

Datenstrukturen und Algorithmen

Louis Graßmann (22 47 748)

Joshua Stripp (22 44 198)

Simon Hain (22 41 762)

Aufgabe 1 – Wörterbuch

Beschreibung

Die Stadt Schilda floriert und hat sich zu einem ordentlichen Publikumsmagneten entwickelt. Leider kommt es immer wieder zu Problemen, da die Touristen die Bürger Schildas nicht verstehen – kein Wunder, denn die Sprache Schildas ist sehr ungewöhnlich: Der Sprachschatz Schildas ist immens. Die Bürger Schildas sind sehr kreativ und erfinden täglich neue Wörter und erweitern damit ihren Wortschatz. Sie sollen nun ein Wörterbuch entwickeln, welches sehr effizient neue Wörter aufnehmen und die Bedeutung von Wörtern suchen kann.

Welche Daten sollen verarbeitet werden?

- Übersetzung der Schilda-Wörter zu den Deutschen

Was sind die Eingaben?

- Neue Wortverbindungen Schilda <-> Deutsch
 - `setMeaningSchildaGerman(SchildaWort, DeutschesWort)`
- Schilda-Wort, das übersetzt werden soll
 - `getSchildaToGerman(SchildaWort)`
- Deutsches-Wort, das übersetzt werden soll
 - `getGermanToSchilda(DeutschesWort)`

Was sind die Ausgaben?

- Die jeweiligen Übersetzungen

Welcher Algorithmus eignet sich?

- Kein besonderer Algorithmus

Welche Datenstruktur eignet sich?

- HashMaps mit den Wörtern der einen Sprache als Key und den Wörtern der anderen Sprache als Value
- Eine HashMap mit Schilda-Wörtern als Key, eine zweite mit den deutschen Wörtern als Key

Was ist die Laufzeit?

- HashMaps haben eine durchschnittliche Laufzeit von $O(1)$, in seltenen Fällen, kann die Laufzeit aber auch $O(n)$ betragen

Aufgabe 2 – Parkhaussystem

Beschreibung

Mit den vielen Touristen, die in die Graphschaft kommen, hat sich auch die Parkplatzsituation drastisch verschlechtert. Die Stadt Schilda plant nun, ein kundenfreundliches Parkhaus zu bauen.

Jeder Parkplatz hat eine eindeutige Kennung, die Parkplatznummer. Bei Einfahrt in das Parkhaus können die Kunden einen bestimmten Parkplatz reservieren. Sollte dieser Parkplatz nicht frei sein, schlägt die Software den nächsten freien Parkplatz vor.

Die Graphschaft Schilda beauftragt Sie, ein solches Parkplatzzuordnungssystem zu entwickeln. Dabei soll effizient der nächste freie Parkplatz berechnet werden.

Welche Daten sollen verarbeitet werden?

- Größe des Parkhauses
- Welche Parkplätze sind belegt?
- Welche Parkplätze sind frei?

Was sind die Eingaben?

- Parkplatz Reservierung / Buchungen ggfs. mit gewünschter Parkplatznummer
 - reserve(Nummer)
 - reserve()
- Parkplatzfreigabe
 - unreserve(Nummer)

Was sind die Ausgaben?

- Parkplatznummer, die reserviert wurde
- Es wird in der Konsole gedruckt, welche reserviert wurde, bzw. wenn das Parkhaus voll ist

Welcher Algorithmus eignet sich?

- Kein besonderer Algorithmus

Welche Datenstruktur eignet sich?

- Int-Array, in dem gespeichert wird, welche Parkplätze noch frei sind

Was ist die Laufzeit?

- Im Worst-Case $O(n)$, im Durchschnitt schneller, da Parkplatz mit kleinster Nummer gespeichert wird

Mögliche Verbesserungen

- Statt nur die minimale freie Nummer zu speichern könnte man auch alle Nummern in einem MinHeap speichern.
- Dieser wurde auch implementiert, das Lösungsprogramm war zu diesem Zeitpunkt jedoch schon erstellt worden

Aufgabe 3 – Empfehlungssystem

Beschreibung

In den letzten Jahren wurde die Graphschaft Schilda zum Touristenmagneten. Die Stadt hat über die Jahr genau verfolgt, welche Touristen welche Stätten gemeinsam besuchen: beispielsweise besuchen Touristen, die den Burgfried besuchen häufig auch die historische Ausstellung im Schloss. Touristen, die eine architektonische Führung gebucht haben, gehen anschließend gern im Schlosskeller etwas trinken und so weiter, einige besuchen danach auch noch das Restaurant Hexenkeller. So konnte zwischen allen Sehenswürdigkeiten und Veranstaltungsorten eine mehr oder weniger starke Verbindung festgestellt werden (mit Werten zwischen 1 und 10 belegt). Die Stadt wünscht sich nun ein Empfehlungstool, das aufgrund individueller Wünsche (Kurzbesuch, Wochenendbesuch, offen für Neues, etc.) Empfehlungen für Freizeitaktivitäten gibt.

Welche Daten sollen verarbeitet werden?

- Sehenswürdigkeiten/Orte
- Verbindungen zwischen Sehenswürdigkeiten

Was sind die Eingaben?

- 2D-Int-Array, welchen einen ungerichteten, ungewichteten Graphen darstellt
 - Knoten = Sehenswürdigkeiten
 - Gewichtete Kanten = Verbindungswege, Gewichtung stellt Länge des Weges dar
- Knoten/Sehenswürdigkeit an dem Benutzer steht
- Nach was sucht Benutzer?

Was sind die Ausgaben?

- Empfehlung, basierend auf den obigen Daten
 - „Kurzzeittrip“: Es wird zufällige Sehenswürdigkeit innerhalb 3 Längeneinheiten empfohlen
 - „Langzeittrip“: Es wird zufällige Sehenswürdigkeit innerhalb 5 Längeneinheiten empfohlen
 - „Offen für neues“: Es wird zufällige Sehenswürdigkeit innerhalb 10 Längeneinheiten empfohlen

Welcher Algorithmus eignet sich?

- Dijkstra berechnet die kürzeste Distanz zwischen den Sehenswürdigkeiten und dem jetzigen Standort
- Basierend auf den errechneten Distanzen wird dann eine Sehenswürdigkeit empfohlen

Welche Datenstruktur eignet sich?

- Graph-Objekt mit Knoten und Kanten
- Selbst erstellt

Was ist die Laufzeit?

- $O(n^2)$

Aufgabe 4 – Winter in Schilda

Beschreibung

Im Winter sind die Wege in Schilda immer sehr verschneit, vereist und glatt. Damit die Touristen nun gut und sicher vom Parkhaus zu den vielen Sehenswürdigkeiten und historischen Plätzen kommen können, hat sich die Stadt Schilda etwas besonderes überlegt: Vom Parkhaus aus werden beheizte Wege eingerichtet, die die Straßen eisfrei halten.

Dieses Vorhaben ist sehr teuer: ein Meter Weg kostet mehrere 100 Schilda-Gulden. Die Stadt Schilda hat daher die Bitte an Sie, in Ihr Planungstool eine Funktionalität einzufügen, die basierend auf einem bestehenden Stadtplan die günstigste Auswahl an zu beheizenden Wegen identifiziert, so dass alle Plätze und Sehenswürdigkeiten über beheizte Wege erreicht werden können, die Kosten dabei aber möglichst gering gehalten werden.

Welche Daten sollen verarbeitet werden?

- Orte
- Verbindungen zwischen den Orten

Was sind die Eingaben?

- 2D-Int-Array, welchen einen ungerichteten, gewichteten Graphen darstellt
 - Knoten = Orte
 - Gewichtete Kanten = Verbindungswege, Gewichtung stellt Länge des Weges dar

Was sind die Ausgaben?

- Adjazenzmatrix, welche angibt, welche Wege beheizt werden sollten
- Die Matrix wird auch auf der Konsole ausgegeben

Welcher Algorithmus eignet sich?

- Prim-Algorithmus berechnet den minimalen Spannbaum, indem er immer den kürzesten Weg auswählt
- So wird, Stück für Stück, die insgesamt kürzeste Menge an Wegen ausgewählt, die alle Orte verbindet

Welche Datenstruktur eignet sich?

- Graph-Objekt mit Knoten und Kanten
- Selbst erstellt

Was ist die Laufzeit?

- $O(n^2)$

Aufgabe 5 – Bierlieferung

Beschreibung

Aufgrund der vielen Touristen in der Stadt, wurden laufend das gastronomische Restaurant erweitert. Da die Graphschaft Schilda große Freude an innovativen Ideen hat, hat sie ein Bierliefernetzwerk installiert, über das laufend Bier von der Brauerei zu den einzelnen Gaststätten und Restaurants geliefert wird.

Natürlich ist nicht jede Gaststätte und Kneipe direkt an die Brauerei angeschlossen, sondern teilt sich das Netzwerk mit anderen gastronomischen Stätten. Nun soll ein großer neuer Biergarten am Ortsrand eröffnet werden und auch mit Bier versorgt werden.

Hier stellt sich die Frage, ob das bestehende Netzwerk noch ausreicht, oder wo Engpässe entstehen und gegebenenfalls die Kapazität erweitert werden muss. – Hierzu ist zu beachten, dass jede gastronomische Stätte wie auch der neue Biergarten einen individuellen Bierbedarf haben, der auch geliefert werden soll.

Welche Daten sollen verarbeitet werden?

- Verbindungen zwischen Kneipen und der Brauerei
- Benötigte Menge an Bier der Kneipen
- Kapazitäten der Rohre

Was sind die Eingaben?

- 2D-Int-Array, welchen einen ungerichteten, gewichteten Graphen darstellt
 - Knoten 0 = Brauerei
 - Knoten 1 bis (n-1) = Kneipen
 - Knoten n = Supersenke
 - Gewichtete Kanten = Rohre, Gewichtung stellt Kapazität dar
 - Verbindungen zu Knoten n = Verbrauch der einzelnen Kneipen

Was sind die Ausgaben?

- flowGraph, der zeigt, wo wieviel Bier durchfließt, um den maximalen Flow zu erreichen
 - Dieser wird auch auf der Konsole ausgegeben
- Auf der Konsole wird außerdem ausgegeben, ob der Bedarf aller Kneipen gedeckt wurde

Welcher Algorithmus eignet sich?

- Ford-Fulkerson berechnet den max-Flow zwischen Brauerei -> Kneipen -> Supersenke

Welche Datenstruktur eignet sich?

- Graph-Objekt mit Knoten und Kanten
- Selbst erstellt

Was ist die Laufzeit?

$O(\text{Summe der Bedarfe})$

Aufgabe 6 – Projekt-Aufgaben

Beschreibung

In der Graphschaft Schilda gibt es viel zu tun. Die Graphschaft hat viele Mitarbeiter:innen mit unterschiedlichen Kompetenzen (Finanzen, Planung, Straßenbau, Festorganisation, und so weiter).

Viele Projekte sind zu bewältigen, viel mehr, als es Mitarbeitende gibt.

Gleichzeitig müssen einige Projekte von mehreren Mitarbeitenden gemeinsam bearbeitet werden um erfolgreich zu sein.

Die Graphschaft hat nun, um die Mitarbeitenden vor Überlastung zu schützen, die Policy aufgestellt, dass jede Mitarbeiterin und jeder Mitarbeiter an maximal 3 Projekten arbeiten darf.

Damit nun die Kräfte möglichst gut eingesetzt werden, alle Mitarbeiter:innen entsprechend ihrer Kompetenzen Projekten zugeordnet sind, aber kein Mitarbeiter/keine Mitarbeiterin mehr als 3 Projekte bearbeitet, bittet die Stadt Schilda Sie ein Planungstool für das Matching zwischen Mitarbeiter:innen und Projekten zu erstellen.

Welche Daten sollen verarbeitet werden?

- Minimale Anzahl an Mitarbeiter, die Projekte benötigen
- Welche Projekte kann ein einzelner Mitarbeiter ausführen
- Maximale Anzahl der Projekte pro Mitarbeiter

Was sind die Eingaben?

- 2D-Int-Array, welchen einen gerichteten, gewichteten Graphen darstellt
 - Knoten 0 = Superquelle
 - Knoten 1 bis (n-1) = Mitarbeiter und Projekte
 - Knoten n = Supersenke
 - Verbindungen von Knoten 0 = Anzahl der Projekte, die ein Mitarbeiter erledigen kann. Angabe hierfür war 3
 - Verbindungen zu Knoten n = Anzahl der benötigten Mitarbeiter, die ein Projekt benötigt
 - Restlichen Verbindungen = Mitarbeiter kann Projekt erledigen, Gewichtung immer 1

	Q	M1	M2	M3	P1	P2	M4	P3	...	S
Q	0	3	3	3	0	0	3	0	...	0
M1	0	0	0	0	1	0	0	1	...	0
M2	0	0	0	0	1	1	0	1	...	0
M3	0	0	0	0	0	0	0	1	...	0
P1	0	0	0	0	0	0	0	0	...	2
P2	0	0	0	0	0	0	0	0	...	1
M4	0	0	0	0	1	0	0	0	...	0
P3	0	0	0	0	0	0	0	0	...	2
...
S	0	0	0	0	0	0	0	0	...	0

Was sind die Ausgaben?

Die Ausgabe ist ein zweidimensionales Array. Jede Zeile entspricht ein Mitarbeiter. Jede Spalte entspricht ein Projekt. Falls ein Mitarbeiter einem Projekt zugeordnet ist, so beträgt der Wert in deren Kreuzfeld 1. Wenn ein Mitarbeiter dem jeweiligen Projekt nicht zugeordnet ist, da die benötigten Kompetenzen nicht übereinstimmen, so befindet sich in deren Schnittfeld eine 0.

Welcher Algorithmus eignet sich?

- Ford-Fulkerson berechnet die minimale Zuordnungen, der Bedarf aller Projekte wird gedeckt. Es können Mitarbeiter existieren, die nicht 3 Projekten zugewiesen sind. Da jedoch die minimale Anzahl

aller Projekte bereits gedeckt ist, können übrig gebliebene Kapazitäten der Mitarbeiter frei gewählt werden.

Welche Datenstruktur eignet sich?

- Graph-Objekt mit Knoten und Kanten
- Selbst erstellt

Was ist die Laufzeit?

$O(\text{Summe der minimalen Anzahl der Mitarbeiter})$

Aufgabe 7 – Aktivitätsdiagramm

Beschreibung

Weil es nicht nur viele Projekte gibt, sondern in den Projekten auch noch viel zu tun ist, wünscht sich die Graphschaft auch Unterstützung bei der Planung der einzelnen Aktivitäten in den Projekten. Die Erfahrung zeigt, dass ohne Planung, viele Projekte im Chaos enden.

Es gibt Aktivitäten, die erst ausgeführt werden können, wenn andere Aktivitäten bereits beendet sind, andere Aktivitäten hängen nicht voneinander ab. Aber wie die ganzen Aktivitäten in die richtige Reihenfolge bringen?

Wieder ein Auftrag für Sie: Sie werden beauftragt, das Planungstool so zu erweitern, dass damit auch die Aktivitätenreihenfolge für die einzelnen Projekte geplant werden kann.

Welche Daten sollen verarbeitet werden?

- Aktivitäten, welche vor anderen ausgeführt werden müssen

Was sind die Eingaben?

- 2D-Int-Array, welchen einen gerichteten, ungewichteten Graphen darstellt
 - Knoten = Aktivitäten
 - Kante von Knoten x zu Knoten y = Aktivität x muss vor Aktivität y ausgeführt werden

Was sind die Ausgaben?

- Reihenfolge, in der Aktivitäten ausgeführt werden sollten, damit korrekte Reihenfolge eingehalten wird

Welcher Algorithmus eignet sich?

- DFS besucht alle Knoten, fängt immer bei Knoten an, zu denen keine Kante führt (keine Aktivität muss vor dieser ausgeführt werden)

Welche Datenstruktur eignet sich?

- Graph-Objekt mit Knoten und Kanten
- Selbst erstellt

Was ist die Laufzeit?

- $O(|E|+|V|)$
- $|E|$ = Anzahl Kanten
- $|V|$ = Anzahl der Knoten

Aufgabe 8 – Eiszeit

Beschreibung

So kalt es im Winter ist, so warm ist Schilda im Sommer. Nun möchte die Graphschaft ihren Bürgern den Zugang zu leckerem, erfrischendem Eis erleichtern.

Die Stadt beschließt ein Grundrecht, dass jeder Bürger maximal einen Block zum nächsten Eisstand laufen muss.

Nun überlegt die Stadt, wie sie dieses Grundrecht umsetzen kann: An den Straßenkreuzungen sollen Eisstände so verteilt werden, dass an jeder Straßenkreuzung entweder ein Eisstand vorhanden ist oder man nur einen Block zur nächsten Straßenkreuzung laufen muss um dort einen Eisstand zu finden.

Nachdem die Stadt dieses Grundrecht beschlossen hat, stellt sie fest, dass da ja doch immense Kosten auf die Stadt zukommen, so dass zwar das Grundrecht durchgesetzt werden soll, aber mit der minimalen Anzahl von Eisständen.

Sie werden wieder beauftragt, ein Tool zu entwickeln, dass die besten Standorte für die Eisstände berechnet.

Welche Daten sollen verarbeitet werden?

- Orte
- Verbindungen zwischen den Orten

Was sind die Eingaben?

- 2D-Int-Array, welchen einen ungerichteten, ungewichteten Graphen darstellt
 - Knoten = Orte
 - Kanten = Verbindungswege

Was sind die Ausgaben?

- Liste der Knoten, an denen Eisstände aufgebaut werden sollten
 - Wird auch auf Konsole ausgegeben

Welcher Algorithmus eignet sich?

- Set Cover Algorithmus berechnet die minimale Menge der Knoten, so dass jeder Knoten maximal eine Verbindung entfernt ist

Welche Datenstruktur eignet sich?

- Graph-Objekt mit Knoten und Kanten
- Selbst erstellt

Was ist die Laufzeit?

- $O(|E| + |V|)$
- $|E|$ = Anzahl Kanten
- $|V|$ = Anzahl der Knoten

Aufgabe 9 – Historische Funde

Beschreibung

Beim Ausheben der Wege während des Straßenbaus wurde ein antiker Feuerwerksplan gefunden. Die Lage der pyrotechnischen Effekte und die Zündschnüre sind noch sehr gut zu erkennen.

Wie aber ist die Choreographie des Feuerwerks? In welcher Reihenfolge zünden die Bomben? Können Sie den Bürgern der Graphschaft Schilda helfen? (Unter der Annahme, dass die Zündschnur immer mit gleichbleibender Geschwindigkeit abbrennt...)

Welche Daten sollen verarbeitet werden?

- Feuerwerkskörper
- Zündschnüre zwischen den Feuerwerkskörper

Was sind die Eingaben?

- 2D-Int-Array, welchen einen ungerichteten, gewichteten Graphen darstellt
 - Knoten = Feuerwerkskörper
 - Gewichtete Kanten = Zündschnüre zwischen den Feuerwerkskörper. Gewichtung stellt die Zünddauer dar
- Feuerwerkskörper 0 wird zuerst gezündet

Was sind die Ausgaben?

- Reihenfolge, in denen die Feuerwerkskörper explodieren

Welcher Algorithmus eignet sich?

- Dijkstra berechnet die Zünddauer zwischen Feuerwerkskörper 0 und den anderen Feuerwerkskörper

Welche Datenstruktur eignet sich?

- Graph-Objekt mit Knoten und Kanten
- Selbst erstellt

Was ist die Laufzeit?

- $O(n^2)$

Benutze (selbst implementierte) Datenstrukturen

- Graph
 - Vertex: Stellt Knoten dar
 - Enthält eine Liste an Kanten, die reingehen (unbenutzt)
 - Enthält eine Liste an Kanten, die rausgehen
 - Edge: Stellt Kanten dar
 - Zeigt „von“ und „zu“ Knoten an
 - Graph
 - Erstellt bei Initiierung alle Knoten und Kanten und weist diese zu
 - Enthält Informationen über alle Knoten und Kanten
- CustomArrayList
- CustomLinkedList
- Dijkstra
- FordFulkerson
- Prim
- SetCover
- TopologicalSort

Unbenutzte (selbst implementierte) Datenstrukturen

- CustomMinHeap
- CustomQueue
- CustomTreeNode
- search