

Effiziente Algorithmen I

3. Programmieraufgabe WiSe 2015/16 Abgabe bis zum 21.01.2016

Implementieren Sie den in Kapitel 6.5 beschriebenen Preflow-Push-Algorithmus zur Bestimmung maximaler Flüsse. Realisieren Sie dabei möglichst viele der folgenden Varianten zur Festlegung der Reihenfolge, in welcher die Push- und Lift-Operationen ausgeführt werden:

- FIFO-Preflow-Push,
- Highest-Label-Preflow-Push,
- Excess-Scaling,
- eine eigene Variante.

Wenden Sie die implementierten Varianten auf die Testinstanzen an, die wir auf der Übungsseite bereitstellen. Halten Sie die Laufzeiten in einer Tabelle fest und interpretieren Sie diese Ergebnisse.

- Schreiben Sie Ihren Code in C oder C++ (die neuen Standards von 2011 sind gestattet). Er soll mit einem aktuellen GCC-Compiler kompilierbar sein. Sie dürfen einfache Datenstrukturen aus den Standardbibliotheken (Listen, Vektoren, usw.), jedoch keine externen Graphenbibliotheken verwenden.
- Der Aufruf des Programms soll in der Konsole mit `./programm inputdatei` erfolgen. Das Ergebnis soll auf dem Bildschirm ausgegeben werden. Zwischenschritte werden nicht benötigt. Die Ergebnisanzeige muss den Wert des maximalen Flusses enthalten. Außerdem sind Laufzeit sowie Statistiken zur Instanz und zum Verlauf des Algorithmus denkbar. Der maximale Fluss selbst muss berechnet, aber nicht ausgegeben werden. Er kann also beim Beenden des Programms gelöscht werden.
- Sie erhalten eine Textdatei pro Problem Instanz $D = (V, A, c)$. In der ersten Zeile stehen die beiden Zahlen $|V|$ und a (Bedeutung siehe unten). In Zeile 2 stehen die beiden Zahlen s und t , also die Knotennummern von Quelle bzw. Senke. In Zeile 3 bis $a + 2$ stehen jeweils vier ganzzahlige Werte: $u \ v \ c_{uv} \ c_{vu}$. Jede dieser Zeilen spezifiziert die Kapazitäten der Kanten zwischen den beiden Knoten, die zu Beginn der Zeile genannt werden. Die Kapazitäten c sind allesamt nicht-negativ, können also insbesondere 0 sein. Als Knotennummern kommen nur Zahlen zwischen 1 und $|V|$ vor. Mehrere Zahlen pro Zeile sind durch einfache Leerzeichen getrennt. Beispiel:

```
4 4
1 4
1 2 5 0
1 3 2 0
2 3 5 3
3 4 7 0
```

Das Netzwerk hat 4 Knoten, Quelle ist 1, Senke ist 4. Es gibt 5 Kanten: $(1, 2)$, $(1, 3)$, $(2, 3)$, $(3, 2)$, $(3, 4)$ mit Kapazitäten 5, 2, 5, 3, 7.

- Es kann in Gruppen aus bis zu drei Personen abgegeben werden.
- Laden Sie Ihren Code in Form einer einzigen Quelldatei (`nachnamen.c` bzw. `nachnamen.cpp`) über das Moodle-Interface hoch. Schreiben Sie Ihre vollständigen Namen sowie sämtliche Hinweise an Frau Kolb als Kommentar oben in diese Datei. Geben Sie Ihre Laufzeitexperimente schriftlich in der Vorlesung ab.