

7. Übung

1. Sei $G = (V, E)$ ein gerichteter Graph mit $V = \{1, \dots, n\}$. Der Pfadgraph $G_P = (V, E^*)$ von G ist folgendermaßen erklärt: Die Knoten von G_P sind jene von G , weiters ist $(i, j) \in E^*$ genau dann falls $i, j \in V$ durch einen Pfad in G verbunden sind.

Eine leichte Modifikation des Floyd-Warshall Algorithmus liefert eine binäre Matrix $M \in \{0, 1\}^{n \times n}$ mit $m_{ij} = 1$ genau falls ein Pfad von i nach j in G führt.

Die Matrix M wird (analog zu FW) algorithmisch rekursiv berechnet, M^0 sei jene binäre $n \times n$ Matrix mit 1 an m_{ij}^0 , falls $i = j$ oder falls $(i, j) \in E$ und 0 sonst. M^k erfülle $m_{ij}^k = 1$ falls es einen Pfad von i nach j in G gibt, welcher (nur) Knoten in $\{1, \dots, k\}$ durchläuft, und = 0 sonst. Diese Definition liefert die Rekursion:

$$m_{ij}^k = m_{ij}^{k-1} \text{ ODER } \left(m_{ik}^{k-1} \text{ UND } m_{kj}^{k-1} \right),$$

wobei das ODER und UND als logische Operationen auf $\{0, 1\}$ zu verstehen sind.

- Erklären Sie die Rekursion und warum $M^n = M$ gilt.
- Geben Sie einen Algorithmus an, welcher M in $\Theta(n^3)$ Schritten bestimmt.
- Berechnen Sie die Matrizen M^k für den Graphen $G = (V, E)$, mit $V = \{1, 2, 3, 4\}$ und

$$E = \{(1, 2), (1, 4), (2, 3), (3, 4), (4, 2)\}.$$

2. Der Pfadgraph G_P eines Graphen G soll dynamisch bestimmt werden, d.h. zum ursprünglichen Graphen G werden sukzessive Kanten hinzugefügt und nach jeder zugefügten Kante soll der Pfadgraph aktualisiert werden.

Angenommen G hat initial keine Kanten und G_P werde wie in (a) durch eine binäre Matrix dargestellt.

- Zeigen Sie, wie der Pfadgraph G_P in $O(V^2)$ Zeit aktualisiert werden kann, wenn zu G eine Kante hinzugefügt wird.
 - Geben Sie einen Graphen G und eine Kante e an, sodass $\Omega(V^2)$ Zeit benötigt wird, um G_P zu bestimmen, sobald e zu G hinzugefügt wird.
 - Geben Sie einen Algorithmus an, welcher schrittweise den Pfadgraph aktualisiert. Dieser Algorithmus soll in $O(V^3)$ Zeit laufen (unabhängig davon, wie die (a priori unbekannt) Folge an eingefügten Kanten aussieht).
3. Wiederholen Sie die Funktionsweise des Ford-Fulkerson Algorithmus am Beispiel $G = (V, E)$ mit $V = \{s, v_1, v_2, v_3, v_4, t\}$ und den Kapazitäten $c(s, v_1) = 18$, $c(s, v_2) = 5$, $c(v_1, v_2) = 16$, $c(v_1, v_3) = 7$, $c(v_2, v_4) = 16$, $c(v_4, v_3) = 9$, $c(v_3, t) = 20$, $c(v_4, t) = 6$, genauer: Finden Sie einen maximalen Fluss von s nach t mit dem FF-Algorithmus. Geben Sie die auftretenden Restkapazitätsgraphen an.
4. Die aktivste Kante eines Netzwerks (Graph inkl. Kapazitäten für Fluss von s nach t) sei jene Kante, deren Entfernung die größte Abnahme des maximalen Flusses von s nach t bewirkt. Sei f nun ein maximaler s - t -Fluss. Zeigen oder widerlegen (durch ein passendes Gegenbeispiel) Sie folgende Behauptungen:
- Eine aktivste Kante ist eine Kante e mit maximalem Fluss $f(e)$.
 - Eine aktivste Kante ist eine Kante e mit maximaler Kapazität $c(e)$.
 - Eine aktivste Kante ist eine Kante e mit maximalem Fluss $f(e)$ unter jenen Kanten, welche zu einem minimalen Schnitt gehören.
 - Eine Kante, welche zu keinem minimalen Schnitt gehört, kann keine aktivste Kante sein.
 - Ein Netzwerk kann mehrere aktivste Kanten enthalten.
5. p Familien mit a_i , $i = 1, 2, \dots, p$, Familienmitgliedern besuchen gemeinsam ein Restaurant, q Tische mit b_j , $j = 1, 2, \dots, q$, Plätzen seien für das Essen reserviert. Im Sinne einer optimalen Durchmischung der einzelnen Familienmitglieder (oder weil sich die jeweiligen Familienmitglieder aus dem Weg gehen wollen), sollen niemals zwei Familienmitglieder der selben Familie am selben Tisch sitzen.

Formulieren Sie ein Maximales Fluss Problem, welches dieses Problem löst.