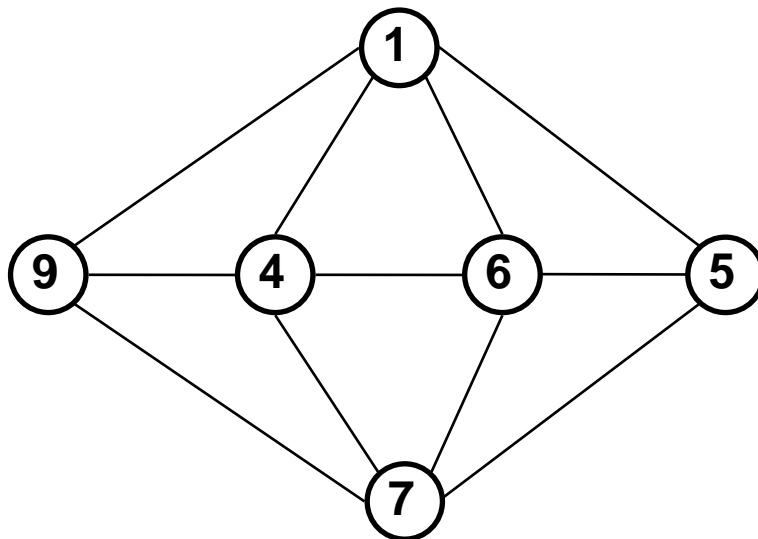


Übungsserie 11

Aufgabe 1: Graphen-Traversierung

Gegeben ist nachfolgender Graph:



Es soll mit dem **Start-Vertex 1** eine *Tiefensuche (Depth-First-Search DFS)* und eine *Breitensuche (Breadth-First-Search BFS)* durchgeführt werden und der resultierende *aufspannende Baum* entsprechend eingezeichnet werden.

Hinweise:

- `incidentEdges()` liefert die Kanten aufsteigend sortiert aufgrund der Inhalte der gegenüberliegenden Vertices (z.B. für Vertex 1 : 4, 5, 6, 9)
- die Listen bei BFS haben FIFO-Verhalten

Aufgabe 2: Implementation

Es soll die `search()` resp. `_search()`-Methode in der Klasse `DepthFirstSearch` implementieren werde.

Die Test-Applikation befinden sich in `DepthFirstSearchTest.java` resp. `depth_first_search_test.py`.

Modul ADS: Algorithmen & Datenstrukturen

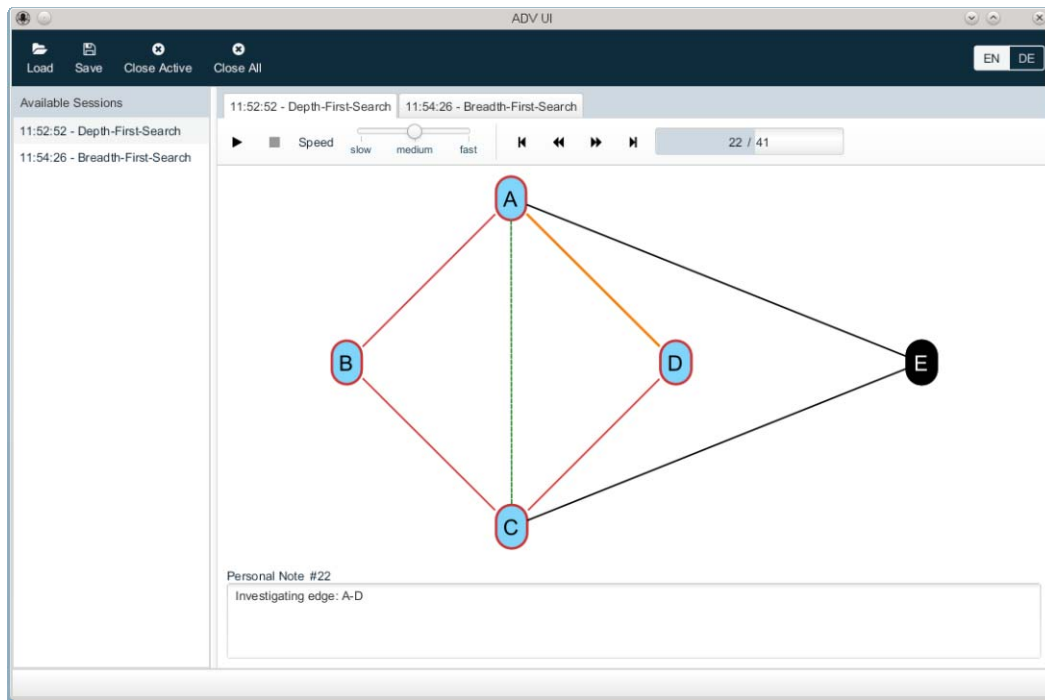
Für die Verwendung des "Algorithm & Data Structure Visualizer (ADV)" wird jeweils zu Beginn bei `main()` die entsprechenden Zeilen umkommentiert.

Der Setup für ADV ist sonst gleich wie bei der Übung 4.

Bei Verwendung von ADV mit *Python* ist NACH dem Starten des ADV-Visualisierungs-Server mit `start_ADV-UI` noch der Python-Adapter zu starten mit `start_PyAdapter`.

Die Datei `DepthFirstSearchTest.adv` ist eine *ADV-Session's* und kann mit dem *ADV-UI* abgespielt werden (\Rightarrow *Load*).

Ziel ist, dass das eigene Programm im *ADV-UI* eine gleiche Session erzeugt.



Testat

Eine Nachreichung für das Testat 1 von letzter Woche ist möglich bis:

- Di 13.05.24 07:00 Uhr

Die Abgabe erfolgt wie gehabt mittels Check-In wie in der letzten Woche.