Lars Porth

JOHANNES GUTENBERG UNIVERSITÄT MAINZ

JG U

Parmaetrisierte Algorithmen Seminar

16.11.2015



nhaltsverzeichnis

- 1 Einführung
 - Allgemeines

- 2 6.1 Dynamische Programmierung über Subsets
 - 6.1.1 Set Cover
 - 6.1.2 Steiner Tree

Einführung

inführung

- In Kapitel 6 geht es um verschiedene algorithmischen Werkzeuge, die zu sonst nicht so gut in irgend ein anderes Kapitel gepasst hätten.
- Als erstes geht es um Expotentialzeit dynamische Programmierung
- Als zweites geht es um Ganzahlige Lineare Programmierung

6.1 Dynamische Programmierung über Subsets

6.1 Dynamische Programmierung

■ TODO Hier kommt eine Beschreibung zu dynamischer Programmierung hin.

6.1.1 Set Cover

Überdeckung

Sei $\mathcal F$ eine Familie von Mengen in einem Universum $\mathcal U$. Für eine Unterfamilie $\mathcal F'\subseteq \mathcal F$ und einer Teilmenge $\mathcal U'\subseteq \mathcal U$ sagen wir, dass $\mathcal F'$ $\mathcal U'$ überdeckt, wenn jedes Element von $\mathcal U'$ zu mindestens einer Menge von $\mathcal F'$ gehört $(\mathcal U'\subseteq \bigcup \mathcal F')$.

Set Cover

Im Set Cover Problem ist eine Familie von Mengen $\mathcal F$ gegeben in einem Universum $\mathcal U$ und eine positive ganze Zahl k. Die Aufgabe ist zu überprüfen ob eine Unterfamilie, mit maximal k Elementen, $\mathcal F'\subseteq \mathcal F$ existiert, so dass $\mathcal F'$ $\mathcal U$ überdeckt.

Theorem 6.1

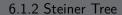
Theorem 6.1

Gegeben eine Instanz des SET COVER Problems $(\mathcal{U}, \mathcal{F}, k)$, kann die minimal mögliche Größe einer Unterfamilie $\mathcal{F}' \subseteq \mathcal{F}$, die \mathcal{U} überdeckt, in Zeit $2^{|\mathcal{U}|}(|\mathcal{U}|+|\mathcal{F}|)^{\mathcal{O}(1)}$ gefunden werden.

- lacksquare Sei $\mathcal{F} = \{F_1, F_2, ..., F_{|\mathcal{F}|}\}$
- Wir definieren die dynamische Programmierungs Tabelle wie folgt:
- Für jede Untermenge $X \subseteq \mathcal{U}$ und jede ganze Zahl $0 \le j \le |\mathcal{F}|$, definieren wir T[X,j] als die minimale Größe einer Untermenge $\mathcal{F}' \subseteq \{F_1,F_2,...,F_j\}$, die X überdeckt. Falls keine solche Teilmenge \mathcal{F}' existiert setzten wir $T[X,j] = +\infty$
- In unserem dynamischen Algorithmus Programm berechnen wir alle $2^{|\mathcal{U}|}(\mathcal{F}+1)$ Werte T[X,j]
- Basis Fall: $T[\emptyset, 0] = 0$, $T[X, 0] = +\infty$, für $X \neq \emptyset$
- Rekusiver Fall: Für $X \subseteq \mathcal{U}$ und $0 \le j \le |\mathcal{F}|$ zeigen wir, dass $T[X,j] = min(T[X,j-1],1+T[X\setminus F_j,j-1])$

- $T[X,j] = min(T[X,j-1],1+T[X\setminus F_j,j-1])$
- Dafür zeigen wir die Ungleichheit in beide Richtungen
- Für "≤": Sei $\mathcal{F}' \subseteq \{F_1, F_2, ..., F_j\}$ eine Familie mit minimaler Größe, die X überdeckt, wir unterscheiden zwei Fälle:
 - $F_j \notin \mathcal{F}'$, dann ist \mathcal{F}' auch ein zulässiger Kandidat für T[X,j-1]
 - $F_j \in \mathcal{F}'$, dann ist $\mathcal{F}' \setminus F_j$ ein zulässiger Kandidat für $T[X \setminus F_j, j-1]$
- Für ≥:
 - jeder zulässige Kandidat \mathcal{F}' für T[X,j-1] ist auch ein zulässiger Kandidat für T[X,j]
 - für jeden zulässigen Kandidaten \mathcal{F}' für $T[X \setminus F_j, j-1]$ gilt, dass $\mathcal{F}' \cup F_j$ ein zulässiger Kandidat für T[X, j] ist

- Mit diesem dynamischen Programm können wir für alle $X \subseteq U$ und $0 \le j \le |\mathcal{F}|$ den Wert T[X,j] in versprochener Zeit berechnen.
- Der Wert den wir suchen ist $T[\mathcal{U}, |\mathcal{F}|]$



Steiner Tree

Sei G ein ungerichteter Graph mit n Knoten und $K \subseteq V(G)$ von Endpunkten (terminals). Ein Steiner Tree für K in G ist ein zusammenhängender Teilgraph H von G, der K enthält ($K \subseteq H$)



Lemma 6.2

Lemma 6.2

Für jedes $D \subseteq K$ von einer Größe von mindestens 2 und jedes $v \in V(G) \setminus K$ gilt folgendes:

$$T[D, v] = \min_{\substack{u \in V(G) \setminus K \\ \emptyset \neq D' \subseteq D}} \{T[D', u] + T[D \setminus D', u] + dist(u, v)\}$$



Theorem 6.3

Theorem 6.3

Steiner Tree kann in Zeit $3^K n^{\mathcal{O}(1)}$

