1. **Graphs and their representation, DFS, BFS.** The first question refers to the following directed graph:

![Directed Graph](image)

Answer the following questions:

(a) Give the degrees of the vertices A, B, and E.
(b) What are the direct neighbors of vertex B?
(c) Are there any loops in the graph?
(d) Are there any cycles in the graph? What about simple cycles?
(e) Find a trail from vertex A to vertex E.
(f) Find a path from vertex A to vertex E that is no trail.
(g) Is this graph a simple directed graph? Explain why or why not.
(h) Does this graph have a source and/or a sink? Explain why or why not.
(i) Is this graph (strongly or weakly) connected or disconnected? Explain your answer.
(j) Give the adjacency matrix representation of the graph.
(k) Give the adjacency list representation of the graph.
(l) Let A be the starting node. Traverse the graph with *Depth-First-Search (DFS)* and give each node a number when it is detected (for example, A gets number 1, the next node that is detected gets number 2 and so on). Always break ties in alphabetical order (for example, if you must choose between B and C, choose B).
(m) Do the same as in the question before, but now with *Breadth-First-Search (BFS)*.
2. Graphs and their representation. The second question refers to the following undirected graph:

![Graph Diagram]

Answer the following questions:

(a) Give the degrees of the vertices A, B, and E.
(b) What are the direct neighbors of vertex B?
(c) Are there any loops in the graph?
(d) Are there any cycles in the graph? What about simple cycles?
(e) Find a trail from vertex A to vertex E.
(f) Find a path from vertex A to vertex E that is no trail.
(g) Is this graph a simple undirected graph? Explain why or why not.
(h) Does this graph have a source and/or a sink? Explain why or why not.
(i) Is this graph (strongly or weakly) connected or disconnected? Explain your answer.
(j) Give the adjacency matrix representation of the graph.
(k) Give the adjacency list representation of the graph.

3. Trees. Draw a full binary tree of height 3. Number the nodes alphabetically starting with A in the root and then level for level from the left to the right in increasing order.

(a) What is the parent of node L?
(b) What are the children of node C?
(c) Is the tree balanced? Explain your answer.
(d) Name the leaves of three. How many leaves does this tree have? Can you generalize your answer?
(e) How many internal vertices does the tree have? Can you generalize your answer?

4. Trees. For each of the following statements, either prove it’s true or provide a counter example.

(a) Every tree with n nodes has exactly n – 1 edges.
(b) Every graph with exactly n – 1 edges is a tree.

5. DFS. Please use exactly the algorithm of the lecture.

(a) Prove the correctness of DFS.
(b) Perform a time analysis for DFS.

6. Euler and Hamilton Tours.

(a) Describe briefly what the problem of the 7 Bridges of Königsberg is.
(b) What is an Euler tour? What is an Euler Cycle? What is a Hamilton Tour? What is a Hamilton Cycle? Is it (from a computationally point of view) harder to compute Hamilton Tours than to compute Euler Tours? Explain why or why not.
(c) Right or false? If a graph G has 3 or more even vertices ↔ G has no Euler tour.
7. Minimum Spanning Tree.

(a) Perform the algorithm of **Prim** on the following graph. Start node is node $E$.

(b) Remember that $n = |V|$ and $m = |E|$. **Prim**’s algorithm needs running time $O(n \log n + m)$ if the priority queue (that is needed to keep track of the nodes that are not yet included in the minimum spanning tree) is implemented with e.g. a Fibonacci-Heap. **Kruskal**’s algorithm needs running time $O(m \log m)$ Which algorithm is better, i.e. needs less time? Formulate your answer carefully and take into consideration different possible numbers $m$ of edges in comparison to the number $n$ of nodes.

8. Modelling problems with graphs: Topological Sorting. Imagine you would have to perform in a project jobs $A, B, C, D, E, F$ with the following time dependencies:

- $B$ must be finished before you can start with $D, E, F$.
- $C$ must be finished before you can start with $B, E$.
- $D$ must be finished before you can start with $A, E$.
- $E$ must be finished before you can start with $A$.
- $F$ must be finished before you can start with $D$.

(a) Draw the corresponding graph to that problem (Why do you have to draw a directed graph?).

(b) When you want to answer if all of the time dependencies can be fulfilled: what graph theoretic problem do you have to solve? Apply the corresponding algorithm to that problem. In what order will you have to perform the jobs?

9. Modelling problems with graphs: Directed Acyclic Graphs. A daily flight schedule is a list of all the flights taking place that day. In a daily flight schedule, each flight $F_i$ has an origin city $OC_i$, a destination city $DC_i$, a departure time $d_i$ and an arrival time $a_i > d_i$. This is an example of a daily flight schedule for August 21, 2016, listing flights as $F_i = (OC_i, DC_i, d_i, a_i)$:

- $F_1 = (Portland, Los Angeles, 7:00am, 9:00am)$
- $F_2 = (Portland, Seattle, 8:00am, 9:00am)$
- $F_3 = (Los Angeles, San Francisco, 8:00am, 9:30am)$
- $F_4 = (Seattle, Los Angeles, 9:30am, 11:30am)$
- $F_5 = (Los Angeles, San Francisco, 12:00pm, 1:00pm)$
- $F_6 = (San Francisco, Portland, 1:30pm, 3:00pm)$

(a) Describe how to construct a Directed Acyclic Graph (DAG) so that paths in the DAG represent possible sequences of connecting flights a person could take. What are the vertices, and when are two vertices connected with an edge?

(b) Why is your graph always a DAG?

(c) Draw the DAG you described for the given example of August 21, 2016.

(d) Use your DAG to help you determine the maximum number of connecting flights a person could take on August 21, 2016.