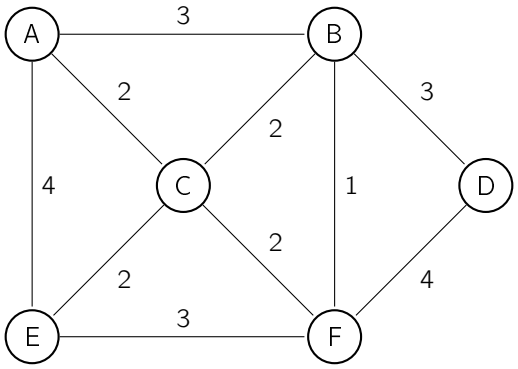
Prof. Dr. Leif Kobbelt
Stefan Dollase, Ira Fesefeldt, Alexandra Heuschling, Gregor Kobsik

Übung 10

Aufgabe 4 (Kruskals Algorithmus und Union-Find): 10 Punkte

Gegeben ist folgender Graph G:



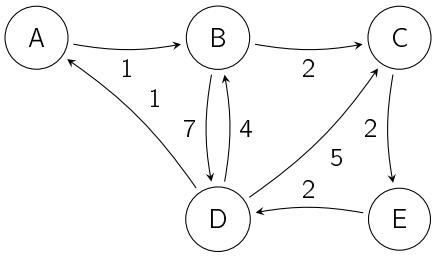
Führen Sie Kruskals Algorithmus auf den Graphen G aus, um einen Minimalen Spannbaum zu berechnen. Geben Sie dabei nach jeder Operation auf der Union-Find Struktur die entsprechende Operationen und den neuen Zustand der Union-Find Struktur an falls diese sich verändert hat. Dabei soll die Union-Operation bei **gleicher Höhe der Wurzeln immer die Wurzel des zweiten Parameters** als neue Wurzel wählen. Benutzen Sie hierfür sowohl Pfadkompression als auch Höhenbalancierung. Es ist nicht nötig, alle Zwischenschritte beim Ausführen der MakeSet Operationen anzugeben. Benutzen Sie die folgende Sortierung der Kanten:

$$(B,F),(A,C),(B,C),(C,E),(C,F),(A,B),(B,D),(E,F),(A,E),(D,F)$$

Geben Sie außerdem den resultierenden Minimalen Spannbaum an.

Aufgabe 5 (Floyd-Warshall Algorithmus): 10 Punkte

Betrachten Sie den folgenden Graphen:



Führen Sie den *Algorithmus von Floyd* auf diesem Graphen aus. Geben Sie dazu nach jedem Durchlauf der äußeren Schleife die aktuellen Entfernungen in einer Tabelle an. Die erste Tabelle enthält bereits die Adjazenzmatrix nach Bildung der reflexiven Hülle. Der Eintrag in der Zeile i und Spalte j ist also ∞, falls es keine Kante vom Knoten der Zeile i zu dem Knoten der Spalte j gibt, und sonst das Gewicht dieser Kante. Beachten Sie, dass in der reflexiven Hülle jeder Knoten eine Kante mit Gewicht 0 zu sich selbst hat.

①	A	B	C	D	E
A	0	1	∞	∞	∞
B	∞	0	2	7	∞
C	∞	∞	0	∞	2
D	1	4	5	0	∞
E	∞	∞	∞	2	0

②	A	B	C	D	E
A					
B					
C					
D					
E					

③	A	B	C	D	E
A					
B					
C					
D					
E					

④	A	B	C	D	E
A					
B					
C					
D					
E					

⑤	A	B	C	D	E
A					
B					
C					
D					
E					

⑥	A	B	C	D	E
A					
B					
C					
D					
E					

Aufgabe 6 (Programmierung in Python - Graphalgorithmen):

2 + 3 + 5 + 6 + 4 = 20 Punkte

Bearbeiten Sie die Python Programmieraufgaben. In dieser praktischen Aufgabe werden Sie sich mit Graphalgorithmen auseindersetzen. Diese Aufgabe dient dazu einige Konzepte der Vorlesung zu wiederholen und zu vertiefen. Zum Bearbeiten der Programmieraufgabe können Sie einfach den Anweisungen des Notebooks *blatt10-python.ipynb* folgen. Das Notebook steht in der .zip-Datei zum Übungsblatt im Lernraum zur Vergügung.
Ihre Implementierung soll immer nach dem # *YOUR CODE HERE* Statement kommen. Ändern Sie keine weiteren Zellen.
Laden Sie spätestens bis zur Deadline dieses Übungsblatts auch Ihre Lösung der Programmieraufgabe im Lernraum hoch. Die Lösung des Übungsblatts und die Lösung der Programmieraufgabe muss im Lernraum an derselben Stelle hochgeladen werden. Die Lösung des Übungsblatts muss dazu als .pdf-Datei hochgeladen werden. Die Lösung der Programmieraufgabe muss als .ipynb-Datei hochgeladen werden.

Übersicht der zu bearbeitenden Aufgaben:

a) Implementierung von Dijkstra
- initialize_single_source()
- relax()
- dijkstra()

b) Labyrinthe: Generieren und Lösen
- generate_maze()

- solve_maze()