

# EE4371: Assignment 1

August 12, 2016

1. Read Chapter 1 of Aho, Hopcroft and Ullman
2. Read Chapters 1 & 2 in Tanenbaum for Friday's lecture
3. Create static arrays to hold upto 30 nodes. What properties of nodes are required? Create the corresponding arrays.
4. Create a static incidence matrix (see page 18 of textbook) of size upto 30 by 30.
5. Add other data structures and variables as required to implement the function greedy defined as follows

```
int greedy( ); // no args since static variables
// greedy assigns to newcolor a set of nodes of G that
// are as yet uncoloured and may be given the same
// colour
// greedy should print out all nodes that have newcolor
// or are "free turns" for this colour.
```

6. Write code to implement greedy
7. Write the main program that will call greedy in a loop till the entire problem is solved. The input to the program is a file containing the following:

**line 1** The number of nodes

**line 2** to line  $n + 1$ : rows defining the arrays created in part 3.

**line  $n + 2$  ff:** The incidence matrix like the one on page 18

The output should be like the table on page 20, with nodes names as input, and colours being numbers.

8. Estimate the time complexity of the algorithm you have implemented. What are the bottlenecks where you feel a speedup should be possible.

This code uses no structures and can surely be improved. But first implement this quick and dirty algorithm. The *input.dat* file contains the following:

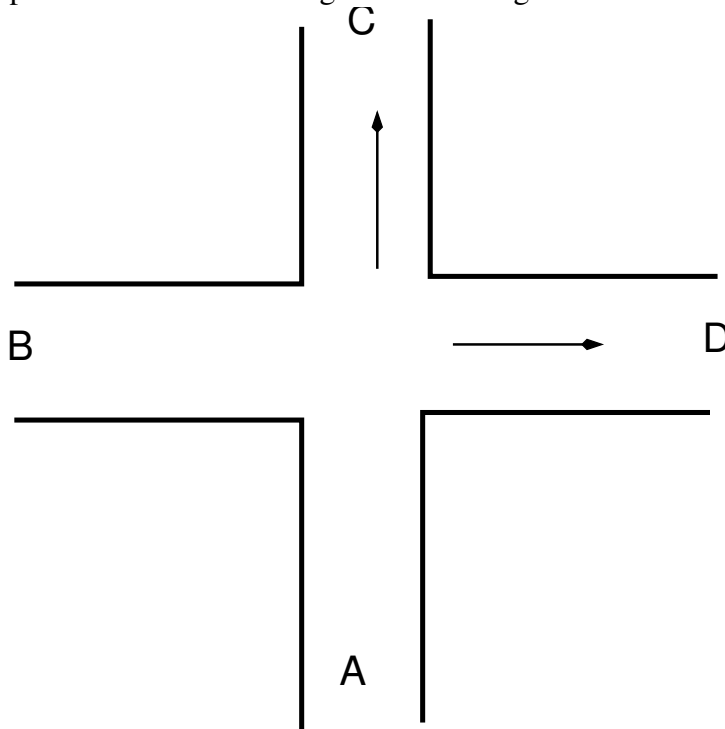
```
6
AB
AC
AD
BA
BC
BD
```

```

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

```

It represents the traffic through the following intersection with traffic keeping to the left (as in India)



**Note:** Remember to add traffic flows that may have another number, but may be consistent with the current flow. For example, AB and BC are permitted under all circumstances.

Upload the code to moodle. Please do not copy - I will check.