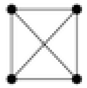
Minimale Spannbäume

Vollständiger Graph:

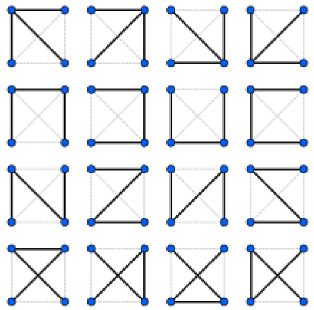
Ungerichteter, zusammenhängender Graph ohne Mehrfachkanten, in dem jeder Knoten mit jedem anderen Knoten durch eine Kante verbunden ist.



Vollständiger Graph
mit 4 Knoten

Spannbaum (Spanning tree):

Teilgraph eines ungerichteten, zusammenhängenden Graphen, der ein <u>Baum</u> ist (zyklenfrei), und alle Knoten des Graphen enthält.

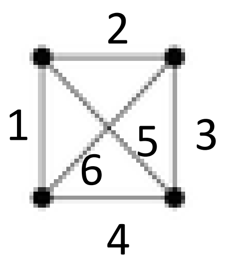


Alle Spannbäume eines Graphen
mit 4 Knoten

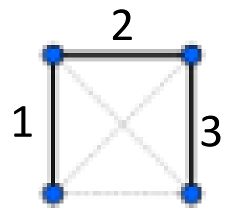
Minimale Spannbäume

Minimaler Spannbaum (Minimal Spanning tree), MST:

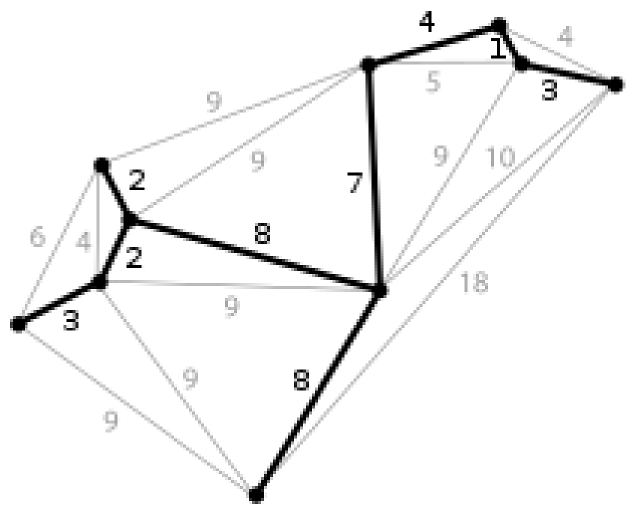
Ein Spannbaum eines <u>kantengewichteten</u> Graphen heißt *minimal*, wenn in demselben Graphen kein anderer Spannbaum mit geringerem Gewicht existiert.



Vollständiger kantengewichteter Graph mit 4 Knoten



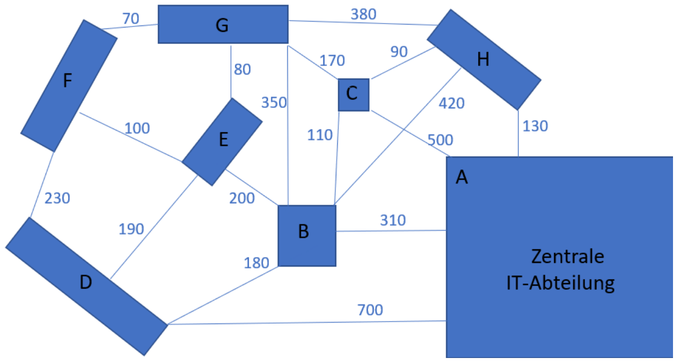
Minimaler Spannbaum dieses kantengewichteten Graphen



Ein Graph mit einem minimalen Spannbaum

Formale Beschreibung

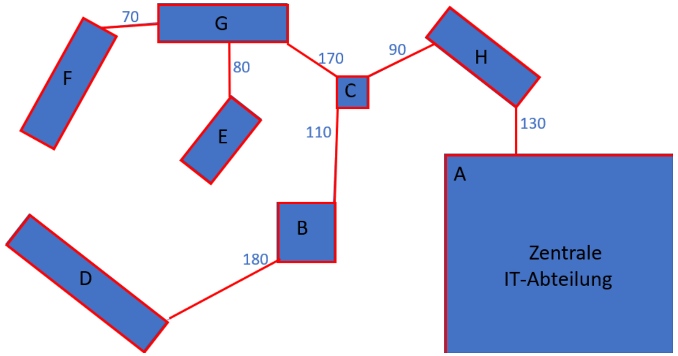
Wenn wir eine endliche Menge von **Knoten über Kanten verbinden**, sprechen wir von einem <u>Graphen</u>.



G = (K, V)   K = Knoten, V = Verbindungen

Für Berechnungen innerhalb eines Graphen müssen die Kanten gewichtet sein (Gewichte: Längen, Kosten, Zeiten ...)

Wenn die **Kanten des Graphen zusammenhängend und kreisfrei** sind, sprechen wir von einer Spezialform der Graphen, den <u>Bäumen</u>



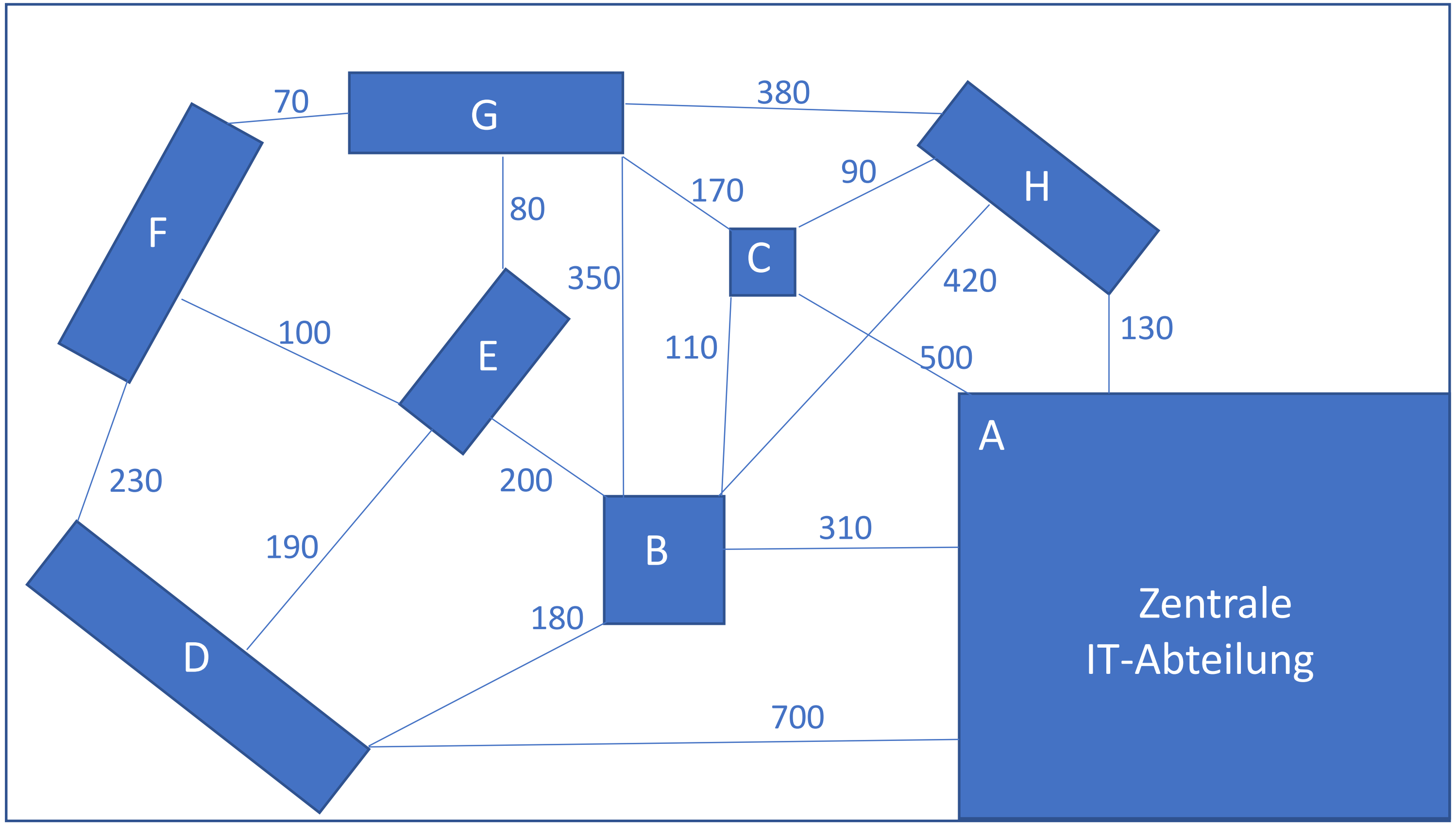
Ein „Minimal spanning tree“ (minimaler Spannbaum) ermittelt den Teilgraph eines Graphen, für den die Gesamtlänge der Kanten beim Verbinden aller Knoten minimal ist.

Gesucht: Tree T = (K, V‘)   V‘ = minimale Verbindungen, V‘ $\subseteq$/

wobei: $W(T) = \sum_{v \in T} w(v)$ minimal $v \in V'$ w = weights

Berechnung Minimaler Spannbäume

Hochschulcampus bekommt Neuverkabelung…



Start beim Rechenzentrum

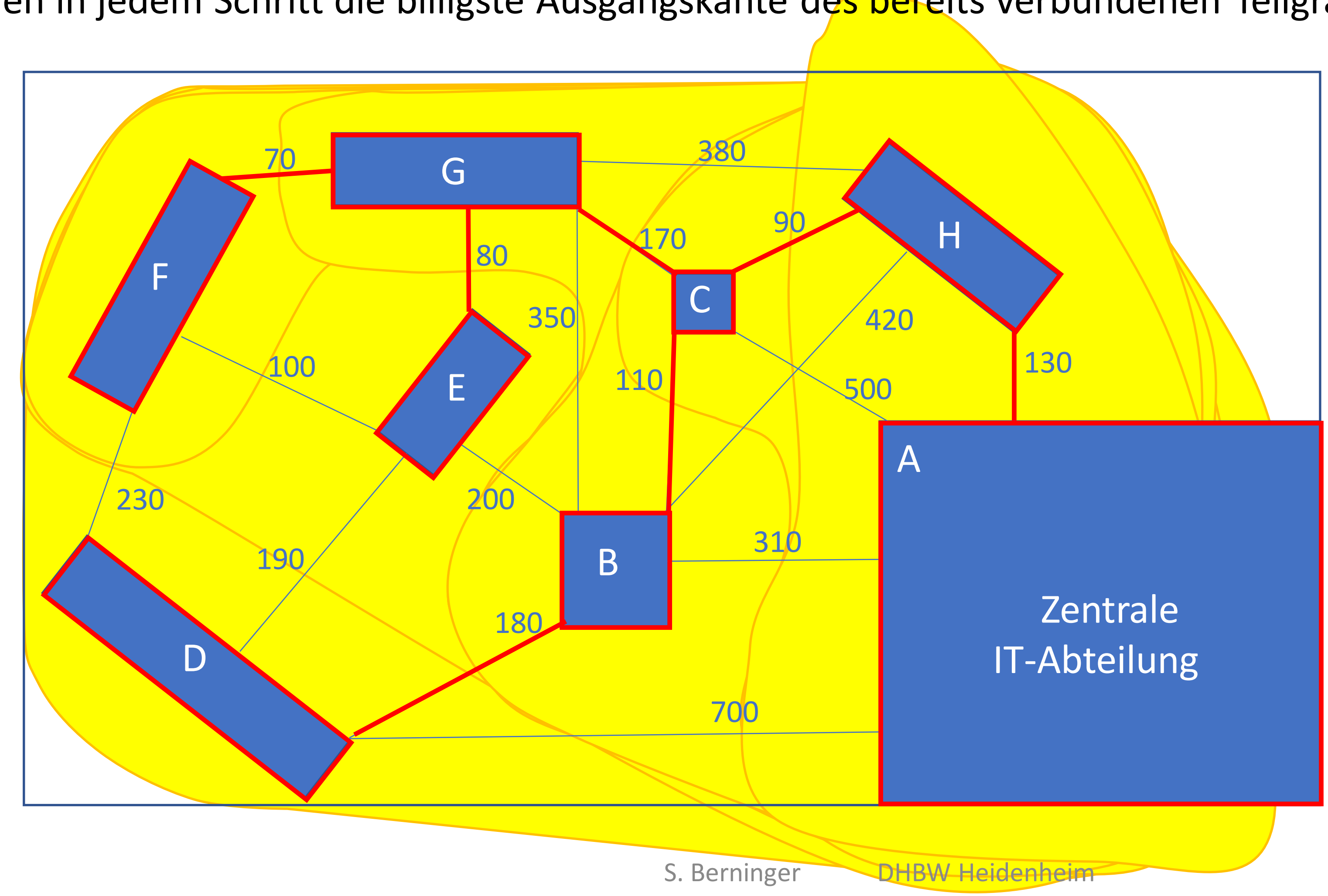
1 Kosteneinheit / lfdm,

pro Jahr ist eine Teilstrecke finanzierbar

(Distanzen zwischen allen Gebäuden vermessen)

Ausbau über mehrere Schritte – das Budget… Die finale Verkabelung

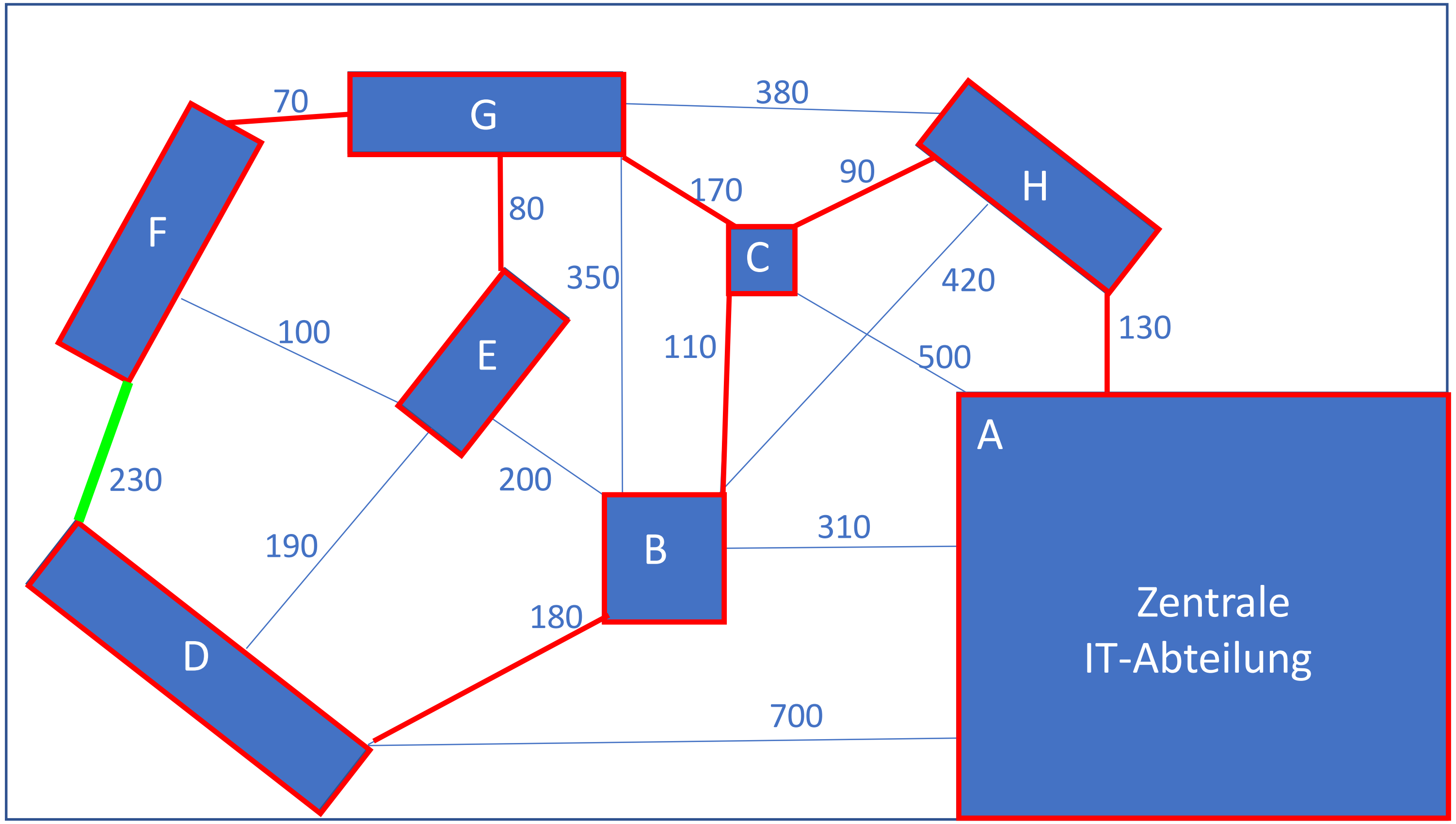
Algorithmus von Prim:
Wir suchen in jedem Schritt die billigste Ausgangskante des bereits verbundenen Teilgraphen zu einem neuen Knoten.



Kosten pro Jahr

Gesamt-kosten

Ist es wirklich der minimale Baum?



Wenn wir eine beliebige Kante hinzunehmen, die nicht zu unserem MST gehört, entsteht ein Kreis (z.B. D-F).
Wir müssen deshalb eine andere Kante dieses Kreises entfernen, um wieder einen Baum zu erhalten.

Nur wenn diese Kante größer wäre, würde der neue Baum „minimaler“ sein.

In diesem Kreis kann es aber keine größere Kante geben als die neue, sonst wäre sie nicht für die Anbindung ihres Knotens verwendet worden.

Algorithmus von Prim

Algorithmus von Jarnik/ **Prim** - 1930 vom tschechischen Mathematiker Vojtěch Jarník entwickelt, 1957 wurde er von Robert C. Prim wiederentdeckt und dann 1959 auch von Edsger W. Dijkstra weiterentwickelt

Die Vergrößerung eines Graphen durch sequentielle Anbindung von Kanten **von einem Startknoten aus** (beginnend mit einem kantenfreien Graphen)

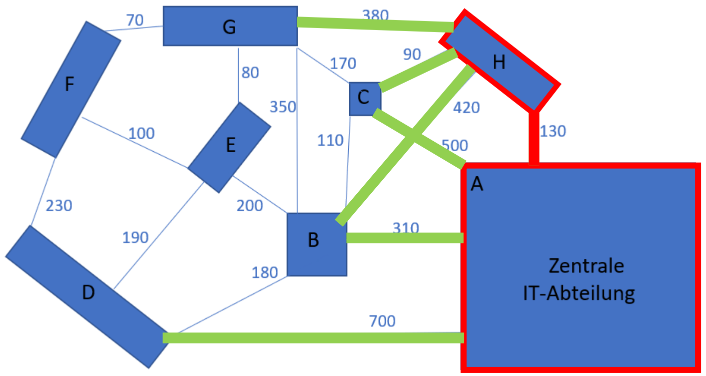
- Algorithmus sucht kurzfristig größten Gewinn, nicht vorausschauend, nennt man deshalb auch „gierig" oder „Greedy-Algorithmus"
- Auswahl einer lokalen, freien Kante vermeidet bei der Konstruktion die Bildung von Kreisen
- Effizienz des Algorithmus hängt von Effizienz der Listenberechnung der möglichen Kanten ab
- Den Beispielcode finden Sie im Moodle zur Vorlesung

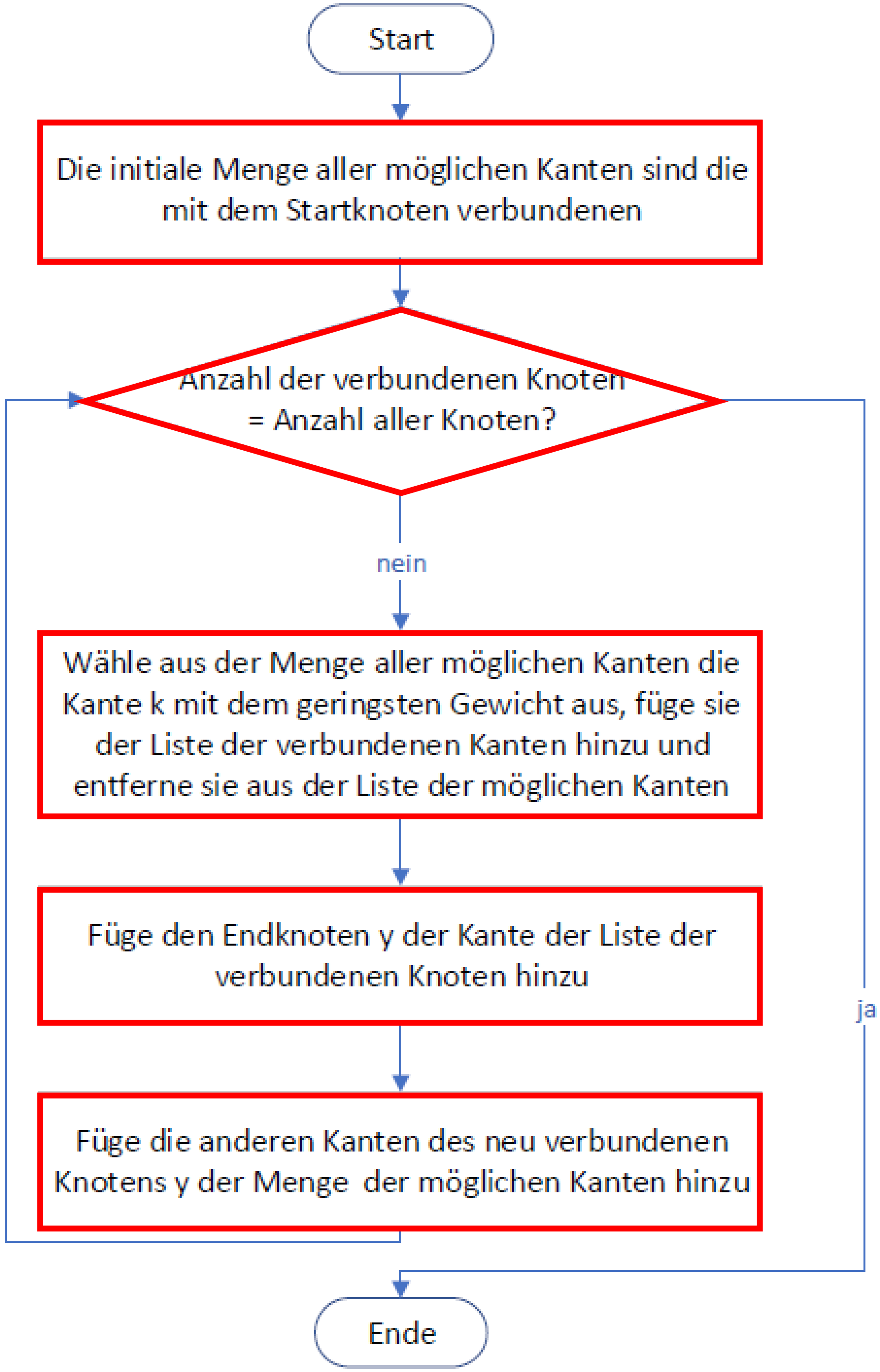
Anwendungsmöglichkeiten:

- „Spanning tree-Verfahren" für das Aufbauen von schleifenfreien Routen im Netzwerk
- Pipelineverbindungen zwischen erdölverarbeitenden Werken
- Bewässerungssysteme von einem Stausee aus …

DHBW
Duale Hochschule
Baden-Württemberg
Heidenheim

Algorithmus von Prim





Algorithmus von Prim

Aufwand/ Laufzeit des Algorithmus:

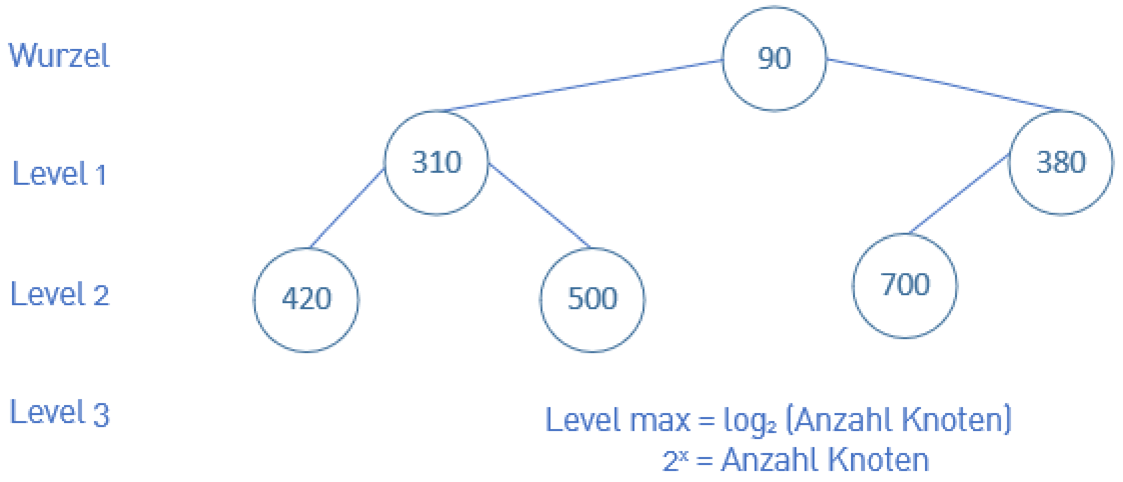
Kosten – Rechenzeiten:

Wie skaliert der Algorithmus bei steigender Menge an Eingangswerten?

Liste aller möglichen Kanten muss effizient sein:
Kante **entfernen**, Kanten des neuen Knotens **hinzufügen**, **sortieren**, kürzeste Kante **finden**:

Liste als binärer Min-Heap, minimaler Eintrag an der Wurzel :
Tiefe = $\log_2$ (Knoten)

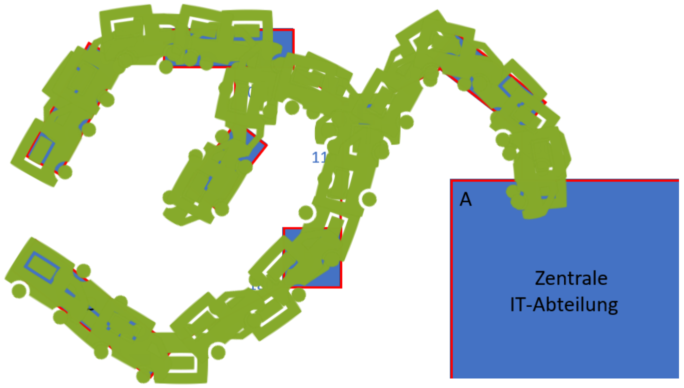
Laufzeit = (Kanten+Knoten)*$\log_2$ (Knoten)   (N+M) * O(log N)



Weitere Anwendungsgebiete

- Minimal spanning trees bilden die Grundlage für endliche Labyrinthe – es gibt einen eindeutigen Weg heraus durch Traversierung („immer an der gleichen Wand lang“)

- Basis für Lösungen des „Travelling salesman“-problems, und Routenplanung



Algorithmus von Kruskal

Algorithmus von Joseph Kruskal - 1956 in der Zeitschrift „Proceedings of the American Mathematical Society“ veröffentlicht.

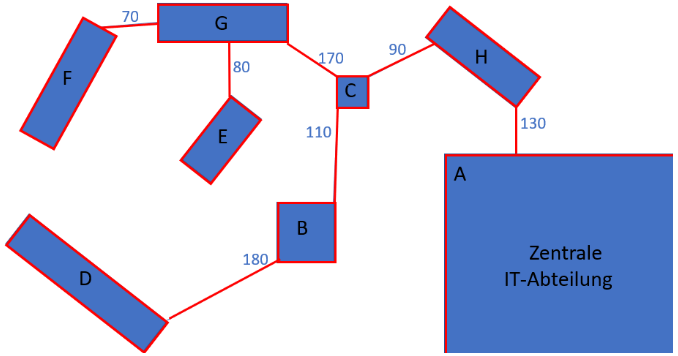
Algorithmus sucht die billigste aller noch nicht verbundenen Kanten, die mindestens einen neuen Knoten verbindet – und nimmt temporär nicht verbundene Teilgraphen in Kauf.

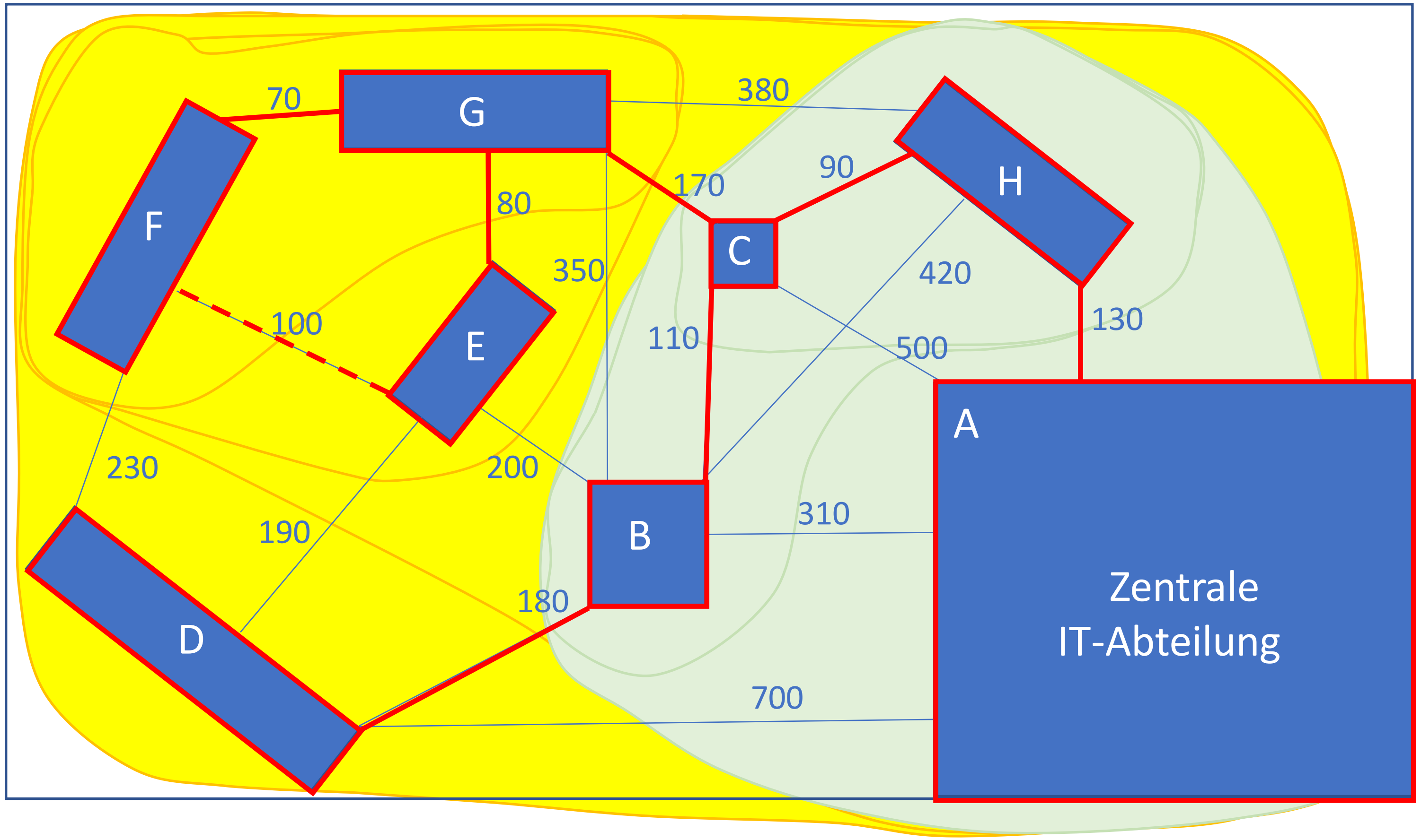
„Die Vergrößerung eines Graphen durch sequentielle Anbindung von Teilgraphen, bestehend aus mindestens einer Kante, **von einem leeren Graphen aus …“**

Es ist am Jahresende Budget übrig:

Wir leisten uns überdachte Verbindungswege…

Algorithmus von Kruskal



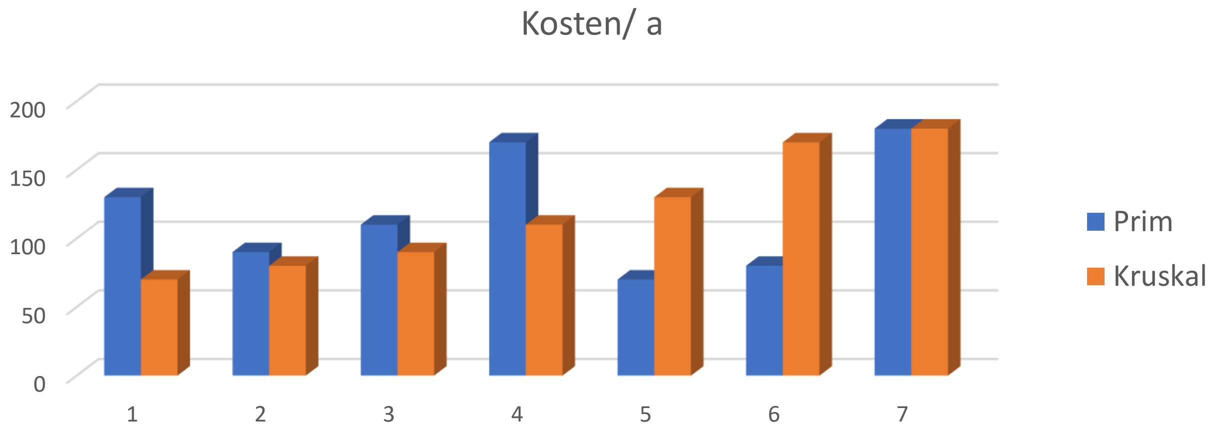


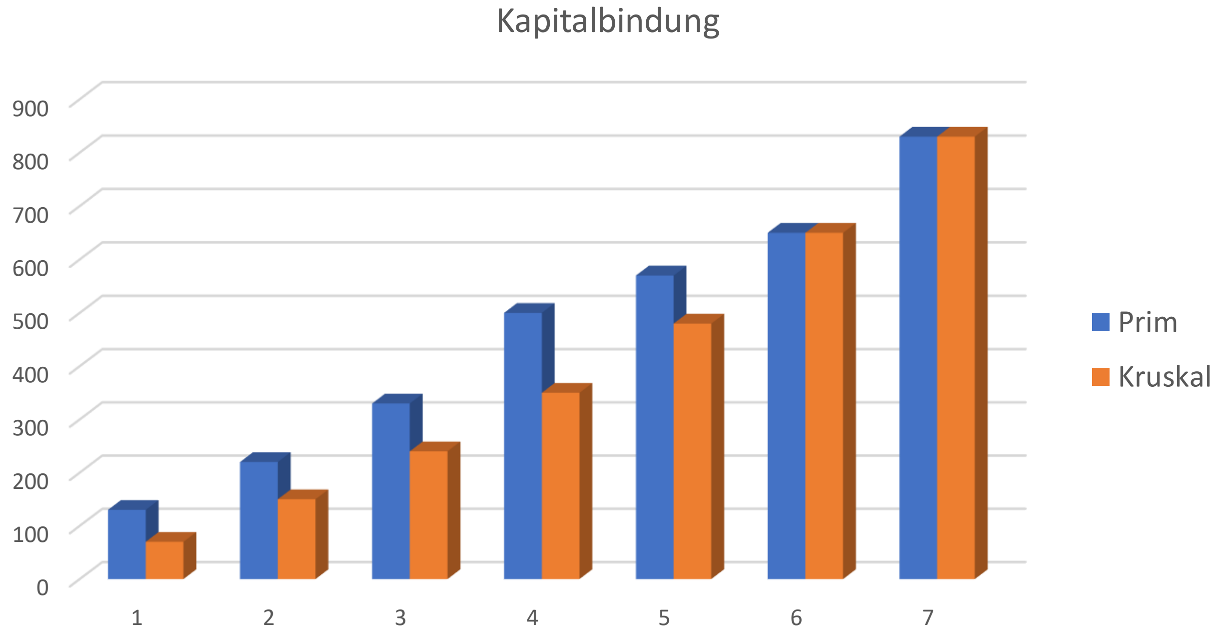
Es ist am Jahresende Budget übrig:

Kosten Gesamt-
pro Jahr kosten

Wir leisten uns überdachte Verbindungs-wege…

Kosten und Kapitalbindung





Unterschiede der Algorithmen

Kriterium	Algorithmus von Prim	Algorithmus von Kruskal
Anbindung an Startpunkt/ Basisstation	Startpunkt wählbar	Startkante durch minimale Kosten vorgegeben
Kreisvermeidung	Per Konstruktion durch Nutzung „möglicher“ Kanten	Muss aktiv geprüft und vermieden werden
Kostenakkumulierung	Kosten statistisch gleichverteilt	Spätestmögliche Budgetbindung
Parallelisierbarkeit	Knoten können ihre „möglichen“ Kanten parallel und unabhängig berechnen	Wenig Möglichkeiten zur Parallelisierung

Zusammenfassung

Sie sollten heute besser verstanden haben:

- … was einen Graphen zum Baum macht
- … welche Anforderungen an die Knoten und Kanten eines Graphen gestellt werden, damit ein Spannbaum berechnet werden kann
- … und was das Spanning tree–Protokoll in geswitchten Netzwerken verhindert…