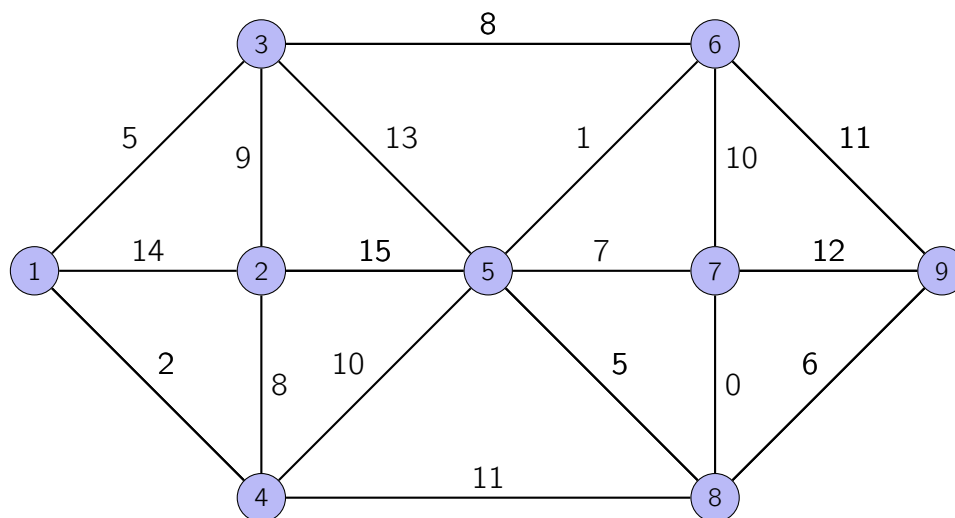


Hinweise:

- Die Übungsblätter sind in Gruppen von je 3 Studierenden der gleichen Kleingruppenübung zu bearbeiten.
- Die Lösungen müssen bis Montag, den 12. Juli um 11:00 Uhr in die Übungskasten eingeworfen werden.
- Namen und Matrikelnummern, sowie die Nummer der Übungsgruppe sind auf jedes Blatt zu schreiben.

Aufgabe 1 (Prim-Algorithmus):**(5+2+4+6+4 Punkte)**

- a) Bestimmen Sie mit dem Algorithmus von Prim einen minimalen Spannbaum auf dem nachfolgenden Graphen und geben Sie an, in welcher Reihenfolge Sie die Kanten in den Spannbaum aufnehmen. Starten Sie beim Knoten 1.



- b) Geben sei der folgende Algorithmus, der zu einem zusammenhängenden, gewichteten, ungerichteten Graphen $G = (V, E)$ einen MST berechnet.

```

1  mst(Graph G = (V, E) ){
2      // sort gibt eine absteigend sortierte Folge der übergebenen Menge zurück
3      E' = sort(E);
4      foreach (k in E')
5          if ( zusammenhaengend( (V, E'\{k}) ) ) // E'\{k} ist die Menge E' ohne Kante k
6              E' := E'\{k}
7      return (V, E')
8  }
```

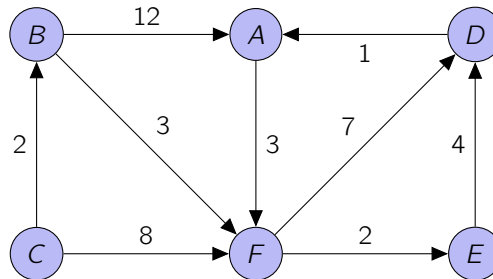
Geben Sie ein Verfahren aus der Vorlesung an, mit dem der Test auf Zusammenhang in Zeile 5 realisiert werden kann.

- c) Bestimmen Sie die Laufzeit des Algorithmus aus b) in Abhängigkeit von $n = |V|$ und $m = |E|$.
- d) Zeigen Sie, dass der obige Algorithmus einen Spannbaum berechnet, indem sie zeigen,
- dass das Ergebnis zusammenhängend ist und
 - dass das Ergebnis zyklfrei ist.
- e) Zeigen Sie, dass der obige Algorithmus einen minimalen Spannbaum berechnet.

Aufgabe 2 (Bellman-Ford-Algorithmus):

(6+6 Punkte)

- a) Bestimmen Sie, für den Startknoten C , die Längen der kürzesten Wege zu jedem Knoten im unten gegebenen Graphen G mit Hilfe des Bellman-Ford Algorithmus. Dabei sei die Reihenfolge der Knoten gegeben durch die alphabetische Sortierung (A, B, C, \dots). Geben Sie für jeden Knoten K die Kosten an, die dem Knoten während der Berechnung zugewiesen werden. Z.B. im Format $K : \infty, 10, 7, 6, 3$:



- b) Passen Sie die Implementierung von Bellman-Ford, die Sie in der Vorlesung kennengelernt haben (Vorlesung 17, Folie 13), so an, dass der kürzeste Pfad zwischen zwei Knoten i und j ausgegeben wird. Sie dürfen davon ausgehen, dass der übergebene Graph keine negativen Zyklen besitzt.

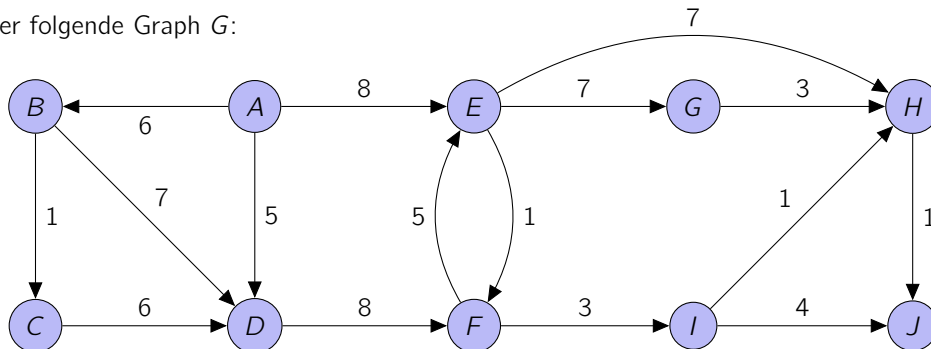
Ihre Funktion soll die folgende Signatur haben:

```
void bellFord(List adjLst[n], int n, int i, int j)
```

Aufgabe 3 (Dijkstra-Algorithmus):

(6 Punkte)

Gegeben sei der folgende Graph G :



Ermitteln Sie mit Hilfe des Dijkstra-Algorithmus, unter Verwendung einer Tabelle der folgenden Form, die kürzesten Wege von Startknoten A zu allen anderen Knoten des Graphen. Notieren Sie für jeden Rechenschritt den aktuell gewählten Knoten zur Verbesserung der Wege und die Länge der bis zu diesem Zeitpunkt möglichen kürzesten Wege für jeden noch nicht abgeschlossenen Knoten ($D[\dots]$). Streichen Sie Felder der Tabelle die nicht mehr benötigt werden durch.

Schritt	0	1	2	3	4	5	6	7	8	9
Knoten										
$D[A]$										
$D[B]$										
$D[C]$										
$D[D]$										
$D[E]$										
$D[F]$										
$D[G]$										
$D[H]$										
$D[I]$										
$D[J]$										