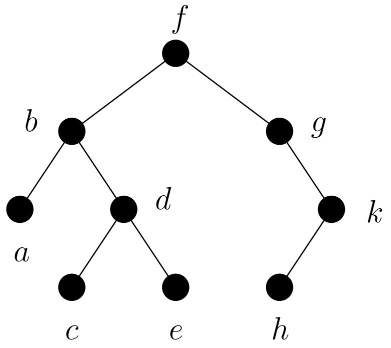
COMP108 Algorithmic Foundations
Tutorial 9

w/c 24th April 2017

Tutorial participation contributes to 5% of overall marks. For this tutorial, make sure you have scanned your ID card.

1. [**Do this before tutorial**] Consider the following binary tree T rooted as f. Give the order of traversal of preorder, inorder, and postorder traversal of the tree.



2. [**Puzzle**] Form groups of two persons with your fellow classmates to play the following game.

   There are 26 coins on the table. Two players take turns removing 1, 2, 3 or 4 coins. The winner is the player who removes the last coin. Design a winning strategy for the player making the first move.

   If there are 25 coins on the table, does the player making the first move have a winning strategy?

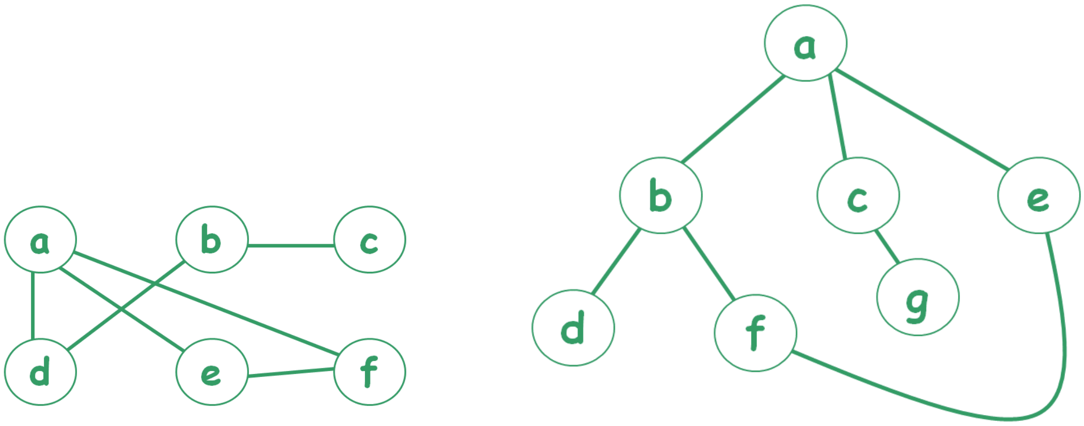
   What is the relationship between the initial number of coins and which player having a winning strategy?

3. **[Do this during tutorial] Programming on BFS/DFS**

Download three java files **GraphApp.java**, **Graph.java** and **Vertex.java** and two input files **graph1.txt** and **graph2.txt** from the tutorial page
`http://www.csc.liv.ac.uk/~pwong/teaching/comp108/201617/tutorial.html`
(Use right mouse click to save the files.)

The two files refer to the graphs in Figure 1.

You can refer to the lecture notes for the pseudo codes.
`http://www.csc.liv.ac.uk/~pwong/teaching/comp108/201617/notes.html` (Graph Theory)



(a) graph1.txt

(b) graph2.txt

Figure 1: The two graphs.

(a) Compile and run the program; then enter options 1 and 3 to generate some new graph and see the adjacency matrix.

(b) Try option 2 to input a new graph. Copy the content of one of the .txt file as the adjacency matrix. Note that graph1.txt contains 6 vertices and graph2.txt 7 vertices.

(c) Try options 4 and 5 and note that these functions are NOT working yet.

(d) Fill in the program Graph.java the methods **bfs( )** for breadth first search, **dfs( )** for depth first search.

Remember to read the comments in the methods. Also refer to the pseudo code on slides #45 and #65 of the lecture notes on Graph Theory.

(e) Checking the answer:

- For graph1.txt, bfs gives **a d e f b c** and dfs gives **a d b c e f**.
- For graph2.txt, bfs gives **a b c e d f g** and dfs gives **a b d f e c g**.