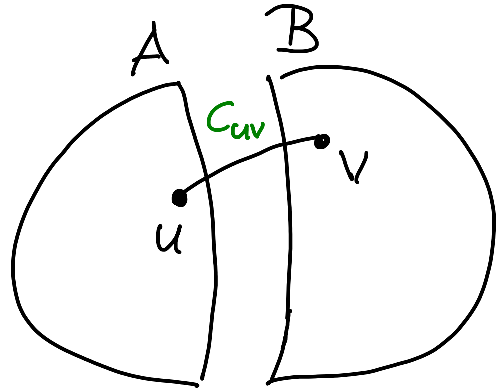
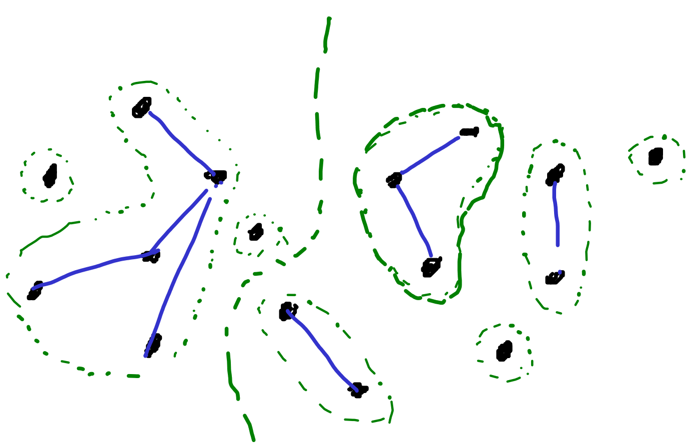
Der Algorithmus von Prim für kürzeste Spannbäume

Lemma: $A \subseteq V(G)$, $B = V(G) \setminus A$

uv sei die Kante mit $u \in A$, $v \in B$ mit den kleinsten Kosten c_{uv}.

Dann gehört uv zum kürzesten Spannbaum T^*.





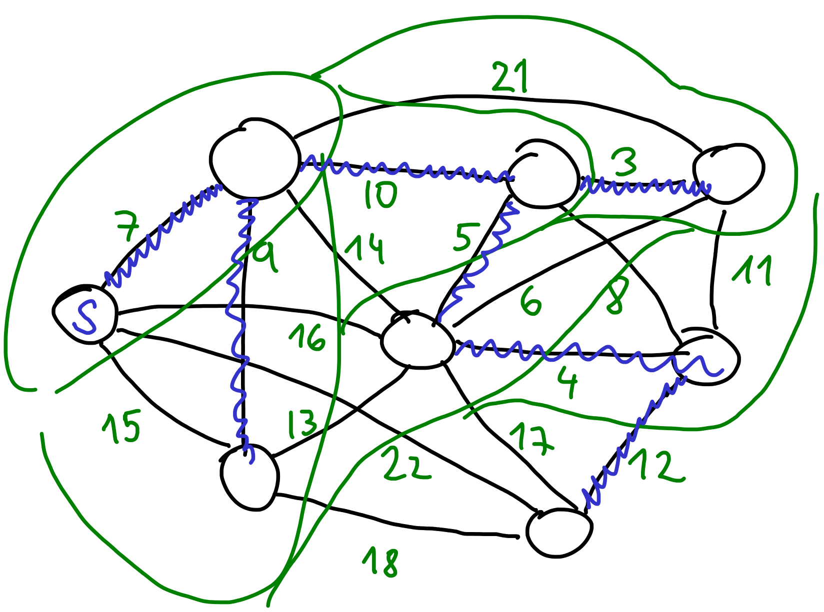
Wähle einen beliebigen Startknoten s.

$A := \{s\}$ $T := \emptyset$

<u>Schleife</u> $(n-1)$ mal.

Bestimme die kürzeste Kante uv mit $u \in A$, $v \notin A$.

$T := T + uv$; $A := A \cup \{v\}$



Wir speichern für jeden Knoten $v \notin A$

$$d[v] := \min\{c_{uv} \mid u \in A\}$$

$$\min\{d[v] \mid v \notin A\}$$

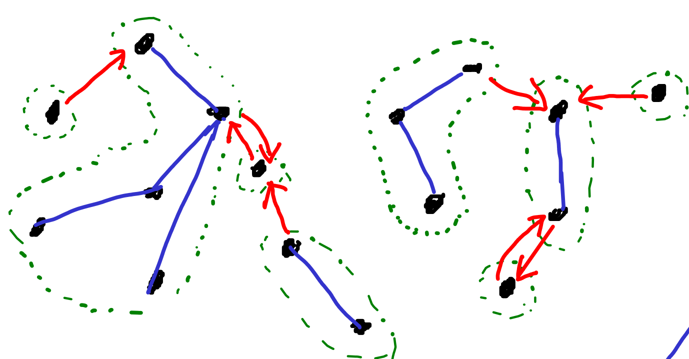
$$d[v] = c_{f(v)v} \quad f(v) \in A$$

$d[v] := \infty$ für $v \in V$
$d[s] := 0$
$A := \emptyset$; $T := \emptyset$
while $|A| < n$:
    Bestimme $u \notin A$ mit kleinstem Wert $d[u]$ deletemin $O(\log n)$ $n\times$
    $T := T + (f[u], u)$ falls $u \neq s$
    $A := A \cup \{u\}$
    for v in $E[u]$:
        if $v \notin A$ **and** $c_{uv} < d[v]$:
            $d[v] := c_{uv}$
            $f[v] := u$ } decreasekey | einfügen $O(\log n)$
$\leq m$-mal

$$d[v] \equiv \min\{c_{uv} \mid u \in A\} = c_{f(v)v} \quad (v \notin A)$$

Laufzeit: $O((m+n)\log n)$

Der Algorithmus von Borůvka (1926) für kürzeste Spannbäume

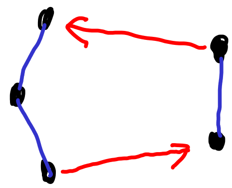


Bestimme die Komponenten, z.B. durch Tiefensuche in T $O(n)$

$T := \emptyset$; n Einzelkomponenten
while mehr als eine Komponente:
    Für jede Komponente:
    - Bestimme die kürzeste Kante, die aus der Komponente hinausführt.   } $O(m)$
    - Füge alle diese Kanten dazu.

- Die Anzahl der Komponenten wird in jedem Durchlauf mindestens halbiert.
- Anzahl der Schleifendurchläufe $\leq \log_2 n$

Laufzeit: $O((m+n)\log n)$



Es ist wichtig, dass die Kosten verschieden sind!

Gleichheiten müssen konsistent aufgelöst werden!

Knoten nummern

$$(u,v) \prec (a,b) \;:\Leftrightarrow\; c_{uv} < c_{ab} \lor \Big[c_{uv} = c_{ab} \land \big(\min(u,v) < \min(a,b) \lor \left[\min(u,v) = \min(a,b) \land \max(u,v) < \max(a,b) \right] \big) \Big]$$