

Anmerkungen zu Aufgabenblatt 5

Der zu verwendende Programmrahmen steht als Zip-Archiv `uebung5.zip` im Moodle unter der Adresse <https://moodle.hpi3d.de/course/view.php?id=137> zum Download zur Verfügung.

Aufgabe 5.1: Prim-Algorithmus (8 Punkte ²⁺¹⁺⁵)

Ein minimaler Spannbaum (Minimal Spanning Tree, *MST*) verbindet alle Knoten eines Graphen in Form eines zyklensfreien Teilgraphen (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).

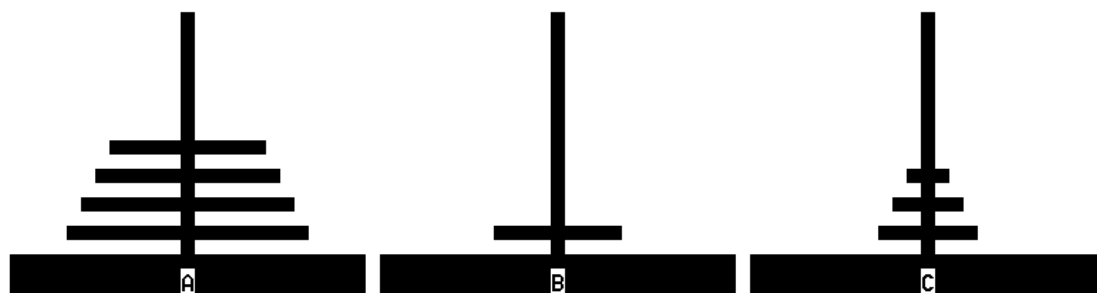
- Implementieren Sie die Funktion `createGraph`, die aus der Entfernungstabelle mittels `getWeight` die Vertices und Edges erzeugt und diese in den per Referenz übergebenen Graphen speichert. Nutzen Sie die vorgegebenen Datenstrukturen `Graph`, `Vertex` und `Edge`. Testen Sie diese Funktion für die beiden Entfernungstabellen des Programmrahmens.
- Implementieren Sie die Funktion `totalWeight` zur Berechnung der Gesamtkosten einer Kantenliste.
- Implementieren Sie den Prim-Algorithmus in der Funktion `prim`. Diese wird in der `main`-Methode aufgerufen, nachdem zunächst der Graph generiert wurde und im Anschluss werden die Kanten des MST sowie die damit verbundenen Kosten ausgegeben. Überprüfen Sie insbesondere die Ausgabe des Vorlesungsbeispiels.

Aufgabe 5.2: Türme von Hanoi (8 Punkte)

Der Türme-von-Hanoi-Algorithmus wurde in der Vorlesung vorgestellt. Erweitern die in `toh.cpp` vorgegebene Implementierung, indem Sie für die drei Stapel den Zustand im Verlauf der Umlagerungen bestimmen.

Auf der Konsole soll jedes Umstapeln mit einem Tastendruck auf die Eingabetaste ausgelöst und eine aktualisierte Sicht auf die drei Stapel und der zuletzt durchgeführte Zug dargestellt werden. Geben Sie die Stapel mittels ASCII-Grafik aus und nutzen Sie zum Löschen der Konsole das Kommando `std::system`. Die Gesamtanzahl der Züge soll am Ende ausgegeben werden. Stellen Sie sicher, dass bis zu 10 Scheiben dargestellt werden können.

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 für $N = 8$ Scheiben:



Last Move: A to B

Aufgabe 5.3: Greedy Scheduling (5 Punkte)

Im Programmrahmen `schedule.cpp` werden per Zufall Zeitintervalle erzeugt. Jedes Intervall I_n besitzt einen Identifier, einen Startzeitpunkt S_n und einen Endzeitpunkt E_n mit $I_n = [S_n, E_n]$. 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 sind¹.

```

intervals (randomized):
#0 : .....
#1 : .....
#2 : .....
#3 : .....
#4 : .....
#5 : .....
#6 : .....
#7 : .....
#8 : .....
#9 : .....
#10 : .....
#11 : .....
#12 : .....
#13 : .....
#14 : .....
#15 : .....
#16 : .....
#17 : .....
#18 : .....
#19 : .....

intervals (sorted):
#3 : .....
#11 : .....
#10 : .....
#9 : .....
#17 : .....
#13 : .....
#4 : .....
#7 : .....
#2 : .....
#0 : .....
#15 : .....
#1 : .....
#12 : .....
#19 : .....
#8 : .....
#16 : .....
#5 : .....
#6 : .....
#18 : .....
#14 : .....

intervals (scheduled, 10 of 20 possible)
#3 : .....
#11 : .....
#10 : .....
#17 : .....
#13 : .....
#4 : .....
#7 : .....
#2 : .....
#0 : .....
#15 : .....

```

Zusatzaufgabe: Klausuraufgabe (1 Bonuspunkt)

Formulieren Sie eine mögliche Klausuraufgabe zu den Inhalten aus Kapitel 5 (Algorithmenarten) der Vorlesung und skizzieren Sie kurz eine zugehörige Lösung. Die Aufgabe soll nicht einfach konkretes Wissen, bzw. Folieninhalte abfragen, sondern das Gesamtverständnis eines Sachverhalts und die Befähigung zur Übertragung auf neue Probleme zum Gegenstand haben.

¹Sie können für die Ausgabe „Extended ASCII Codes“ nutzen (<http://www.rapidtables.com/code/text/ascii-table.htm>).

Allgemeine Hinweise zur Bearbeitung und Abgabe

- Die Aufgaben können allein oder zu zweit bearbeitet werden. Je Gruppe ist nur eine Abgabe notwendig.
- Bonuspunkte können innerhalb des Übungsblattes verlorene Punkte ausgleichen. Sie sind nicht auf andere Übungsblätter übertragbar und werden nicht über die reguläre Punktzahl hinaus angerechnet.
- Bitte reichen Sie Ihre Lösungen bis spätestens **Donnerstag, den 2. Juli um 12:00** ein.
- Die Implementierung kann auf einer üblichen Plattform (Windows, Linux, OS X) erfolgen, darf aber keine plattformspezifischen Elemente enthalten, d. h. die Implementierung soll plattformunabhängig entwickelt werden.
- Die Lösungen von theoretischen Aufgaben können als Textdatei oder PDF abgegeben werden. Benennen Sie die Dateien im Format **<Aufgabenblatt>_<Aufgabe>_<Teilaufgabe>**.
- Bestehen weitere Fragen und Probleme, kontaktieren Sie den Übungsleiter oder nutzen Sie das Forum im Moodle.
- Archivieren Sie zur Abgabe Ihren bearbeiteten Programmrahmen als Zip-Archiv und ergänzen Sie Ihre Namen im Bezeichner des Zip-Archivs im folgenden Format: **uebung5_vorname1_nachname1_vorname2_nachname2.zip**. Beachten Sie, dass dabei nur die vollständigen Lösungen sowie eventuelle Zusatzdaten gepackt werden (alle Dateien, die im gegebenen Programmrahmen vorhanden waren). Laden Sie keine Kompilate und temporären Dateien (*.obj, *.pdb, *.ilk, *.ncb, *.exe, etc.) hoch. Testen Sie vor dem Hochladen, ob die Abgabe fehlerfrei kompiliert und ausgeführt werden kann.
- Reichen Sie Ihr Zip-Archiv im Moodle ein:
<https://moodle.hpi3d.de/course/view.php?id=137>.