

Übungsserie 12

Aufgabe 1: Gerichtete Tiefensuche

Ein gerichteter Graph $G = (\{1, \dots, 10\}, E)$ sei durch folgende Adjazenzliste gegeben:

E : 1 \rightarrow 8; 10
 2 \rightarrow 6
 3 \rightarrow 1; 6; 7
 4 \rightarrow 5; 10
 5 \rightarrow 10
 6 \rightarrow 7
 7 \rightarrow 2; 8
 8 \rightarrow 9
 9 \rightarrow 8
 10 \rightarrow 3

- Zeichnen Sie den angegebenen Graphen und stellen Sie ihn mithilfe einer Adjazenzmatrix dar.
- Wenden Sie den Tiefensuche-Algorithmus, startend bei Knoten mit Key = 1, auf den Graphen an und geben Sie zu jedem Knoten v seine DFS-Nummer (Start-Knoten: DFS-Nummer = 1) an.
Benutzen Sie dabei die in Graph G und in der Adjazenzliste E angegebenen Reihenfolgen.
- Geben Sie an, welche Kanten Baum-, Rück-, Vorwärts- oder Kreuzungskanten sind.

Aufgabe 2: Implementation gerichtete Tiefensuche

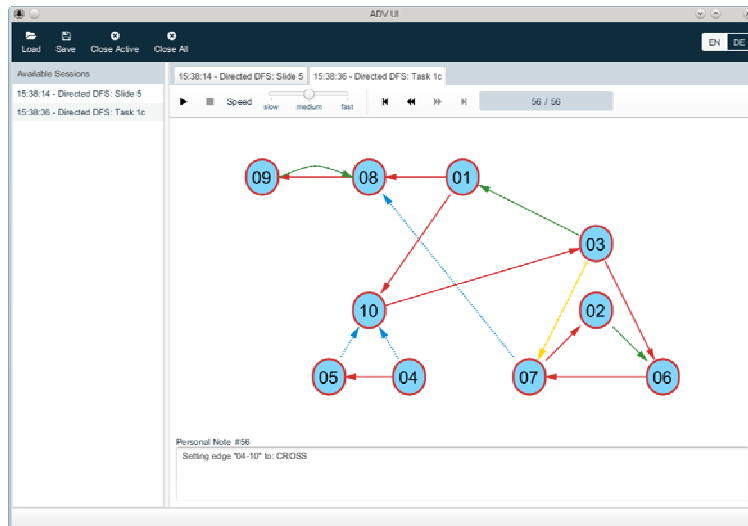
Es soll die gerichtete Tiefensuche implementiert werden.

Dazu soll die `search()`-Methode in der Klassen `DirectedDFS` implementiert werden. Die Test-Applikation befinden sich in `DirectedDFS_TestTask1c` (Beispiel von vorheriger Aufgabe 1c).

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

Bei Verwendung von ADV mit Python ist NACH `start_ADV-UI` noch `start_PyAdapter` zu starten

Die Datei `DirectedDFS_TestTask1c.adv` ist eine *ADV-Session* und kann mit dem *ADV-UI* abgespielt werden (\Rightarrow Load).
Ziel ist, dass das eigene Programm im *ADV-UI* eine gleiche Session erzeugt.



Testat

Diese Aufgabe 2 wird als Testat abgegeben.

Geprüft wird die korrekte Identifizierung der *DISCOVERY*- und *BACK*-Kanten (nicht der *FORWARD*- und *CROSS*-Kanten).

Abgabe

Die Abgabe erfolgt analog zum *Testat 1* bei *Übung 10*.

Hinweise:

- Den Check-In für dieses Testat im '*Exercise Repository Client*' mit *Exercise:Uebung12Testat2* machen (nicht mit *Uebung10Testat1*!).
- Eine VPN-Verbindung zur HSLU ist nötig.