**Batch: B3 Roll No.: 121**

**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.Rajasekaran,” Fundamentals of computer algorithm”, University Press**
2. **T.H.Cormen ,C.E.Leiserson,R.L.Rivest and C.Stein,” Introduction to algorithms”,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**](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**](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**](http://www.mathcs.emory.edu/~cheung/Courses/170.2010/Syllabus/Backtracking/8queens.html)
7. [**http://www.geeksforgeeks.org/backtracking-set-3-n-queen-problem/**](http://www.geeksforgeeks.org/backtracking-set-3-n-queen-problem/)
8. [**http://www.hbmeyer.de/backtrack/achtdamen/eight.htm**](http://www.hbmeyer.de/backtrack/achtdamen/eight.htm)
9. [**https://www.tutorialspoint.com/Graph-Coloring**](https://www.tutorialspoint.com/Graph-Coloring)
10. [**https://www.gyaanibuddy.com/assignments/assignment-detail/graph-coloring-backtracking-algorithm/**](https://www.gyaanibuddy.com/assignments/assignment-detail/graph-coloring-backtracking-algorithm/)
11. [**https://www.shiksha.com/online-courses/articles/introduction-to-backtracking/**](https://www.shiksha.com/online-courses/articles/introduction-to-backtracking/)
12. [**https://www.scaler.com/topics/graph-coloring-problem/**](https://www.scaler.com/topics/graph-coloring-problem/)

**Pre Lab/ Prior Concepts:**

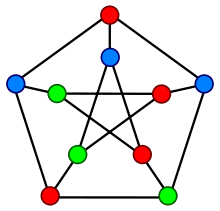
Data structures, Concepts of algorithm analysis

**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 the same color. Here coloring of a graph means assignment of colors to all vertices.

***Input:***1) A 2D array graph [V][V] where V is the number of vertices in the graph and graph[V][V] is the 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 that can be colored with 3 colors.  


**New Concepts to be learned:**

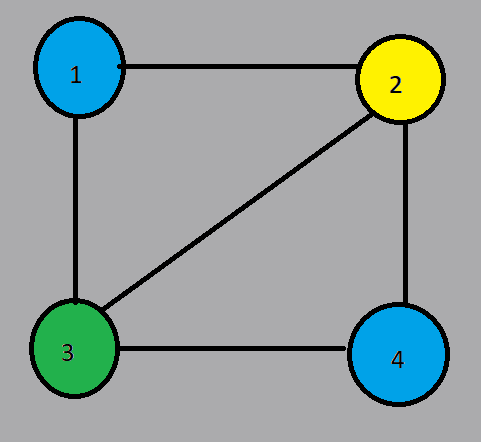
Application of algorithmic design strategy to any problem, Backtracking method of problem solving Vs other methods of problem solving problem graph colouring and its applications.

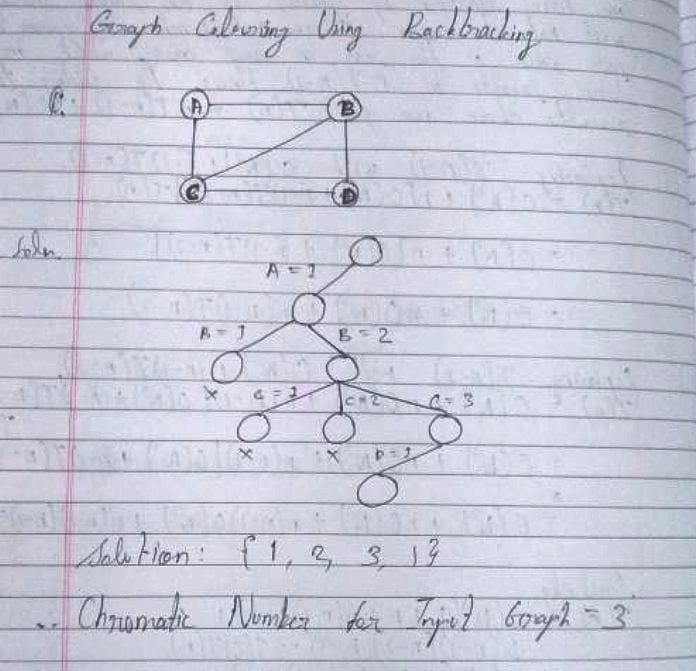
**Algorithm Graph colouring Problem:-**

**Text, letter

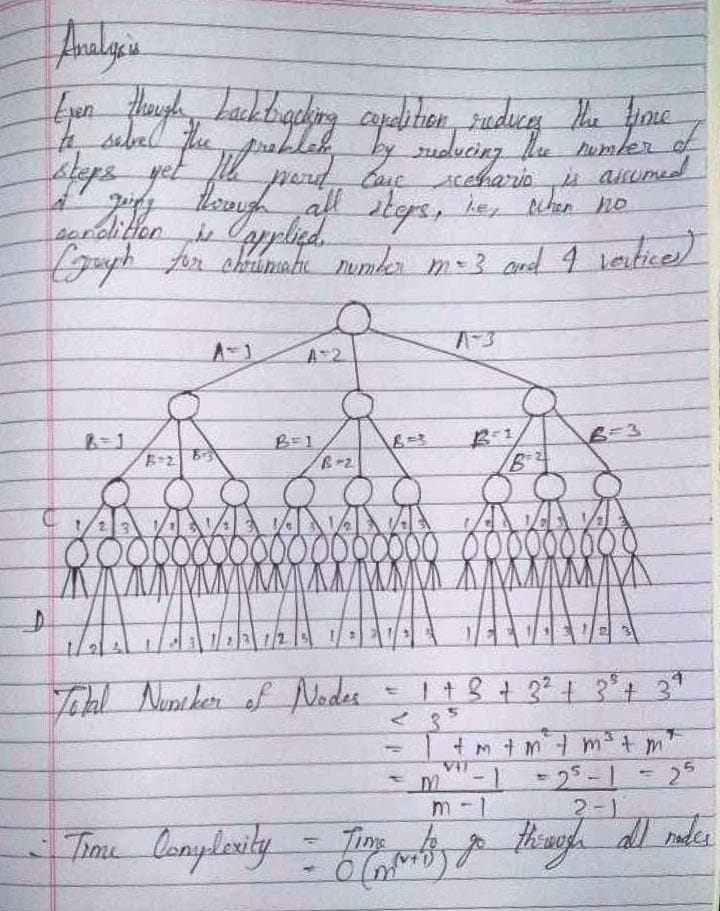
Description automatically generated**

**Example Graph Colouring Problem:**

****

****

**Analysis of Backtracking solution for Graph Colouring Problem:**

****

In the backtracking approach to the graph coloring problem, we are not using any extra space but we are using the recursive stack for the recursive function call. So, the space complexity is O(V).

**Conclusion:**

Thus, in this experiment, the concept of Backtracking has been used to solve Graph Coloring Problem. Using Backtracking is helpful as Backtracking has a brute-force nature; due to this reason, it can solve maximum problems. Also, the step-by-step representation of the backtracking solution is straightforward to understand. Further, backtracking code can easily be debugged. However, when compared to other options, it is quite slow. Also, due to the usage of recursion and stack storage for function information, there is a high space complexity. Thus, we can conclude that Backtracking should be used in appropriate situation where its advantages are more apparent than its disadvantages.