Der linearzeit MST-Algorithmus

Maximilian Springenberg

1 Motivation

1.1 Konventionen

In dieser Ausarbeitung wird sich an die üblichen Konventionen zur Notation von Variablen in Graphen orientiert. So definieren wir einen Graphen als $G = (V, E)$, mit der Anzahl von Knoten $n = |V|$ und Kanten $m = |E|$. Ferner betrachten wir ungerichtete gewichtete Graphen und bezeichnen die Gewichtungsfunktion als $w : E \to \mathbb{R}$.

1.2 MST und MSF

Der minimale Spannbaum, oder auch MST, stellt einen azyklischen zusammenhängenden Teilgraph aus G, der alle Knoten verbindet und dessen Summe von Kantengewichte $\sum_{e \in E_{MST}} w(e)$ minimal ist dar.
Ist G selbst nicht zusammenhängend, so werden wir einen minimalen Spannwald als nächst beste Lösung betrachten. Ein minimaler Spannwald besteht aus einer Sammlung von minimalen Spannbäumen für die verbundenen Komponenten in G.

1.3 Warum ein nichtdeterministischer Anzatz von Vorteilen sein kann

Wir werden in dieser Ausarbeitung zum Kapitel 10.3 aus dem Buch 'Randomized Algorithms' von Motwani, R., Raghavan, P. einen randomisierten Ansatz für einen Algorithmus, der einen minimalen Spannbaum in erwarteter Linearzeit aproximiert betrachten.
Das MST Problem ist in P, und es sind bereits deterministische Algorithmen wie die von Prim, Kruskal, Borůvka bekannt , die mit einer Worst-Case Laufzeitschranke von $O(m * log(n))$ das Problem lösen. Zudem existiert der Algorithmus von Bernard Chazelle, für den eine Worst-Case Laufzeitschranke von $O(m * log\beta(m, n))$ bekannt ist, wobei mit $\beta(m, n) = \{i | log^{(i)} n \leq m/n\}$ die inverse Ackermann Funktion verwendet wird. $log^{(i)} n$ ist hierbei die i-te Anwendung von log auf n. Dementsprechend steigt die Funktion β so schwach, dass die aus ihr resultierenden Faktoren für die Worst-Case Laufzeit hinsichtlich der Größe von Graphen in der Praxis als nahezu konstant angesehen werden.
Wozu dient also ein nichtdeterministischer Algorithmus, der im Erwartungswert linear verläuft? - Die Antwort auf diese Frage kann sowohl durch die komplexe Implementierung des Algorithmus von Chazelle, als auch der Stabilität, bzw. Güte, des vorgestellten Algorithmus hinsichtlich seiner Laufzeit und nicht zuletzt durch das Betrachten des Algorithmus als akademisches Beispiel für kreative Laufzeitverbesserungen begründet werden.