Assignment 1

Sanjana S Prabhu

September 8, 2019

1 Introduction

This is an assignment to implement a traffic intersection problem from the textbook Aho,Hopcroft and Ullmann. We have 5 roads and a junction common to them. The problem is to design a traffic system between them keeping in mind that the traffic system is keep left.

2 Mathematical model of the problem

We can model this problem with a mathematical structure known as a graph. A graph consists of a set of points called vertices, and lines connecting the points, called edges. For the traffic intersection problem we can draw a graph whose vertices represent turns and whose edges connect pairs of vertices whose turns cannot be performed simultaneously. This graph can now be converted into a table of incompatible turns where the rows and columns represent the vertices and if there is an edge between any two vertices, the entry in the table is 1. The table is shown in Figure 1.

3 Algorithm to solve the problem

We can use the greedy algorithm to solve this problem. The approach is to scan all the vertices and assign them a same colour if they are compatible. By the end of the iterations we must end up with the fewest number of groups of vertices which represent the turns that can be made simultaneously.

4 The Pseudocode

The pseudocode is as follows:

```
procedure greedy ( var G: GRAPH; var newcolour: LIST );
{ greedy assigns to newcolour those vertices that may be
given the same colour }
var marker: boolean;
```

x	AB	AC	AD	BA	BC	BD	DC	DB	DA	EA	EB	EC	ED
AB	0	0	0	0	0	0	0	0	0	0	0	0	0
AC	0	0	0	1	0	1	0	1	0	0	1	0	0
AD	0	0	0	1	0	1	1	0	0	0	1	1	0
BA	0	1	1	0	0	0	0	1	0	0	1	0	0
BC	0	0	0	0	0	0	0	0	0	0	0	0	0
BD	0	1	1	0	0	0	1	0	0	0	0	1	0
DC	0	0	1	0	0	1	0	0	0	0	0	0	1
DB	0	1	0	1	0	0	0	0	0	0	0	1	1
DA	0	0	0	0	0	0	0	0	0	0	1	1	1
EA	0	0	0	0	0	0	0	0	0	0	0	0	0
EB	0	1	1	1	0	0	0	0	1	0	0	0	0
EC	0	0	1	0	0	1	0	1	1	0	0	0	0
ED	0	0	0	0	0	0	1	1	1	0	0	0	0

Figure 1: Table of incompatible turns

```
i,j: integer;
begin
newcolour := n;
i := first uncolored vertex in G;
while v<nodes do begin
found = false;
j = first vertex in newcolour;
while j<nodes do begin
if there is an edge between v and w in G then found = true;
j = next vertex in newcolour
end;
if found = false do begin
mark j colored;
add j to newclr
end;
j = next uncolored vertex in G
end
end; { greedy }
```

5 The C Code

The greedy algorithm is implemented in C as follows:

```
int greedy(int *array,int nodes,int *colour){
int i;
int j;
int m;
int newcolour=1;//indicates the initial colour,
the uncoloured vertices have colour zero
int marker=2;//true
for(i=0;i<nodes;i++){
if(colour[i]==0){//assign a new colour to the first
uncoloured vertex
colour[i]=newcolour;
newcolour++;//newcolour gets updated with the next
number representing the next colour
}
for(j=0;j<nodes;j++){
if(colour[j]==0){//check for next uncoloured vertex
for(m=0;m<nodes;m++){
if(colour[m]==newcolour-1){
if(*(array+j*nodes+m)==1){
/*check if previously coloured vertices of the colour in
question have an incompatibility with the next uncoloured vertex*/
marker=1;//incompatibility exists
}}}
if(marker==2){
colour[j]=newcolour-1;//if not assign the colour to this vertex}
else
marker=2;//else reinitialise the marker}}}
return newcolour-1;//return the total number of colours
}
```

6 Output

The output of the code when it is inputted a .dat file of the format corresponding to the example input file and which consists of the number of vertices, the names of the vertices and the table of incompatible turns, gives an output as shown in Figure 2. Therefore the following turns can be performed at the same time and so we require 4 traffic light colours in order to accommodate this intersection, keeping in mind that the traffic is keep left.

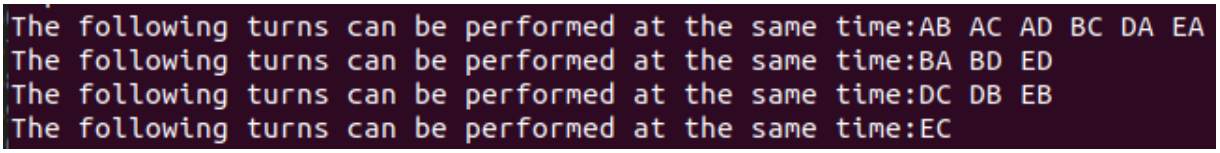

Figure 2: Output

7 Conclusions

We can make the following conclusions.

- The data structure used to represent the traffic system is called a graph. It associates a vertex with one or more vertices by edges. In this problem the turns are represented by vertices and their incompatibility is denoted by the presence of an edge in between them.
- The algorithm used to segregate these vertices into the fewest possible groups which can move together without causing collisions is known as the Greedy Algorithm. This algorithm is a paradigm that builds up a solution piece by piece, always choosing the next piece that offers the most obvious and immediate benefit. So the problems where choosing locally optimal also leads to global solution are best fit for this algorithm.
- Also the input file is a .dat file which is read by the 'fgetc' command from file handling functions.
- This code is written in a way such that atmost 99 nodes can be analaysed if the required input file is written and provided.