TIMETABLE CREATION USING GRAPH COLORING

FINAL REPORT

SAI KETAN KALLURI
MASTER OF COMPUTER SCIENCE
UNIVERSITY OF NORTH TEXAS

TARUNENDRA MALEPATI
MASTER OF COMPUTER SCIENCE
UNIVERSITY OF NORTH TEXAS

Project Objective:

The objective of our project is to build an algorithm that can build a timetable in the least possible resources. For this implementation graph colouring is used.

Introduction:

This project had an algorithm that could implement the timetable scheduling, there may be n number of constraints for a problem and an algorithm is built upon this algorithm. we are using a single constraint which is a subject cannot be taken with other at same time and that may be because of like prerequisites.

Our algorithm works fine with these conditions. The input for this is a CSV file and the output is the course number algorithm the semester in which they must be taken. The output is printed as a group wise list as we require the subjects to be taken according to the semester so that they don’t merge we divide them into semester.

Applications:

There are wide range of applications for this algorithm because scheduling is a basic aspect in any sector. The advantage with our algorithm is that we are using graph colouring and then we are scheduling. Problem case can have n number of constraints, even then our algorithm is good because we are converting into graph, so the edges are designed based on the constraints.

This can also be implemented for:

1. Designing of Flights routes. There can be various flights travelling in same route, but the problem arises to schedule the flights at this stage we can use the above algorithm.

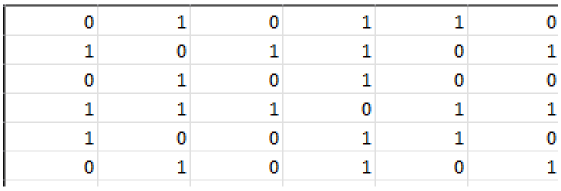
2. We can even complete sudoku using this algorithm. But for this we have 27 constraints for a 3*3 sudoku. So, with increasing the constraints we can even implement this sudoku.

3. We can use this for colouring maps. This can be implemented directly with the above algorithm with least changes.

Previous Works:

There are multiple algorithms that are instate of art for the current algorithms, but we stand out different because of the input for the algorithm. We are sending a CSV file which is of table style. Giving the data in this format makes users easy to create a schedule and this are what much needed in current scenarios.

And, if we see the below inputs, we can observe symmetry. So, reading inputs will be less complicated comparatively. We observed this, in our logic we implemented it. Because of this the time and storage are reduced.



0	1	0	1	1	0
1	0	1	1	0	1
0	1	0	1	0	0
1	1	1	0	1	1
1	0	0	1	1	0
0	1	0	1	0	1

Theory:

Graph colouring is a subset of graph labelling in graph theory; it involves the assignment of labels to elements of a graph, commonly referred to as colours according to specific constraints. In its most basic form, vertex colouring is a method of colouring the vertices of a graph so that no two neighbouring vertices are the same colour. Similarly, edge colouring lends a colour to each edge so that no two adjacent edges have the same colour, and face colouring assigns a colour to each face or region in a planar graph so that no two faces that share a border have the same colour.

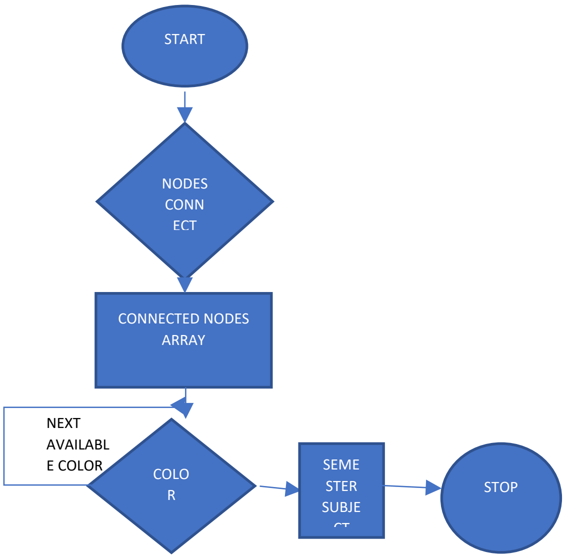
We have used edge colouring in this algorithm.

Algorithm:

The steps involved in the algorithm which we built are:

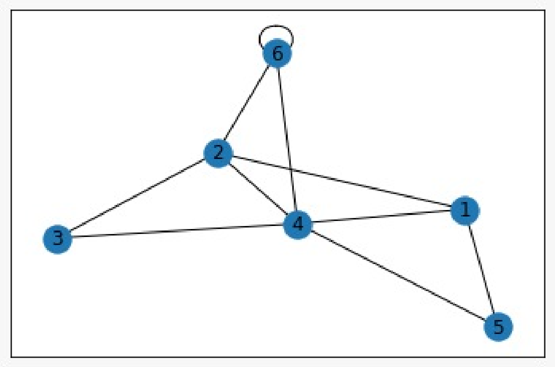
1) The input dataset must read, and it must be converted into graph. This is done through reading the CSV file, first it checks if the cell has a value if it possesses a values then that makes a edge and after all edges are formed the graph is built.
2) The graph built must be coloured and this is achieved by a simple logic,
   a) First it starts with colour zero and checks for its availability.
   b) This is done by the nodes connected, that is it first checks if the nodes are connected are having this colour if not the colour is assigned.
   c) Else next colour is checked and then repeated until a colour is assigned.
3) Next the data is printed according to the semester-wise and the data is printed.

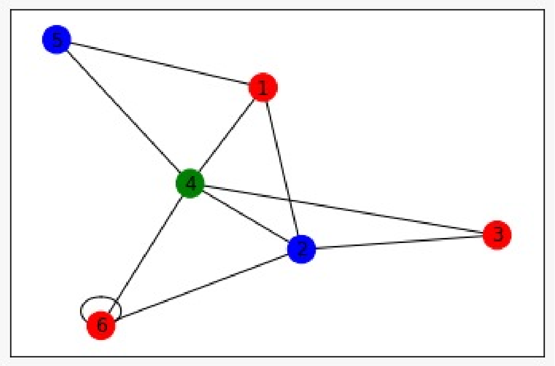
Flow Chart:



Dataset:

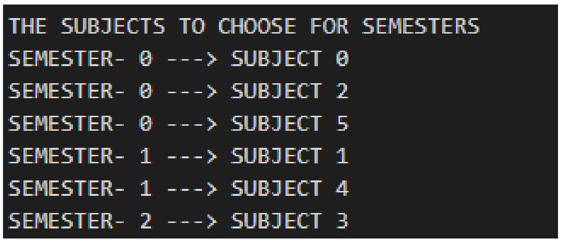
The dataset which we used for our project was a simple CSV file which has 6 subjects. We could build a very large dataset and have performed experiments on that, but we did not do that because we again must verify the graph plotted. But with these experiments we can assure that the algorithm is accurate enough to use in practical cases and is also effective.





Experimental Results:

From the results, the following observations are done the output is as follows



```
THE SUBJECTS TO CHOOSE FOR SEMESTERS
SEMESTER- 0 ---> SUBJECT 0
SEMESTER- 0 ---> SUBJECT 2
SEMESTER- 0 ---> SUBJECT 5
SEMESTER- 1 ---> SUBJECT 1
SEMESTER- 1 ---> SUBJECT 4
SEMESTER- 2 ---> SUBJECT 3
```

That means in first semester we can take subject 0, 2 &5.

We are in next semester we can take 1 & 4 subjects.

And in last semester we can take subject 3.

Based on the graph formed as shown below,

Analysis:

From the inputs and algorithm, we observed the time complexity for the above example is 3.017 seconds.

Time complexity for

Graph colouring is n(n+1)/2

And for reading the inputs and adding the edges is n*n as they are in csv file all the data is read at once and then it is split into a 2d array.

We observed that the time complexity has increased because of the input format, i.e., reading the data from CSV file. But will be of much help than the drawback in the real-world cases.

Conclusion:

We conclude by saying that the algorithm which we built is best suited for a any case of scheduling and this has less implementations in terms of time and storage. With the implementation of the Csv file, we are a step ahead of the current implementations and this could save a lot of time and energy. This eases the use of algorithm and could be even done by a naïve user.

References:

[1] https://en.wikipedia.org/wiki/Graph_coloring

[2] https://www.interviewbit.com/tutorial/graph-coloring-algorithm-using-backtracking/

GITHUB LINK:

https://github.com/m-tarunendra/SCHEDULING-USING-GRAPH-COLORING