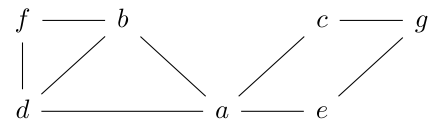
Exercises Problem Set 4

Question 1. Consider the following undirected graph G:



a. (Levitin, exercise 3.5.1.b)
Complete a DFS traversal through G, starting in node a. Use alphabetic order as a tie-breaker. Additionally, construct the corresponding DFS tree. Make sure to indicate in which order the nodes are being visited for the first time (added to the stack) and in which order they are completely handled (removed from the stack).

b. (Levitin, exercise 3.5.4)
Complete a BFS traversal through G, starting in node a. Use alphabetic order as a tie-breaker. Additionally, construct the corresponding BFS tree. Indicate in which order the nodes are being visited for the first time.

Question 2. (Levitin, exercise 3.5.6)
a. Adjust the pseudocode for BFS[1] over an undirected graph in such a way, that the algorithm checks whether a graph is acyclic. If this is the case, the algorithm has to return **True**. If this is not the case (i.e., the graph has at least one cycle), the algorithm has to return **False**.

b. Same as **a.** but for DFS with the added constraint that for any cycle your algorithm print outs the nodes of the cycle, in their order of occurrence.
c. Note that we can use either DFS or BFS to detect cycles in an undirected graph. Is one always preferable over the other? If so, which of the two faster, and why is that the case? If not, give an example of a graph in which BFS is preferable, and an example of a graph in which DFS is preferable.

Question 3. Design a variant of DFS to check whether (i) an given undirected graph is connected and, (ii) to provide a marking to indicate which connected component[2] a node belongs to. Your algorithm should add a marking to each node and return **True** if the graph is connected and **False** otherwise.

Question 4. We consider the game tic-tac-toe, played by cross (X) and circle (O). Suppose that cross starts.

a. How would you define a state? Draw part of the state-space.

b. Give and motivate a reasonable upper bound on the total number of states.

c. Consider an expanded version of tic-tac-toe, with a 9×9-grid where the goal is 9 tokens of the same kind in a line. Which number is larger: the potential number of states in this expanded tic-tac-toe, or the number of atoms in the observable universe, estimated to be around 10^{80}? Motivate your answer.

Question 5. *Alternating Disks* (Levitin 3.1.14)
You have a row of $2n$ disks of two colours, n dark and n light. They are placed in an alternating fashion: dark, light, dark etc. You want to get all the dark disks to the right-hand end (and all the light disks to the left-hand end). The only moves you are allowed to do is to swap two neighbouring disks.

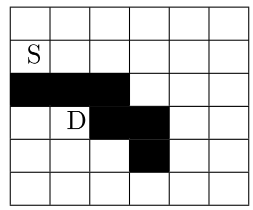
○●○●○●○ ●	$\longrightarrow$	○●○●○○●●
Starting position		After one move

a. Design a pseudocode algorithm for solving this puzzle, for any n.

[1] For the algorithm, see the slides or Levitin 3.5
[2] For a graph $G = (V, E)$, a connected component is a subgraph $H \subset G$ such that there is a path from every node in H to every other node.

b. What is the time complexity of your algorithm?

Question 6. *2D Pathfinding*
Consider the following 6x6 board:



You can move in one step to each adjacent square (horizontal, vertical and diagonal). The black squares are not accessible.

a. Execute a BFS on this board, starting at square S. Note for each (non-shaded) square what the resulting distance is from S.

b. Determine, based on the answer of **a.**, all shortest paths from S to D. How many of these shortest paths are there?

c.* Considering the general case where black squares occur randomly, would it be possible to use a DFS algortihm to obtain the same results? Explain your reasoning.