# EE4371 Assignment 1 - Traffic Intersection Problem

Rishhanth Maanav V EE16B036

11th September 2019

## 1 Introduction

Implementation of the greedy algorithm for the graph colouring problem applied to the traffic intersection problem as described in Aho, Hopcroft and Ullman. The code has been written in C++.

## 2 Pseudocode

The pseudocode for the greedy algorithm to colour nodes of a graph is presented here.

```
procedure greedy(var G: GRAPH; var newclr: LIST)
{
    bool found;
    int v,w;
    int newclr[];
    for v in uncoloured node of G{
        found := false;
        for w in newclr{
            if v,w are connected in G
                found := true;
        }
        if found == false{
            mark v coloured;
            add v to newclr
        }
    }
}
```

## 3 Description of Code

A class for graphs has been defined and used in the code. This facilitates adding nodes and edges, and accessing them. This is especially useful in our case since we need to maintain a list of already coloured coloured nodes. Every function used has been described in comments in the code.

## 4 Usage

The code takes in a command line argument as the name of the input file containing the graph description.

```
./a.out input.dat
```

The output is contains row-wise lists of nodes each row representing a group of nodes to be allowed simultaneously.

## 5 Sample Output

For the given input file, the output of the greedy algorithm is,

```
AB AC AD BC
BA BD
```