ETH zürich
Departement Informatik

Algorithmik für schwere Probleme

Prof. Dr. Dennis Komm
Dr. Hans-Joachim Böckenhauer
`https://courses.ite.inf.ethz.ch/schwere_prob_23/`

Lösungsvorschläge – Blatt 13

Zürich, 30. Mai 2023

Lösung zu Aufgabe 15

a) Da $U = \dot\bigcup_{i=1}^{k} D_{\mathcal{C},i}$, gilt

$$\begin{aligned} w_{\mathcal{C}}(U) &= \sum_{x\in U} w_{\mathcal{C}}(x) = \sum_{i=1}^{k} \sum_{x\in D_{\mathcal{C},i}} w_{\mathcal{C}}(x) \\ &= \sum_{i=1}^{k} \sum_{x\in D_{\mathcal{C},i}} \frac{1}{|D_{\mathcal{C},i}|} = \sum_{i=1}^{k} \frac{|D_{\mathcal{C},i}|}{|D_{\mathcal{C},i}|} \\ &= k = |\mathcal{C}|\,. \end{aligned}$$

b) Nehmen wir umgekehrt an, dass es ein x_j gibt mit $w_{\mathcal{C}}(x_j) > 1/(l-j+1)$. Sei $1 \le i \le k$, sodass $x_j \in D_{\mathcal{C},i}$. Dann ist $|D_{\mathcal{C},i}| < l-j+1$. Das bedeutet, S_i deckt höchstens $l-j$ neue Elemente ab. Aufgrund der Reihenfolge der Elemente würde die Menge S aber mindestens die $l-j+1$ Elemente $x_j, \ldots, x_l$ neu abdecken. Da S das Element $x_j \in D_{\mathcal{C},i}$ enthält, kann S keine der Mengen $S_1,\ldots,S_{i-1}$ sein. Dann hätte der Algorithmus aber statt S_i die Menge S wählen müssen, da sie mehr neue Elemente abdeckt. Somit kann es kein solches Element x_j geben.

c) Es gilt nach Teilaufgabe a),

$$|\mathcal{C}| = w_{\mathcal{C}}(U) = \sum_{x\in U} w_{\mathcal{C}}(x) \le \sum_{S\in\mathcal{C}_{\text{opt}}} w_{\mathcal{C}}(S)\,.$$

Für jede Menge $S \in \mathcal{F}$ gilt aber mit $S = \{x_1, \ldots, x_l\}$ wie in Teilaufgabe b),

$$\begin{aligned} w_{\mathcal{C}}(S) &= \sum_{j=1}^{l} w_{\mathcal{C}}(x_j) \\ &\le \sum_{j=1}^{l} \frac{1}{l-j+1} \\ &= \sum_{i=1}^{l} \frac{1}{i} \end{aligned}$$

$$
\begin{aligned}
&= \mathrm{har}(l) \\
&\leq \mathrm{har}(|S_{\max}|)\,.
\end{aligned}
$$

Somit ist

$$
\begin{aligned}
|\mathcal{C}| &\leq \sum_{S \in \mathcal{C}_{\mathrm{opt}}} \mathrm{har}(|S_{\max}|) \\
&= |\mathcal{C}_{\mathrm{opt}}| \cdot \mathrm{har}(|S_{\max}|)\,.
\end{aligned}
$$

d) Es gilt $\mathrm{har}(n) \leq \ln(n) + 1$. Damit ist

$$
\frac{|\mathcal{C}|}{|\mathcal{C}_{\mathrm{opt}}|} \leq \ln(|S_{\max}|) + 1 \leq \ln(|U|) + 1 = \ln(|U| \cdot e)\,.
$$

Falls $|\mathcal{F}| \geq 3$, gilt $e \leq |\mathcal{F}|$, also $|\mathcal{C}|/|\mathcal{C}_{\mathrm{opt}}| \leq \ln(|U| \cdot |\mathcal{F}|)$. Wenn $\mathcal{F}$ aber nur ein oder zwei Elemente enthält, lässt sich das Problem in polynomieller Zeit exakt lösen.

e) Ein Durchlauf der Schleife des Algorithmus lässt sich in Zeit $|\mathcal{F}| \cdot |U|$ berechnen. Die Schleife wird höchstens $|\mathcal{F}|$ mal durchlaufen. Sie wird aber auch höchstens $|U|$-mal durchlaufen, weil in jedem Schritt die Menge U_{free} um mindestens ein Element kleiner wird. Wir können die Anzahl Durchläufe der Schleife also abschätzen durch

$$
\min\{|U|, |\mathcal{F}|\} \leq (|U| \cdot |\mathcal{F}|)^{1/2}\,.
$$

Insgesamt ist die Laufzeit des Algorithmus also in $\mathcal{O}((|U| \cdot |\mathcal{F}|)^{3/2}) = \mathcal{O}(n^{3/2})$.