

# Der Linearzeit MST Algorithmus

Der schnellste Algorithmus für das MST/ MSF Problem

---

Max Springenberg

Proseminar: Randomisierte Algorithmen, TU Dortmund

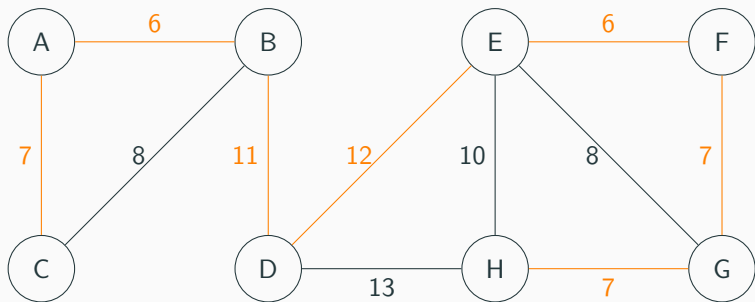
# MST in gewichteten Graphen

---

# 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

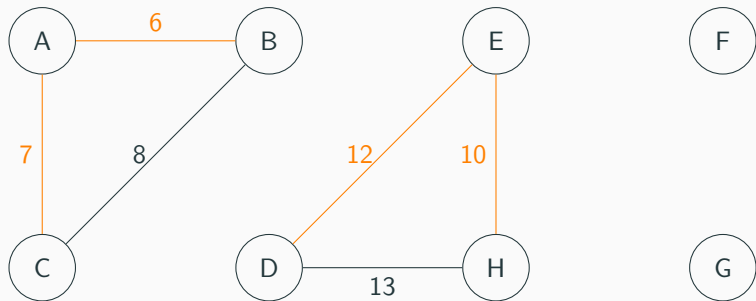
$\sum_{e \in E_T} w(e)$  **minimal** ist.



# Bäume vs. Wälder

---

Teilgraph aus disjunkten MSTs



# Borůvka Phasen

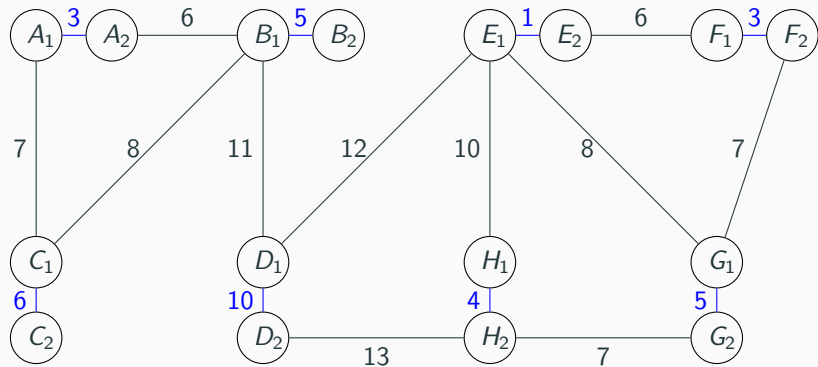
---

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

## **Was bedeutet das für den reduzierten Graphen?**

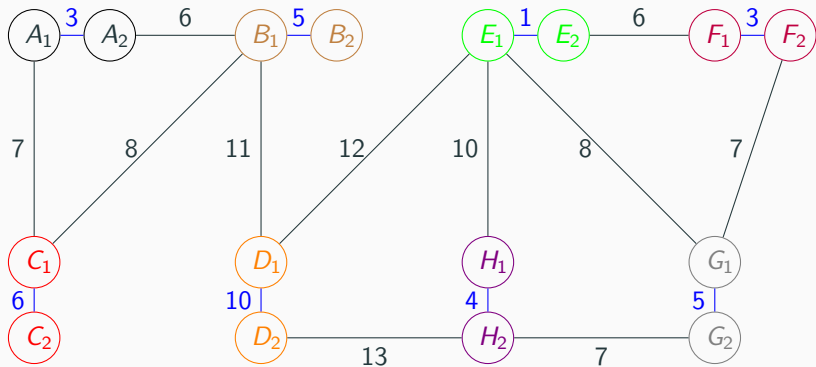
⇒ Knoten werden auf maximal  $n/2$ ,  $n = |V|$  reduziert!

# 1. Kontraktierende Kanten markieren

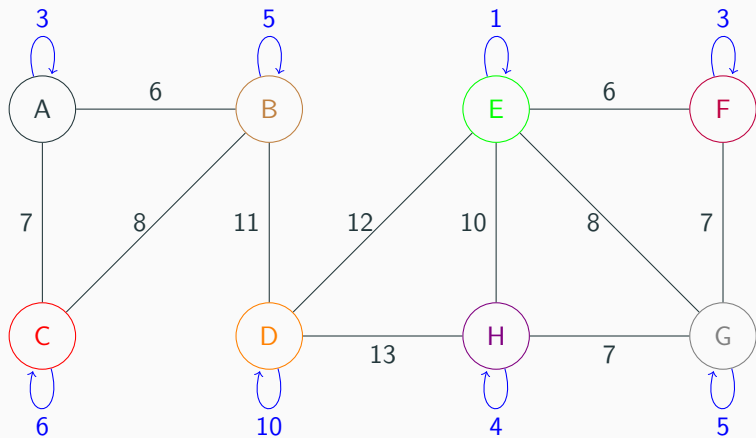




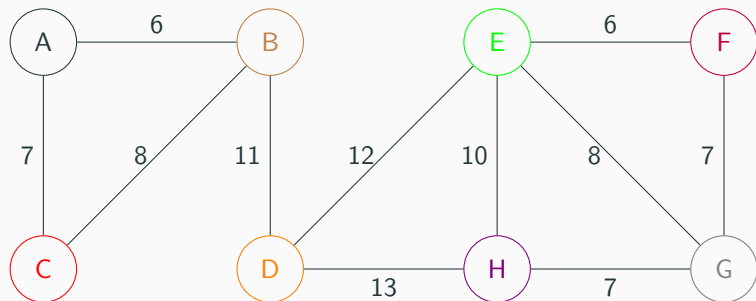
## 2. Verbundene Komponenten bestimmen



### 3. Verbundene Komponenten durch einzelne Knoten ersetzen



## 4. Selbstschleifen entfernen



# Randomisierte Stichproblem

---

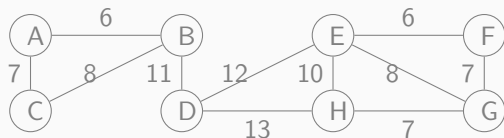
Wirf eine Münze!



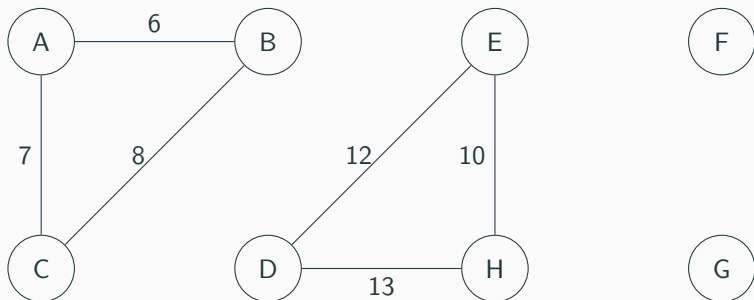
Quelle: <https://melbournechapter.net/explore/coin-flip-clipart/>

# Kanten 'würfeln'

G:



$G(p = 0,5)$  :

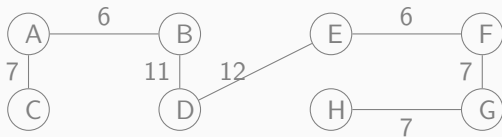


# Erkenntnis

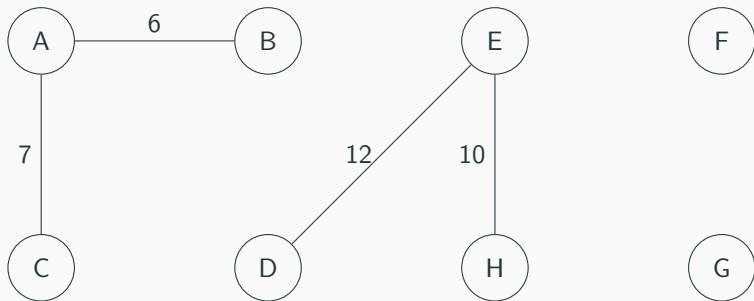
---

# MST vs. MSF

$MST_G$ :



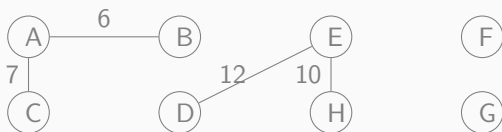
$MSF_{G(0,5)}$ :



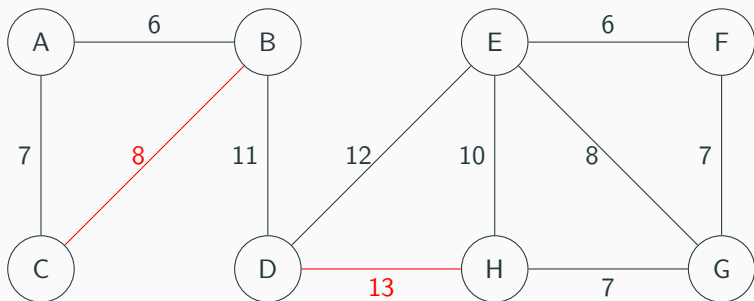


# Eliminierung von F-schweren Kanten

$MSF_{G(0,5)}$ :



$G$ :



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

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