Übungsaufgabe

Kürzeste Pfade und Flussprobleme: Dijkstra, Bellman-Ford, Ford-Fulkerson/Edmonds-Karp

Universität:	Technische Universität Berlin
Kurs/Modul:	Algorithmen und Datenstrukturen
Erstellungsdatum:	September 6, 2025

Aufgabe 1: Kürzeste Pfade in gewichteten Graphen (Dijkstra, Bellman-Ford)

Betrachten Sie gewichtete gerichtete Graphen. Die Aufgaben bedecken den Einsatz von Dijkstra auf Graphen mit nicht-negativen Gewichten und Bellman-Ford auf Graphen mit möglichen negativen Kantengewichten (aber ohne negative Zyklen).

Gegeben Graph G1:

$$V = \{S, A, B, C, T\}, \quad E = \{(S, A, 2), (S, B, 4), (A, B, 1), (A, C, 7), (B, C, 1), (C, T, 3), (B, T, 5)\}.$$

Gegeben Graph G2:

$$V = \{S, P, Q, R, T\}, \quad E = \{(S, P, 6), (S, Q, 7), (P, Q, 5), (Q, P, -4), (P, R, 8), (Q, R, 6), (R, T, 2), (Q, T, 3)\}.$$

a) Bestimme die kürzesten Entfernungen von S zu allen Knoten in Graph G1 durch Anwendung des Dijkstra-Algorithmus. Gib die entstehenden Abstände $d(S, A)$, $d(S, B)$, $d(S, C)$, $d(S, T)$ an und notiere, falls möglich, die Vorgängerpfade für jeden Knoten.

b) Bestimme die kürzesten Entfernungen von S zu allen Knoten in Graph G2 durch Anwendung des Bellman-Ford-Algorithmus. Gib die entstehenden Abstände $d(S, P)$, $d(S, Q)$, $d(S, R)$, $d(S, T)$ an und notiere die Vorgängerpfade für jeden Knoten.

c) Vergleiche kurz die Situation der beiden Graphen hinsichtlich der Eignung von Dijkstra bzw. Bellman-Ford und erläutere, warum Bellman-Ford hier (ggf.) die korrekten Ergebnisse liefert, während Dijkstra bei Graphen mit negativen Kantengewichten nicht verwendet werden sollte. (Hinweis: keine Lösungen, nur Aufgabenstellung.)

Aufgabe 2: Flussprobleme – Ford-Fulkerson/Edmonds-Karp

Betrachten Sie das folgende Netzwerk mit Quelle S und Senke T. Die Kanten tragen Kapazitäten in Knoten-Paaren.

Graph H:

$$S \to A: \ 8$$
$$S \to B: \ 6$$
$$A \to B: \ 3$$
$$A \to T: \ 7$$
$$B \to T: \ 5$$

a) Formuliere das Max-Flow-Problem für diesen Graphen mit Quelle S und Senke T. Gib die Variablen für Flüsse entlang der Kanten an und die Bedingung der Netzflussgleichungen (Flusskonservierung an den inneren Knoten) an.

b) Wende den Edmonds-Karp-Algorithmus (BFS-basierte Augmentierung) an, um den maximalen Fluss von S nach T zu bestimmen. Gib am Ende den Max-Flow-Wert an.

Aufgabe 3: Komplexität und Implementierungsüberlegungen

Betrachte die drei vorgestellten Algorithmen. Beantworte die folgenden Fragen zu deren Komplexität und praktischer Implementierung.

a) Gebe die Worst-Case-Laufzeitkomplexität von Dijkstra mit Min-Heap-Implementierung an (in Abhängigkeit von $|V|$ und $|E|$).

b) Gebe die Worst-Case-Laufzeitkomplexität von Bellman-Ford an (in Abhängigkeit von $|V|$ und $|E|$).

c) Gebe die Worst-Case-Laufzeitkomplexität von Edmonds-Karp (eine Implementierung von Ford-Fulkerson mit BFS-Augmentierung) an (in Abhängigkeit von $|V|$ und $|E|$).

Lösungen

Aufgabe 1: Kürzeste Pfade in gewichteten Graphen (Dijkstra, Bellman-Ford)

a) Dijkstra auf Graph G1

Lösung: - Endgültige kürzeste Entfernungen von S zu den Knoten:

$$d(S,A) = 2, \quad d(S,B) = 3, \quad d(S,C) = 4, \quad d(S,T) = 7.$$

- Vorgängerpfade (Vorgänger jedes Knotens im kürzesten Pfad):

$$\text{pred}(A) = S, \quad \text{pred}(B) = A, \quad \text{pred}(C) = B, \quad \text{pred}(T) = C.$$

- Daraus ergeben sich die kürzesten Pfade: - $S \to A$ (Länge 2) - $S \to A \to B$ (Länge $2 + 1 = 3$) - $S \to A \to B \to C$ (Länge $2 + 1 + 1 = 4$) - $S \to A \to B \to C \to T$ (Länge $2 + 1 + 1 + 3 = 7$)

b) Bellman-Ford auf Graph G2

Lösung: - Endgültige kürzeste Entfernungen von S zu den Knoten:

$$d(S,P) = 3, \quad d(S,Q) = 7, \quad d(S,R) = 11, \quad d(S,T) = 10.$$

- Vorgängerpfade:

$$\text{pred}(P) = Q, \quad \text{pred}(Q) = S, \quad \text{pred}(R) = P, \quad \text{pred}(T) = Q.$$

- Daraus ergeben sich die kürzesten Pfade: - $S \to Q$ (Länge 7) - $S \to Q \to P$ (Länge $7+(-4) = 3$) - $S \to Q \to P \to R$ (Länge $7 + (-4) + 8 = 11$) - $S \to Q \to T$ (Länge $7 + 3 = 10$)

c) Vergleich Dijkstra vs. Bellman-Ford

- Dijkstra (mit Min-Heap) setzt grundsätzlich voraus, dass alle Kantengewichte nicht negativ sind. Bei negativen Kantengewichten kann Dijkstra frühzeitig Knoten als endgültig klassifizieren, wodurch spätere relaxationen über negative Kanten zu einer kürzeren Distanz zu einem bereits festgelegten Knoten führen können. Dadurch sind falsche Ergebnisse möglich. - Bellman-Ford (mit $|V| - 1$ Relaxationen über alle Kanten) ermöglicht es, auch negative Gewichte korrekt zu berücksichtigen, da jeder Pfad mit maximal $|V| - 1$ Kanten in der Relaxationsphase vollständig evaluiert wird. Es kann zudem negative Zyklen erkennen. - In Graph G2 existiert zwar ein negativer Kantengewichtszyklus zwischen P und Q, dieser Zyklus hat jedoch eine positive Summe (5 + (-4) = 1), sodass kein negativer Zyklus vorliegt und Bellman-Ford korrekte Ergebnisse liefert. Dijkstra hätte hier potenziell falsche Distanzwerte liefern können, da negative Kanten existieren.

Aufgabe 2: Flussprobleme – Ford-Fulkerson/Edmonds-Karp

Betrachten Sie das folgende Netzwerk mit Quelle S und Senke T. Die Kanten tragen Kapazitäten in Knoten-Paaren.

Graph H:

$$S \to A: \ 8$$
$$S \to B: \ 6$$
$$A \to B: \ 3$$
$$A \to T: \ 7$$
$$B \to T: \ 5$$

a) Maximiere den Fluss von S nach T (Formuliere das Max-Flow-Problem).

Lösung (Lineare Programm-Formulierung): - Variablen für Flüsse entlang der Kanten:

$$f_{SA}, \quad f_{SB}, \quad f_{AB}, \quad f_{AT}, \quad f_{BT}.$$

- Kapazitätsbeschränkungen:

$$0 \le f_{SA} \le 8, \quad 0 \le f_{SB} \le 6, \quad 0 \le f_{AB} \le 3, \quad 0 \le f_{AT} \le 7, \quad 0 \le f_{BT} \le 5.$$

- Flussgleichungen (Konzentration an inneren Knoten S, A, B, T beachtet): - Zielgröße: Maximiere den Gesamtausfluss aus S, d. h. $f_{SA} + f_{SB}$. - Flow-Konservierung an A: $f_{SA} = f_{AB} + f_{AT}$. - Flow-Konservierung an B: $f_{SB} + f_{AB} = f_{BT}$. - Am Knoten S und T gelten die Randbedingungen durch die Maximierung des Outflows bzw. Inflows.

b) Edmonds-Karp-Algorithmus (BFS-basierte Augmentierung) – maximaler Fluss S→T

Hilfswegweiser: - Augmentationspfade in zeitlich sinnvoller Reihenfolge (kürzeste Pfade in BFS-Ebene): 1) Pfad S → A → T mit Fluss 7. - Flüsse: $f_{SA} = 7, \ f_{AT} = 7$. 2) Pfad S → A → B → T mit Fluss 1. - Flüsse aktualisiert: $f_{SA} = 8, \ f_{AB} = 1, \ f_{BT} = 1$. 3) Pfad S → B → T mit Fluss 4. - Flüsse aktualisiert: $f_{SB} = 4, \ f_{BT} = 5$.

- Nach diesen Augmentierungen ist kein weiterer S→T-Pfad in der Restkapazität vorhanden; der maximale Fluss ist erreicht.

Endergebnis: - Max-Flow-Wert: 12. - Endzustandsflusswerte:

$$f_{SA} = 8, \quad f_{SB} = 4, \quad f_{AB} = 1, \quad f_{AT} = 7, \quad f_{BT} = 5.$$

- Bestimmter maximale Fluss liefert:

$$|F| = f_{SA} + f_{SB} = 12 = f_{AT} + f_{BT}.$$

Aufgabe 3: Komplexität und Implementierungsüberlegungen

Betrachte die drei vorgestellten Algorithmen. Beantworte die folgenden Fragen zu deren Komplexität und praktischer Implementierung.

a) Gebe die Worst-Case-Laufzeitkomplexität von Dijkstra mit Min-Heap-Implementierung an (in Abhängigkeit von $|V|$ und $|E|$).

Lösung: - $\mathcal{O}((|V| + |E|) \log |V|)$ bzw. $\mathcal{O}(|E| \log |V|)$ bei $|E| \geq |V|$.

b) Gebe die Worst-Case-Laufzeitkomplexität von Bellman-Ford an (in Abhängigkeit von $|V|$ und $|E|$).

Lösung: - $\mathcal{O}(|V| \cdot |E|)$.

c) Gebe die Worst-Case-Laufzeitkomplexität von Edmonds-Karp (eine Implementierung von Ford-Fulkerson mit BFS-Augmentierung) an (in Abhängigkeit von $|V|$ und $|E|$).

Lösung: - $\mathcal{O}(|V| \cdot |E|^2)$.