



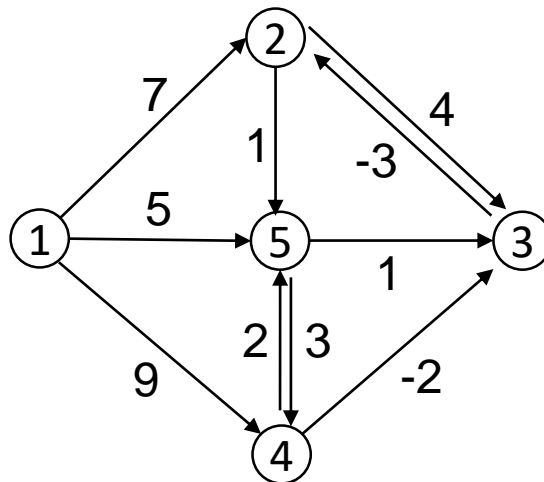
Grundzüge der Informatik 1

Vorlesung 22 - flipped classroom

Graphalgorithmen

Aufgabe 1

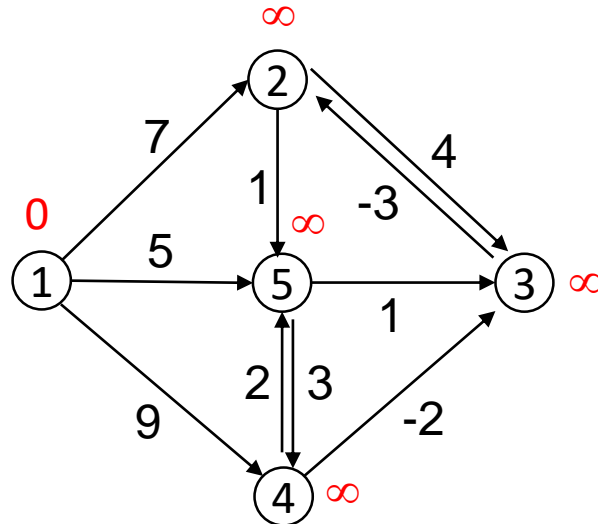
- Führen Sie den Bellman-Ford-Algorithmus (verbesserte Version) auf unten stehendem Graph mit Startknoten 1 aus. Dabei werden Knoten und Kanten in aufsteigender Reihenfolge betrachtet.



Graphalgorithmen

Aufgabe 1

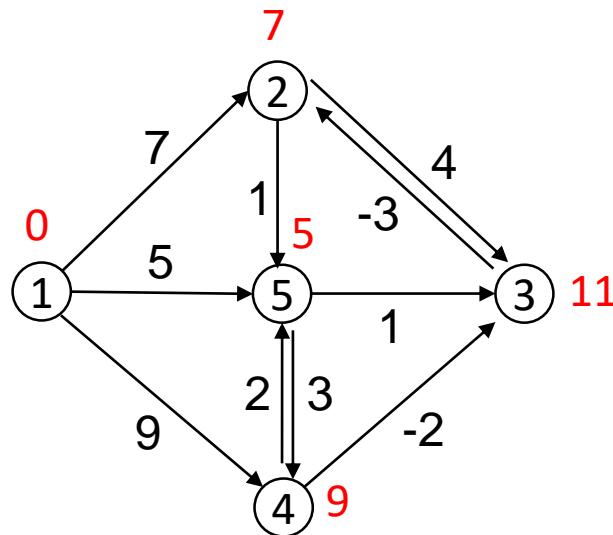
- Führen Sie den Bellman-Ford-Algorithmus (verbesserte Version) auf unten stehendem Graph mit Startknoten 1 aus. Dabei werden Knoten und Kanten in aufsteigender Reihenfolge betrachtet.



Graphalgorithmen

Aufgabe 1

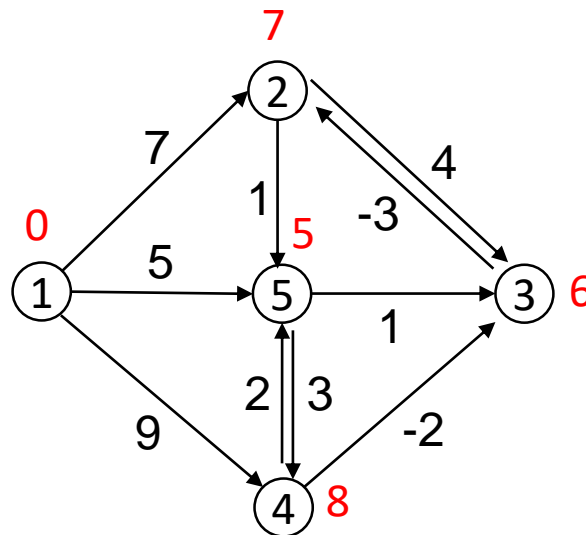
- Führen Sie den Bellman-Ford-Algorithmus (verbesserte Version) auf unten stehendem Graph mit Startknoten 1 aus. Dabei werden Knoten und Kanten in aufsteigender Reihenfolge betrachtet.



Graphalgorithmen

Aufgabe 1

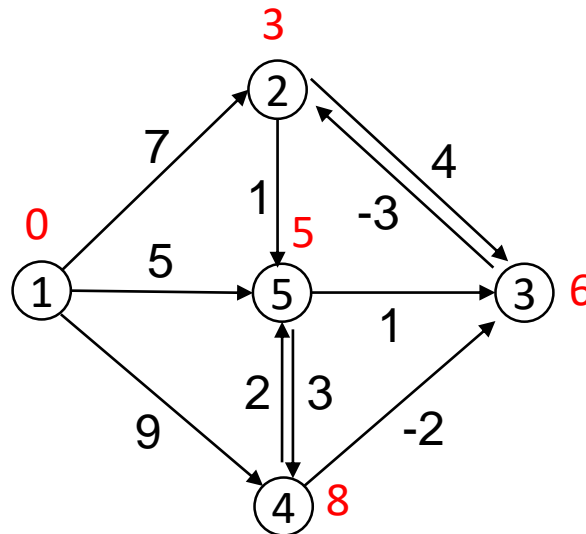
- Führen Sie den Bellman-Ford-Algorithmus (verbesserte Version) auf unten stehendem Graph mit Startknoten 1 aus. Dabei werden Knoten und Kanten in aufsteigender Reihenfolge betrachtet.



Graphalgorithmen

Aufgabe 1

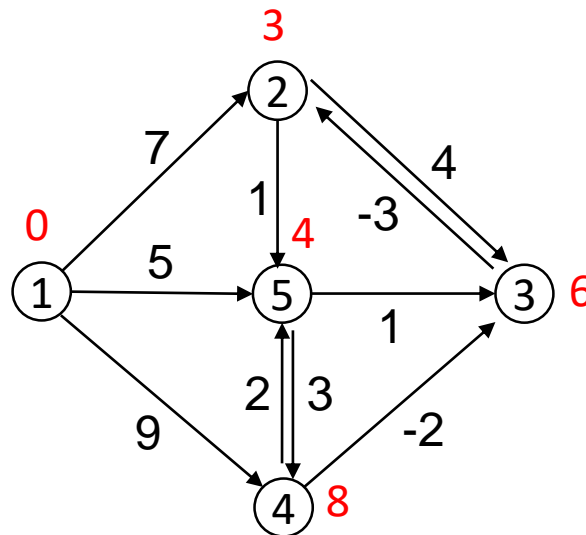
- Führen Sie den Bellman-Ford-Algorithmus (verbesserte Version) auf unten stehendem Graph mit Startknoten 1 aus. Dabei werden Knoten und Kanten in aufsteigender Reihenfolge betrachtet.



Graphalgorithmen

Aufgabe 1

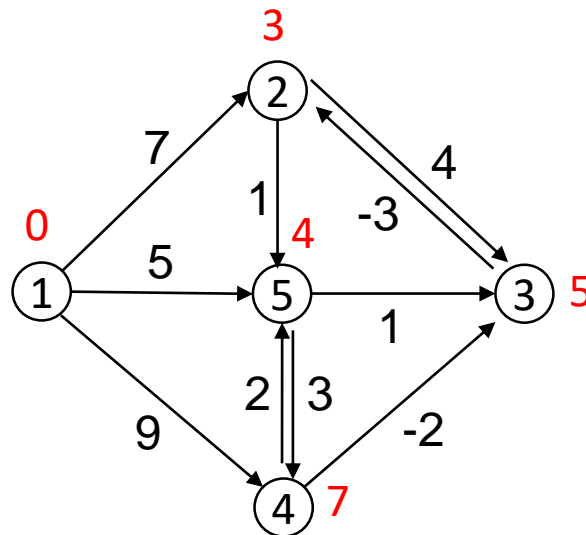
- Führen Sie den Bellman-Ford-Algorithmus (verbesserte Version) auf unten stehendem Graph mit Startknoten 1 aus. Dabei werden Knoten und Kanten in aufsteigender Reihenfolge betrachtet.



Graphalgorithmen

Aufgabe 1

- Führen Sie den Bellman-Ford-Algorithmus (verbesserte Version) auf unten stehendem Graph mit Startknoten 1 aus. Dabei werden Knoten und Kanten in aufsteigender Reihenfolge betrachtet.



Graphalgorithmen

Aufgabe 2

- Sei G ein gerichteter, gewichteter Graph ohne negative Kreise. Entwickeln Sie einen Algorithmus mit Laufzeit $O(|V|^2 + |V| \cdot |E|)$, der für jeden Knoten v aus V die Entfernung zu dem Knoten u aus V bestimmt, der die geringste Entfernung zu v hat (der nächste Nachbar von v)
- Entwickeln Sie dazu zunächst eine Rekursion für die Entfernung zum Knoten mit geringster Entfernung, der über einen Weg mit maximal i Kanten erreicht werden kann

Graphalgorithmen

Aufgabe 2

- Sei G ein gerichteter, gewichteter Graph ohne negative Kreise. Entwickeln Sie einen Algorithmus mit Laufzeit $O(|V|^2 + |V| \cdot |E|)$, der für jeden Knoten v aus V die Entfernung zu dem Knoten u aus V bestimmt, der die geringste Entfernung zu v hat (der nächste Nachbar von v)
- Entwickeln Sie dazu zunächst eine Rekursion für die Entfernung zum Knoten mit geringster Entfernung, der über einen Weg mit maximal i Kanten erreicht werden kann

Rekursion

- $i > 0$: $\text{Opt}(i, v) = \min_{(v, u) \in E} \{\text{Opt}(i-1, v), w(v, u) + \text{Opt}(i-1, u)\}$
- $i = 0$: $\text{Opt}(0, v) = 0$

Graphalgorithmen

Aufgabe 3

- Sei G ein gerichteter, gewichteter Graph ohne negative Kreise. Entwickeln Sie einen Algorithmus mit Laufzeit $O(|V|^2 + |V| \cdot |E|)$, der für jeden Knoten v aus V die Entfernung zu dem Knoten u aus V bestimmt, der die geringste Entfernung zu v hat (der nächste Nachbar von v)
- Gehen Sie dann ähnlich wie bei den Optimierungen des Bellman-Ford Algorithmus vor, um einen Algorithmus mit der gewünschten Laufzeit zu erhalten

Graphenalgorithmen

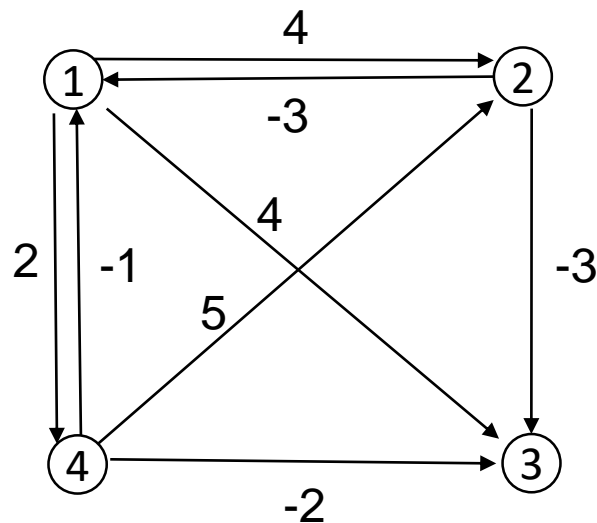
NächsterNachbar(G)

1. **d = new array** [1.. $|V|$]
2. **for each** $v \in V$ **do** $d[v]=0$
3. **for** $i=1$ **to** $|V|-1$ **do**
4. **for each** $v \in V$ **do**
5. **for each** $(v,u) \in \text{Adj}[v]$ **do**
6. **if** $w(v,u)+d[u]<d[v]$ **then** $d[v]=w(v,u)+d[u]$
7. **return** d

Graphalgorithmen

Aufgabe 4

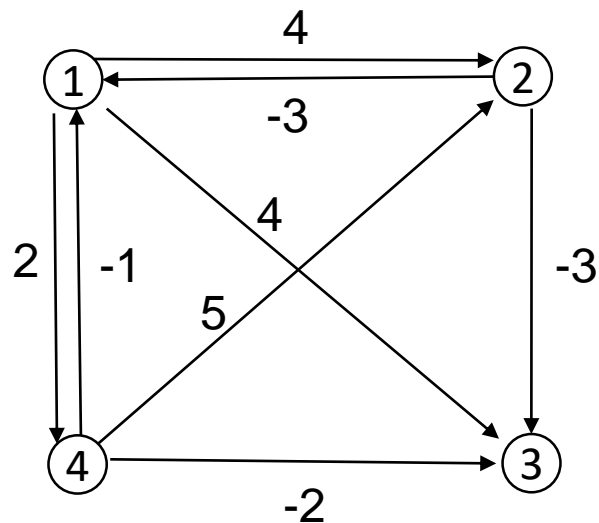
- Führen Sie den Floyd-Warshall-Algorithmus auf unten stehendem Graph aus. Transformieren Sie den Graph dazu zunächst in die Adjazenzmatrixdarstellung.



Graphalgorithmen

Aufgabe 4

- Führen Sie den Floyd-Warshall-Algorithmus auf unten stehendem Graph aus. Transformieren Sie den Graph dazu zunächst in die Adjazenzmatrixdarstellung.



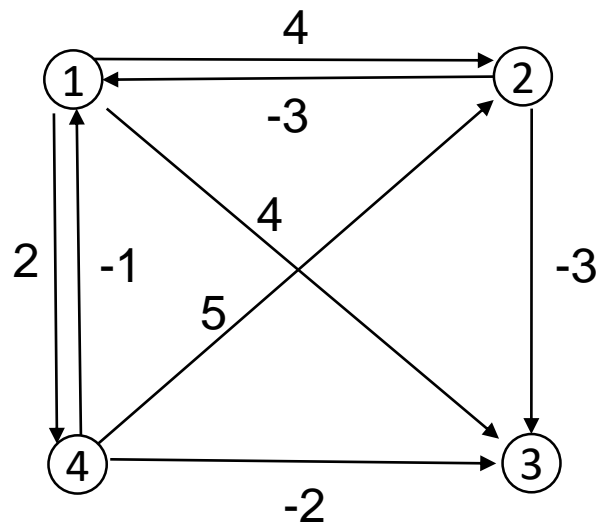
$D^{(0)}$

0	4	4	2
-3	0	-3	∞
∞	∞	0	∞
-1	5	-2	0

Graphalgorithmen

Aufgabe 4

- Führen Sie den Floyd-Warshall-Algorithmus auf unten stehendem Graph aus. Transformieren Sie den Graph dazu zunächst in die Adjazenzmatrixdarstellung.



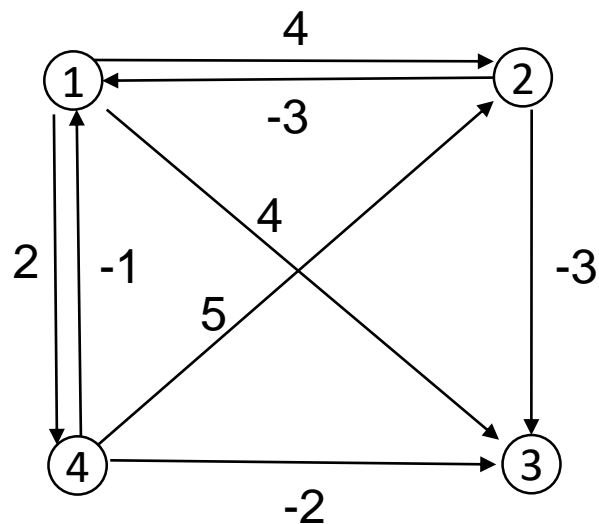
$D^{(1)}$

0	4	4	2
-3	0	-3	-1
∞	∞	0	∞
-1	3	-2	0

Graphalgorithmen

Aufgabe 4

- Führen Sie den Floyd-Warshall-Algorithmus auf unten stehendem Graph aus. Transformieren Sie den Graph dazu zunächst in die Adjazenzmatrixdarstellung.



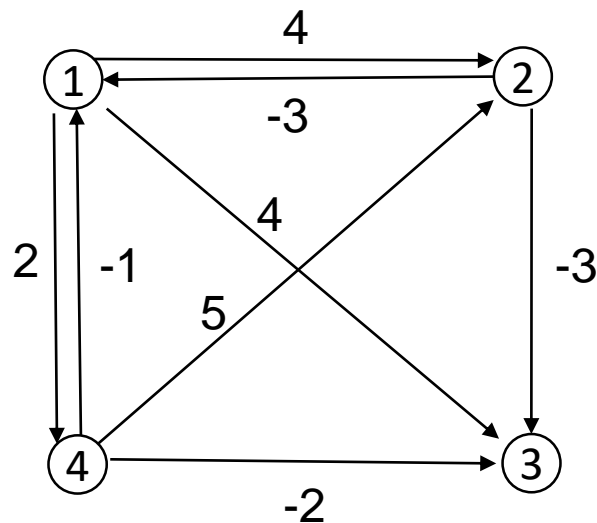
$D^{(2)}$

0	4	1	2
-3	0	-3	-1
∞	∞	0	∞
-1	3	-2	0

Graphalgorithmen

Aufgabe 4

- Führen Sie den Floyd-Warshall-Algorithmus auf unten stehendem Graph aus. Transformieren Sie den Graph dazu zunächst in die Adjazenzmatrixdarstellung.



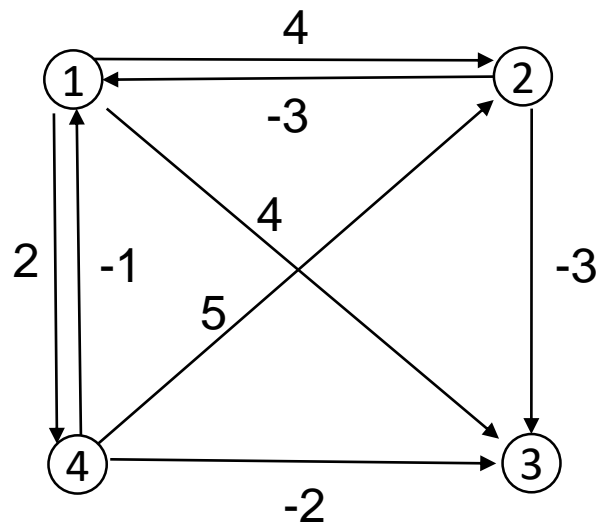
$D^{(3)}$

0	4	1	2
-3	0	-3	-1
∞	∞	0	∞
-1	3	-2	0

Graphalgorithmen

Aufgabe 4

- Führen Sie den Floyd-Warshall-Algorithmus auf unten stehendem Graph aus. Transformieren Sie den Graph dazu zunächst in die Adjazenzmatrixdarstellung.



$D^{(4)}$

0	4	0	2
-3	0	-3	-1
∞	∞	0	∞
-1	3	-2	0