

Algorithmen I - Tutorium 12

Sebastian Schmidt – *isibboi@gmail.com*

Arbeitsgruppe Kryptographie und Sicherheit

Sei $V = 1..n$

```
Tc : UnionFind(n)           // encodes components of forest T
foreach (u, v) ∈ E in ascending order of weight do           // sort
    if Tc.find(u) ≠ Tc.find(v) then
        output {u, v}
        Tc.union(u, v)           // link reicht auch
```

Union-Find

Struktur: Array

Operationen: `union(i, j)`, `find(i)`

Gegeben Graph G .

Finde minimale Menge $V' \subseteq V$ mit $\forall (u, v) \in E : u, v \cap V' = \emptyset$.

Formuliere Vertex Cover als ILP.

Ein **lineares Programm** mit n **Variablen** und m **Constraints** (NB) wird durch das folgende Minimierungs-/Maximierungsproblem definiert:

- ▶ Kostenfunktion $f(\mathbf{x}) = \mathbf{c} \cdot \mathbf{x}$
 \mathbf{c} ist der **Kostenvektor**
- ▶ m **Constraints** der Form $\mathbf{a}_i \cdot \mathbf{x} \bowtie_i b_i$ mit $\bowtie_i \in \{\leq, \geq, =\}$, $\mathbf{a}_i \in \mathbb{R}^n$.

Wir erhalten:

$$\mathcal{L} = \{\mathbf{x} \in \mathbb{R}^n : \forall j \in 1..n : x_j \geq 0 \wedge \forall i \in 1..m : \mathbf{a}_i \cdot \mathbf{x} \bowtie_i b_i\} \quad .$$