

Algorithmen I - Tutorium 11

Sebastian Schmidt – *isibboi@gmail.com*

Arbeitsgruppe Kryptographie und Sicherheit

Sei $G = (V, E)$ ein Graph.

Schnitteigenschaft:

Sei $S \subset E$ ein Schnitt in G . Dann gehört eine Kante mit dem kleinsten Gewicht in S zum MST.

Kreiseigenschaft:

Sei $K \subset E$ ein Kreis in G . Dann gehört eine größte Kante auf K nicht zum MST.

Was ist der MST zum gegebenen Beispielgraphen? (Tafel)

Sei $G = (V, E)$ ein Graph.

Schnitteigenschaft:

Sei $S \subset E$ ein Schnitt in G . Dann gehört eine Kante mit dem kleinsten Gewicht in S zum MST.

Kreiseigenschaft:

Sei $K \subset E$ ein Kreis in G . Dann gehört eine größte Kante auf K nicht zum MST.

Was ist der MST zum gegebenen Beispielgraphen? (Tafel)

Jarník-Prim

Function jpMST : Set of Edge // weitgehend analog zu Dijkstra

pick any $s \in V$

$d = \{\infty, \dots, \infty\}$; $\text{parent}[s] := s$; $d[s] := 0$; $Q.\text{insert}(s)$

while $Q \neq \emptyset$ **do**

$u := Q.\text{deleteMin}$

$d[u] := 0$

// scan u

foreach edge $e = (u, v) \in E$ **do**

if $c(e) < d[v]$ **then**

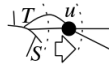
$d[v] := c(e)$

$\text{parent}[v] := u$

if $v \in Q$ **then** $Q.\text{decreaseKey}(v)$

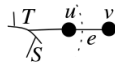
else $Q.\text{insert}(v)$

return $\{(v, \text{parent}[v]) : v \in V \setminus \{s\}\}$



// relax

// update tree



► $d[u] = 0 \Leftrightarrow u \in S$

► \Rightarrow am Ende jeder **while**-Iteration: endliches $d[v] > 0$ speichert

Gewicht der leichtesten Kante von $v \notin S$ über den Schnitt

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
```