



Algorithmische Mathematik I

Wintersemester 19/20

Prof. Dr. J. Gedicke

Johannes Rentrop und Jannik Schürg



Programmieraufgabenblatt 8. Abgabedatum: 09.12.2019–13.12.2019

Programmieraufgabe 1. (Kürzeste Wege)

Schon wieder Stromausfall in der Mensa! Da sich der Einsatz von antiquitierten Waagen nicht bewährt hat, bleibt die Mensa diesmal geschlossen. Den Studierenden bleibt also nichts anderes übrig, als den weiten Weg zur Nassemensa anzutreten. Um als erste da zu sein und so einen Vorteil gegenüber denjenigen zu erlangen, die die Wegfindung von gewissen kommerziellen Anbietern nutzen, wollen wir in dieser Aufgabe selbstständig den kürzesten Weg vom Hörsaalgebäude zur Nassemensa berechnen.

Dazu finden Sie auf der Vorlesungswebsite eine Datei, die das Kartenmaterial der Bonner Innenstadt als Graph im richtigen Format für die Graphenklasse enthält. Die Kantengewichte sind jeweils die Abstände der verbundenen Knoten in km (Wir wollen annehmen, dass alle Studierenden zu Fuß unterwegs sind, sodass es nur auf die reine Länge des Weges ankommt). Außerdem finden Sie zwei Dateien, die jeweils eine Liste mit den x - bzw. y -Koordinaten der Knoten enthalten. In Abbildung 1 sehen Sie eine Darstellung der Karte. Schreiben Sie nun ein Programm, das folgende Funktionalität enthält.

- a) Der Graph sowie die Knotenkoordinaten sollen aus den Dateien eingelesen werden. Letztere können entweder jeweils in einem Array gespeichert werden, oder Sie erweitern die Klasse `Node` so, dass sie die Koordinaten in zwei *Member*-Variablen speichert.
- b) Es soll ein Bild in eine Datei geschrieben werden, das die Stadtkarte darstellt. Das Bild soll wieder durch ein Array der Länge *Breite* mal *Höhe* vom Typ `uint8_t`, das heißt mit Graustufenwerten, repräsentiert werden. Zum Anlegen und Ausgeben des Bildes können Sie wie in Aufgabe 2 auf Programmieraufgabenblatt 5 verfahren. In dieses Bild sollen nun alle Kanten des Graphen eingezeichnet werden. Zum zeichnen einer Linie zwischen zwei gegebenen Punkten können Sie die Funktion `draw_line` aus dem Header „drawline.h“, den Sie auf der Vorlesungswebsite finden, verwenden. Eine Beschreibung ihrer Parameter befindet sich in der Datei „drawline.cpp“.
- c) Der Code soll eine Funktion enthalten, die den *Dijkstra*-Algorithmus implementiert, um den kürzesten Weg zwischen gegebenem Start- und Zielknoten sowie dessen Länge zu berechnen. Der Weg kann selbst als Graph gespeichert werden.
- d) Als letztes soll der kürzeste Weg zwischen Knoten 1758 (Eingang zum Hörsaalzentrum) und Knoten 584 (Ecke Nassemensa) berechnet werden. Außerdem soll dieser mit einem anderen Farbwert und/oder einer höheren Linienstärke in das auszugebende Bild gezeichnet werden. Zusätzlich soll die berechnete Distanz ausgegeben werden.

(20 Punkte)



Abbildung 1: Die Bonner Innenstadt