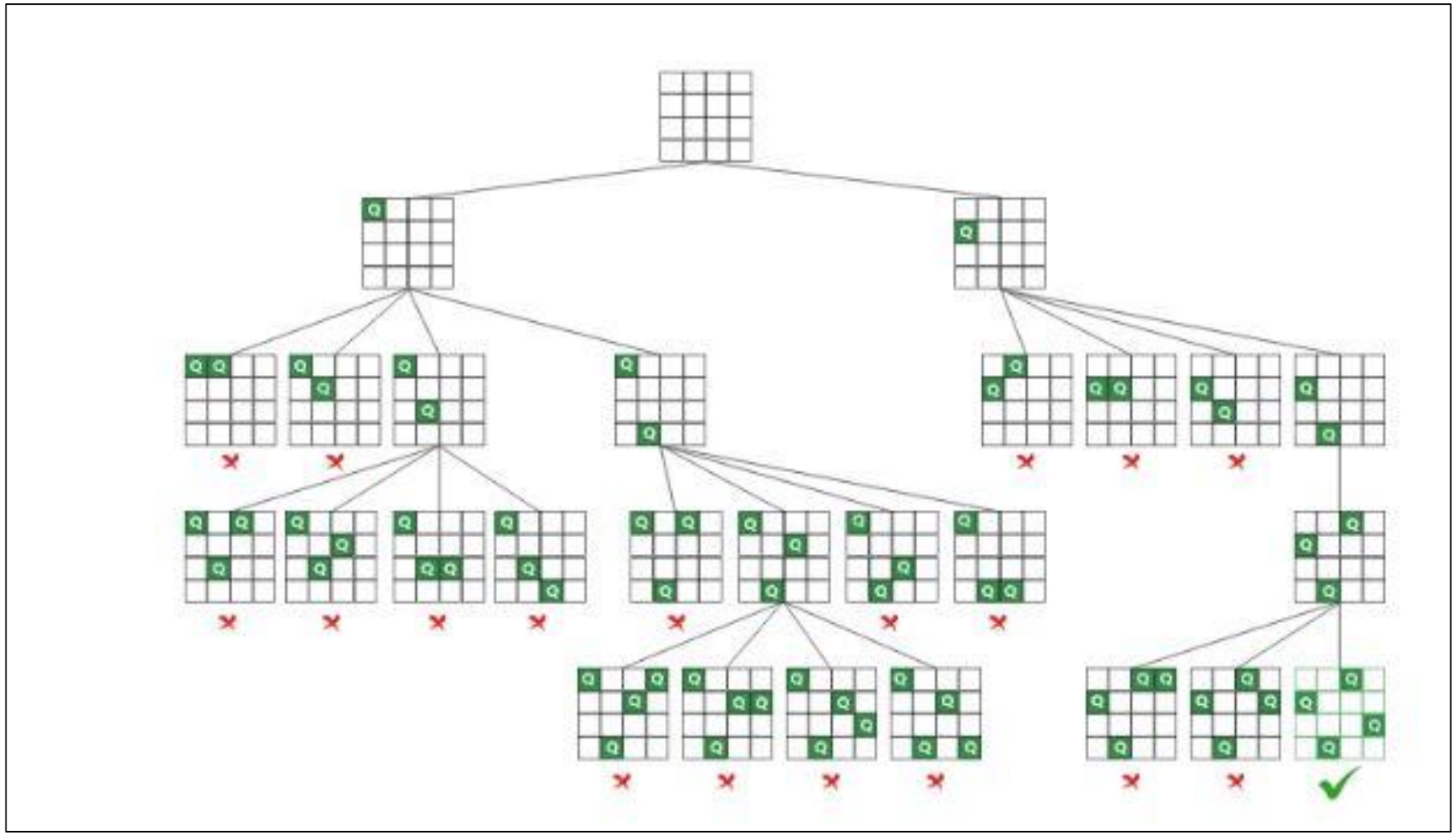
- The N Queen is the problem of placing N chess queens on an $N \times N$ chessboard so that no two queens attack each other.



n-Queens Problem using Backtracking

- Start in the leftmost column
- If all queens are placed return true
- Try all rows in the current column. Do the following for every row.
  - If the queen can be placed safely in this row
    - Then mark this **[row, column]** as part of the solution and recursively check if placing queen here leads to a solution.
    - If placing the queen in **[row, column]** leads to a solution then return **true**.
    - If placing queen doesn't lead to a solution then unmark this **[row, column]** then backtrack and try other rows.
  - If all rows have been tried and valid solution is not found return **false** to trigger backtracking.

State Space Tree



Backtracking

22CS420 Design and Analysis of Algorithms

Subset Sum Problem

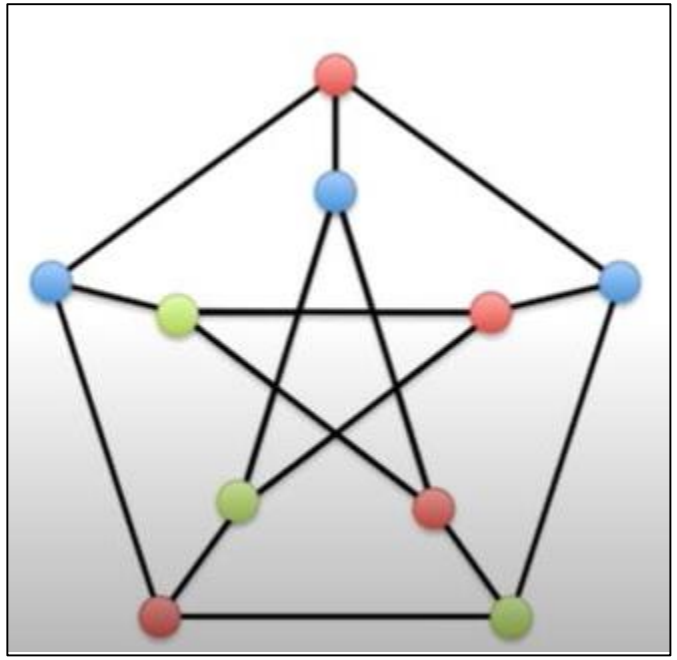
Problem Statement:

Find a subset of a given set $A = \{a_1, a_2, \ldots, a_n\}$ of n positive integers whose sum is equal to a given positive integer d.

Example: $A = \{3,5,6,7\}$; $d = 15$

Graph Coloring Problem

- Given an undirected graph and a number **m**, the task is to color the given graph with **at most m colors** such that **no two adjacent vertices** of the graph are colored with the **same color.**



Graph Coloring Problem using Backtracking

```
Algorithm graphcolor(k)
 for c = 1 to m
        if(isSafe(k,c))
          x[k] = c
          if(k+1<n)
            graphcolor(k+1)
         else
           print x[]
```

Red = 1
Green = 2
Blue = 3

x[k]

0	1	2	3
1	2	1	3

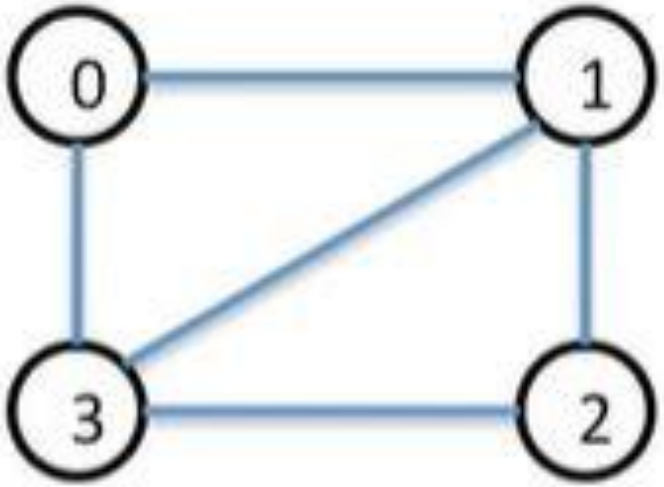
x[0] = 1, x[1] = 2, x[2] = 1, x[3] = 3

```
isSafe(k,c)
 for i from 0 to n-1
   Check if G[k][i] AND c==x[i]
       return false
return true
```

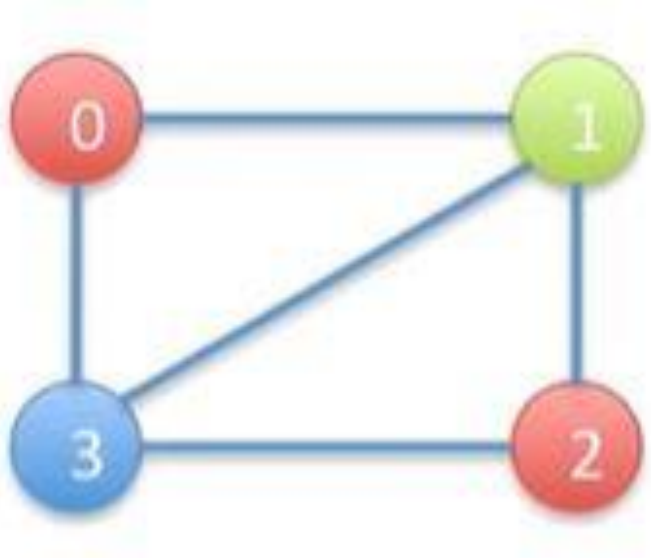
Graph Coloring Problem using Backtracking

- The recursion continues for all the nodes in the graph, trying the different colours
- If no colour is safe, and not all nodes are filled, it will backtrack and try a different colour on the last node set

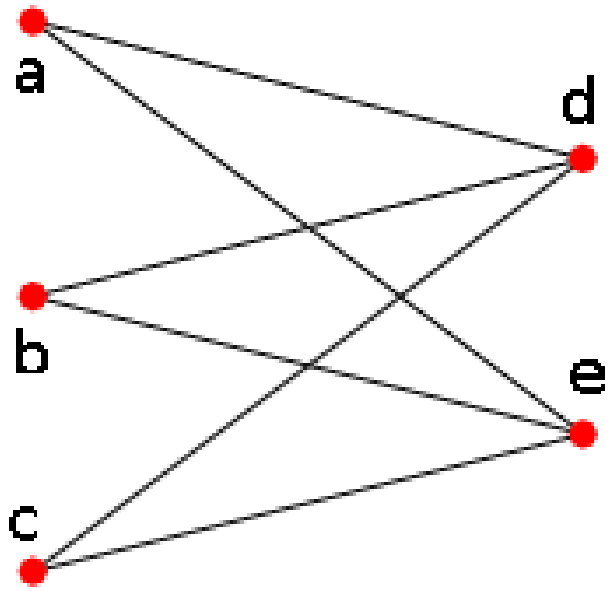
Example:



Example



What is m for the following graph?



Analysis

- It is an NP-hard problem
- It takes $O(m^n)$ time,

  where m is the number of colours and

  n is the number of vertices in a graph