ELL781: Assignment 1

Submission deadline: **18 September, 23:59**

Maximum Marks: 10

1 Problem statement

The problem statement for this assignment is a generalised version of the first question of Minor I. Consider a cricket tournament being held involving n teams. Each pair of teams is supposed to play twice, once at each of teams' home grounds. Assuming that each team can play only one game every 2 days, find a schedule for the tournament such that it is completed in the minimum number of days.

2 Procedure

Here are the steps you should follow for this assignment:

1. Show how the problem can be transformed into a graph colouring problem (the nodes of the graph will correspond to games to be played, and the links should be such that in a valid colouring of the graph, each colour will represent games that can be scheduled together). Draw the graph for a simple example (say for $n = 3$) to illustrate the transformation. **[1]**

2. A greedy algorithm for the graph colouring problem has already been discussed in class, and this is what you need to implement. That algorithm made use of two abstract data types (ADTs): `GRAPH` and `LIST` (see Figure 1.8 of the AHU book). You need to implement these two ADTs as actual data structures (you are NOT permitted to use existing implementations or libraries for these). Describe a plan for doing so, in the form of pseudocode. Remember, to fully describe a data structure, you need to specify what your *cells* will consist of, what the *aggregation mechanism* over these cells will be, and what associated operations or *functions* will be available to manipulate the data structure. Please show all of these clearly for your data structures corresponding to both ADTs. **[3]**

3. Now, implement your data structures as actual code, and then use them to implement the greedy graph colouring algorithm to solve the given problem. All your code should be well-structured and properly commented, such that it can be easily understood by another person. You may use a programming language of your choice; one of C/C++/Java is highly recommended. Logically, you should aim to structure your code into 4 components/modules:

   - An implementation of a data structure which implements the `LIST` ADT.
   - An implementation of a data structure which implements the `GRAPH` ADT.
   - An impelentation of the greedy graph colouring algorithm, which makes use of the above two data structures.
   - A top-level program, which takes as input the problem parameter n, constructs the corresponding graph colouring problem, solves it using the above, and returns the final result in a given output format, which is specified below.

   [6]

2.1 Output format

Your top-level program should return a schedule for the tournament. If the schedule is over m days, the number of lines in the output should be m. Each line should contain a list of pairs of teams (each team indicated by an integer from 1 to n), with the first team being the home team and the second team being the away team. If no games are scheduled on a given day, the corresponding line should be blank. For example, if the program is called with $n = 4$, one possible output could be as follows:

```
(1 2); (3 4)

(1 3); (2 4)

(1 4); (2 3)

(2 1); (4 3)

(3 1); (4 2)

(4 1); (3 2)
```

3 What to submit

- For items 1 and 2 above, you need to submit a written report, describing your process of going from problem to program, including an illustrative example and the required pseudocode for your data structures. This report should be submitted as a PDF document. If you wish to include some scans of hand-drawn or hand-written portions, that is fine.
- For item 3 above, you need to submit all your code, properly documented. Please package everything into a single zip/tar/rar file. You should also include a README file which explains to the user exactly how they should use your code to generate a schedule for a given tournament.

Submission will be via Moodle (`http://moodle.iitd.ac.in/`); the exact submission procedure will be announced there in due course. The submission deadline for this assignment will be **18 September, 23:59**. Any late submissions will be penalised, and may not be evaluated at all, depending on the extent of delay.

4 Collaboration policy

You are free to discuss any aspect of the assignment with others; however, both your report and your code must be entirely written by you, without any copying from anywhere. We will be using plagiarism-detection tools to ensure that this is the case, and any violations will lead to the loss of all marks for the assignment, plus further disciplinary action depending on the severity of the offence.