**Problem set 4.**

Graph representation, DFS.

1. Use the adjacency matrix below to determine the number of vertices and edges in the undirected graph? Determine the degrees of all vertices.

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \end{bmatrix}$$

2. (a) How do we know that the adjacency matrix below belongs to a directed graph?
   (b) Determine the in and out-degree of each vertex.

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

3. What is the degree of each vertex in a directed graph whose adjacency list is as follows: **a**: b, c; **b**: e, d; **c**: d; **d**: f; **e**: f; **f**:- ? What is the in-degree of the vertex $f$? How can this be determined from the adjacency list?

4. Run the DFS from vertex 1 in the directed graph below (given by the adjacency list) so that it follows the order given in the edge list when examining the neighbours. Also give the discovery and finishing numbers of the vertices.

   $$\mathbf{a}: c, d, e; \quad \mathbf{b}: a, e, f, g; \quad \mathbf{c}: b, d, f; \quad \mathbf{d}: e; \quad \mathbf{e}: a, c; \quad \mathbf{f}: -; \quad \mathbf{g}: e$$

   State which edges will be tree edges, forward-edges, back-edges and cross-edges.

5. Given an undirected graph $G$ and a vertex $s$, give an algorithm to find the vertices in the component containing $s$. If instead your graph was directed, what would your algorithm return?

---

6. We are given an inverse adjacency list of a directed graph where for each vertex we have a list of the edges entering the vertex. Give an algorithm with running time $O(n + m)$ that produces the outgoing edge list for each vertex from the given edge lists.

7. In a directed graph on six vertices, the order in which vertices are encountered in a run of DFS is $a, c, f, e, d, b$, while the finishing numbers are $a$: 6; $b$: 5; $c$: 4; $d$: 3; $e$: 2; $f$: 1. Is it possible that there is an edge in the graph (a) from $f$ to $e$? (b) from $d$ to $e$?

8. We are given a directed graph $G$ with its adjacency list. We are further given an array $C$ indexed by vertices which stores the color of each vertex as red or green. We are also given two red vertices $s$ and $t$ in the graph.
   (a) Give an algorithm with running time $O(n + m)$ that decides whether there is a path from $s$ to $t$ that passes only through red vertices.
   (b) Give an algorithm with running time $O(n(n + m))$ that decides whether there is a path from $s$ to $t$ that passes through at most one green vertex.

9. Given a directed graph $G$ with an adjacency matrix, we want to determine all vertices from which a given vertex $t$ is reachable by a directed path. Give an algorithm with running time $O(n^2)$ for this task.

---

10. We are given an undirected graph $G$ with its adjacency matrix. Give an algorithm that decides in $O(n^2)$ steps,
    (a) if there is a vertex in the graph with degree 2.
    (b) what is the largest degree in the graph.
    (c) what is the most frequent degree in the graph.

11. A novice driver wants to choose the best route to drive to work. The road network is given as an undirected graph where the vertices are the intersections and the edges are the roads. Also, some of the vertices are labelled as difficult intersections to navigate through. (Difficulty of navigation through an intersection does not depend on where he arrives from or where he leave to from it.)

    Give an algorithm with $O(n+m)$ running time, to determine the vertices he can drive to from his home at a given vertex such that his path will not have two consecutive vertices labelled as difficult. Assume that you are given the graph with an adjacency list.

12. Give an algorithm with running time $O(n+m)$ for Problem 8*b*.

13. We are given the adjacency list of a connected directed graph $G$ whose edges have an integer value weight from the set $1, 2, \dots, k$. Let the value of a path be the *maximum* of the weights of the edges on the path. Given two vertices of the graph, $x$ and $y$, give an algorithm with running time $O(m \log k)$ to determine the path from $x$ to $y$ with the smallest possible value.

14. We are given the adjacency list of a directed graph $G$. Each vertex of the graph is given a label (number) from 1 to 100, a label can occur more than once. Find (if there exists) a *colorful* path in the graph in which each label $1 \leq i \leq 100$ occurs exactly once. The running time of the algorithm should be $O(m+n)$.

15. We are given a path on $n$ vertices, where each vertex contains a positive integer. We want to select a subset of vertices that has no adjacent vertices and where the sum of the numbers is as large as possible. Give an algorithm with running time $O(n)$ for this task.

16. We are given the adjacency list of a tree on $n$ vertices, where each vertex has degree at most 3. Give an algorithm with running time $O(n)$ to find the length of the longest path (with most edges) in the tree.

## Final Answers

1. $5, 7$, degrees: $2, 4, 2, 3, 3$.
2. not a symmetric matrix, In-degrees: $0, 2, 1, 2, 2$, Out-degrees: $2, 1, 3, 1, 0$
3. $2, 2, 1, 1, 1, 0$, In-degree of $f$ is 2.