K. J. Somaiya College of Engineering

(A Constituent College of Somaiya Vidyavihar University)

Batch: A1 **Roll No.: 16010120015**

Experiment No. ___9___

Grade: AA / AB / BB / BC / CC / CD /DD

Signature of the Staff In-charge with date

Title: Implementation of Graph Colouring Backtracking Algorithm

Objective: To learn the Backtracking strategy of problem solving for Graph Colouring problem

CO to be achieved:

CO 2 Analyze and solve problems for divide and conquer strategy, greedy method, dynamic programming approach and backtracking and branch & bound policies.

Books/ Journals/ Websites referred:

1. **Ellis horowitz, Sarataj Sahni, S.Rajsekaran," Fundamentals of computer algorithm", University Press**
2. **T.H.Cormen ,C.E.Leiserson,R.L.Rivest and C.Stein," Introduction to algortihtms",2nd Edition ,MIT press/McGraw Hill,2001**
3. **http://www.math.utah.edu/~alfeld/queens/queens.html**
4. **http://www-isl.ece.arizona.edu/ece175/assignments275/assignment4a/Solving%208%20queen%20problem.pdf**
5. **http://www.slideshare.net/Tech_MX/8-queens-problem-using-back-tracking**
6. **http://www.mathcs.emory.edu/~cheung/Courses/170.2010/Syllabus/Backtracking/8queens.html**
7. **http://www.geeksforgeeks.org/backtracking-set-3-n-queen-problem/**
8. **http://www.hbmeyer.de/backtrack/achtdamen/eight.htm**

Pre Lab/ Prior Concepts:

Data structures, Concepts of algorithm analysis

Historical Profile:

Given an undirected graph and a number m, determine if the graph can be colored with at most m colors such that no two adjacent vertices of the graph are colored with same color. Here coloring of a graph means assignment of colors to all vertices.

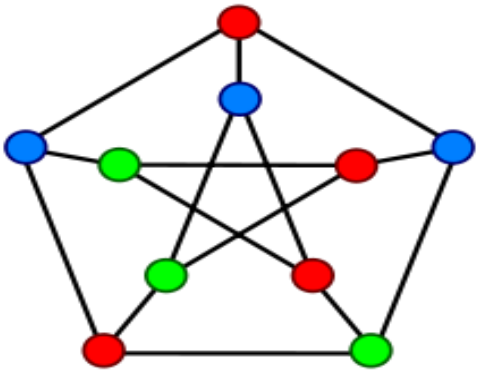
nput:

1) A 2D array graph [V][V] where V is the number of vertices in graph and graph[V][V] is adjacency matrix representation of the graph.

Output:

An array color [V] that should have numbers from 1 to m. color[i] should represent the color assigned to the ith vertex. The code should also return false if the graph cannot be colored with m colors.

Following is an example graph can be colored with 3 colors.



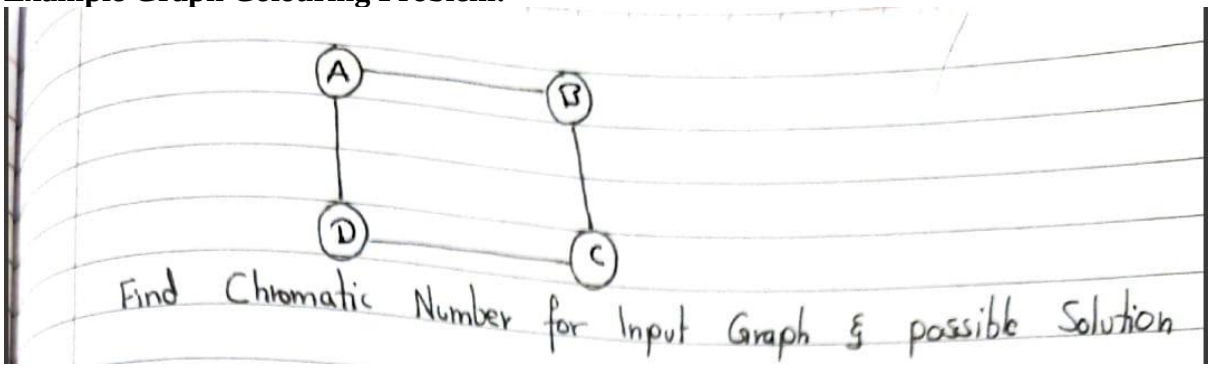
New Concepts to be learned:

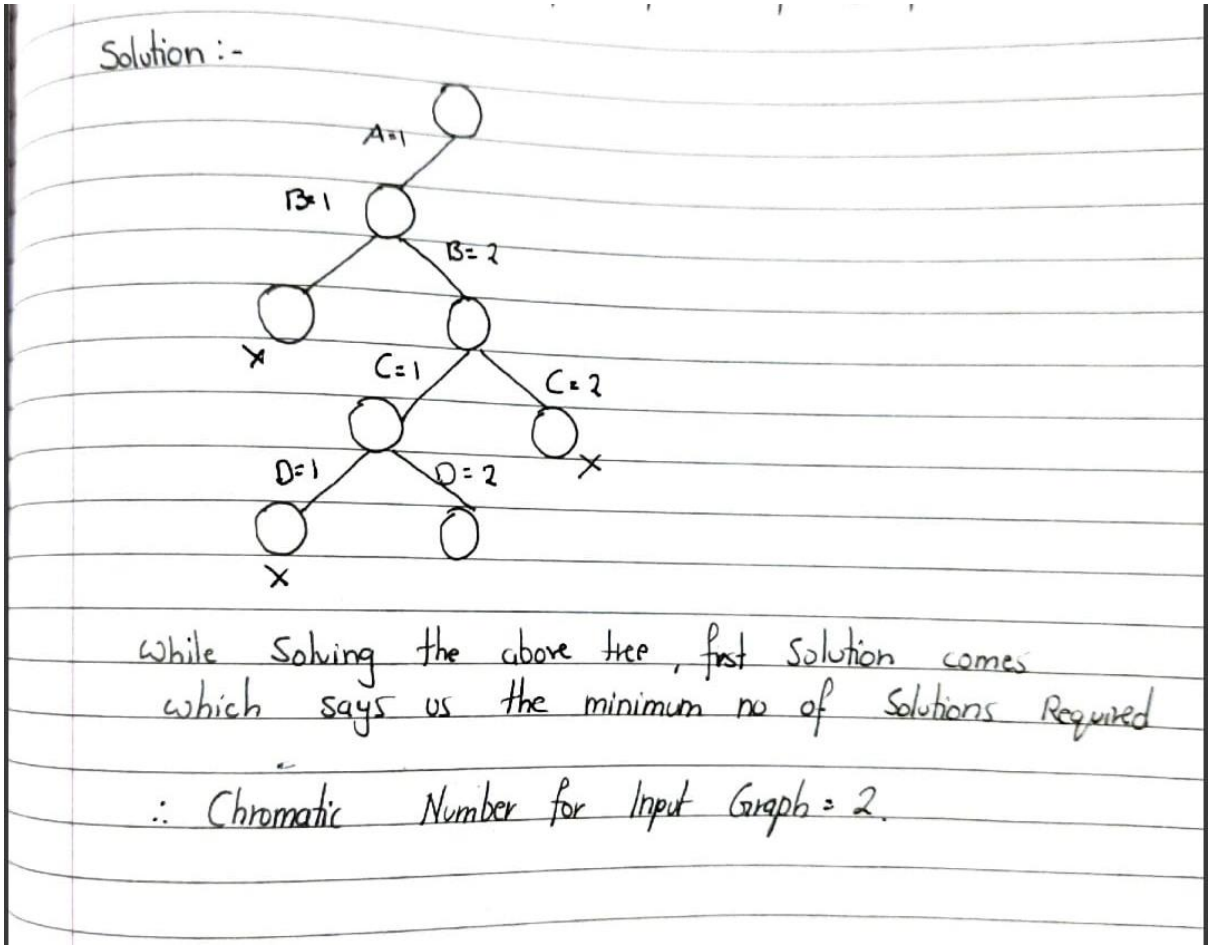
Application of algorithmic design strategy to any problem, Backtracking method of problem solving Vs other methods of problem solving problem graph colouring and its applications.

Algorithm Graph colouring Problem:-

```
1   Algorithm mColoring(k)
2   // This algorithm was formed using the recursive backtracking
3   // schema. The graph is represented by its boolean adjacency
4   // matrix G[1 : n, 1 : n]. All assignments of 1, 2, ..., m to the
5   // vertices of the graph such that adjacent vertices are
6   // assigned distinct integers are printed. k is the index
7   // of the next vertex to color.
8   {
9       repeat
10      {// Generate all legal assignments for x[k].
11          NextValue(k); // Assign to x[k] a legal color.
12          if (x[k] = 0) then return; // No new color possible
13          if (k = n) then     // At most m colors have been
14                              // used to color the n vertices.
15              write (x[1 : n]);
16          else mColoring(k + 1);
17      } until (false);
18  }
```

Example Graph Colouring Problem:





CODE:

```
#include<stdio.h>
int G[50][50],x[50];
void next_color(int k){
 int i,j;
 x[k]=1;
 for(i=0;i<k;i++){
 if(G[i][k]!=0 && x[k]==x[i])
 x[k]=x[i]+1;
 }
}
int main(){
 int n,e,i,j,k,l;
 printf("Enter no. of vertices : ");
 scanf("%d",&n);
 printf("Enter no. of edges : ");
```

```
 scanf("%d",&e);

 for(i=0;i<n;i++)
 for(j=0;j<n;j++)
 G[i][j]=0;

 printf("Enter indexes where value is 1-->\n");
 for(i=0;i<e;i++){
 scanf("%d %d",&k,&l);
 G[k][l]=1;
 G[l][k]=1;
 }

 for(i=0;i<n;i++)
 next_color(i);
 printf("Colors of vertices -->\n");
 for(i=0;i<n;i++)
 printf("Vertex[%d] : %d\n",i+1,x[i]);
 return 0;
}
```

OUTPUT:

```
C:\Users\yashg\OneDrive\Desktop\AOA\Expt2\main\bin\Debug\main.exe
Enter no. of vertices : 5
Enter no. of edges : 5
Enter indexes where value is 1-->
0 1
1 2
1 3
2 3
3 0
Colors of vertices -->
Vertex[1] : 1
Vertex[2] : 2
Vertex[3] : 1
Vertex[4] : 3
Vertex[5] : 1

Process returned 0 (0x0)   execution time : 36.892 s
Press any key to continue.
```

Analysis of Backtracking solution for Graph Colouring Problem:

If all colors are assigned,
print vertex assigned colors
Else a. Trying all possible colors, assign a color to the vertex
b. If color assignment is possible, recursively assign colors to next vertices
c. If color assignment is not possible, de-assign color, return False

Complexity Analysis:

Time Complexity: O(m^V).
There are total O(m^V) combination of colors.
So time complexity is O(m^V). The upperbound time complexity remains the same but the average time taken will be less.

Space Complexity: O(V).

Conclusion:

- Backtracking provides the hope to solve some problem instances of nontrivial sizes by pruning non-promising branches of the state-space tree.
- The success of backtracking varies from problem to problem and from instance to instance.
- Backtracking possibly generates all possible candidates in an exponentially growing state-space tree.

Hence , We were able to implement C program to implement the Backtracking strategy of problem solving for Graph Colouring problem.