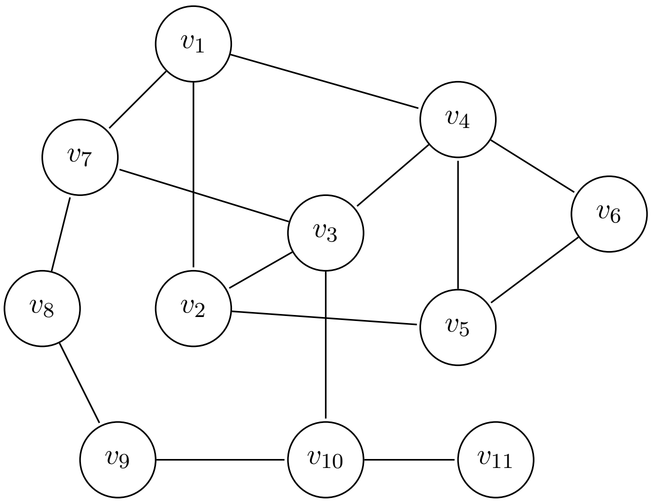
University of Freiburg
Dept. of Computer Science
Prof. Dr. F. Kuhn
M. Fuchs, G. Schmid

Algorithms and Datastructures
Summer Term 2024
Exercise Sheet 8

Due: Wednesday, June 19th, 2pm

Exercise 1: BFS *(5 Points)*

Given the following undirected graph G:



a) Provide G as an adjacency matrix. *(2 Points)*

b) Provide G as an adjacency list. *(2 Points)*

c) Perform a breadth-first search on G starting from node v_1. Write the order in which the nodes are marked (i.e., colored gray) in the algorithm. To obtain a deterministic result, always add the node with the smaller index to the FIFO-queue first, that is, v_i before v_j if $i < j$. *(3 Points)*

Exercise 2: DFS *(6 Points)*

We define 2 timestamps for each node (as in Slide 29):

- $t_{v,1}$: Time when node v is colored gray by the DFS search
- $t_{v,2}$: Time when node v is colored black by the DFS search

Additionally, consider the following *directed* graph $G = (V, E)$ given with

- $V = \{u_1, u_2, u_3, u_4, u_5\}$
- $E = \{(u_1, u_2), (u_1, u_3), (u_2, u_3), (u_3, u_4), (u_4, u_1), (u_5, u_1), (u_5, u_3), (u_5, u_4)\}$

a) Draw G. *(2 Points)*

b) Write the processing interval $[t_{v,1}, t_{v,2}]$ for each node in G. Similar to part 1c), if multiple nodes could be visited next by the depth-first search, always choose the one with the smallest index (and thus we also start with u_1). *(2 Points)*

c) For each edge, indicate whether it is a **Tree Edge**, **Backward Edge**, **Forward Edge**, or **Cross Edge**. *(2 Points)*

Exercise 3: Cycle search *(9 Points)*

a) How many edges m can an undirected connected graph with n nodes have at most? Justify your answer. *(2 Points)*

b) Show that every undirected connected graph which contains no cycle[1] has exactly $n-1$ edges (where n is the number of nodes of the graph). *(4 Points)*
Hint: You can prove this statement, for example, by induction on $n \geq 1$.

c) Given an undirected connected graph $G = (V, E)$ with $n = |V|$. Provide an algorithm that decides in $\mathcal{O}(n)$ time whether G contains a cycle or not. Specify explicitly in which data structure G should be given. *(3 Points)*

[1]A cycle is a path $v_1, \ldots, v_k \in V$ in a graph where there is also an edge between the start and the end node, i.e., $\{v_1, v_k\} \in E$.