

Algorithmen & Datenstrukturen

Übungsstunde 11

Miniquiz 8

- Passwort: **bellmanford3**
- 10 Fragen
- 11 Minuten
- 0.1 Punkte pro richtige Antwort

Programm

- Miniquiz
- Theory Recap
- Prüfungsaufgaben
- Programmieraufgabe

Besprechung Miniquiz

Nehmen Sie an, dass ein gerichteter Graph einen negativen Kreis hat. Dann ist die Distanz zwischen zwei beliebigen Knoten $-\infty$.

- ☐ Wahr
- ☐ Falsch

Nehmen Sie an, dass während der Ausführung des Algorithmus von Dijkstra, um einen Pfad in einem Graphen mit nicht-negativen Kantenkosten von s nach t zu finden, das Array d der Distanzen wie folgt aussieht:

v	s	v_1	v_2	v_3	v_4	t
$d(v)$	0	6	4	3	8	5

Welche der folgenden Aussagen sind wahr **nachdem der Algorithmus von Dijkstra terminiert ist**?

- ☐ a. $d(t) = 5$
- ☐ b. $d(v_3) = 3$
- ☐ c. $d(v_4) < 8$

On directed graphs, Dijkstra's algorithm can be implemented in time $O((n + m) \log(n))$ (in the worst case) where n is the number of vertices and m is the number of edges.

☐ True

☐ False

Betrachten Sie einen gewichteten und gerichteten Graphen G mit möglicherweise negativen Kantenkosten aber ohne negative Kreise. Wie oft muss die *Schranken verbessern* Funktion im Algorithmus von Bellman und Ford mindestens ausgeführt werden um zu garantieren, dass die korrekten Distanzen berechnet wurden (im Worst Case; n beschreibt die Anzahl der Knoten)?

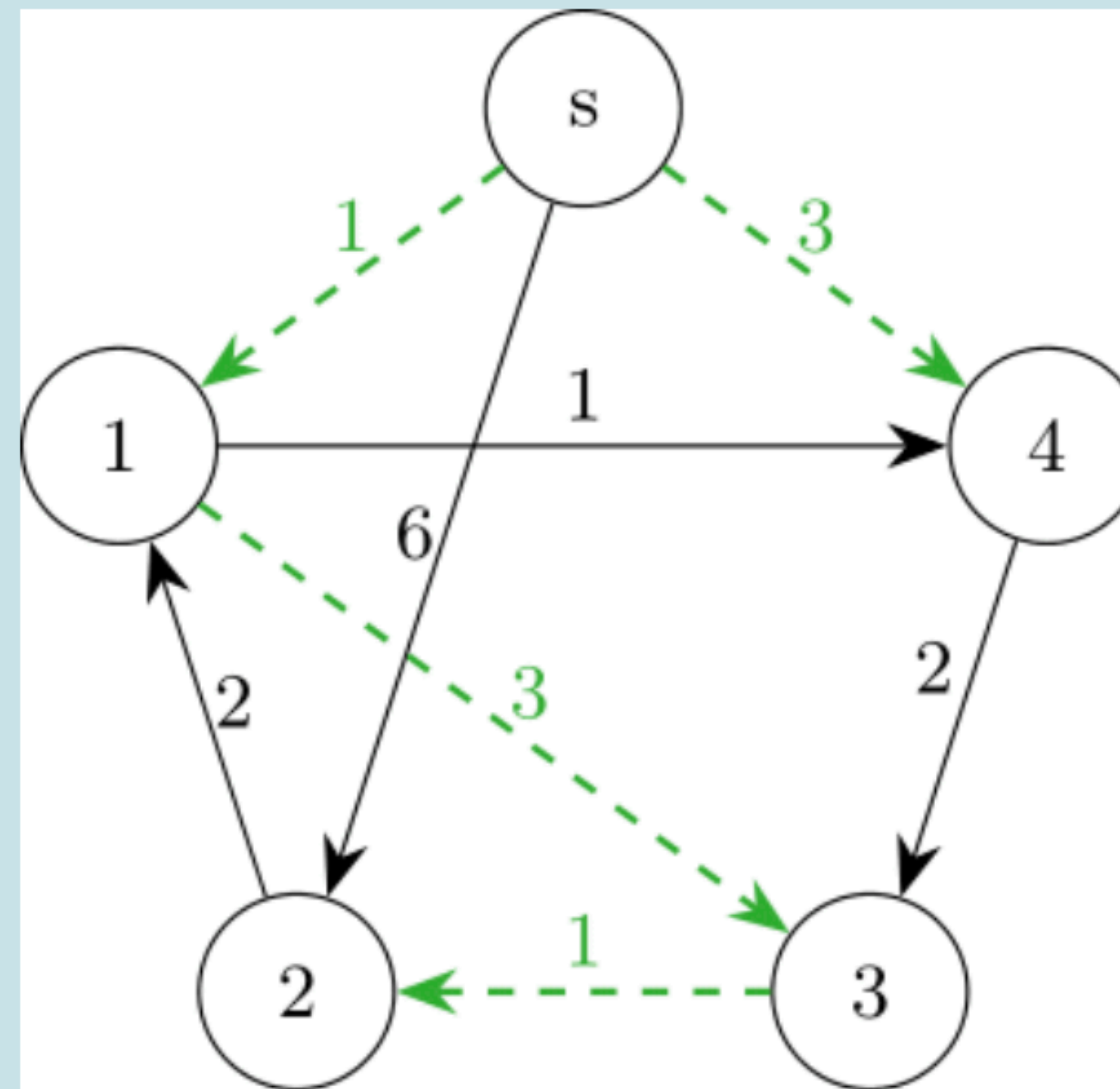
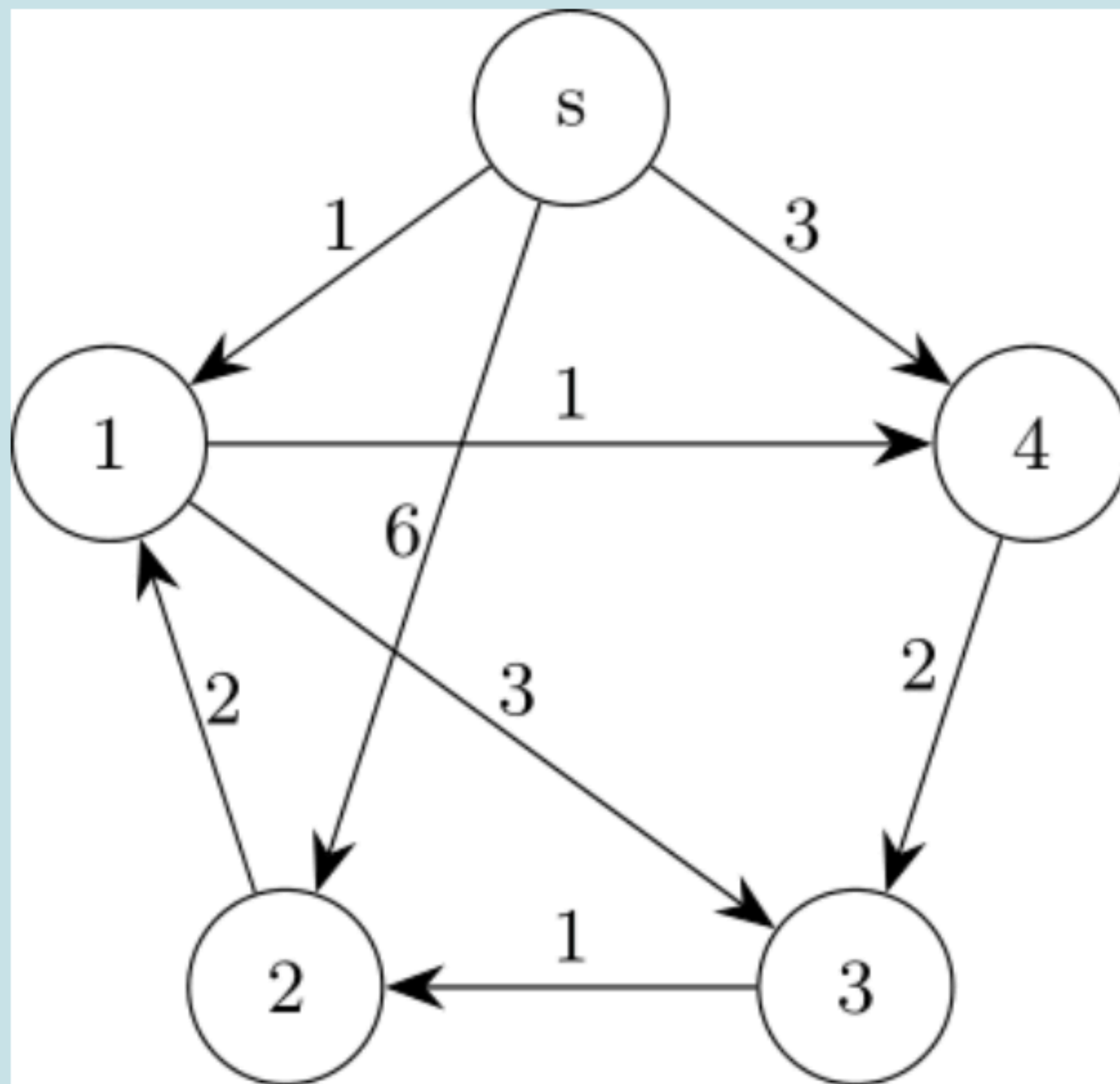
☐ a. 1

☐ b. 2

☐ c. $n - 1$

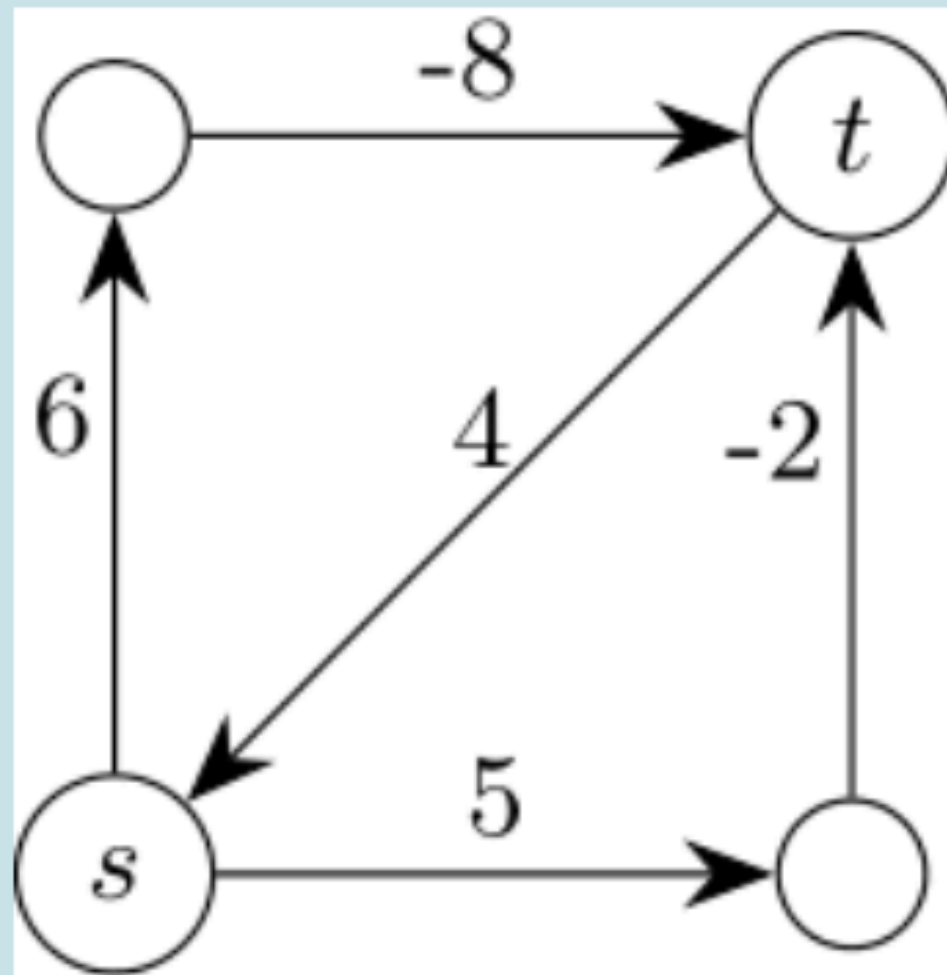
☐ d. n

☐ e. $n + 1$



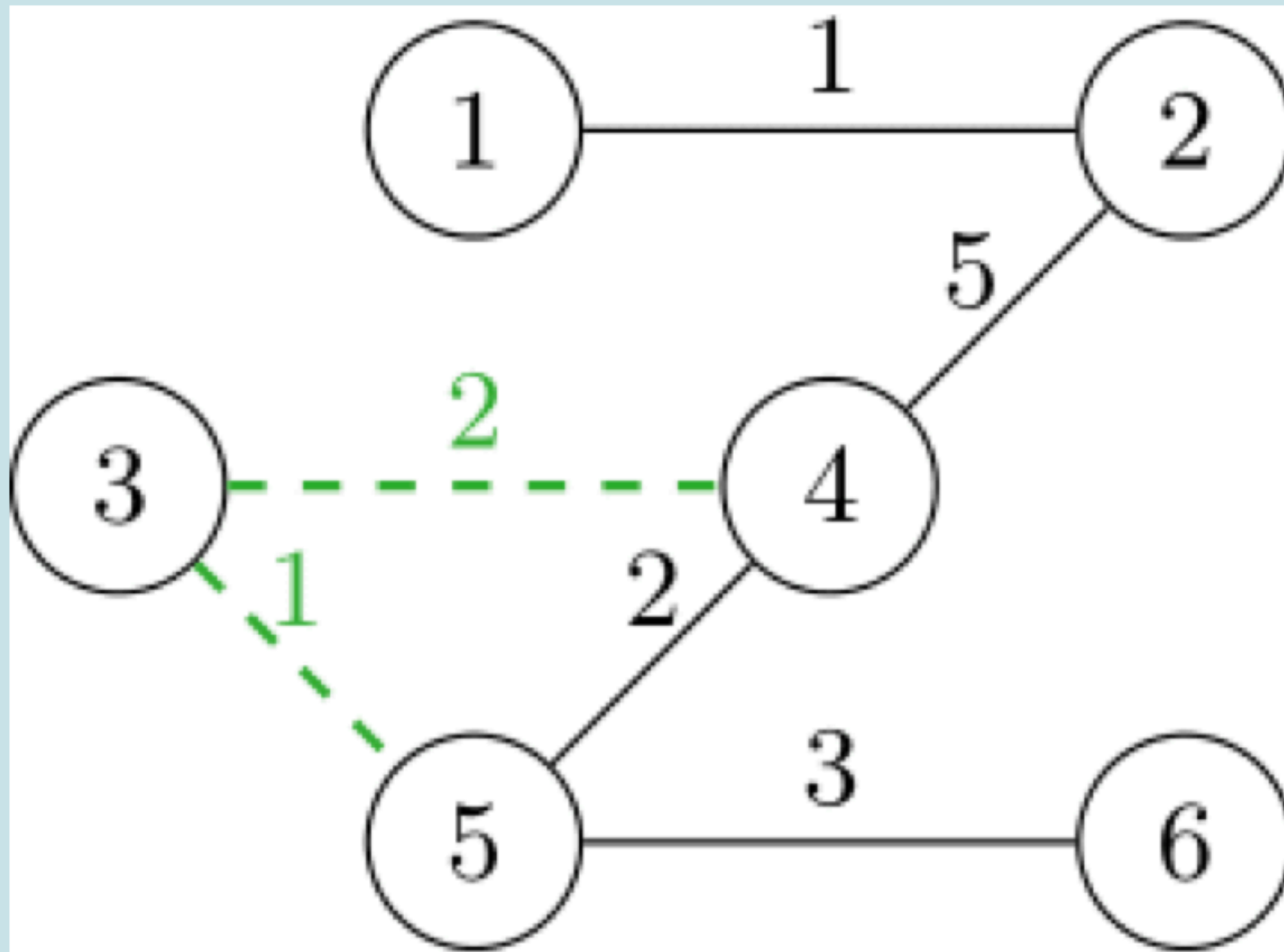
Betrachten Sie den Graphen im ersten Bild. Bilden die grün gestrichelten Kanten im zweiten Bild einen von s ausgehenden Shortest Path Tree?

- ☐ Wahr
- ☐ Falsch



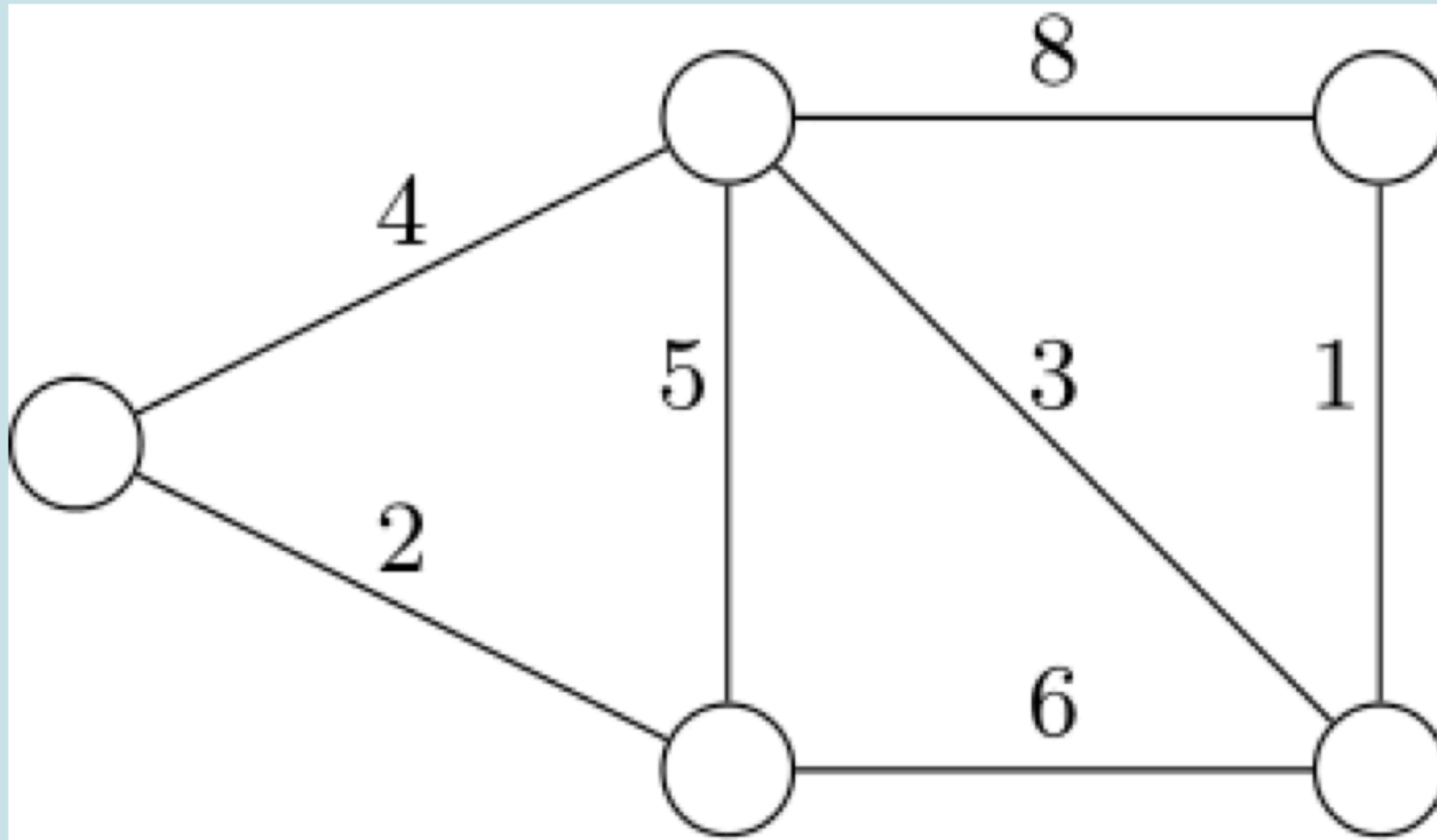
Nehmen Sie an, dass wir den Algorithmus von Bellman und Ford auf den obigen gewichteten und gerichteten Graphen anwenden um einen kürzesten (hier im Sinne von günstigsten) Pfad von s nach t zu finden. Wahr oder falsch: wir haben nach dem dreimaligen Ausführen der *Schranken verbessern* Funktion (aus der Vorlesung) die korrekte Distanz nach t .

- ☐ a. Wahr
- ☐ b. Falsch
- ☐ c. Das hängt von der Reihenfolge, in der die Knoten in der *Schranken verbessern* Funktion verarbeitet werden, ab.



In the weighted graph above, the edges of an (incomplete) minimum spanning tree constructed during the execution of Prim's Algorithm are marked (dashed and green). Which edge will be added to the tree next?

- ☐ a. $\{1, 2\}$
- ☐ b. $\{2, 4\}$
- ☐ c. $\{4, 5\}$
- ☐ d. $\{5, 6\}$



Boruvka's algorithm finds a minimum spanning tree by adding groups of edges in steps. How many edges does Boruvka's algorithm add in the first step when applied to the weighted graph above?

Answer:

Let G be an undirected, weighted, connected graph. Suppose that there is a unique edge e with weight 1 and all other edges have weight strictly greater than 1.

Then any MST of G contains the edge e .

☐ True

☐ False

Let $G = (V, E)$ be an undirected, connected graph with positive edge weights. Suppose that G has a unique minimum spanning tree $T \subseteq E$. Consider any *spanning* subset $S \subseteq E$ of G . Recall that a subset $S \subseteq E$ is called *spanning* if the graph (V, S) is a connected graph. Then S must share an edge with T .

☐ True

☐ False

Theory Recap

Bellman-Ford

- 1-to-all shortest paths mit negativen Kanten
- n-1 Iterationen durch alle Kanten
- $\min\{d(u) + c(u, v)\}$ für alle Vorgänger von v
- mit n-ter Iteration prüfen ob negativer Kreis
 - Wenn sich noch mind. 1 Knoten verbessert

```
public static void bellmanFord(int v, int[][] graph) {
    dists[v] = 0;
    for(int i = 0; i < dists.length; i++) {
        for(int j = 0; j < dists.length; j++) {
            for(int k = 0; k < dists.length; k++) {
                if(dists[k] > dists[j] + graph[j][k]) {
                    dists[k] = dists[j] + graph[j][k];
                }
            }
        }
    }

    //check for negative cycle

    for(int j = 0; j < dists.length; j++) {
        for(int k = 0; k < dists.length; k++) {
            if(graph[j][k] != 0) {
                if(dists[k] > dists[j] + graph[j][k]) {
                    throw new IllegalArgumentException();
                }
            }
        }
    }
}
```


Minimum Spanning Trees

- Sichere Kanten:
 - Alle, welche in keinem Kreis enthalten
 - Kante mit kleinstem Gewicht über Schnitt in Kreis

Boruvka

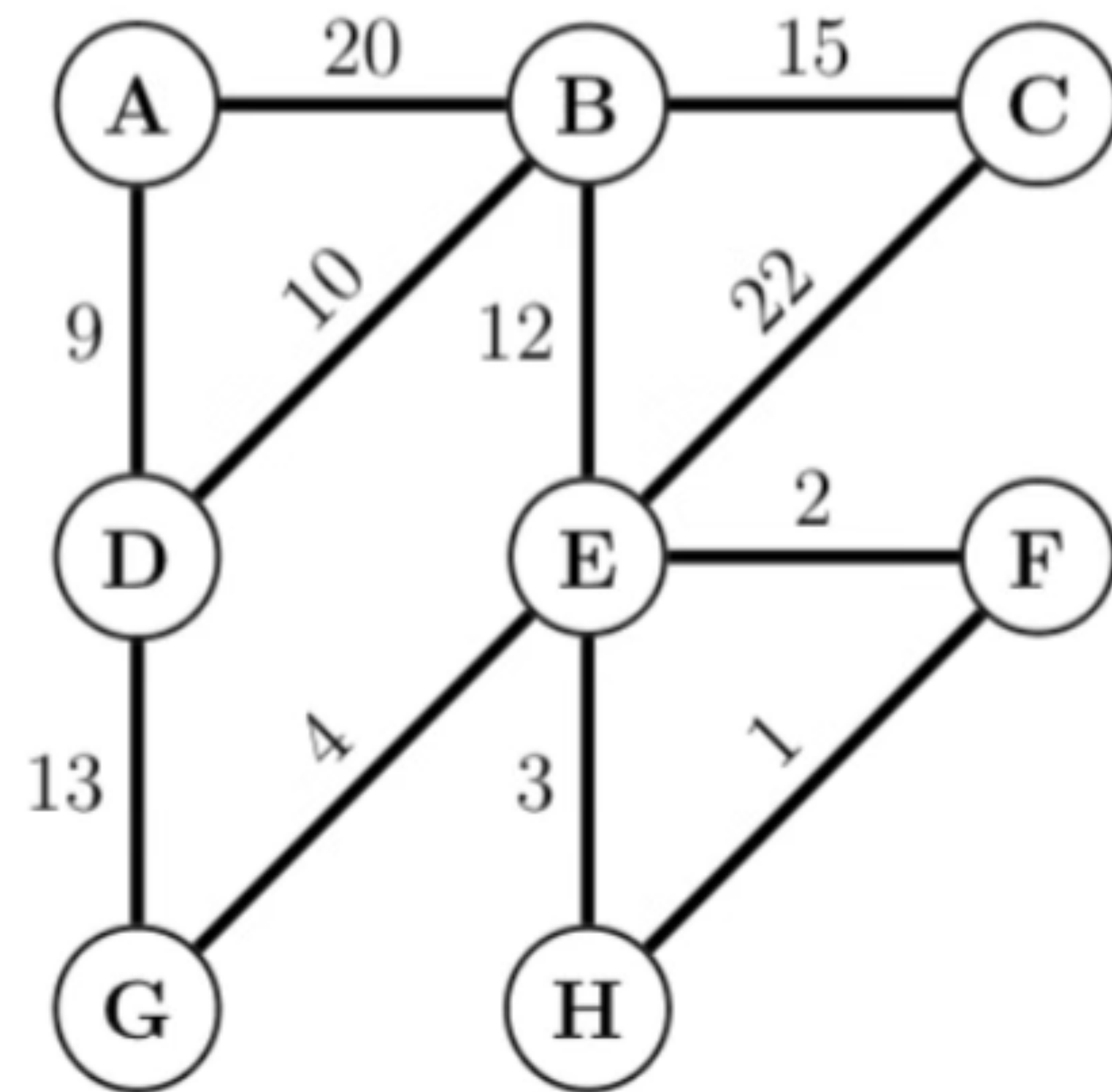
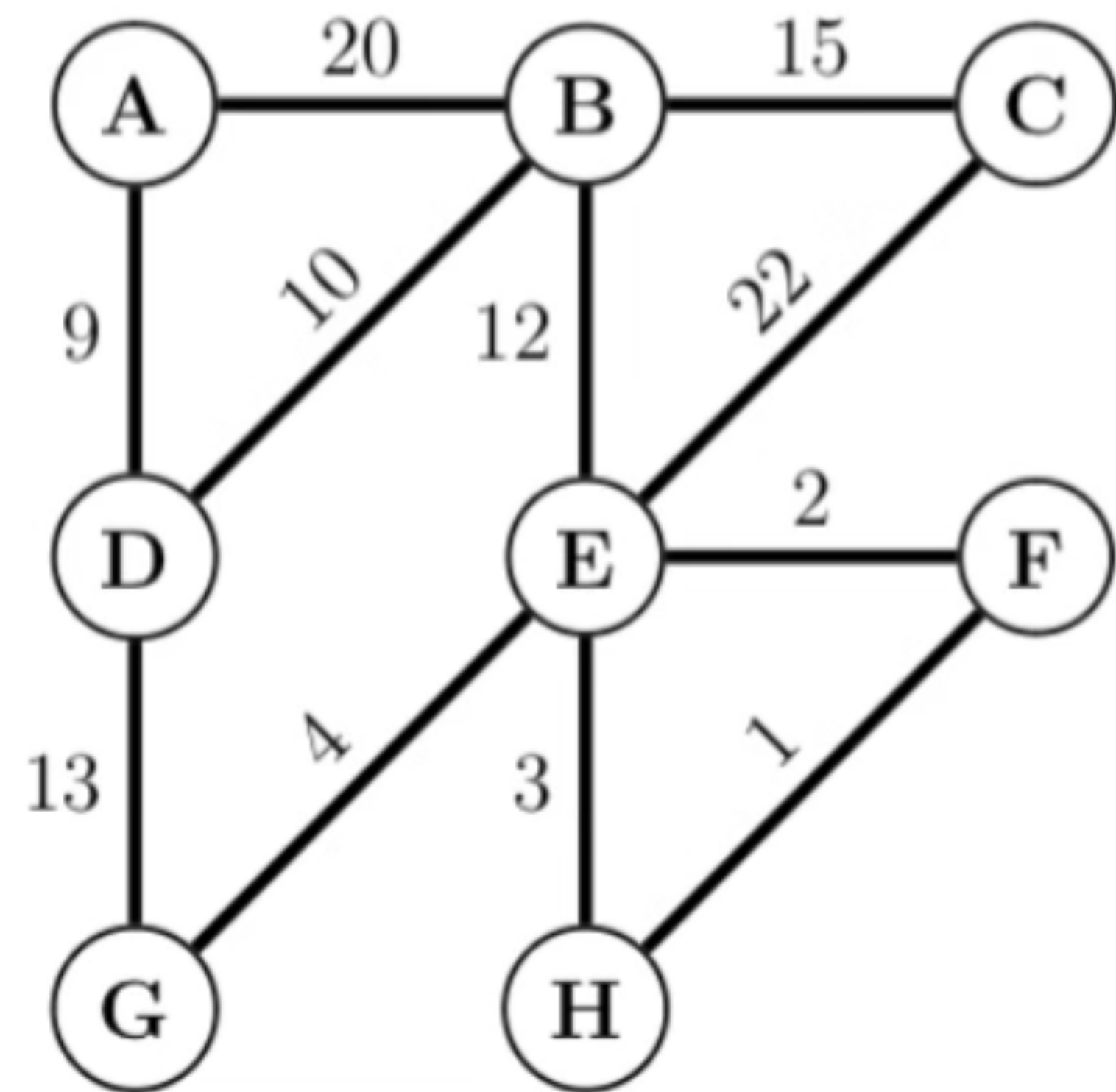
- Minimum Spanning Tree finden
- Jede ZHK wählt Kante um hinzuzufügen
- Laufzeit $O((n + m) \cdot \log(n))$

Prim

- Minimum Spanning Tree finden
- Wir starten mit 1 Knoten (ZHK s)
- fügen minimale Kante an ZHK s hinzu, welche neuen Knoten hinzufügt
- implementierung mit Min-heap
- Laufzeit $O((n + m) \cdot \log(n))$


```
public static void prim(int v){  
    PriorityQueue<Integer> q = new PriorityQueue<Integer>();  
  
    q.add(v);  
  
    while(count < mst.length) {  
        int curr = q.poll();  
        count++;  
  
        for(int i = 0; i < Ev.get(curr).size(); i++) {  
            q.add(Ev.get(curr).get(i));  
        }  
    }  
}
```


- d) Below you find two copies of the same graph. In the left graph, highlight the first 5 edges that Prim's algorithm adds to the minimum spanning tree, when starting at vertex B . In the right graph, highlight the first 5 edges that ~~Kruskal's~~ algorithm adds to the minimum spanning tree. *in the first iteration* *Boruvka's*



Minimum spanning trees: Given is a weighted undirected connected graph $G = (V, E, w)$ with positive and pairwise different edge weights $w: E \rightarrow \mathbb{R}^+$. Suppose that there exists a vertex $v \in V$ of degree 4 with incident edges e_1, e_2, e_3, e_4 such that $w(e_1) = 2, w(e_2) = 3, w(e_3) = 4, w(e_4) = 7$. Further, suppose that the weight of the minimum spanning tree of G is 72.

What are the possible weights of a minimum spanning tree if we decrease the weight of the edge e_1 from 2 to 1? The weights of all other edges remain unchanged. Justify your answer.

Exercise Recommendations

Aufgabenblatt 11

- 11.1 Bonus
- 11.2 Bonus
- 11.3 Witzige Aufgabe, viel schreibarbeit
- 11.4 Bonus
- 11.5 Gut scih zu überlegen, eher einfach

Peergrading

- Aufgabe 10.4
- Grading Scheme in Lösungen beachten