Design and Analysis of Algorithms

Lecture-40

Dharmendra Kumar (Associate Professor)
Department of Computer Science and Engineering
United College of Engineering and Research,
Prayagraj

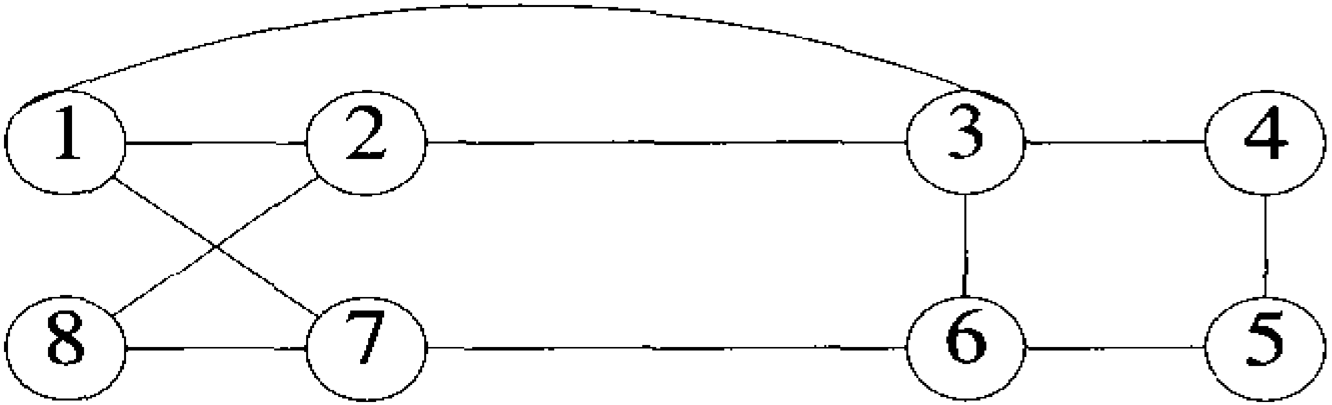
Hamiltonian Cycle Problem

Hamiltonian cycle: A cycle in a graph is said to be Hamiltonian cycle if it contains all the vertices of the graph and no vertex is repeated.

Statement: In Hamiltonian cycle problem, we have to find a Hamiltonian cycle present in the given graph.

Hamiltonian Cycle Problem

Example: Consider the following graph.



Find a Hamiltonian cycle in this graph.

Solution: Hamiltonian cycles are

(1, 3, 4, 5, 6, 7, 8, 2, 1)

(1, 2, 8, 7, 6, 5, 4, 3, 1)

Backtracking algorithm for Hamiltonian Cycle Problem

- The backtracking solution vector $(x_1, x_2, \ldots\ldots, x_n)$ is defined so that x_i represents the i^{th} visited vertex of the proposed cycle.

- First we choose, x_1, can be any vertex of the n vertices.

- We will take $x_1 = 1$. To avoid printing same cycle n-times, we require that $x_1 = 1$.

Backtracking algorithm for Hamiltonian Cycle Problem

- If $1 < k < n$, then x_k can be any vertex v that is distinct from x_1, x_2, ……, x_{k-1} and v is connected by an edge to x_{k+1}. The vertex x_n can only be the one remaining vertex and it must be connected to both x_{n-1} and x_1.

- This algorithm is started by first initializing the adjacency matrix G[1:n][1:n], then x[2:n] to 0 and x[1] to 1 and then executing algorithm Hamiltonian(2).

- Time complexity of this algorithm = **$O(2^n n^2)$.**

Backtracking algorithm for Hamiltonian Cycle Problem

```
Algorithm Hamiltonian(k)
// This algorithm uses the recursive formulation of
// backtracking to find all the Hamiltonian cycles
// of a graph. The graph is stored as an adjacency
// matrix G[1 : n, 1 : n]. All cycles begin at node 1.
{
    repeat
    { // Generate values for x[k].
        NextValue(k); // Assign a legal next value to x[k].
        if (x[k] = 0) then return;
        if (k = n) then write (x[1 : n]);
        else Hamiltonian(k + 1);
    } until (false);
}
```

Backtracking algorithm for Hamiltonian Cycle Problem

```
Algorithm NextValue(k)
// x[1 : k − 1] is a path of k − 1 distinct vertices. If x[k] = 0, then
// no vertex has as yet been assigned to x[k]. After execution,
// x[k] is assigned to the next highest numbered vertex which
// does not already appear in x[1 : k − 1] and is connected by
// an edge to x[k − 1]. Otherwise x[k] = 0. If k = n, then
// in addition x[k] is connected to x[1].
{
    repeat
    {
        x[k] := (x[k] + 1) mod (n + 1); // Next vertex.
        if (x[k] = 0) then return;
        if (G[x[k − 1], x[k]] ≠ 0) then
        { // Is there an edge?
            for j := 1 to k − 1 do if (x[j] = x[k]) then break;
                        // Check for distinctness.
            if (j = k) then // If true, then the vertex is distinct.
                if ((k < n) or ((k = n) and G[x[n], x[1]] ≠ 0))
                    then return;
        }
    } until (false);
}
```

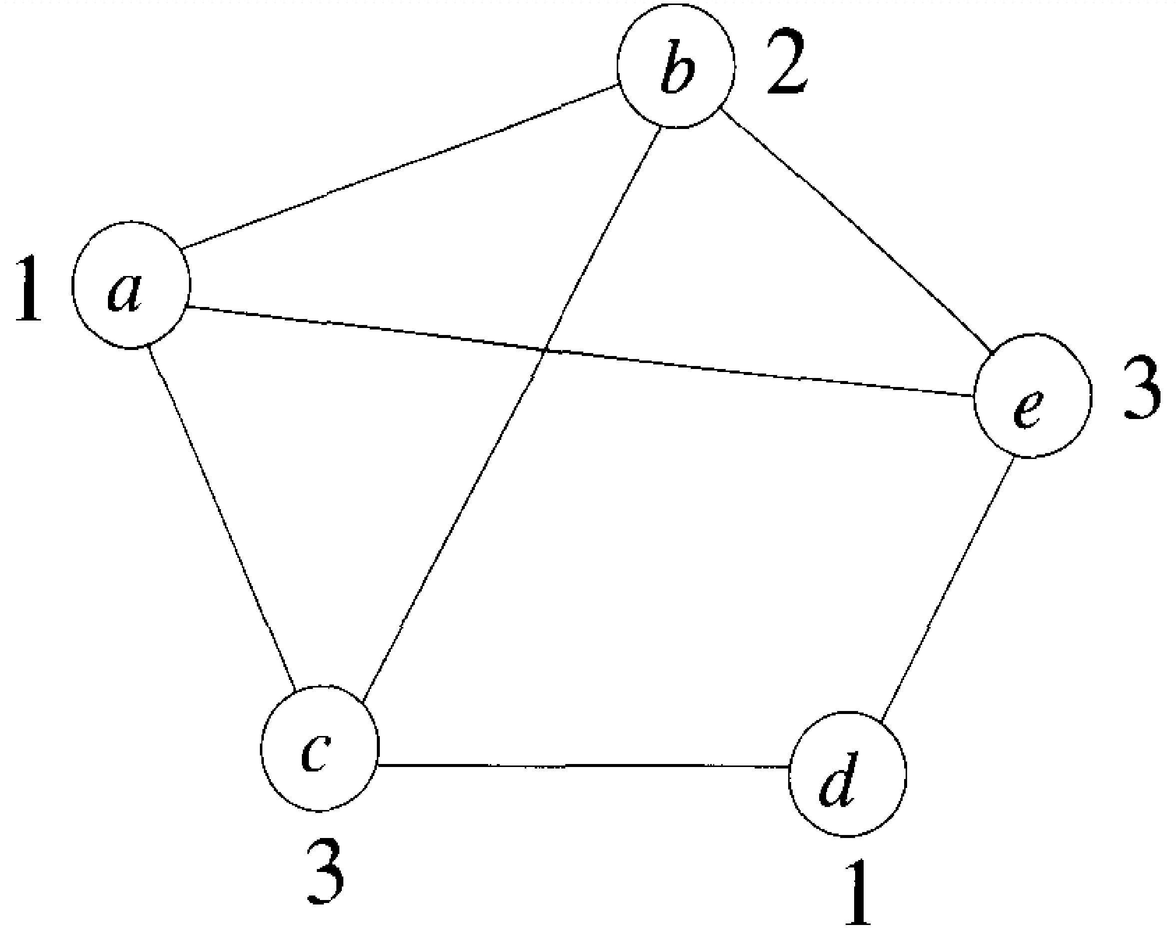
Graph Coloring Problem

- Let G be a graph and m be a given positive integer. We want to discover whether the nodes of G can be colored in such a way that no two adjacent nodes have the same color yet only m colors are used. This is termed the **m-colorability decision problem.**

- The **m-colorability optimization problem** asks for the smallest integer m for which the graph G can be colored. This integer is referred to as the **chromatic number** of the graph.

Graph Coloring Problem

Example: Consider the following graph



Find the chromatic number of this graph.

Backtracking algorithm for Graph Coloring Problem

- Following algorithm determines all the different ways in which a given graph can be colored using at most m colors.

- In this algorithm, a graph is represented by its adjacency matrix G[1:n, 1:n]. The colors are represented by the integers 1, 2, ……., m and the solutions are represented by the n-tuple $(x_1, x_2, ……., x_n)$, where x_i is the color of node i.

- Function **mColoring** is begun by first assigning the graph to its adjacency matrix, setting the array x[] to 0, and then invoking the statement **mColoring(1)**.

Backtracking algorithm for Graph Coloring Problem

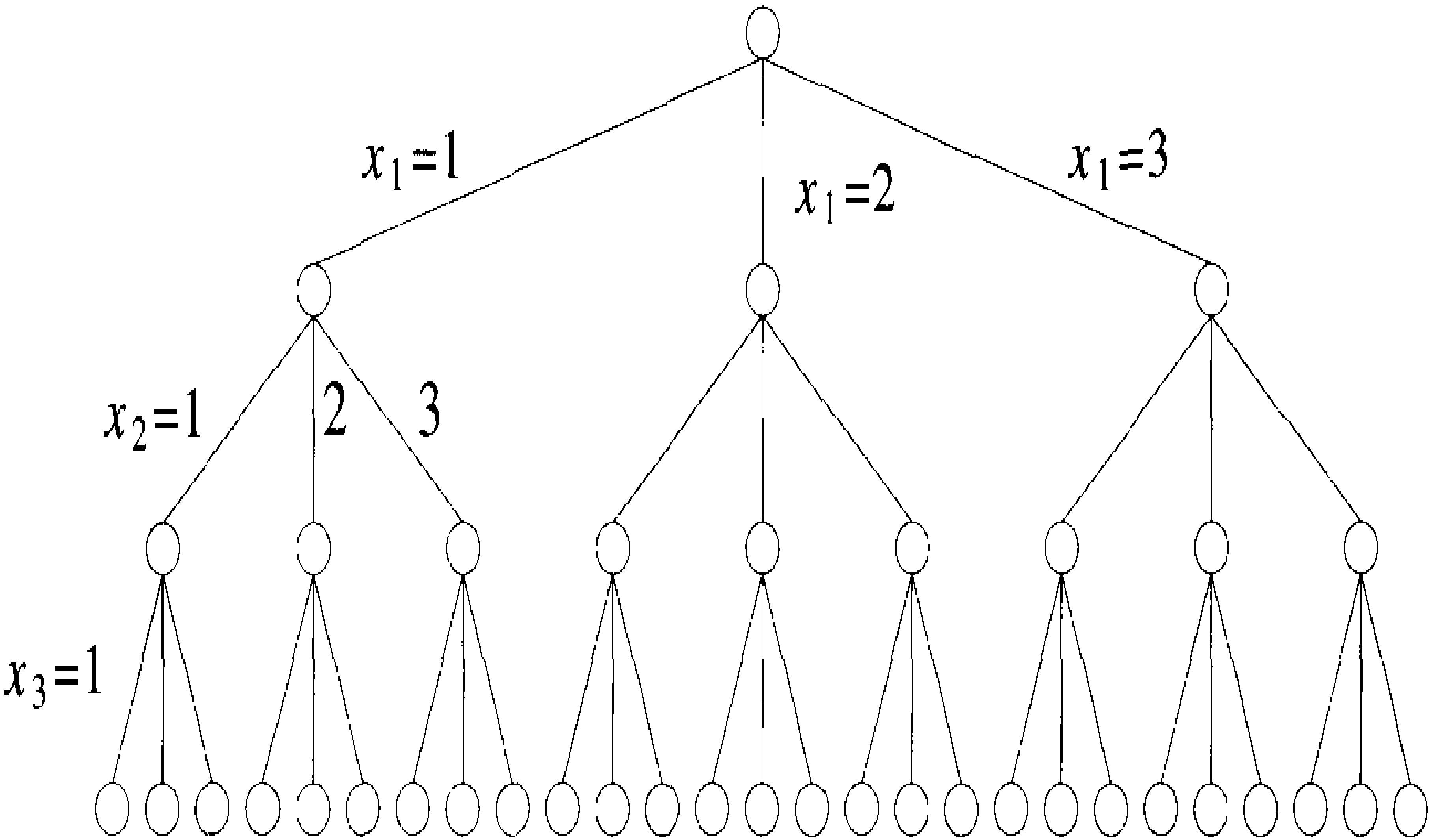
```
mColoring(k)
{
        while(1)
        {
                NextValue(k)
                if (x[k] = 0)
                        return
                if(k=n)
                        print x[1:n]
                else
                        mColoring(k+1)
        }
}
```

Backtracking algorithm for Graph Coloring Problem

```
NextValue(k)
{
        while(1)
        {
                x[k] ← (x[k] + 1) mod (m+1)
                if (x[k] = 0)
                        return
                for ( j← 1  to  n )
                        if(G[k,j] ≠ 0 and (x[k] = x[j]))
                                break
                if (j=n+1)
                        return
        }
}
```

Backtracking algorithm for Graph Coloring Problem

State space tree when n = 3 and m = 3 is



Backtracking algorithm for Graph Coloring Problem

- An upper bound on the computing time of mColoring can be arrived at by the number of internal nodes in the state space tree is $\sum_{i=0}^{n-1} m^i$.

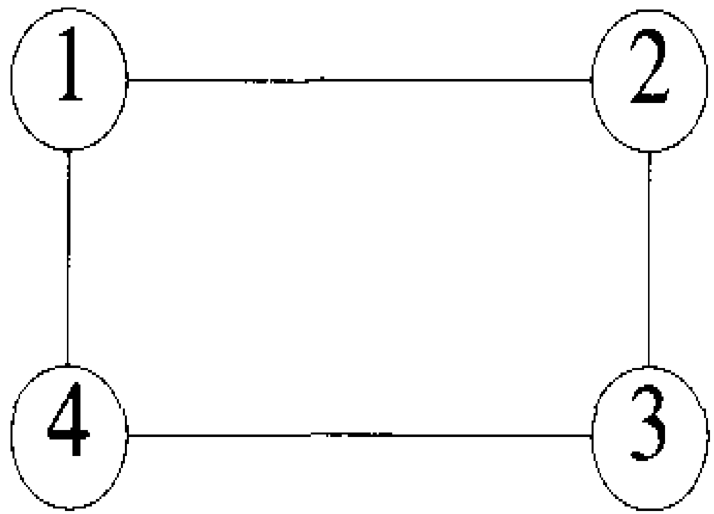
- At each internal node, O(mn) time is spent by NextValue to determine thechildren corresponding to legal colorings. Hence the total time is bounded by

$$\sum_{i=0}^{n-1} m^{i+1} n = \sum_{i=1}^{n} m^i n = n(m^{n+1} - m)/(m-1)$$

$$= O(nm^n)$$

Backtracking algorithm for Graph Coloring Problem

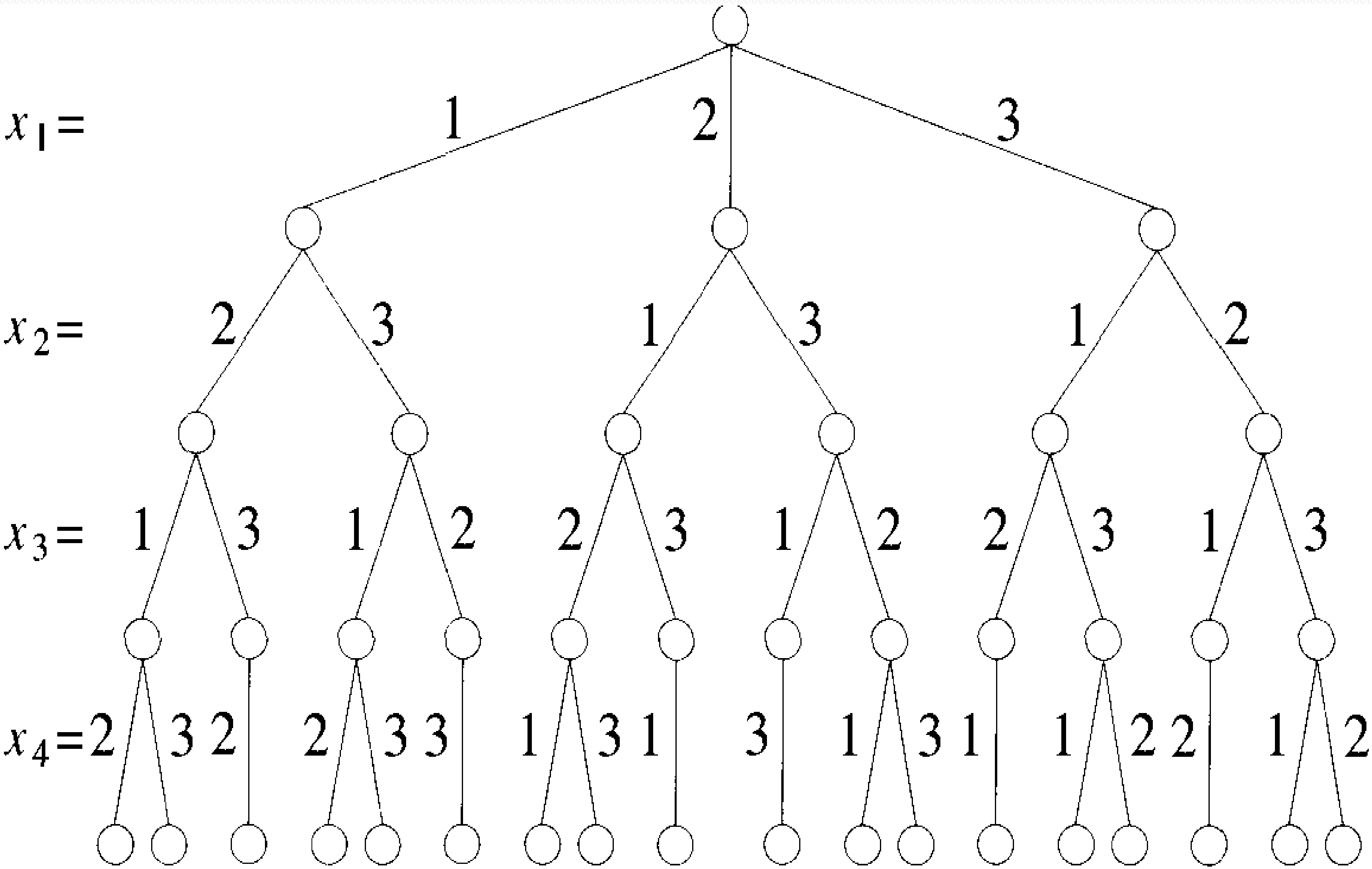
Example: Consider the following graph:-



- Here m = 3.
- Draw state space tree for this graph using mColoring.

Backtracking algorithm for Graph Coloring Problem

Solution: State space tree for the graph will be



AKTU Examination Questions

1. Explain application of graph coloring problem.
2. Solve the Subset sum problem using Backtracking, where

   n=4, m=18, w[4] = {5, 10, 8, 13}
3. Define Graph Coloring.
4. What is backtracking? Discuss sum of subset problem with the help of an example.
5. Explain Implicit and Explicit constraints of N-queen Problem.

AKTU Examination Questions

6. What is the difference between Backtracking and Branch & Bound? Write Pseudo code for Subset Sum Problem using Backtracking. Give example for the same.
7. Consider a graph G=(V,E). We have to find a Hamiltonian cycle using backtracking method.