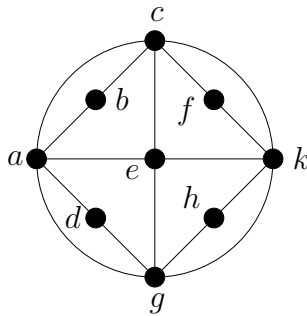


COMP108 Algorithmic Foundations — Tutorial 8

w/c 27th March 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 undirected graph G below.



- (a) List all the vertices adjacent to vertex k . _____
- (b) What is the degree of the vertex d ? _____
- (c) What is the degree of the graph G (i.e., maximum degree of the vertices)? _____
- (d) Give the adjacency matrix of the graph G .
- (e) State the conditions for a graph to contain an Euler circuit.
- (f) Does G contain an Euler circuit? If yes, write down the sequence of the vertices in one of these Euler circuits; if no, explain why and suggest the minimum number of edges needed to add to the graph so that an Euler circuit exists.

2. [Do this during tutorial] Programming on Merge Sort

Download two java files MSortApp.java and MSort.java from the tutorial page <http://www.csc.liv.ac.uk/~pwong/teaching/comp108/201617/tutorial.html> (Use right mouse click to save the files.)

You can refer to the lecture notes (divide-and-conquer) for the pseudo codes. <http://www.csc.liv.ac.uk/~pwong/teaching/comp108/201617/notes.html> (Divide and Conquer Method)

- (a) Compile and run the program; then enter some numbers, one per line, followed by `-1` to terminate the input. Try the option to sort the numbers using the merge sort algorithm. Note that these two function is NOT working yet.
- (b) Check the program MSort.java to read the method `copy()` to see what they do.
- (c) In the program MSort.java read the method `msort()` which invokes the recursive merge sort algorithm `rec_msort()`. Fill in the method `rec_msort()` to sort the numbers in ascending order using the **merge sort** algorithm. You will also need to fill in the method `merge()`. You can make use of the `copy()` method if necessary.

Remember to read the comments in the methods.

You can refer to the pseudo code on slides #25 and #28 of the lecture notes on Divide and Conquer.

Test cases:

- i. **10, 30, 20, 40, 50**
- ii. **50, 30, 10, 40, 20**
- iii. **50, 40, 30, 20, 10**
- iv. **40, 20, 60, -30, -40, 10, -5, -50**

3. [**Puzzle**] Suppose there are 10 people in a room. Each person shakes hands with some other people in the room. Prove that the number of people having an odd number of handshakes must be even. *Hint: relate this to a graph and the degree of vertices.*