Der Linearzeit MST Algorithmus

Der schnellste Algorithmus für das MST/ MSF Problem

Max Springenberg

Proseminar: Randomisierte Algorithmen, TU Dortmund

Table of contents

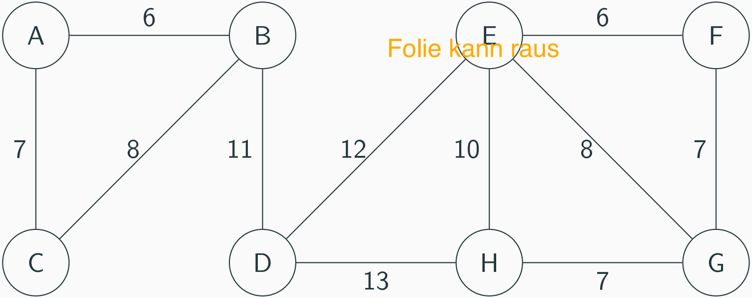
MST in gewichteten Graphen

Definition MST

Ein Teilgraph T ist genau dann ein minimaler Spannbaum von G, wenn er ein Spannbaum in G ist und die Summe seiner Kantengewichte $\sum\limits_{e \in E_T} w(e)$ minimal ist.

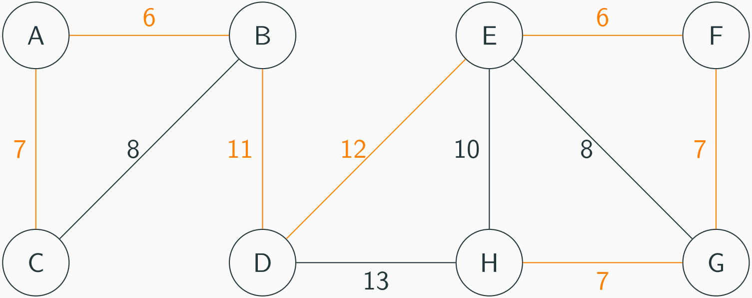
Definition MST

Ein Teilgraph T ist genau dann ein minimaler Spannbaum von G, wenn er ein Spannbaum in G ist und die Summe seiner Kantengewichte $\sum_{e \in E_T} w(e)$ minimal ist.



Definition MST

Ein Teilgraph T ist genau dann ein minimaler Spannbaum von G, wenn er ein Spannbaum in G ist und die Summe seiner Kantengewichte $\sum_{e \in E_T} w(e)$ minimal ist.

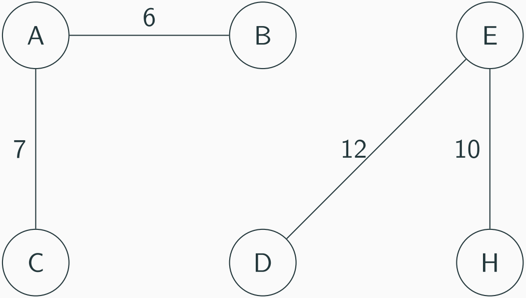


Bäume vs. Wälder

Überschrift: Definition Wald

Ein Wald F ist ein Teilgraph von G, der aus disjunkten Bäumen besteht.

F : PLATZ FEHLT



Borůvka Phasen

Ablauf

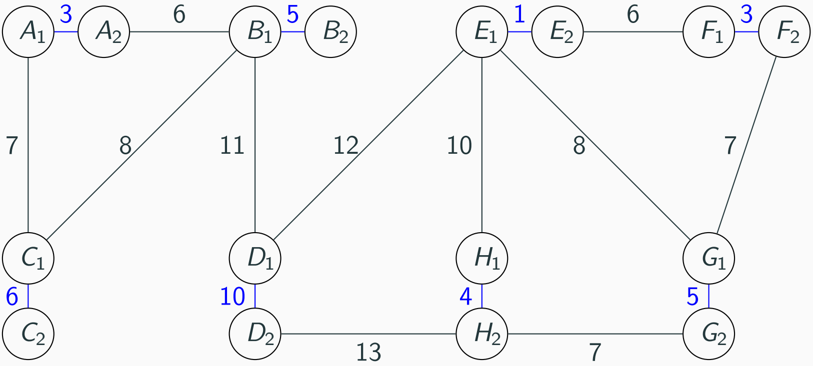
1. Kontraktierende Kanten markieren
2. Verbundene Komponente bestimmen
3. Verbundene Komponenten durch einzelnen Knoten ersetzen
4. Selbstschleifen entfernen

vllt. die Schlussfolgerung eine Folie später wenn noch zeit

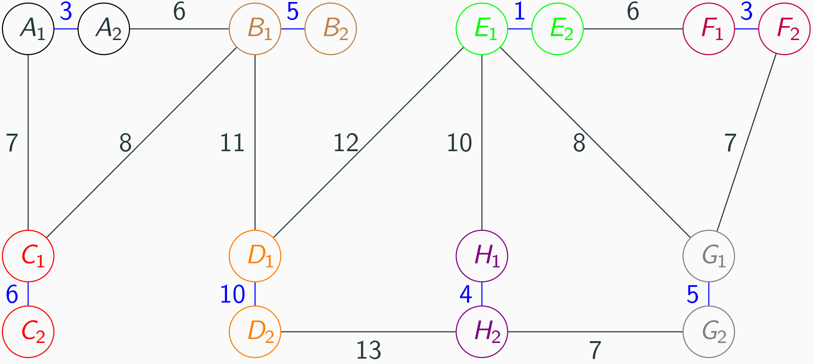
Was bedeutes das für den reduzierten Graphen?

$\Rightarrow$ Knoten werden auf maximal $n/2, n = |V|$ reduziert!

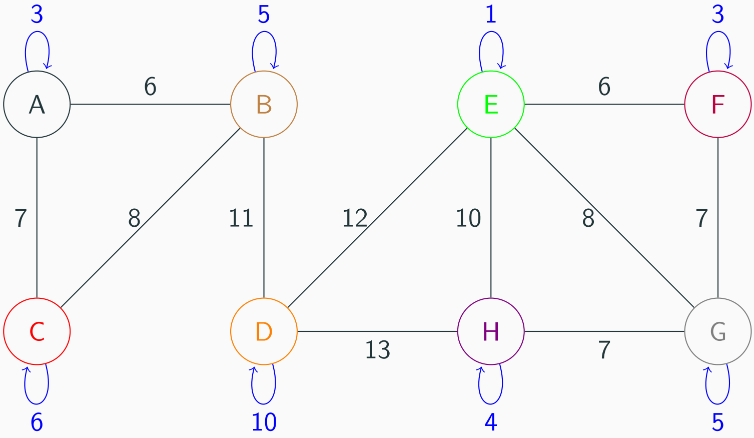
1. Kontraktierende Kanten markieren



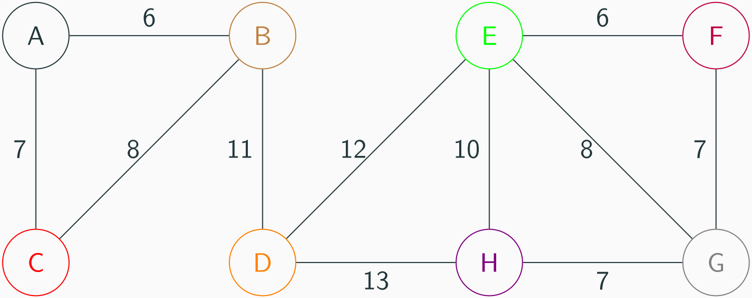
2. Verbundene Komponente bestimmen



3. Verbundene Komponenten durch einzelnen Knoten ersetzen



4. Selbstschleifen entfernen



F-schwere/ -leichte Kanten

Definition

Sei $P(e = \{u, v\})$ der Pfad, der die Knoten im MSF verbindet, in Kanten
Sei $w : E \to \mathbb{R}$, die Gewichtsfunktion von G
Sei ferner definiert $w(E) = \{w(e_1), \ldots, w(e_m)\}$

Eine Kante ist F-schwer, wenn gilt:

$$w(e) > w_F(e)$$

, wobei:

$$w_F(e = (u, v)) = \begin{cases} \infty, \text{ u und v sind in verschiedenen Komponenten} \\ \text{max\{w(P(e))\}}, \textit{sonst} \end{cases}$$

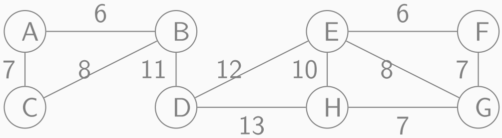
Randomiserte Stichprobem

Wirf eine Münze!

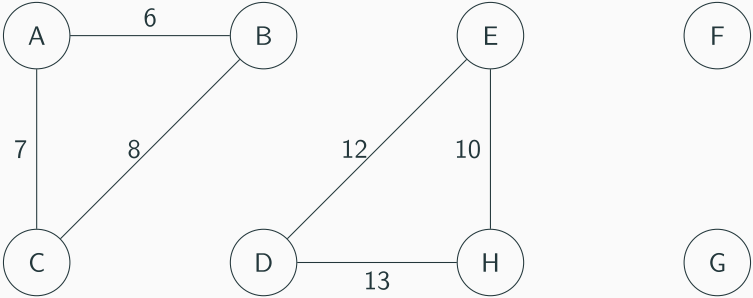
Quelle: https://melbournechapter.net/explore/coin-flip-clipart/

Kanten 'würfeln'

G:

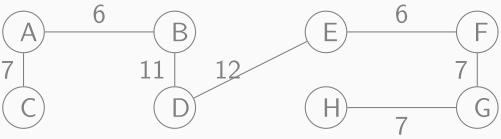


$G(p = 0,5)$:

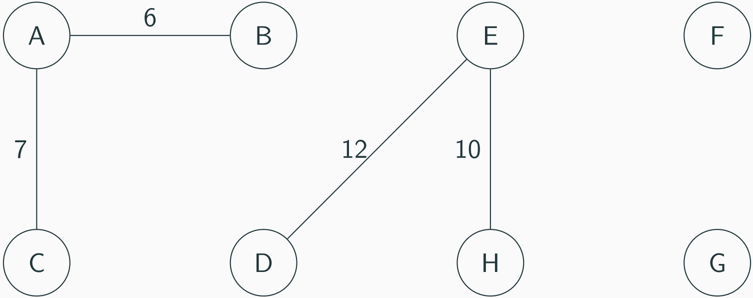


MSF

MST: _G



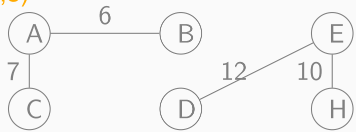
MSF: _G(0,5)



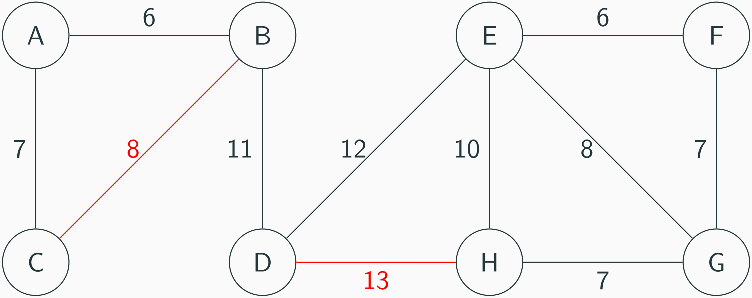
Erkenntnis

Eleminierung von unnützen Kanten

MSF: G(0,5)



G :



Idee

1. Nutze Borůvka-Phasen, um Knoten zu reduzieren
2. Nutze Stickproben, um die Kanten zu reduzieren
3. Entferne alle F-schweren Kanten
4. **Rekursion**

Teaser

- Wie Fassen wir die Erkenntnis geschickt in einem Algorithmus?
- Wie erhalten wir trotz rekursiven Aufrufen eine erwartete lineare Laufzeit?