Assignment 4

Due date: 10:00am, Wednesday, the 14th of September 2016.
This assignment is worth of 5% of the total course mark.

General Instructions

You may do this assignment as a **team of two students** if you both are undergraduate students. Individual work of an undergraduate student is also allowed and will be treated in the same way as a team work. Team members may be from different tutorial groups. You have to do it **individually** if you are a postgraduate student.

Submissions have to include coversheet including names, student ids, and your tutorial groups such that submissions can get marked.

Hand in your solutions to the box "ADSA" on level 4, Ingkarni Wardli (close to reception) **by the deadline**. **No late submissions will be accepted.**

Exercise 1 *Skip Lists (15 points for 1.1 + 15 points for 1.2)*

1. Recall from lectures that the height of a list at the insertion of an element into a skip-list is h where h is the number of flips of a coin that it takes until a head is achieved. Given the information above, draw the development of a skip-list resulting from the insertion of the elements:

   $$22, 9, 15, 8, 65, 86, 28, 11$$

   assume that the corresponding sequence of coin flips is:

   $$HTTHHTHTHTTTHTHHTTHHHTTTTHTH$$

   where H stands for heads and T stands for tails. Show the skip-list after each insertion.

2. Skip-lists have unbounded worst-case behaviour if the coin-flip sequence continually produces tails. One way to avoid worst case behaviour is to limit the height of the skip-list to a maximum of $logn$ where n is the current length of the skip-list. A disadvantage of this height-limiting approach is the need to reallocate the skip-list every time it doubles size to account for the fact that n is larger. Perform a short analysis to demonstrate the worst case cost of performing these reallocations and show whether these costs affect the asymptotic cost of insertion into skip-lists. One way to avoid these reallocation costs would be to set the maximum height to some fixed h. Explain the potential costs of such a fixed height limit?

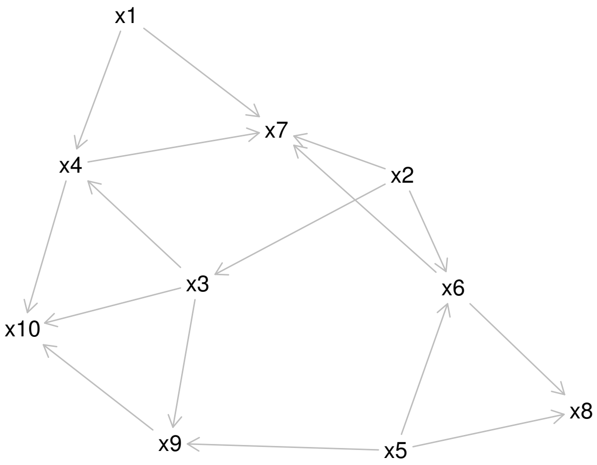
Exercise 2 *Introduction to Graphs (10 points)*

Name and briefly describe five graph applications not mentioned in lectures.

Exercise 3 *BFS and DFS (30 points for 3.1 + 30 points for 3.2)*

1. Topological sort of a directed graph is a linear ordering of its vertices such that for every directed edge (u, v) from vertex u to vertex v, u comes before v in the ordering. For instance, the vertices of the graph may represent tasks to be performed, and the edges may represent constraints that one task must be performed before another; in this application, a topological ordering is just a valid sequence for the tasks.

   - Which of the graphs search algorithms (BFS or DFS) is/are suitable for finding the topological order of the nodes? Explain your answer.
   - Write a program to find all topological answers for the graph shown below.



2. a bipartite graph (or bigraph) is a graph whose vertices can be divided into two disjoint sets U and V (that is, U and V are each independent sets) such that every edge connects a vertex in U to one in V. Vertex sets U and V are usually called the parts of the graph.

   To test the bipartiteness of a given graph $G = (V, U)$, the two sets U and V may be thought of as a colouring of the graph with two colors: if one colors all nodes in U blue, and all nodes in V red, each edge has endpoints of differing colors, as is required in the graph colouring problem. In contrast, such a colouring is impossible in the case of a non-bipartite graph, such as a triangle: after one node is coloured blue and another red, the third vertex of the triangle is connected to vertices of both colors, preventing it from being assigned either color. You can find more information about bipartiteness testing in the following URL

   https://www.cs.umd.edu/class/sum2005/cmsc451/bipartite.pdf

   Can the DFS algorithm be used to check the bipartiteness of a graph? how? Write a program that performs this check.

End of Questions