Programmiertechnik II

Sommersemester 2015 – Aufgabenblatt 6

Hasso-Plattner-Institut
Fachgebiet Computergrafische Systeme
HPI

Anmerkungen zu Aufgabenblatt 6

Der zu verwendende Programmrahmen steht als Zip-Archiv `uebung6.zip` im Moodle unter der Adresse `https://moodle.hpi3d.de/course/view.php?id=75` zum Download zur Verfügung.

Die Aufgaben sollen in den benannten und im Programmrahmen zumeist gekennzeichneten Bereichen implementiert werden. Bevor Sie mit der Lösung beginnen, lesen Sie das gesamte Aufgabenblatt und überprüfen Sie, ob der Programmrahmen auf Ihrem System fehlerfrei kompiliert, die Datensätze enthalten sind und sich alle Programme ausführen lassen. Wir wünschen viel Spaß und Erfolg bei der Bearbeitung der Aufgaben!

Aufgabe 6.1: Prim-Algorithmus (7 Punkte $^{1+1+5}$)

Ein minimaler Spannbaum (Minimal Spanning Tree, *MST*) verbindet alle Knoten eines Graphen in Form eines zyklenfreien Teilgraphen (ein Baum) sodass auf den Kanten definierte Kosten minimiert werden. Implementieren Sie den Prim-Algorithmus zur Berechnung des MST für einen Graphen, der aus einer Entfernungstabelle des Programmrahmens `prim.cpp` generiert wird. Es sind zwei Entfernungstabellen enthalten (Vorlesungsbeispiel und Entfernungen der 20 größten deutschen Städte).

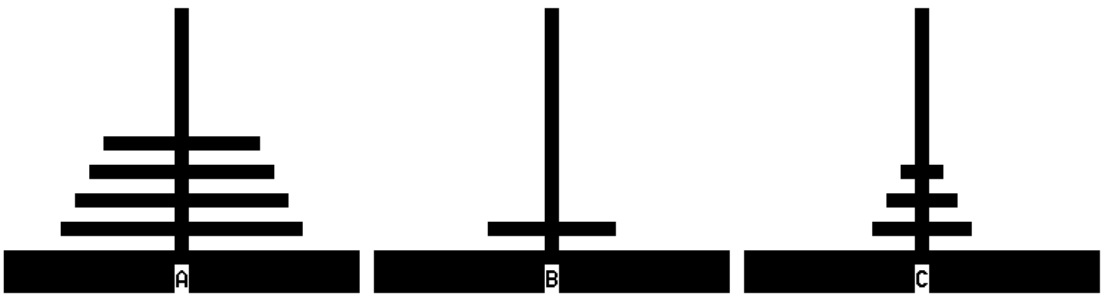
a) Implementieren Sie die Funktion `createGraph()`, die aus der Entfernungstabelle mittels `weights` die Vertices und Edges des Graphen erzeugt und diese in den per Referenz übergebenen Containern speichert. Nutzen Sie die vorgegebenen Datenstrukturen `Vertex` und `Edge`. Testen Sie diese Funktion für die beiden Entfernungstabellen des Programmrahmens.

b) Implementieren Sie die Funktion `totalWeight()` zur Berechnung der Gesamtkosten einer Kantenliste. Nutzen Sie dazu einen Lambda-Ausdruck in Verbindung mit `std::for_each`.

c) Implementieren Sie den Prim-Algorithmus in der Funktion `prim()`, die zunächst den Graphen generiert und die Kanten des MST sowie die damit verbundenen Kosten ausgibt. Verifizieren Sie insbesondere die Ausgabe für das Vorlesungsbeispiel.

Aufgabe 6.2: Türme von Hanoi (7 Punkte $^{2+2+3}$)

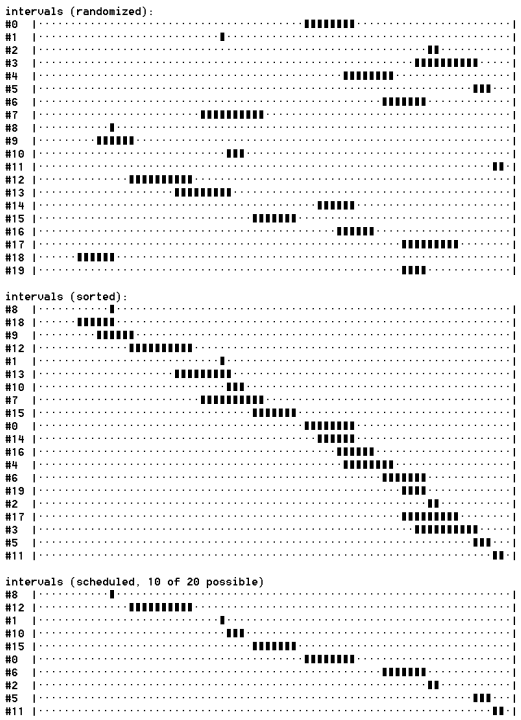
Der Türme-von-Hanoi-Algorithmus wurde in der Vorlesung vorgestellt. Erweitern diese Implementierung, indem Sie für die 3 Stapel den Zustand im Verlauf der Umlagerung verzeichnen. Der Nutzer soll auf der Konsole jedes Umstapeln mit einem Tastendruck auslösen und eine aktualisierte Sicht auf die drei Stapel erhalten. Geben Sie die Stapel mittels ASCII-Grafik aus und nutzen Sie zur Löschung der Konsole das Kommando `std::system()`. Achten Sie bei der Darstellung darauf, dass Elemente, die sich nach einem Zug nicht bewegt haben an gleicher Stelle erscheinen. Nutzen Sie zur Umsetzung die Funktion `print()`. Beispiel für die Ausgabe der Stapel bei `N = 8` Scheiben:




```
Next Move: C to B
```

Aufgabe 6.3: Greedy Scheduling (5 Punkte [2+3])

Im Programmrahmen `schedule.cpp` werden per Zufall Zeitintervalle erzeugt. Jedes Intervall besitzt einen Identifier, einen Startzeitpunkt und einen Endzeitpunkt. Implementieren Sie den Greedy-Scheduling-Algorithmus, der eine maximale Anzahl von nicht überlappenden Zeitintervallen auswählt. Geben Sie, ähnlich der folgenden Abbildung, die ermittelten Intervalle auf der Konsole grafisch aus, sodass der gesamte Zeitraum und die jeweiligen Zeitintervalle unterscheidbar sind[1].



Allgemeine Hinweise zur Bearbeitung und Abgabe

- Die Aufgaben sollen maximal zu zweit bearbeitet werden; Ausnahmen sind nicht vorgesehen. Je Gruppe ist nur eine Abgabe notwendig.
- Bitte reichen Sie Ihre Lösungen bis **Montag, den 29. Juni um 09:00 Uhr** ein.
- Zur Prüfungszulassung muss ein Aufgabenblatt zumindest bearbeitet werden und alle weiteren Aufgabenblätter mit mindestens 50% der Punkte bewertet werden. Aufgaben mit Klausurpunkten werden 1:1 berücksichtigt.
- Die Aufgaben können auf allen wesentlichen Plattformen (Windows, Linux, OS X) bearbeitet werden. Lesen Sie bei Fragen zum Kompilieren und Ausführen bitte genau die den Archiven jeweils beiligende README.TXT. Bestehen weitere Fragen und Probleme, kontaktieren Sie einen der Betreuer direkt oder nutzen Sie das Forum im Moodle.
- Die Durchsicht zur Bewertung kann sowohl auf Windows, Linux oder OS X erfolgen. Es wird folglich eine **plattformunabhängige Lösung** der Aufgaben erwartet.
- Archivieren (Zip) Sie zur Abgabe Ihren bearbeiteten Programmrahmen und ergänzen Sie Ihre Matrikelnummer(n) im Bezeichner des Zip-Archivs entsprechend im folgenden Format: **`uebung6_matrikNr1_matrikNr2.zip`**. Beachten Sie, dass dabei nur die vollständigen Quelltexte, die CMake-Konfiguration sowie eventuelle Zusatzdaten gepackt werden (alles was im gegebenen Programmrahmen entsprechend vorhanden ist). Senden Sie uns keinesfalls Kompilate und temporäre Dateien (*.obj, *.pdb, *.ilk, *.ncb etc.). Testen Sie vor dem Verschicken, ob die Projekte aus den Zipdateien fehlerfrei kompiliert und ausgeführt werden können.
- Reichen Sie Ihr Zip-Archiv im Moodle ein: `https://moodle.hpi3d.de/course/view.php?id=75`.

[1] Sie können für die Ausgabe 'Extended ASCII Codes' nutzen (http://www.rapidtables.com/code/text/ascii-table.htm).