

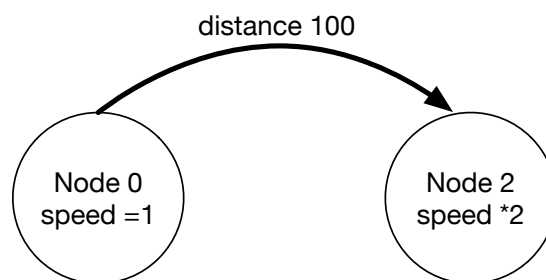


Abgabe 3 Prüfungsvorleistung

Finde den schnellsten Weg

Im Folgenden betrachten wir einen Graphen. Ziel ist es einen Graphen vom Startknoten zum Zielknoten so schnell wie möglich zu durchqueren. Zu diesem Zweck müssen Sie den schnellsten Weg/Pfad ermitteln. Schreiben Sie zu diesem Zweck ein Java-Programm und führen Sie dieses aus.

Aufbau:



- Jeder Knoten hat eine *ID*
- Der Startknoten hat die *ID=0*
- Der Zielknoten hat die *ID=21*
- Der Abstand zwischen zwei Knoten ist die *distance*
- Knoten sind mit Bögen verbunden
- Die Bögen sind gerichtet und können nur in diese Richtung navigiert werden
- Der *speed* gibt an wie schnell ein Bogen passiert werden kann
- Der Startknoten setzt den *speed* auf 1
- Jeder weitere Knoten verändert den *speed* (zB. *5, +3, -100)
- Die Zeit zum Passieren eines Bogens ist *distance/speed* (Zeitschritte)
- Ab einem *speed* von 100 fließt Reibung mit in die Berechnung ein:
 - $speed = speed - (speed * (-1 / (speed / 100) + 1.001))$
 - Der Reibungswiderstand fließt erst nach Verrechnung des *speed* mit dem entsprechenden Modifier am Knoten in die Berechnung ein

Als Ausgabe, geben Sie den schnellsten Pfad an. Geben Sie aus, welche Knoten Sie in welcher Reihenfolge besuchen, wie schnell Sie zum jeweils nächsten Knoten gelangen und wie lange jeder Schritt dauert.

Die Eingabedaten werden aus der Datei nodes.csv (enthält die Definition der Knoten) und arcs.csv (enthält die Definition aller Bögen) eingelesen. Je Zeile wird jeweils ein Knoten oder Bogen definiert (siehe Beispiel). Die beiden Dateien sind in OPAL verfügbar.

Hinweis: Knoten dürfen auch mehrfach besucht werden, sofern es die Richtung der Bögen zulässt.

Beispiel:

nodes.csv

| ID | modifiziert | Erklärung |
|----|----------------|---|
| 0 | 1 | setzt den <i>speed</i> auf 1 |
| 3 | *101 | multipliziert den <i>speed</i> mit dem Faktor 101 |
| 5 | * <i>speed</i> | quadriert den <i>speed</i> ($speed^2$) |
| 7 | +5 | erhöht den aktuellen <i>speed</i> um 5 |
| 15 | -3 | reduziert den aktuellen <i>speed</i> um 3 |
| 21 | | verändert den <i>speed</i> nicht |

arcs.csv

| from | to | distance | Erklärung |
|------|----|----------|--|
| 0 | 3 | 10 | Es gibt ein Bogen von Knoten 0 nach Knoten 3 mit einer <i>distance</i> 10 |
| 3 | 21 | 50 | Es gibt ein Bogen von Knoten 3 nach Knoten 21 mit einer <i>distance</i> 50 |

Beispielausgabe Ausgabe:

Die Suche hat: 3,56 Sekunden gedauert, dabei wurde folgender kürzester Weg gefunden:

- 1.) von 0 nach 3 mit *speed* 1 in 10 Zeitschritten
- 2.) von 3 nach 21 mit *speed* 101 in $\approx 0,5005$ Zeitschritten

Die Gesamtzeit beträgt: 10,5005 Zeitschritte

Abgabe:

Für die Abgabe nutzen Sie bitte das OPAL. In der OPAL-Gruppe finden Sie den Menüpunkt *Abgabe 3*. Laden Sie hier **EINE** ZIP-Datei pro Gruppe hoch. Benennen Sie diese Datei mit Ihrer Gruppennummer. Die ZIP-Datei, sollte mindestens folgende Elemente enthalten:

- Quellcode
- Lauffähiges .jar-Datei
- Ausgabe des Programms als .txt oder PDF (mit schnellstem Weg inkl. aller Teilschritte, Dauer der Suche und Summe der Zeitschritte)
- Liste der Gruppenmitglieder inklusive Matrikelnummern

Abgabe bis spätestens 24. Juni 23:00 MEZ!

Anmerkung: Programme, die lediglich die Textausgaben erzeugen, aber keine Objekte anlegen und verknüpfen, werden mit nicht bestanden bewertet.

Technische Universität Chemnitz, Institut für Informatik, Professur Softwaretechnik

Straße der Nationen 62, Böttcher Bau, B-Bau, B216B, D-09126 Chemnitz

Telefon: +49 (0) 371 / 531 – 39745, Internet: www.tu-chemnitz.de/informatik/ST/