Backtracking | Set 5 (m Coloring Problem)

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.

Input:
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. A value graph[i][j] is 1 if there is a direct edge from i to j, otherwise graph[i][j] is 0.
2) An integer m which is maximum number of colors that can be used.

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 (from Wiki page) that can be colored with 3 colors.

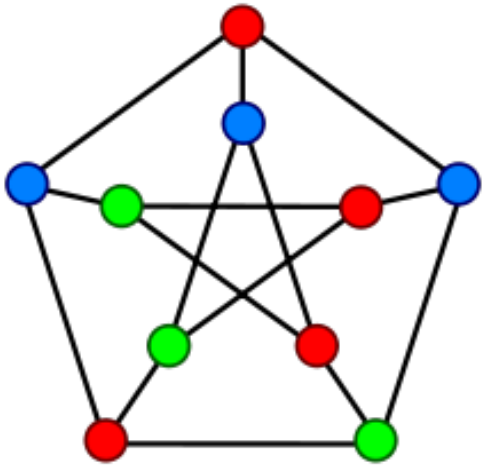
Google™ Custom Search

GeeksforGeeks

53,525 people like GeeksforGeeks.

Interview Experiences

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com



Naive Algorithm
Generate all possible configurations of colors and print a configuration that satisfies the given constraints.

```
while there are untried conflagrations
{
   generate the next configuration
   if no adjacent vertices are colored with same color
   {
      print this configuration;
   }
}
```

There will be V^m configurations of colors.

Backtracking Algorithm
The idea is to assign colors one by one to different vertices, starting from the vertex 0. Before assigning a color, we check for safety by considering already assigned colors to the adjacent vertices. If we find a color assignment which is safe, we mark the color assignment as part of solution. If we do not a find color due to clashes then we backtrack and return false.

Implementation of Backtracking solution

```
#include<stdio.h>
```


Beginners Guide to Git

atlassian.com/Learn-Git

A free tutorial to help introduce you to the Git basics. Learn more.

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

```
// Number of vertices in the graph
#define V 4

void printSolution(int color[]);

/* A utility function to check if the current color assignment
   is safe for vertex v */
bool isSafe (int v, bool graph[V][V], int color[], int c)
{
    for (int i = 0; i < V; i++)
        if (graph[v][i] && c == color[i])
            return false;
    return true;
}

/* A recursive utility function to solve m coloring problem */
bool graphColoringUtil(bool graph[V][V], int m, int color[], int v)
{
    /* base case: If all vertices are assigned a color then
       return true */
    if (v == V)
        return true;

    /* Consider this vertex v and try different colors */
    for (int c = 1; c <= m; c++)
    {
        /* Check if assignment of color c to v is fine*/
        if (isSafe(v, graph, color, c))
        {
           color[v] = c;

           /* recur to assign colors to rest of the vertices */
           if (graphColoringUtil (graph, m, color, v+1) == true)
             return true;

            /* If assigning color c doesn't lead to a solution
               then remove it */
           color[v] = 0;
        }
    }

    /* If no color can be assigned to this vertex then return false */
    return false;
}

/* This function solves the m Coloring problem using Backtracking.
  It mainly uses graphColoringUtil() to solve the problem. It returns
```


stack!

Structure Member Alignment, Padding and Data Packing

Intersection point of two Linked Lists

Lowest Common Ancestor in a BST.

Check if a binary tree is BST or not

Sorted Linked List to Balanced BST

HP Chromebook 11

google.com/chromebook

Everything you need in one laptop.
Made with Google. Learn more.

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

```
  false if the m colors cannot be assigned, otherwise return true and
  prints assignments of colors to all vertices. Please note that there
  may be more than one solutions, this function prints one of the
  feasible solutions.*/
bool graphColoring(bool graph[V][V], int m)
{
    // Initialize all color values as 0. This initialization is needed
    // correct functioning of isSafe()
    int *color = new int[V];
    for (int i = 0; i < V; i++)
       color[i] = 0;

    // Call graphColoringUtil() for vertex 0
    if (graphColoringUtil(graph, m, color, 0) == false)
    {
      printf("Solution does not exist");
      return false;
    }

    // Print the solution
    printSolution(color);
    return true;
}

/* A utility function to print solution */
void printSolution(int color[])
{
    printf("Solution Exists:"
           " Following are the assigned colors \n");
    for (int i = 0; i < V; i++)
      printf(" %d ", color[i]);
    printf("\n");
}

// driver program to test above function
int main()
{
    /* Create following graph and test whether it is 3 colorable
      (3)---(2)
       |   / |
       |  /  |
       | /   |
      (0)---(1)
    */
    bool graph[V][V] = {{0, 1, 1, 1},
        {1, 0, 1, 0},
        {1, 1, 0, 1},
```

705 Subscribe

Recent Comments

Abhi You live US or India?

Google (Mountain View) interview · 8 minutes ago

Aman Hi, Why arent we checking for conditions...

Write a C program to Delete a Tree. · 47 minutes ago

kzs please provide solution for the problem...

Backtracking | Set 2 (Rat in a Maze) · 51 minutes ago

Sanjay Agarwal bool tree::Root_to_leaf_path_given_sum(tree...

Root to leaf path sum equal to a given number · 1 hour ago

GOPI GOPINATH @admin Highlight this sentence "We can easily...

Count trailing zeroes in factorial of a number · 1 hour ago

newCoder3006 If the array contains negative

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

```
        {1, 0, 1, 0},
    };
    int m = 3; // Number of colors
    graphColoring (graph, m);
    return 0;
}
```

Output:

```
Solution Exists: Following are the assigned colors
 1  2  3  2
```

References:
http://en.wikipedia.org/wiki/Graph_coloring

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

Free C# Code Generator

ironspeed.com

Create database & reporting apps straight from your database! Try it

Related Tpoics:

- Backtracking | Set 8 (Solving Cryptarithmetic Puzzles)
- Tail Recursion

numbers also. We...

Find subarray with given sum · 1 hour ago

AdChoices

▶ Java Source Code

▶ Graph Coloring

▶ Graph Theory

AdChoices

▶ Tree Graph

▶ Graph C++

▶ Graph Java

AdChoices

▶ Linear Graph

▶ Graph Program

▶ Graph Search

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

- Find if two rectangles overlap
- Analysis of Algorithm | Set 4 (Solving Recurrences)
- Print all possible paths from top left to bottom right of a mXn matrix
- Generate all unique partitions of an integer
- Russian Peasant Multiplication
- Closest Pair of Points | O(nlogn) Implementation

8 Tweet 2 0

Writing code in comment? Please use **ideone.com** and share the link here.

12 Comments GeeksforGeeks

Sort by Newest

Join the discussion…

sr7 • 3 months ago

you can further bring down the complexity by avoiding the check for each colo
and no of vertices is N .

Checking for each color will take O(n^2) time . Instead

I assume the base color (uncolored node) is -1 . Colors start from 0.

I will check all the adjacent nodes of the given vertex in O(n) time .

If i th node is adjacent to vertex v and color[i]>=0 (colored node) , then temp[c
the color is in use by one of its adjacent nodes) , finally I will search the 'temp'

if temp[i]!=1, then that is the least color not used by any of the adjacent nodes
on the whole, instead of O(n^2).

int temp[V];

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

//Initialize all temp values to zero

for (i=0;i<v;i++) temp[i]="0;" for="" (i="0;i<V;i++)" if="" (graph[v][i]="" &&="" c
colored

temp[color[i]]=1;

for (i=0; i<m ;="" i++)="" {="" only="" 0="" to="" m-1="" colors="" are="" allowe
if="" (graphcoloringutil(graph,m,color,v+1))="" return="" true;="" color[v]="-1;" b

pavi.8081 • 10 months ago

Can we modify this algorithm to find the min. number of colors to color the gra
tried though.

If we can do so then the modified algorithm can be used to solve problems like
halls required to organize parties with given starting and finishing times.

Please comment...

soupboy • 2 years ago

Hi, shouldn't the brute force algorithm's complexity be m^V and not V^m as wri

coolguy soupboy • 2 years ago

You can understand graph coloring using backtracking by watching this
youtu.be/CI3A_9hokjU

Rohit Raj • 2 years ago

bool isSafe (int v, bool graph[V][V], int color[], int c)
{

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

```
for (int i = 0; i < V; i++)
if (graph[v][i] && c == color[i])
return false;
return true;
}
```

as the problem statement says that if (i == j) then also the graph[i][j] = 1; so s
condition in this function?
i.e if(graph[v][i] && c == color[i] && v != i).

kartik ➚ Rohit Raj • 2 years ago

@Rohit Raj

We don't need to check for this as we assign a color c only after isSaf
we assign the color value as 0 which is not a valid value. When we cal
than 0 and value of color[v] will be 0. So color[v] will never be equal to c

gautam • 2 years ago

Can be use some dynamic programming to reduce complexity

```
/* Paste your code here (You may delete these lines if not writing c
```

Venki • 2 years ago

There is small bug in the code. The colour vector should be of size V in lieu of
tries to access elements past the buffer end (while checking validity of colour

What is the worst case complexity? O(V * m V) as we are exploring all configu
call.

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

GeeksforGeeks ➚ Venki • 2 years ago

@Venki: Thanks for pointing this out. This has been fixed. Yes, the wor
) as we may have to try all configurations in worst case.

Guddu sharma • 2 years ago

What is the need of following if condition:
if (graphColoringUtil (graph, m, color, v+1) == true)
return true;

I think when all the vertices have been assigned color,the first condition if(v==\

Please comment if i have understood something wrong..

Venki ➚ Guddu sharma • 2 years ago

@Guddu sharma, We need to find *one* feasible solution. The conditic
end infinite recursion. When we reach the end branch in the state spac
need to backtrack to one level upper and restore the state.

Similarly, if the current assignment is safe, (means the branch generat
state), we recur to next level for next feasible color.

The condition

```
 /* recur to assign colors to rest of the vertices */
if (graphColoringUtil (graph, m, color, v+1) == true)
    return true;
```

checks next color. If this next color reaches last color, we are done wh

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

```
        return true;
```

Try to draw the state space tree for small values of V and m, you can e
procedure.

kartik ➚ Guddu sharma • 2 years ago

@Guddu sharma: The statement serves two purposes:
1) It calls the function recursively for the other vertices.
2) It returns true as soon as one of the color assignments lead to a sol
checked as soon as a color assignment works.

Also, just putting following line is not sufficient. We need to forward the
calls.

```
    if (v == V)
       return true;
```

1 

 Subscribe

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

@geeksforgeeks, **Some rights reserved** **Contact Us!** Powered by **WordPress** & **MooTools**, customized by geeksforgeeks team

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com