Aufgabe 11.1 (4 Punkte) Gegeben sei die Adjazenzmatrix (mit Kantenkosten) eines gerichteten Graphen mit 8 Knoten:

i/j	1	2	3	4	5	6	7	8
1		13			3			
2			4					
3							1	
4			8					
5						5	17	
6		2	7					
7						23		9
8				11				

a) Zeichne den Graphen.

b) Bestimme mit Hilfe des Algorithmus von Dijkstra die Kosten der kürzesten Pfade von $s = 1$ zu den anderen Knoten des Graphen. Gib nach jedem Durchlauf der **while**-Schleife den Inhalt der Arrays d und *parent* an und die Liste der Knoten-Kosten-Paare die die aktuelle adressierbare Prioritätswarteschlange Q bilden (sortiert nach Kantenkosten).

Aufgabe 11.2 (4 Punkte) Gegeben seien n, m mit $n \leq m \leq n(n-1)/2$. Konstruiere einen Graphen G mit n Knoten und m Kanten und einen Satz nichtnegativer Kantengewichte, so dass bei der Ausführung des Algorithmus von Dijkstra $m-(n-1)$ der Kantenrelaxierungen eine *decreaseKey*-Operation auslösen.

Aufgabe 11.3 (4 Punkte) Der Algorithmus von Dijkstra funktioniert nur für Graphen mit nichtnegativen Kantenkosten. Ein Graph G mit beliebigen Kantenkosten kann aber in einen Graphen G' mit nichtnegativen Kantenkosten transformiert werden, indem man zu allen Kantenkosten eine Konstante c hinzuaddiert.
Wieso lässt sich dennoch der Algorithmus von Dijkstra mit dieser Technik nicht auf beliebige Kantenkosten verallgemeinern? Geben Sie ein Gegenbeispiel ohne negative Kreise an.

Aufgabe 11.4 (4 Punkte) Betrachte einen Graphen $G = (V, E)$ mit Kantengewichten. Nehmen wir zusätzlich an, dass die Kantenkosten ganze Zahlen aus $\{0, \ldots, C\}$ sind. In der Vorlesung wurde gezeigt, wie man die kürzesten Wege in G von einem Startknoten s nach allen anderen Knoten mit Hilfe einer Behälterwarteschlange in $O(m + nC)$ Zeit berechnen kann.

Beschreibe eine andere Datenstruktur, mit der die kürzesten Wege von s nach allen anderen Knoten in $O((n + m) \log C)$ Zeit berechnet werden können.

Hinweis: Verwende einen Baum mit höchstens C Einträgen.