Algorithmen und Datenstrukturen 2 **SoSe 2023**
Prof. Dr. Sándor Fekete
Dr. Phillip Keldenich

Präsenzblatt 5

Dieses Blatt dient der persönlichen Vorbereitung. Es wird nicht abgegeben und geht nicht in die Bewertung ein. Die Besprechung der Aufgaben erfolgt in den kleinen Übungen vom 04.–07.07.2023.

Präsenzaufgabe:
Wir betrachten das Problem Set Cover.

Gegeben: Eine endliche Menge U (das *Universum*), eine Menge $\mathcal{F}$ von Teilmengen von U und eine Zahl $k \in \mathbb{N}$.

Gesucht: Ein *Set Cover* von $(U, \mathcal{F})$ der Größe höchstens k. Ein Set Cover ist eine Teilmenge $F \subseteq \mathcal{F}$, die U überdeckt, d.h. für jedes Element $u \in U$ gibt es eine Menge $M \in F$ mit $u \in M$. Die Größe eines Set Covers F ist die Anzahl an Mengen in F, d.h. $|F|$.

Wir nehmen an, dass jedes Element aus U in einer Menge aus $\mathcal{F}$ vorkommt.

Als Beispiel betrachte $U := \{1, 2, 3, 4, 5, 6\}$ und $\mathcal{F} := \{\{1, 2\}, \{1, 4\}, \{3, 6\}, \{2, 3, 4\}, \{1, 2, 5\}, \{2, 3\}\}$, sowie $k = 3$. $F := \{\{1, 4\}, \{3, 6\}, \{1, 2, 5\}\}$ ist ein Set Cover von $(U, \mathcal{F})$. Es kann schnell überprüft werden, dass es für $k = 2$ kein Set Cover gibt. Eine graphische Darstellung dieser Instanz (und einer möglichen Lösung) ist in Abbildung 1 abgebildet.

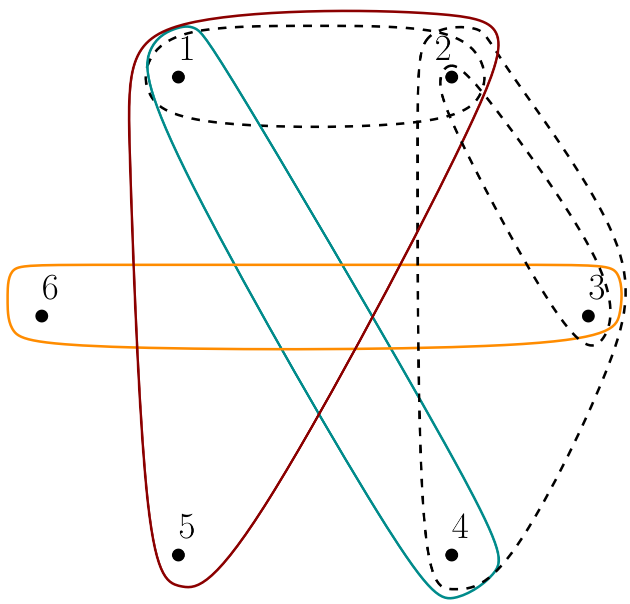


Abbildung 1: Beispiel einer Instanz von Set Cover. Punkte entsprechen den Elementen in U, Kreise entsprechen den Mengen in $\mathcal{F}$. Die farbige Auswahl entspricht einem Set Cover.

a) Zeige, dass Set Cover NP-schwer ist. (Hinweis: Nutze Vertex Cover.)

Da Set Cover also NP-schwer ist, bietet es sich an, Approximationsalgorithmen zu betrachten, um das kleinste k zu finden. Der folgende Algorithmus (GreedySC) versucht ein möglichst kleines Set Cover zu bestimmen.

Algorithmus 1 Algorithmus GreedySC zum Finden eines Set Covers. In jeder Iteration wird diejenige Menge aufgenommen, die die meisten noch nicht überdeckten Elemente besitzt.

```
function GreedySC(U, F)
    C := ∅                                   ▷ Menge der bereits überdeckten Elemente
    C̄ := U                                   ▷ Menge der noch zu überdeckenden Elemente
    SC := ∅                                  ▷ Set Cover

    while C ≠ U do
        S := argmax_{M∈F} |M ∩ C̄|            ▷ Menge mit den meisten nicht überdeckten Elementen
        C := C ∪ S
        C̄ := C̄ \ S
        SC := SC ∪ {S}

    return SC
```

b) Wende GreedySC auf folgende Instanz an: $U := \{1, \ldots, 10\}$, $\mathcal{F} := \{F_1, \ldots, F_5\}$ mit $F_1 = \{1, 2, 3, 7, 9\}$, $F_2 = \{4, 5, 6, 8, 10\}$, $F_3 = \{1, 2, 3, 4, 5, 6\}$, $F_4 = \{7, 8\}$ und $F_5 = \{9, 10\}$.
Gib dabei nach jeder Iteration der while-Schleife S, C sowie $\overline{C}$ an.

c) Betrachte Instanzen der folgenden Form. Sei $3 \leq \ell \in \mathbb{N}$ und $q = 2^\ell - 1$. Das Universum besteht aus zwei Teilmengen mit je q Elementen, also $U = \{x_1, \ldots, x_q\} \cup \{y_1, \ldots, y_q\}$. Die Menge der möglichen Mengen $\mathcal{F}$ enthält die zwei Mengen $X = \{x_1, \ldots, x_q\}$ und $Y = \{y_1, \ldots, y_q\}$ sowie zusätzlich die Mengen

$$M_i = \{x_{2^{i-1}}, \ldots, x_{2^i - 1}\} \cup \{y_{2^{i-1}}, \ldots, y_{2^i - 1}\}$$

für $1 \leq i \leq \ell$. Abbildung 2 zeigt als Beispiel die Instanz für den Fall $\ell = 3$.

Was ist die Größe eines optimalen Set Covers auf dieser Art von Instanz? Welche Größe hat das Set Cover, das GreedySC berechnet, in Abhängigkeit von ℓ? Wie wächst der asymptotische Faktor zwischen OPT und GreedySC in Abhängigkeit der Größe des Universums $n = 2q$ (in O-Notation)? Ist GreedySC ein Approximationsalgorithmus im Sinne der Vorlesung?

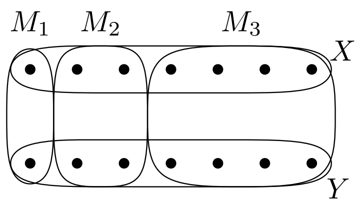


Abbildung 2: Instanz aus Aufgabenteil c) für $\ell = 3$.

z.B. Vertex Cover $G = (V, E)$

$VC \subseteq V$ mit $(u,v) \in E$ $u \in VC \lor v \in VC$.

Z: Set Cover ist NP-Schwer

Es gilt: Vertex Cover ist NP-Schwer.

Reduktion von Vertex Cover auf Set Cover durchführen:

$\Rightarrow$ Idee: Löse Vertex mit Hilfe von Set Cover.
Wenn Set Cover schneller zu lösen wäre, wäre Vertex Cover ebenfalls schneller zu lösen.

<u>Vertex Cover</u> $\xrightarrow{\text{Reduktion}}$ Set Cover

Gegeben: $G = (V, E)$

$V = \{v_1, v_2, v_3, v_4\}$

$E = \{e_1, e_2, e_3, e_4\}$

$U = \underline{E}$

$F = \{\underline{\{e_1, e_2\}}, \{e_2, e_3\}, \{e_1, e_3, e_4\}, \{e_4\}\}$

Gesucht:

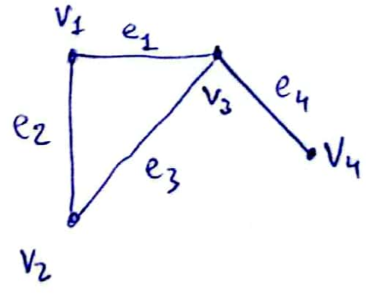
$VC \subseteq V$ mit

$(u,v) \in E \Rightarrow$

$u \in VC \lor v \in VC$ $\leftarrow$ $SC \subseteq F : \bigcup_{M \in SC} = U$

$VC = \{v_3, v_2\}$

$SC = \{\{e_1, e_3, e_4\}, \{e_2, e_3\}\}$



Korrektheit:

Wenn Vertex Cover ein VC der Größe k besitzt, dann besitzt auch Set Cover ein SC der Größe k.

Nimm dazu die Mengen F, die von VC gehören.

(Analog gilt auch die Rückrichtung)

Konstruktion erfolgt in polynomieller Laufzeit.

b) $U = \{1, \ldots, 10\}$

S	C	$\bar{C}$
$F_3 = \{1, 2, \ldots, 6\}$	$\{1, \ldots, 6\}$	$\{7, 8, 9, 10\}$
$F_1 = \{1, 2, 3, 7, 9\}$	$\{1, \ldots, 6, 7, 9\}$	$\{8, 10\}$
$F_2 = \{4, 5, 6, 8, 10\}$	$\{1, \ldots, 10\}$	$\emptyset$

c) Greedy: l

OPT : 2

GreedySC ist Approximationsalgo. wenn $\frac{\text{Greedy}}{\text{OPT}} < c$ für $n \to \infty^+$

$n = 2 \cdot (2^l - 1)$

$n + 2 = 2 \cdot 2^l = 2^{l+2} \quad \Leftrightarrow \quad l + 1 = \log(n+2)$

$\Leftrightarrow \quad l = \log(n+2) - 1$