



Minimale Spannbäume (Woche 9)

Eigenständige Vorbereitung:

Lies Erickson, Kapitel 7 (oder CLRS Kapitel 23) und schau dir das Video der Woche an.

Zeichenlegende:

- Schriftliche Aufgabe, die du fristgerecht in Moodle abgibst. In der Klausur wirst du alle Aufgaben schriftlich bearbeiten, daher ist das Feedback der Tutoren wichtig, damit du deine Schreibfähigkeiten verbessern kannst.
- Diese Art von Aufgabe musst du sicher können, um die Klausur zu bestehen.
- Diese Art von Aufgabe musst du weitgehend können, um die Klausur zu bestehen.
- Diese Art von Aufgabe musst du können, um eine gute Note zu erhalten.
- Diese Aufgabe ist als Knobelspaß gedacht, der das algorithmische Verständnis vertieft.

Aufgabe 9.1 (Algorithmen und Eigenschaften). Gegeben sei der Graph G in Abbildung 1.

- a) Führe Kruskals Algorithmus manuell auf G aus.
- b) Führe Prim's Algorithmus manuell auf G aus (Startknoten 0). Zeige den Inhalt der Prioritätswarteschlange zu jedem Zeitpunkt der Ausführung.
- c) Zeichne *alle* minimalen Spannbäume von G .
- d) Gib einen effizienten Algorithmus an, der einen (nicht notwendigerweise minimalen) Spannbaum berechnet.

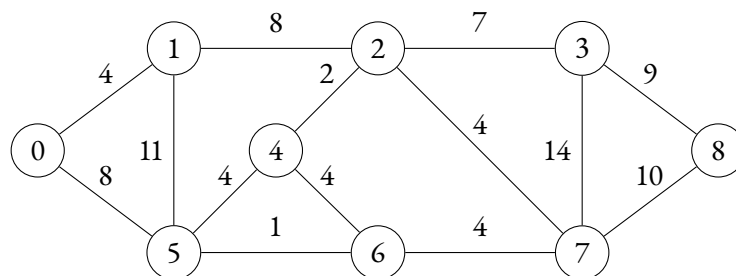


Abbildung 1: Ein ungerichteter, gewichteter Beispielgraph mit neun Knoten.

Aufgabe 9.2 (Kabel für den Riedberg). Nach viel hin und her, sowie einigen Planungsfehlern, hat Fachbereich 12 endlich ganze N Gebäude (nummeriert $1, \dots, N$) am Campus Riedberg fertiggestellt, dabei aber vergessen, Glasfaserkabel verlegen zu lassen. Top-Studentin Algolina entwickelt jetzt einen Plan, um alle Gebäude möglichst kostengünstig mit den neusten Glasfaserkabeln miteinander zu verbinden.

Um zwei Gebäude B_i und B_j mit Glasfaserkabeln zu verbinden, müssen Genehmigungen eingeholt werden, Löcher im Boden gegraben werden, und die Kabel verlegt werden. Algolina hat eine Liste mit M Preisen für das paarweise Verbinden zweier Gebäude B_i und B_j ; Gebäudepaare, die nicht in dieser Liste stehen, können nicht miteinander verbunden werden. Die Gebäude zählen als *miteinander verbunden*, wenn es einen Glasfaserkabel-Weg zwischen allen Gebäude-Paaren gibt (es muss also nicht unbedingt eine direkte Verbindung sein).

Gegeben sei die Liste an Preisen. Entwirf einen Algorithmus, der Algolina dabei hilft, den günstigsten Gesamtpreis zu berechnen, bei dem alle Gebäude miteinander verbunden sind. Du kannst annehmen, dass es möglich ist, alle Gebäude miteinander zu verbinden.

Aufgabe 9.3 (Graphen kappen 🗝️). Zur Bestimmung eines minimalen Spannbaums wird folgender Algorithmus vorgeschlagen: Starte mit einem gewichteten, zusammenhängenden Graphen G . Betrachte die Kanten von G mit absteigendem Gewicht (vom höchstem zu niedrigstem Gewicht). Für jede Kante gilt: Behalte diese Kante, falls das Entfernen dieser Kante den Graphen in zwei Zusammenhangskomponenten aufteilen würde. Entferne sie, falls dies nicht der Fall ist. Gib die finale Menge an Kanten zurück.

- Führe den Algorithmus manuell auf G aus Abbildung 1 aus.
- Argumentiere, warum der Algorithmus immer einen minimalen Spannbaum von G findet.

Aufgabe 9.4 (Eigenschaften von MSTs 🗝️). Sei G ein gewichteter Graph.

- Zeige, dass die Kante mit dem geringsten Gewicht in G immer im minimalen Spannbaum von G enthalten ist. Wie ist es mit der Kante mit dem höchsten Gewicht?
- Wir skalieren alle Kanten-Gewichte in G , indem wir sie mit einem Wert $c > 0$ multiplizieren. Wie sieht der minimale Spannbaum für den neuen Graphen aus?
- Zeige, dass es einen *eindeutigen* minimalen Spannbaum für den Graphen G gibt, wenn alle Kantengewichte verschieden sind. *Hinweis: Erwähne die Eigenschaften von minimalen Spannbäumen.*
- Kruskals Algorithmus kann verschiedene minimale Spannbäume für einen Graphen G zurückgeben, wenn wir zwei Kanten mit gleichem Gewicht anders sortieren. Zeige: Für jeden minimalen Spannbaum T in G gilt, dass es eine Möglichkeit gibt, die Kanten so zu sortieren, dass Kruskals Algorithmus T zurückgibt.

Aufgabe 9.5 (Maximale Spannbäume 🗝️). Gegeben sei ein gewichteter Graph G . Gib einen Algorithmus an, um einen maximalen Spannbaum von G zu bestimmen (also einen Spannbaum mit maximalem Gesamtgewicht). *Hinweis: Transformiere das Problem.*

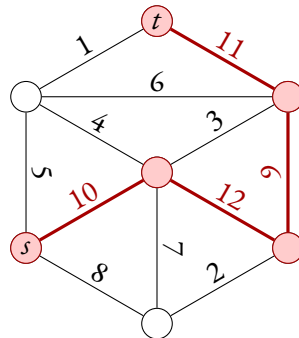
Aufgabe 9.6 (MSTs und kleine Änderungen 🗝️). Wir betrachten nun minimale Spannbäume auf Graphen deren Kantengewichte nicht paarweise verschieden sind.

- Zeige, dass die Schnitt- und Kreiseigenschaften auch dann gelten, wenn die Kantengewichte nicht eindeutig sind (die Eigenschaften müssen entsprechend umformuliert werden).
- Schließe daraus, dass Prim und Kruskals Algorithmus auch in diesem Fall funktionieren.





Aufgabe 9.7 (MSTs mit kleinen Kantengewichten 🗝️). Sei G ein gewichteter Graph mit n Knoten und m Kanten, sodass alle Kantengewichte Werte aus der Menge $\{1, 2, \dots, 10\}$ sind. Gib einen effizienten Algorithmus an, um einen minimalen Spannbaum von G zu bestimmen.

Aufgabe 9.8 (Flaschenhals ). For an English version of this exercise, see [Erickson, page 270].

Betrachte einen einfachen Weg zwischen zwei Knoten s und t in einem ungerichteten, gewichteten Graphen G . Die *Breite* dieses Wegs ist das minimale Gewicht, das auf den Kanten des Wegs vorkommt. Die Flaschenhals-Distanz zwischen s und t ist die Breite des breitesten einfachen Wegs von s nach t . (Wenn es keinen Weg von s nach t gibt, ist die Flaschenhals-Distanz $-\infty$. Die Flaschenhals-Distanz von s nach s ist ∞ .)



Die Flaschenhals-Distanz von s nach t ist 9.

- a)  Beweise, dass der *maximale* Spannbaum von G zwischen *jedem* Paar von Knoten einen breitesten Weg enthält.
- b)  Entwirf einen Algorithmus, der das folgende Problem in $O(n + m)$ Zeit löst: Gegeben ein ungerichteter, gewichteter Graph G mit zwei Knoten s und t sowie einem Gewicht W , ist die Flaschenhals-Distanz von s nach t höchstens W ?
- c)  Sei B die Flaschenhals-Distanz von s nach t .
 - i. Beweise, dass sich die Flaschenhals-Distanz von s nach t nicht ändert, wenn man eine beliebige Kante von Gewicht kleiner B löscht.
 - ii. Beweise, dass sich die Flaschenhals-Distanz von s nach t nicht ändert, wenn man eine beliebige Kante von Gewicht größer B *kontrahiert*. (Um eine Kante $e = \{u, v\}$ in G zu kontrahieren, wird die Kante e gelöscht und die beiden Endpunkte u und v werden zu einem neuen Knoten w verschmolzen. Falls dabei parallele Kanten entstehen, behalten wir nur die *schwerste* Kante zwischen jedem Paar von Knoten, und löschen die anderen. Der neue Graph heißt G/e .)
- d)  (sehr schwer) Entwirf einen Algorithmus, der in Zeit $O(n + m)$ einen breitesten Weg von s nach t berechnet. *Hinweis: Du darfst hierfür annehmen, dass man den Median von m Zahlen in Zeit $O(m)$ berechnen kann.*

Hinweise zur Abgabe. In den Aufgabenteilen a und c werden mathematische Beweise erwartet, und in den Aufgabenteilen b und d die übliche Struktur (grobe Idee, Pseudocode, Korrektheitsbeweis, Laufzeitanalyse).