Application 9: The Chromatic Number

Dumitru Cristina - Nicoleta

June 2018

First Year
Group: C.E.N 1.1 B

1 Problem statement

Chromatic Number. Implement two different algorithms to determine the minimum number of colours needed to colour each node in an undirected graph, such that two adjacent vertex do not have the same colour.

2 Application Design

- The high level architectural overview of the application.
- The application uses one main function for resolving the given task and one auxiliary function called in the main one in order to make the whole program more organized.

The main function is represented by :

- Data introduction.
- The use of auxiliary functions.
- Display the result.

The auxiliary functions are represented by:

- The function that generates the random matrix.
- The function that uses a vector to retain the vertices of the graph and color them.

2.1 Function used and parameters

2.1.1 ** Matrix generator(int NoVertices)

The function creates a random adjacent array and assigns it dynamically to the main function. Because the vertices can not be linked to themselves the main diagonal will be fill with 0 and the rest of the matrix will be generate randomly I used a double pointer to hold the adjacent matrix.

2.1.2 Chromatic number(int ** adjacency matrix, int k)

Using the global variable color vector witch is given the value 1, because there can not be any color 0, this function will go trough the matrix and check if there are two adjacent vertices having the same color. If they have the current vertex will be colored differently. This functions uses backtracking.

3 Definition

A proper vertex coloring of the Petersen graph with 3 colors, the minimum number possible. In graph theory, graph coloring is a special case of graph labeling; it is an assignment of labels traditionally called "colors" to elements of a graph subject to certain constraints. In its simplest form, it is a way of coloring the vertices of a graph such that no two adjacent vertices share the same color; this is called a vertex coloring. Similarly, an edge coloring assigns a color to each edge so that no two adjacent edges share the same color, and a face coloring of a planar graph assigns a color to each face or region so that no two faces that share a boundary have the same color.

A coloring of a graph is almost always a proper vertex coloring, namely a labeling of the graph's vertices with colors such that no two vertices sharing the same edge have the same color. Since a vertex with a loop (i.e. a connection directly back to itself) could never be properly colored, it is understood that graphs in this context are loopless.

4 Pseudocode

```
MatrixGenerator FUNCTION
MatrixGenerator (NoVertices,)
1.tipToIntegerAdjacenctMatrix
2. int line, column (0 to NoVtices)
3. for line ¡ NoVertices
4.      for column ¡ NoVertices
5.          if line := column then
6.             AdjacencyMatrix[line][column] = 0
7.          else
8.             AdjacencyMatrix[line][column] := random(0,1)
9.      AdjacencyMatrix[column][line] := AdjacencyMatrix[line][column]
10. return AdjacencyMatrix

ChromaticNumber FUNCTION
ChromaticNumber (AdjacencyMatrix,k)
11. int line(0 to NoVtices)
12. for line ¡ k
13.         if AdjacenctMatrix[line][k] = 1 and ColorVector[k] = ColorVector[line] then
14.            ColorVector[k] := ColorVector[line] + 1

IN MAIN
15.int AdjacenctMatrix, color
16. int line, column
17. AdjacencyMatrix[line][column] := MatrixGenerator(NoVertices)
18. for line :=0 to NoVertices
19.      for column := 0 to NoVertices
20.          AdjacencyMatrix[line][column] =
21.
22.            write AdjacencyMatrix[line][column] := random(0,1)
23. for line = 0 to NoVertices
24.      ChromaticNumber(AdjacencyMatrix, NoVertices) 25/.  line = 0 to NoVertices
26.      write ColorVector[line]
27.color := ColorVector[0]
28. for line = 0 to NoVertices
29.      if color ¡ ColorVector[line] then
30.         color := ColorVector[line]
31 write color
32. return 0
```

Figure 1: Chromatic Number

5 Conclusions

Working on this project was a really unique experience for me, since it was truly a challenge, both in terms of research and understanding of the topic, as well as in the implementation part. I can say that i have learned a lot of new information regarding programming, as well as writing a .tex document, which was absolutely new for me.

References

[1] https://en.wikipedia.org/wiki/Graph$_c$*oloring*
https : *//www.geeksforgeeks.org/graph* − *coloring* − *applications/*

[2] LaTeXproject site, http://latex-project.org/