

Übungsserie 13

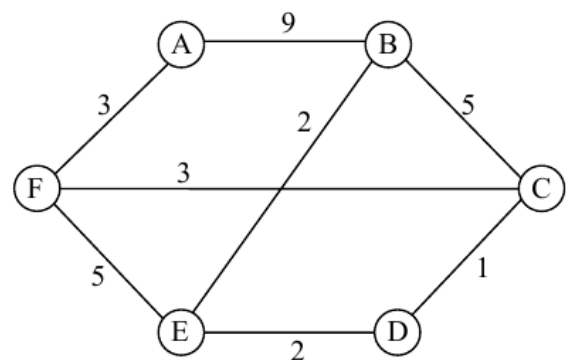
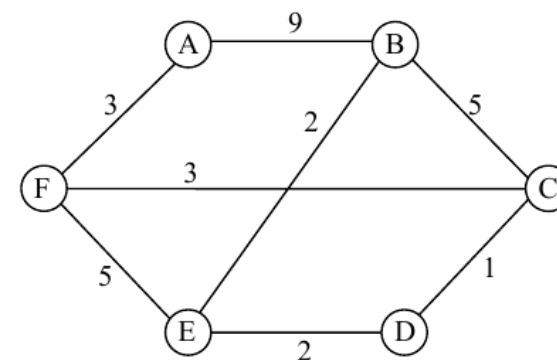
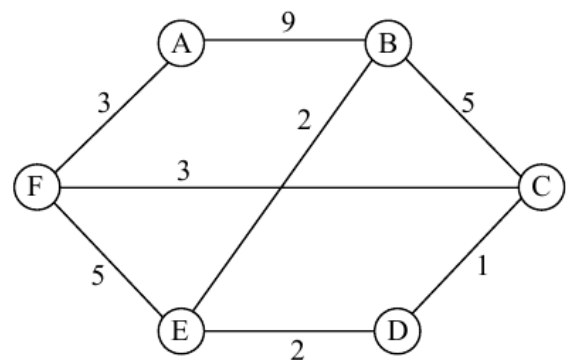
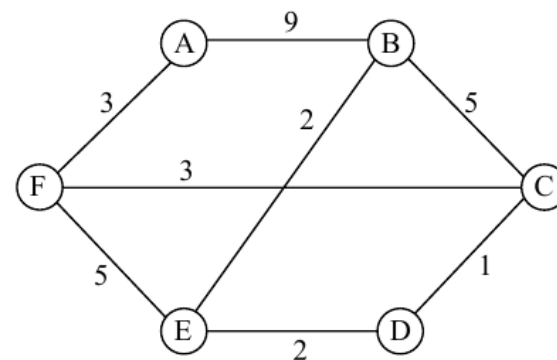
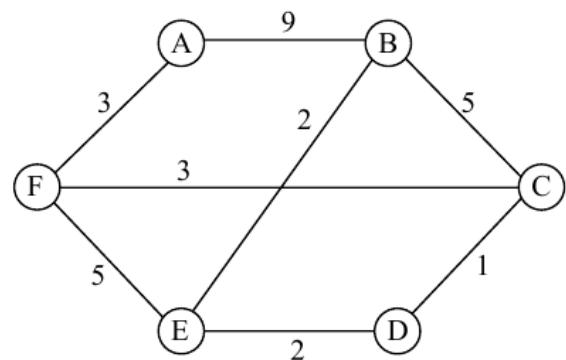
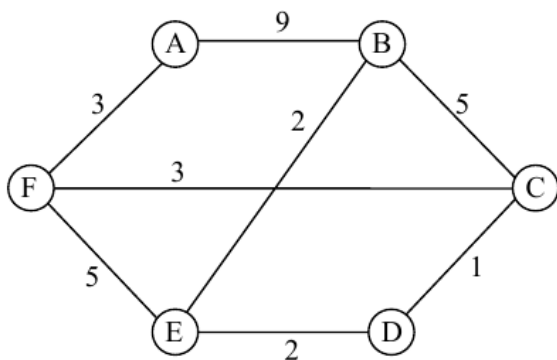
Aufgabe 1: Kürzeste Pfade nach Dijkstra

Gegeben ist nachfolgender Graph.

Berechnen Sie die kürzesten Pfade zu allen Vertices, ausgehend vom **Start-Vertex A**, nach *Dijkstra*.

Zeichnen Sie jeweils den Zustand des Algorithmus inkl. der aktualisierten Distanzen (nach der Kanten-Relaxation) und den Baum mit den kürzesten Pfaden nach jedem Hinzufügen eines Vertex in die *Wolke*.

Start-Vertex: **A**



Aufgabe 2: Kürzeste Pfade nach Dijkstra: Implementation

Es soll der *Dijkstra*-Algorithmus zur Bestimmung des *Shortest-Path-Tree* implementiert werden.

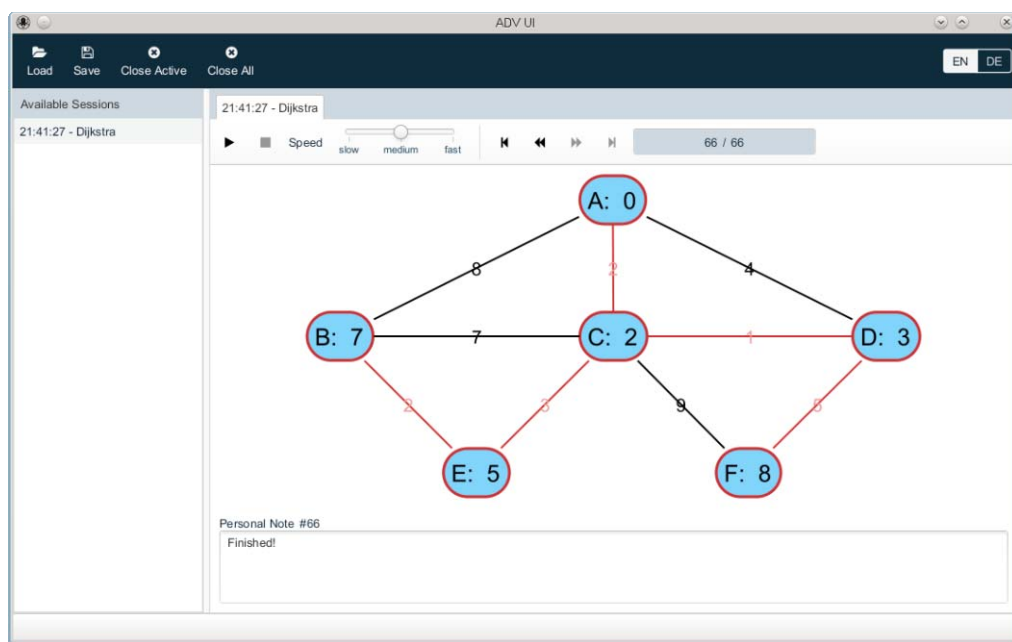
Dazu soll die `distances()`-Methode in der Klasse `Dijkstra` implementiert werden. Die Test-Applikation befindet sich in `DijkstraTest.java` resp. `dijkstra_test.py` (Beispiel aus Folien-Skript) .

Für die Verwendung des "*Algorithm & Data Structure Visualizer (ADV)*" wird zu Beginn der `main()`-Methode die entsprechenden Zeilen umkommentiert. Der Setup sonst gleich wie bei der Übung 4.

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

Die Datei *Dijkstra.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:



Hinweis:

- Für *Unendlich* (∞) wird jeweils verwendet:
 - Java: `Integer.MAX_VALUE`
 - Python: `sys.maxsize`

Testat

Eine Nachreichung für das Testat 2 von letzter Woche ist möglich bis:

- Mo 26.05.25 07:00 Uhr

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