COSC-320 Project 3

Muniru Lamin
Maliak Green

May 12 2019

Map coloring, chromatic number

1 Abstract:

For our final project, we have decided to study the application of graph coloring, and how the concept solves real world problems, with the use of a graph and various of algorithms for coloring n vertices. We plan to implement the algorithms for coloring a graph G using standard C++ along with our currently running graph class. The graph class will contain updated methods for coloring the vertices and showing the status of the graph i.e. if two vertices conflict with one another (a conflict being an edge from u to v $\in G$).

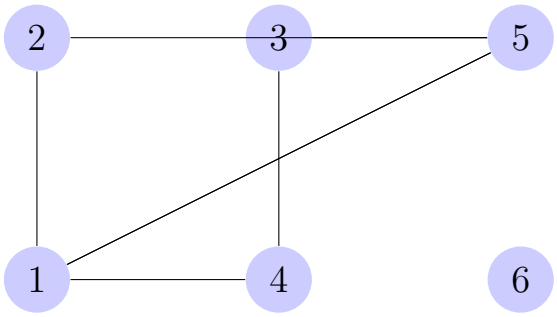
2 Introduction:

Given the various algorithms we have learned in lecture, we wanted to deeper explore the other benefits of graphs in real world problems. Map coloring seemed the most interesting since each color has a significant meaning to the vertices. It is an assignment of labels (traditionally called colors) to elements of a graph subject to certain constraints. Map coloring's origins can be traced back to the 1800s. Back then, mathematicians came up with the principle to help deal with problems related to planar graphs.

3 Topic Details:

The goal of our implementation is to show how map coloring can help efficiently adjust the contents of a graph that represent real-world data. By using C++ OOP principles, we were able to model a graph of various data sets. The first data set we modeled our graph after is a university student's course schedule. Below is a graph representing a student's course schedule. each vertex represents a class. Below are the labeling of each vertex in the graph.

1. COML 1170 10:00 - 10:50 am
2. COSC 3443 10:20 - 11:15 am
3. CHEM 2070 11:00 - 12:15 pm
4. BIOL 1440 12:30 - 1:45 pm
5. MATH 2331 9:30 - 10:40 am
6. ENGL 2001 2:00 - 2:50 pm



In order to assign a proper coloring to this graph of courses, we must first start at an arbitrary vertex and begin assigning colors. The method we used to assign colors to the graph is a greedy algorithm. The goal is to obtain a chromatic number by finding the minimum amount of colors we need to use. Below is our implementation in order to color a graph G.

Algorithm 1 Greedy-Coloring(G)

Create a vector color
make courses base-color: -1
  for $\&[index, v] : Courses$ **do**
    push -1 *onto the vector*

  end for
  Make front of vector 0: first color
  $v.front() = 0$
  Create a temporary Boolean array to store the available colors with size of vector
  for $clr = 0$ *to vector size* **do**
    $available[clr] = false$

  end for
  Assign colors to the rest of class
  $int\ temp = 1$
  for $\&[index, v] : Courses$ **do**
    Make adjacent vertices colors unavailable
    for $inti = 0$ *to* $Courses[u]$ *size* **do**
      if $vector[i] \neq -1$ **then**
        $available$[vector[i]] = $true$

      end if

    end for
    Find the first available color
    int color
    for $color = 0$ *to vector size* **do**
      if $available[color] == false$ **then**
        break

      end if

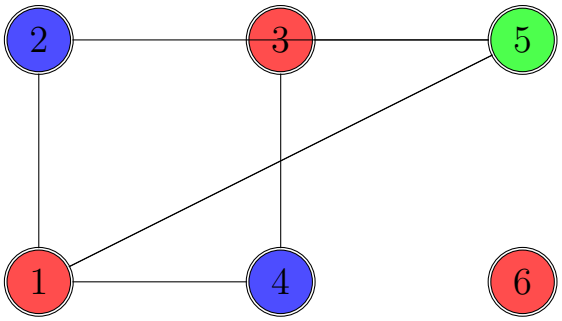
    end for
    Assign the found color to the next Course
    $vector[temp] = color$
    $temp++$
    Reset the rest of available colors to false for next course iteration
    for $inti = 0$ *to vector size* **do**
      if $vector[i] \neq -1$ **then**
        $available$[vector[i]] = $false$

      end if

    end for

  end for
  Print the results of the Algorithm
  int temp
  for $\&[index, v] : Courses$ **do**
    print index and $vector[temp]$

  end for



4 Conclusion:

After running the greedy color algorithm, The graph is modified in order to assign a proper scheduling for the student. Starting from node 1, we continue to visit adjacent nodes and give them the least value color. If all are taken, we make a new one. In the real-world application, the system computing this schedule will adjust the time slots so that the courses occur at different times. This sample schedule of a student reflects how programs use algorithms to determine the best fit for each individual. In conclusion, map coloring is a very effective technique in graph theory that is used to solve not only scheduling problems, but many other issues that different fields face.

5 Bibliography:

1. Graph theory, Part 2. (n. d.). Retrieved from
   $http://web.math.princeton.edu./math_alive/5/Notes2.pdf$

2. "Scheduling, Map Coloring, and Graph Coloring."
   $Http://www-Cgrl.cs.mcgill.ca$,
   $www-cgrl.cs.mcgill.ca/\ godfried/teaching/dm-reading-assignments/Map-Graph-Coloring.pdf$.

3. Verstraete, Jacques. "Map Coloring Theorms."
   $www.math.ucsd.edu$,
   $www.math.ucsd.edu/\ jverstra/mathcircle.pdf$.

4. "Sample Schedules." Sample Schedules from Cornell University. $https://as.cornell.edu/sample-schedules\>$.

5. Cormen, Thomas H. Introduction to Algorithms. MIT Press, 2009.

6 Appendix:

Cornell University students sample Data sets:

Student A:

ANTHR 1182 (FWS): Limits of the Human: Aliens, Apes and Artificial Intelligence, 3 cr.

ARAB 1201: Elementary Arabic, 4 cr.

ASTRO 1101: From New Worlds to Black Holes (PBS-AS), 3 cr.

PHIL 1100: Introduction to Philosophy (HB; KCM-AS), 3 cr.

PE 1412: Swedish Massage

Student B:

MATH 1120: Calculus II (MQR-AS), 4 cr.

SPAN 2090: Intermediate Spanish I, 4 cr.

ASTRO 1102: Our Solar System (PBS-AS), 3 cr.

GERST 1170 (FWS): Marx, Nietzsche, Freud, 3 cr.

PE 1332: Beginning Snowboarding