

# Algorithmik

Q11/12

**Gymnasium der Regensburger Domspatzen**

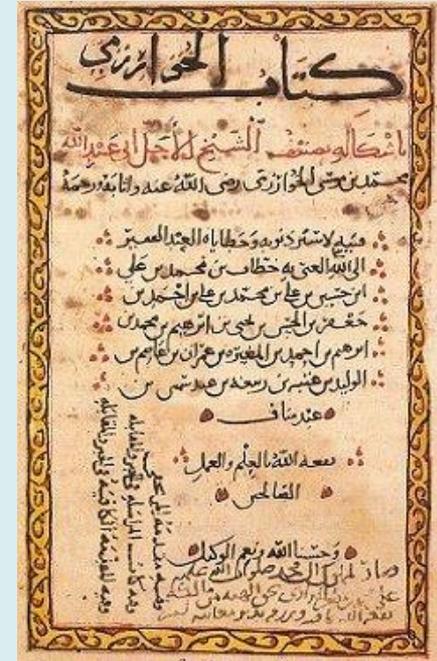
# Algorithmus

## Begriff

„Muḥammad ibn Mūsā al-Khwārizmī“,  
Mathematiker, 780 – ca. 850 n.Chr.  
lateinisiert: *Algorithmi*

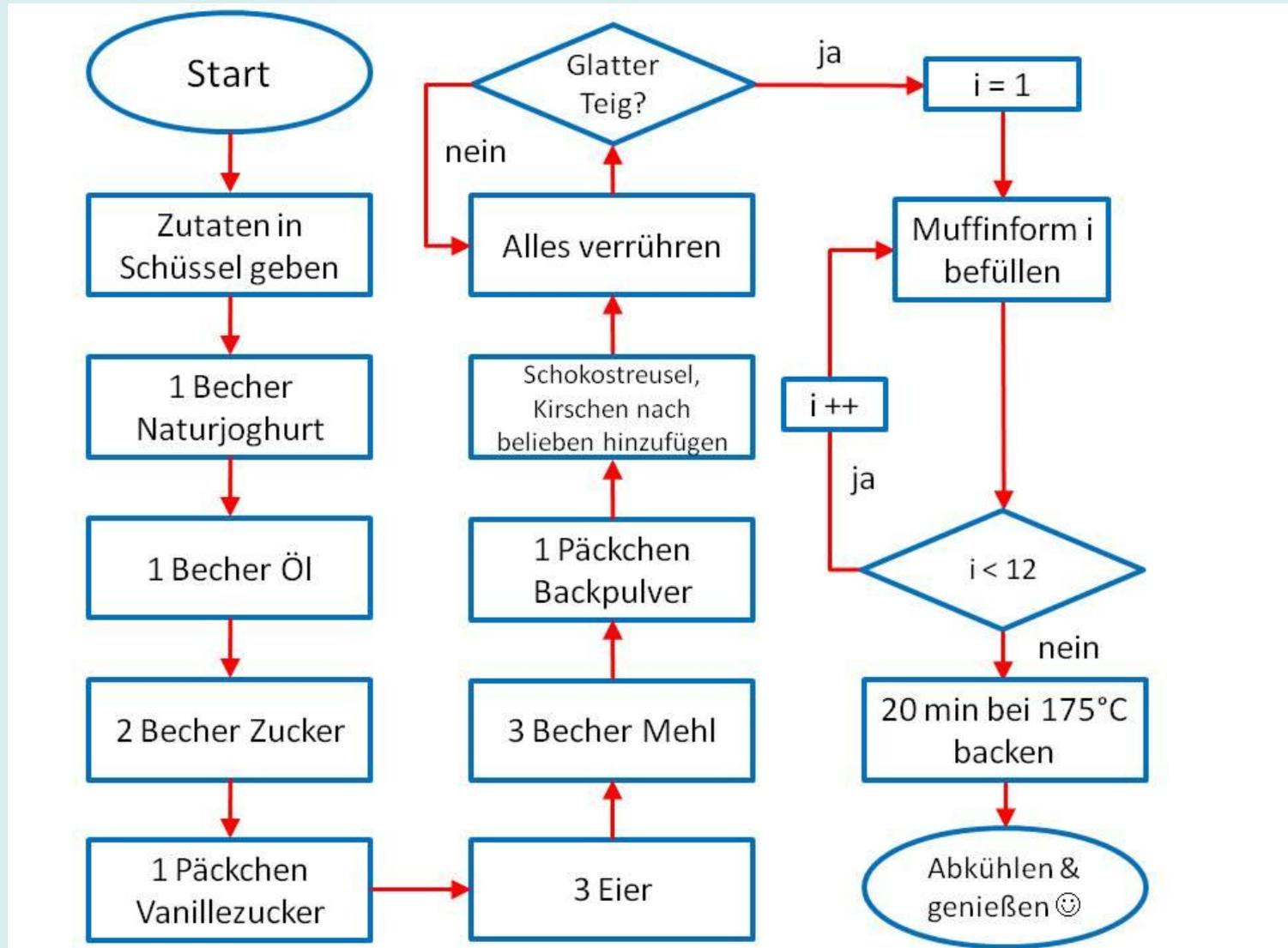
## Definition

**Eindeutige Handlungsvorschrift** zur Lösung eines Problems oder einer Klasse von Problemen.

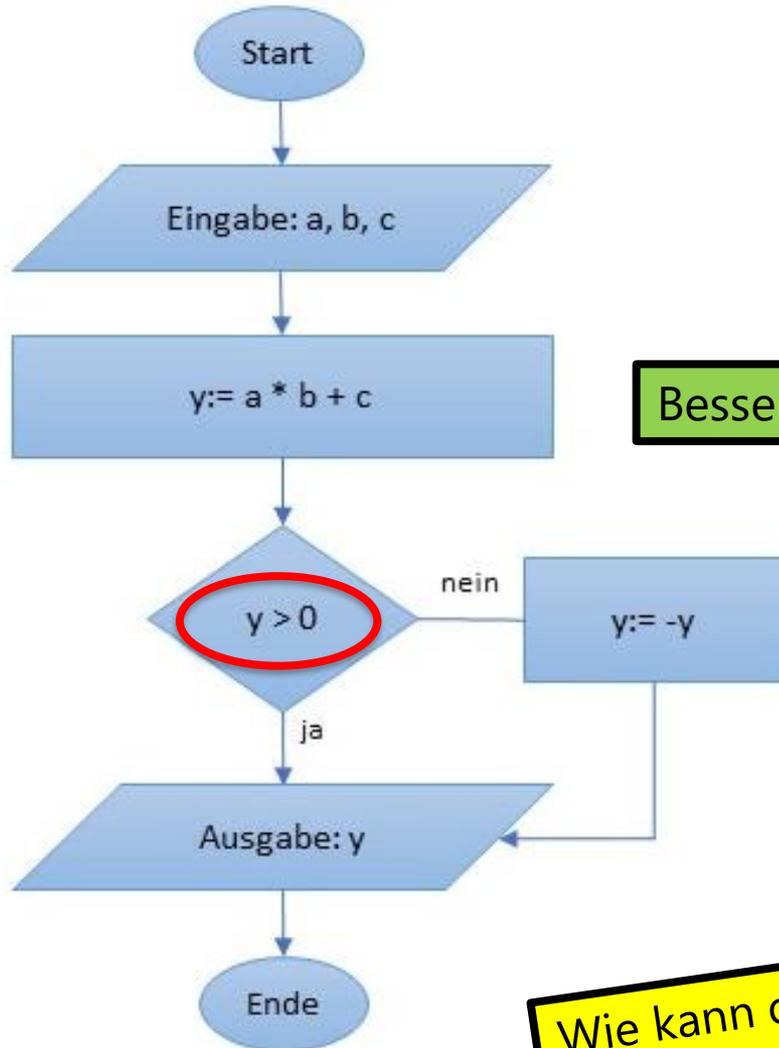


„Das kurz gefasste Buch  
über die Rechenverfahren“

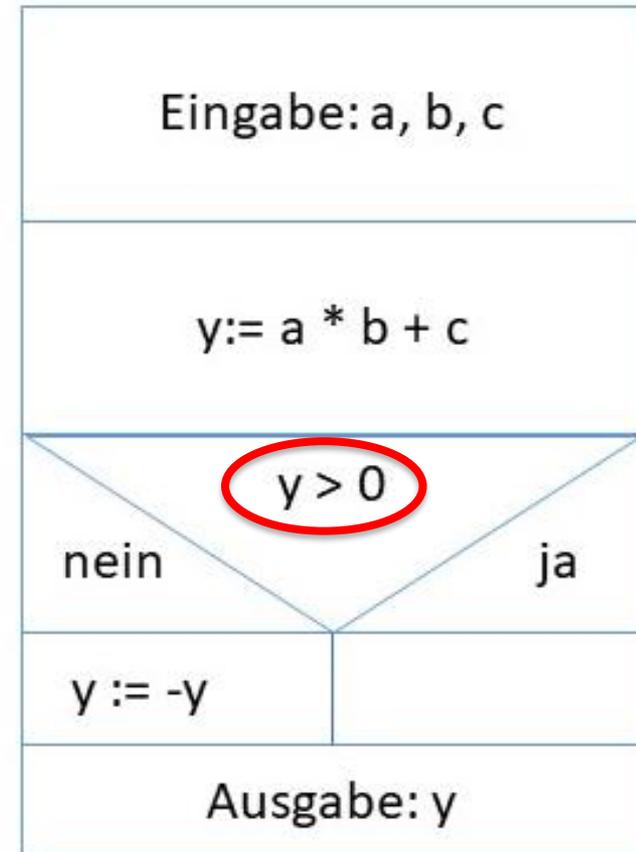
# Algorithmus für das Backen von Muffins



# Flussdiagramm vs. Struktogramm



Besser:  $y \geq 0$

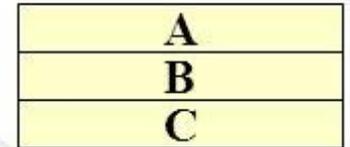
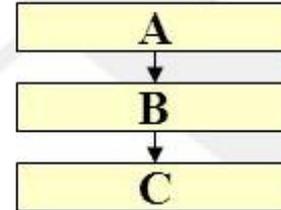


Wie kann das Programm verbessert werden?

# Grundelemente des Algorithmus

Sequenz

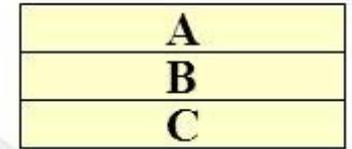
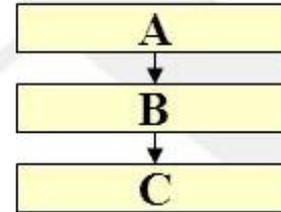
Führe erst A  
dann B dann C  
aus!



# Grundelemente des Algorithmus

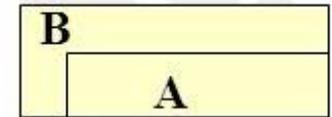
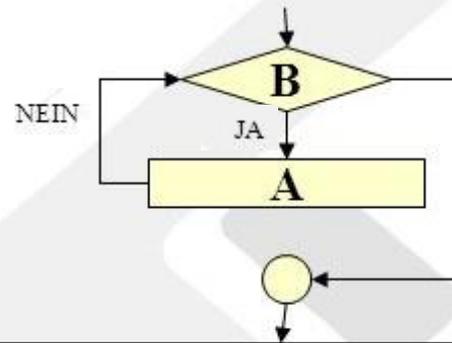
Sequenz

Führe erst A  
dann B dann C  
aus!



Iteration

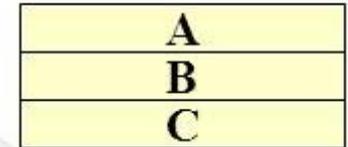
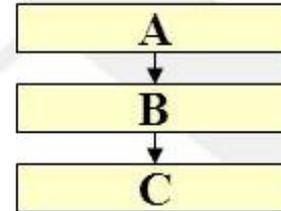
Wiederhole  
A solange  
Bedingung B  
wahr ist!



# Grundelemente des Algorithmus

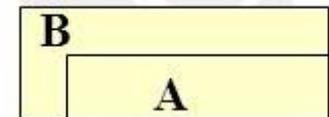
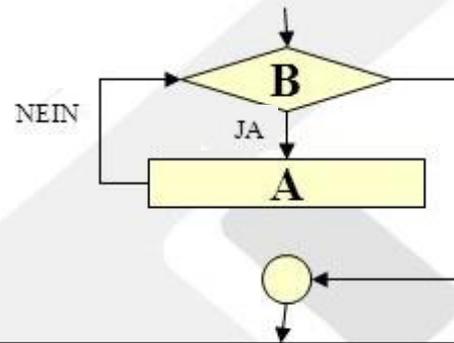
Sequenz

Führe erst A  
dann B dann C  
aus!



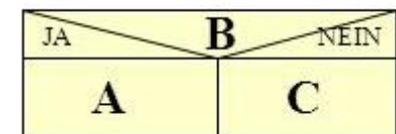
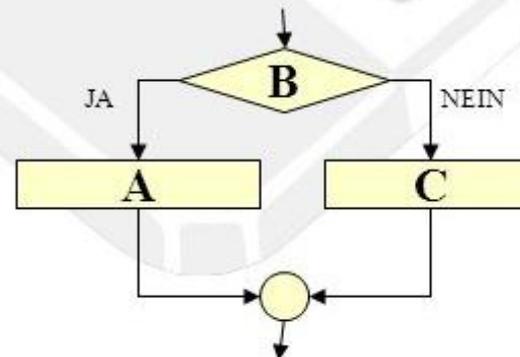
Iteration

Wiederhole  
A solange  
Bedingung B  
wahr ist!



Alternative

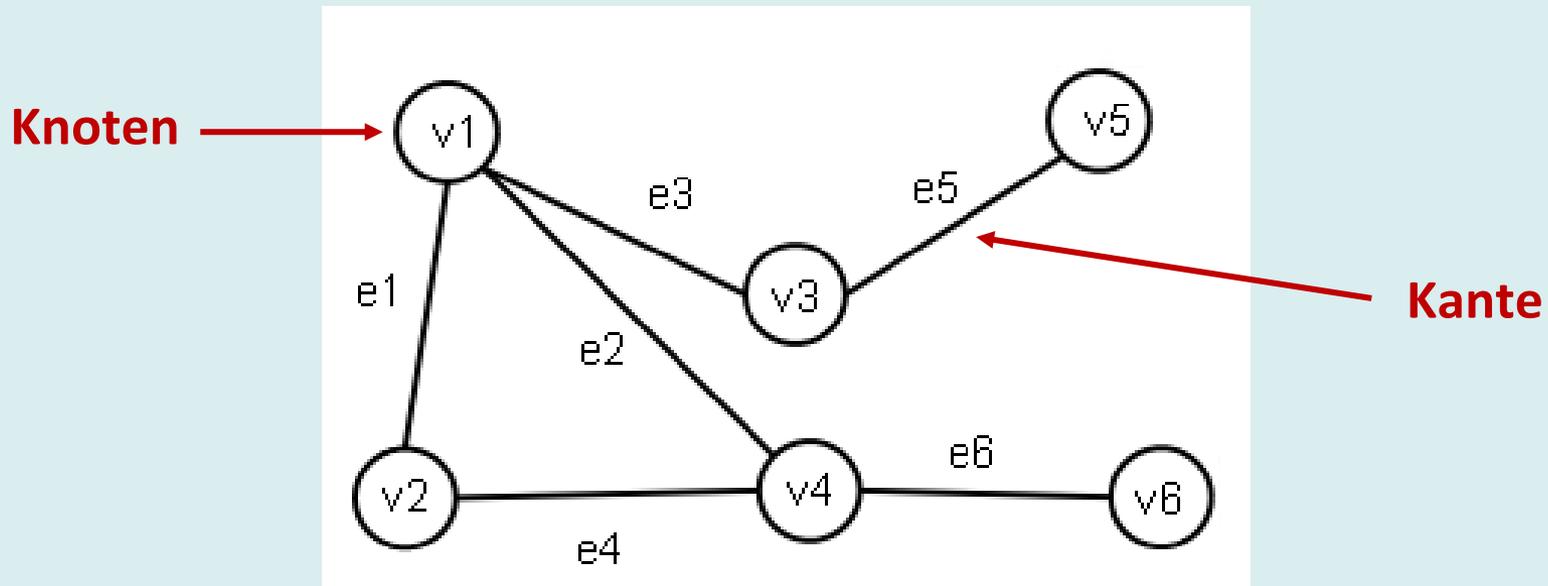
Wenn  
Bedingung B  
wahr ist, führe  
A aus. Sonst  
führe C aus.



Vorwissen:

# Der Graph

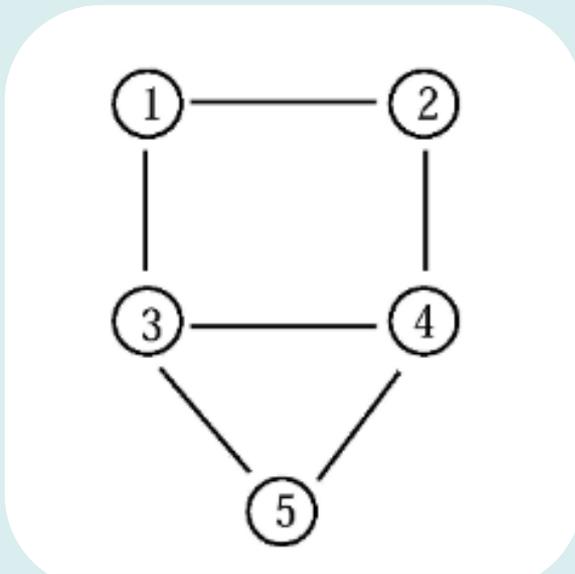
Def.: Repräsentiert eine **Menge von Objekten** mit den zwischen diesen Objekten bestehenden **Verbindungen**.



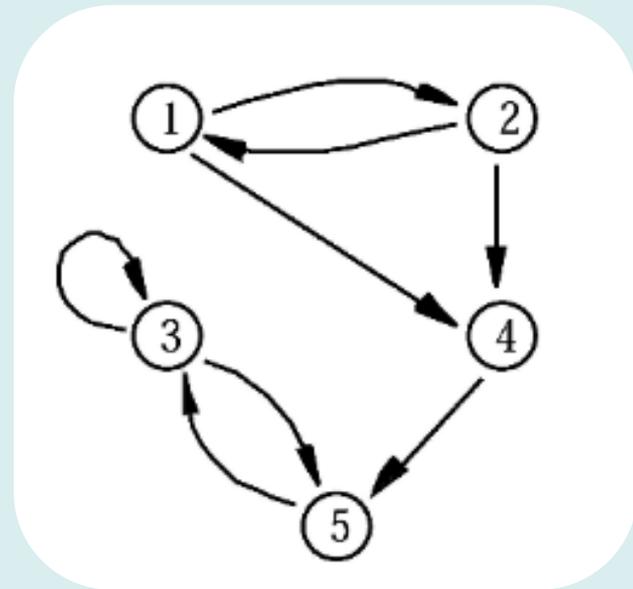
Ein Graph besteht aus **Knoten** (engl. vertex) und **Kanten** (engl. edge)

Vorwissen:

# Der Graph



Ungerichteter Graph

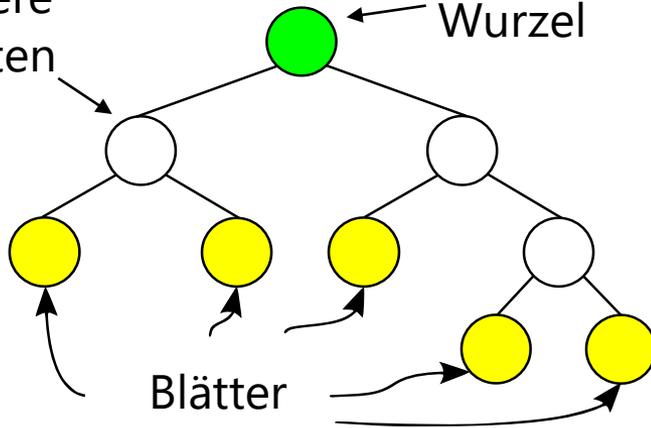


Gerichteter Graph

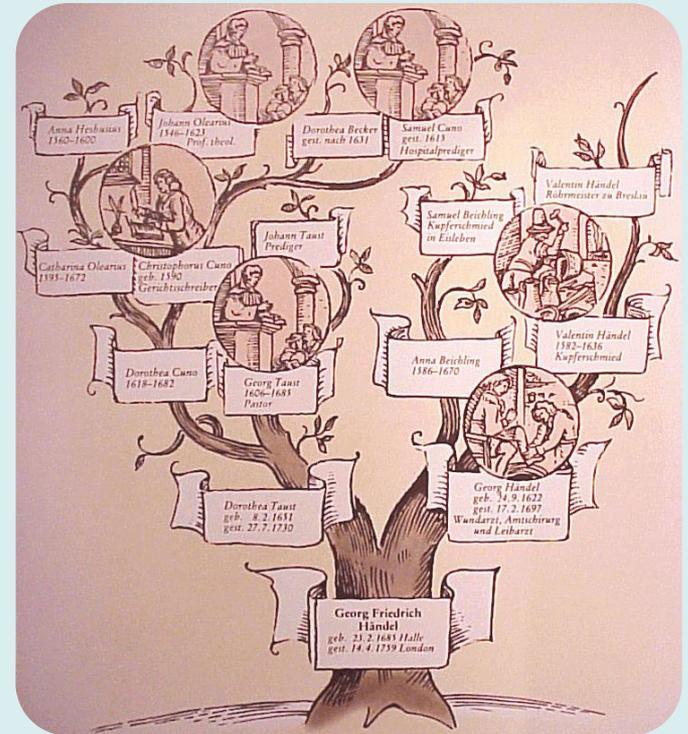
Vorwissen:

# Der Graph

Innere  
Knoten



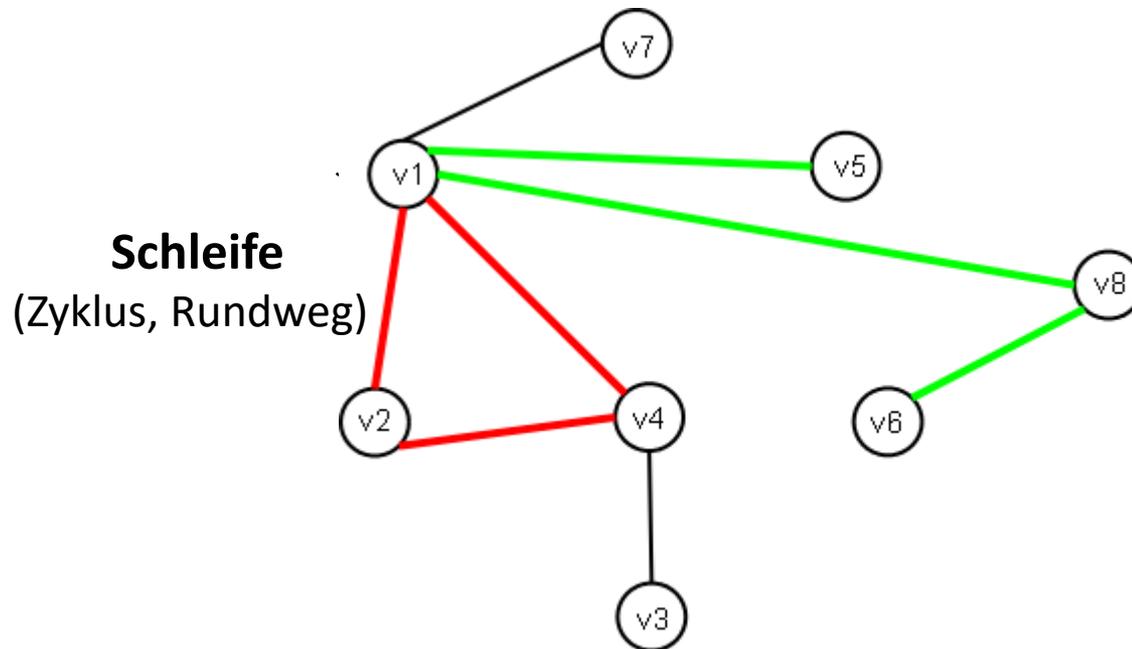
Baum



Beispiel: Stammbaum

Vorwissen:

# Der Graph



**Pfade** durch einen Graphen

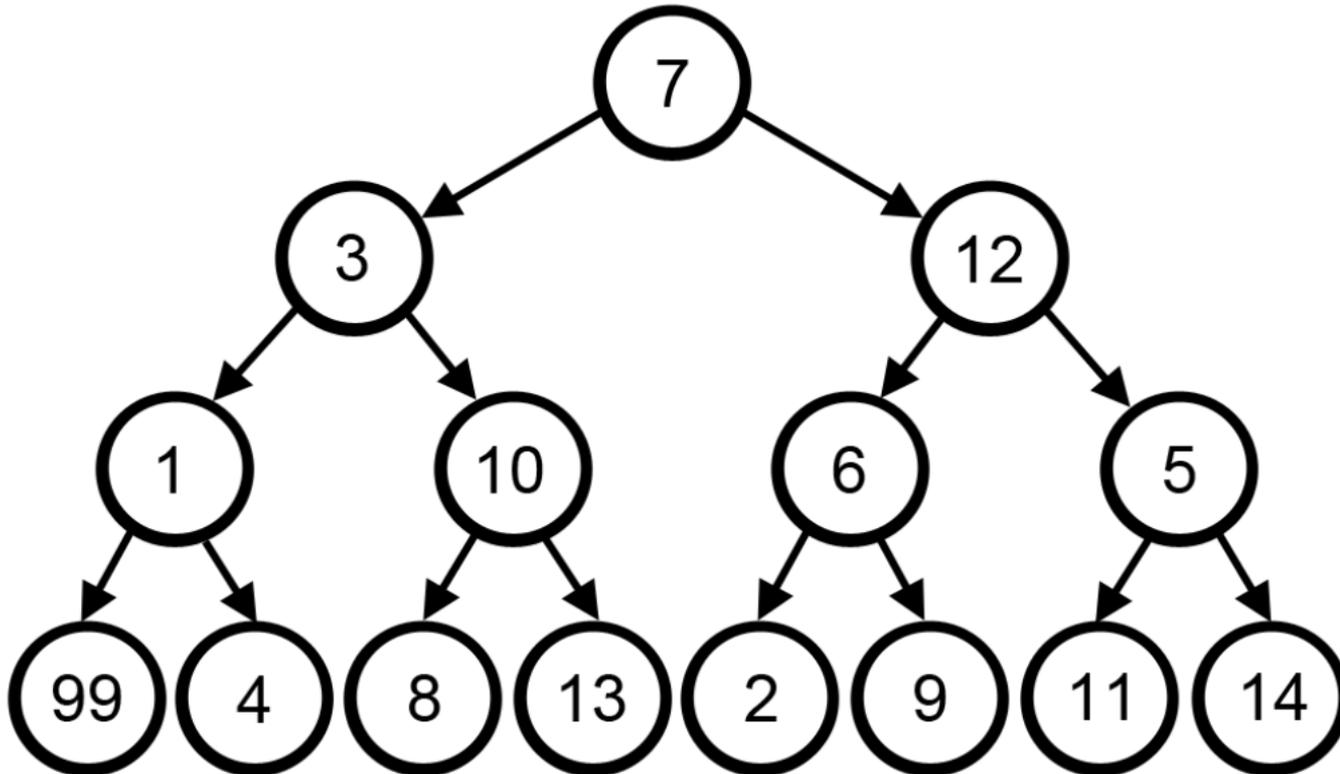
# Optimierungsprobleme

## Greedy Algorithmen

- Die Lösung wird schrittweise zusammengesetzt, wobei in jedem Schritt der **momentan beste** Folgeschritt ausgewählt wird.
- **Vorteil:** schnell und einfach zu implementieren.
- **Nachteil:** Findet selten die beste Lösung.

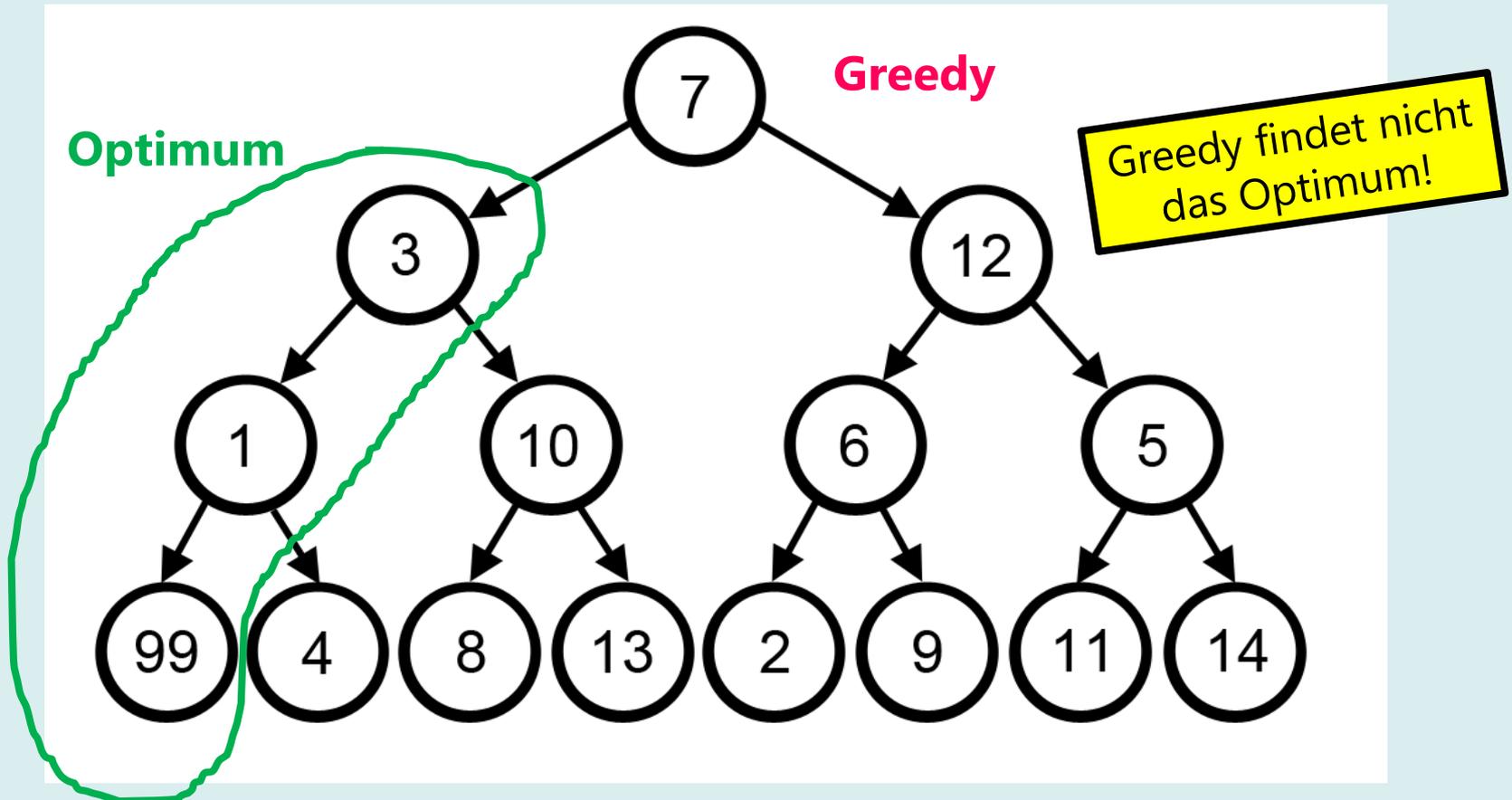
# Greedy Algorithmen

Starte an der Wurzel des Baums, entscheide dich für **rechts** oder **links** und **maximiere** die Summe der besuchten Knoten.



# Greedy Algorithmen

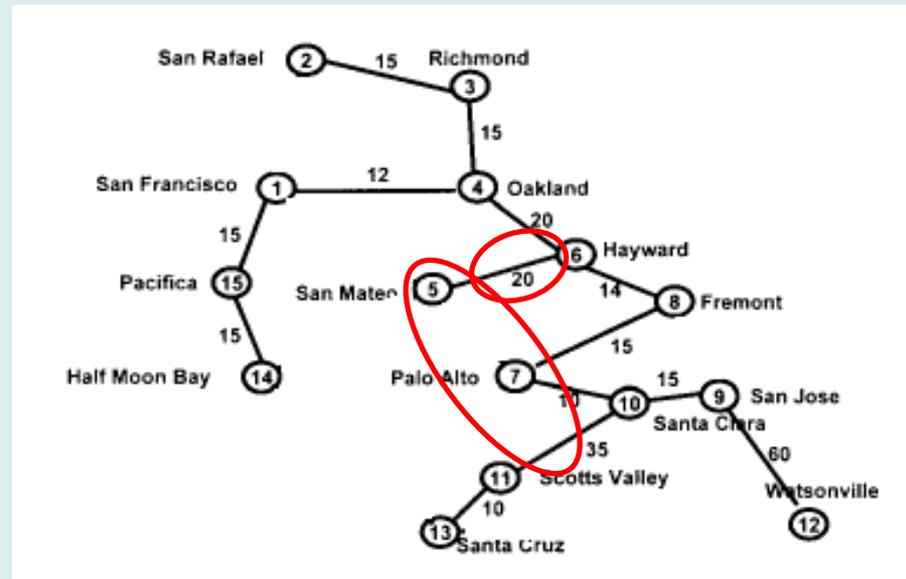
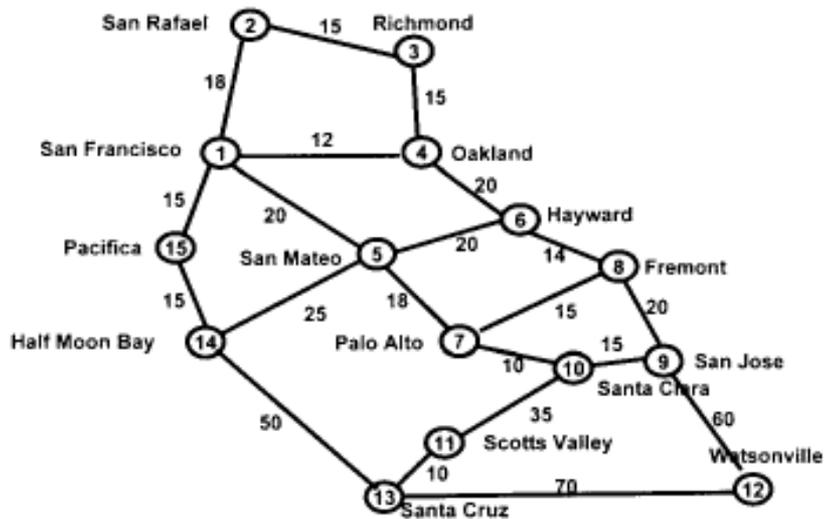
Starte an der Wurzel des Baums, entscheide dich für **rechts** oder **links** und **maximiere** die Summe der besuchten Knoten.



# Minimaler Spannbaum

**geg:** Graph mit Baukosten für Straßen

**ges:** Kostengünstigstes Straßennetz, das alle Städte verbindet



Fast richtige Lösung (Finde den Fehler)

# Minimaler Spannbaum

## Algorithmus vom **Kruskal** (→ greedy)

1. Wähle eine Kante mit **minimalem Kosten** aus.
2. Wähle eine Kante mit minimalem Kosten aus,  
die **keine Schleife** mit dem bereits bestehenden Netz bildet.
3. Gehe zu 2. wenn es noch freie Knoten gibt.

# Minimaler Spannbaum

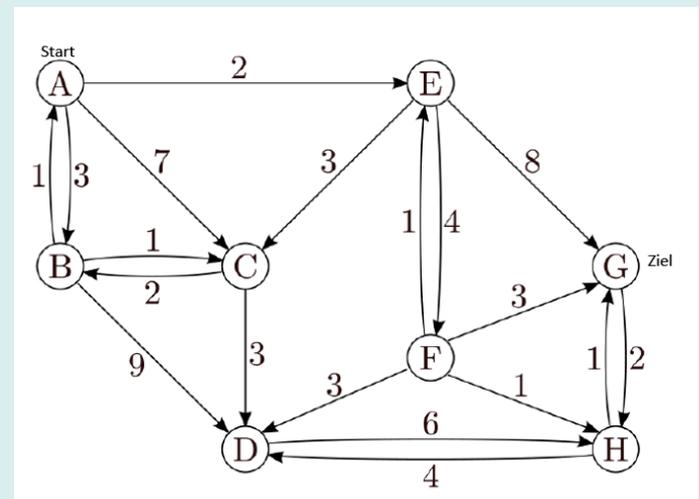
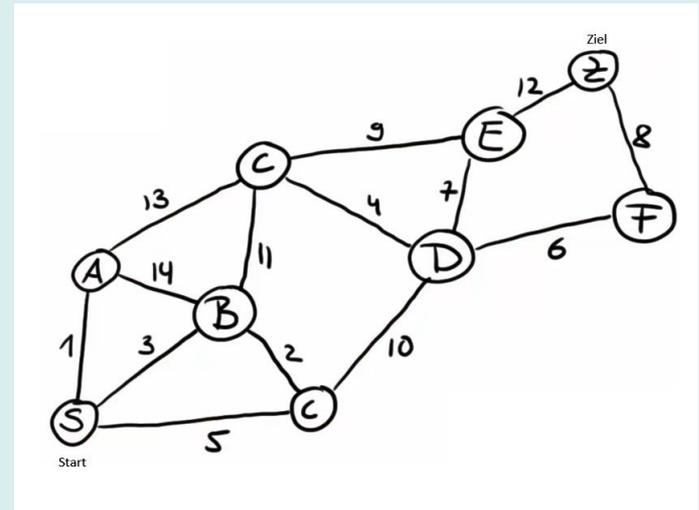
## Algorithmus vom **Prim** (---> greedy)

1. Wähle einen beliebigen Knoten als Startpunkt aus.
2. Wähle eine Kante mit **minimalem Kosten** aus, die einen freien Knoten mit dem bereits bestehenden Netz verbindet.
3. Gehe zu 2. wenn es noch freie Knoten gibt.

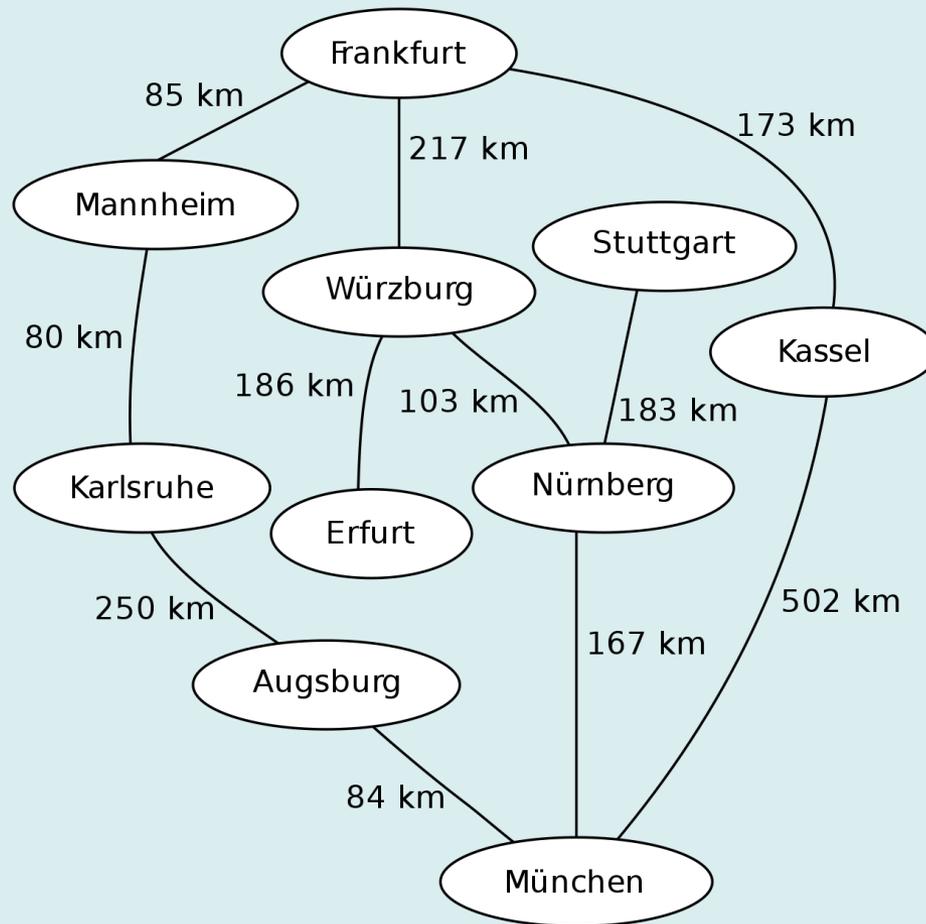
# Der Dijkstra-Algorithmus

## Problemstellung

Finde den kürzesten Pfad zwischen zwei Knoten in einem gerichteten bzw. ungerichteten Graphen.



# Finde den kürzesten Weg von Frankfurt nach München



## **Greedy:**

Frankfurt-Mannheim-Karlsruhe-Augsburg-München (499 km)

## **Optimum:**

Frankfurt-Würzburg-Nürnberg-München (487 km)

---

**Greedy Lösung für  
Nürnberg-Augsburg:  
Sackgasse!!**

---

# Der Dijkstra-Algorithmus

Greedy!

1. Schreibe auf den Startknoten eine **0** als "Distanzwert" und auf alle anderen Knoten  $\infty$ .
2. Gehe zu dem Knoten mit dem **kleinsten Distanzwert** und färbe diesen **rot** - dies signalisiert, dass der Knoten schon **besucht** wurde. (Gibt es mehrere solche Knoten, so wähle einen davon beliebig aus.) Markiere auch den Weg zu diesem Knoten.
3. Berechne für alle noch **unbesuchten** Knoten, die man vom aktuellen Standpunkt aus erreichen kann, die Distanz zum Startknoten. Ist dieser Wert **kleiner** als die dort eingetragene Distanz, so aktualisiere den Distanzwert (= Relaxation).
4. Gibt es noch **unbesuchte** Knoten, so gehe zu Schritt 2.

