Technische Universität München
Institut für Informatik
Lehrstuhl für Effiziente Algorithmen
Prof. Dr. Angelika Steger

Wintersemester 2001/02
Übungsblatt 4
13. November 2001

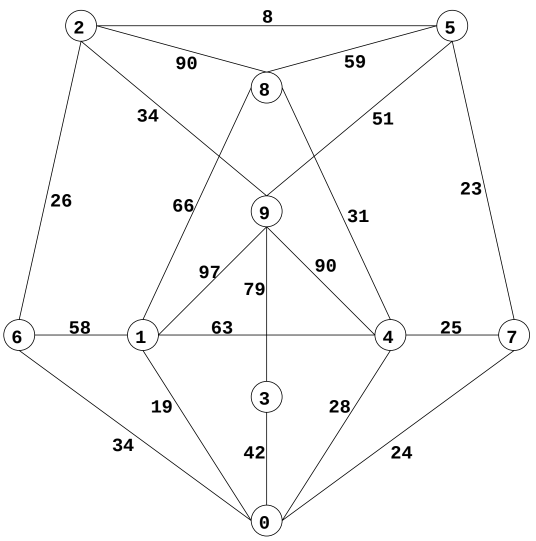
Effiziente Algorithmen und Datenstrukturen I

1. Kontrollblatt !

Abgabetermin: Dienstag, den 20. November 2001 (in der Übung)

Hausaufgabe 1 20 Punkte

Implementieren Sie den Algorithmus von Dijkstra mit Hilfe von *2-Level Buckets*, *Radix Heaps* oder *Fibonacci Heaps*.
Erlaubt sind als Programmiersprachen *C/C++* und *Java*. Benutzen Sie in beiden Fällen keine Datenstruktur-Bibliotheken, (d.h. in *C++* keine Standard Template Library und in *Java* keine Collection Classes oder ähnliches). Benutzen Sie in *C++* keine Templates und kein Operator-Overloading. Ihr Programm sollte in der Sun-Halle ohne Fehler kompilieren.
Die Eingabe des Graphen erfolgt in Form einer ASCII Datei mit einfachem Format: Die Erste Zeile enthält die Anzahl der Knoten, die Zweite Zeile enthält die Anzahl der Kanten. Die dritte Zeile enthält zwei durch Leerzeichen getrennte Zahlen, die den Start- und Zielknoten der Wege-Suche bestimmen. Jede weitere Zeile enthält die Beschreibung einer Kante in Form von drei Zahlen, nämlich Startknoten, Endknoten und Gewicht, jeweils durch ein Leerzeichen getrennt.
Als Ausgabe soll die Entfernung zwischen Start- und Zielknoten ausgegeben werden. Beispiel:




```
10
20
0 9
0 1 19
0 3 42
0 4 28
0 6 34
0 7 24
1 4 63
1 6 58
1 8 66
1 9 97
2 5 8
2 6 26
2 8 90
2 9 34
3 9 79
4 7 25
4 8 31
4 9 90
5 7 23
5 8 59
5 9 51
```

Ausgabe: 89

Diesen und weitere Testgraphen findet man auf den Web-Seiten der Übung.

Hausaufgabe 2 10 Punkte

Die Radix Heaps, die in der Vorlesung besprochen wurden, setzen die Grenzen der Buckets mit Hilfe von zweier Potenzen, d.h. bei der Initialisierung werden die Grenzen der Buckets auf 0, 2^0, 2^1, ... gesetzt.

- Warum kann man die Schranke von $\mathcal{O}(k \log C + l + m)$ für k *Insert*, l *DecreaseKey* und m *ExtractMin* Operationen nicht aufrecht erhalten, wenn man statt dessen dreier Potenzen benutzt (d.h. 0,3^0,3^1,...) ?
- Beschreiben Sie, wie man Radix Heaps erweitern muß, damit man höhere Potenzen als Grenzen benutzen kann, (d.h. 0, k^0, k^1, k^2, ...). Was ergibt sich dann für die Laufzeit ?

  (Tip: Man kann die Buckets in mehrere Segmente aufteilen. Wie ?)

Hausaufgabe 3 10 Punkte

In der Vorlesung wurde beschrieben, daß man bei einer Union Find Struktur, die *Union* Operation durchführt, in dem man den „kleineren“ der beiden Bäume an die Wurzel des größeren hängt. In der Vorlesung war „kleiner“ als der Baum der weniger Knoten entält definiert. Zeigen Sie:

- Das Lemma aus der Vorlesung: Jeder in der Union-Find-Struktur enthaltene Baum mit Höhe h hat mindestens 2^h Knoten. (Wobei ein einzelner Knoten Höhe 0 hat.)
- Warum gilt das Lemma auch, wenn man immer den Baum mit der kleineren Höhe an den Baum mit der größeren Höhe hängt ?
- Was passiert wenn man immer den Baum dessen Wurzel weniger (direkte) Kinder hat an den anderen Baum hängt ?