

# Briefmarkensammeln

Funktioniert der Greedy-Algorithmus auch für Briefmarken aus Manchukuo?



Welche Briefmarken für einen 20 fen-Brief?

Der Greedy-Algorithmus führt nicht zu einer optimalen Lösung und findet manchmal gar keine!

# Briefmarkensammeln

Funktioniert der Greedy-Algorithmus auch für Briefmarken aus Manchukuo?



Welche Briefmarken für einen 20 fen-Brief?

Der Greedy-Algorithmus führt nicht zu einer optimalen Lösung und findet manchmal gar keine!

# Briefmarkensammeln

Funktioniert der Greedy-Algorithmus auch für Briefmarken aus Manchukuo?



Welche Briefmarken für einen 20 fen-Brief?

Der Greedy-Algorithmus führt nicht zu einer optimalen Lösung und findet manchmal gar keine!

# Matroide

Der Korrektheitsbeweis für Greedy-Algorithmen kann sehr trickreich sein. Münzwechsel ist hierfür ein Beispiel.

Viele Beweise können allerdings mit Hilfe von der Theorie der **Matroide** geführt werden.

## Definition (Matroid)

Ein **Matroid**  $M = (S, \mathcal{I})$  besteht aus einer **Basis**  $S$  und einer Familie  $\mathcal{I} \subseteq 2^S$  von **unabhängigen Mengen** mit:

- 1 Falls  $A \subseteq B$  und  $B \in \mathcal{I}$ , dann  $A \in \mathcal{I}$  ( $M$  ist **hereditary**).
- 2 Falls  $A, B \in \mathcal{I}$  und  $|A| < |B|$  dann gibt es ein  $x \in B$  so daß  $A \cup \{x\} \in \mathcal{I}$  ( $M$  hat die **Austauschenschaft**).

# Matroide

## Beispiel – Der graphische Matroid

Sei  $G = (V, E)$  ein ungerichteter Graph.

Sei  $\mathcal{F} = \{ F \subseteq E \mid (V, F) \text{ ist azyklisch} \}$ .

Dann ist  $(E, \mathcal{F})$  ein Matroid.

Zum Beweis machen wir zunächst eine Beobachtung:

Es sei  $G = (V, E)$  ein Wald. Dann verbindet die Kante  $e \in E$  zwei Bäume in  $G$  gdw.

$G = (V, E \cup \{e\})$  kreisfrei ist.

# Matroide

## Beispiel – Der graphische Matroid

Sei  $G = (V, E)$  ein ungerichteter Graph.

Sei  $\mathcal{F} = \{ F \subseteq E \mid (V, F) \text{ ist azyklisch} \}$ .

Dann ist  $(E, \mathcal{F})$  ein Matroid.

Zum Beweis machen wir zunächst eine Beobachtung:

Es sei  $G = (V, E)$  ein Wald. Dann verbindet die Kante  $e \in E$  zwei Bäume in  $G$  gdw.

$G = (V, E \cup \{e\})$  kreisfrei ist.

## Beweis.

- Vererbungseigenschaft: Wegnehmen von Kanten kann keine Kreise schließen.
- Austauscheseigenschaft: Seien  $G_A = (V, A)$  und  $G_B = (V, B)$  Teilwälder von  $G$  und  $|A| < |B|$ 
  - ① Beobachtung: Ein Wald mit  $k$  Kanten besteht aus  $|V| - k$  Bäumen.
  - ②  $G_A$  hat mehr Bäume als  $G_B$
  - ③ Es gibt einen Baum  $T$  in  $G_B$ , der zwei Bäume in  $G_A$  verbindet
  - ④ Es gibt eine Kante in  $T$ , die keinen Kreis in  $G_A$  schließt



# Matroide

Wir nennen eine unabhängige Menge  $A$  **maximal**, wenn keine echte Obermenge von  $A$  unabhängig ist.

Lemma (Lemma G1)

*Alle maximalen unabhängigen Mengen eines Matroids haben die gleiche Größe.*

Beweis.

Angenommen, daß  $A$  und  $B$  maximale unabhängige Mengen sind, aber  $|A| < |B|$ . Nach der Austausch Eigenschaft gibt es ein  $x \in B$ , so daß  $A \cup \{x\}$  unabhängig ist. Das widerspricht der Maximalität von  $A$ . □



# Matroide

Wir nennen eine unabhängige Menge  $A$  **maximal**, wenn keine echte Obermenge von  $A$  unabhängig ist.

## Lemma (Lemma G1)

*Alle maximalen unabhängigen Mengen eines Matroids haben die gleiche Größe.*

Beweis.

Angenommen, daß  $A$  und  $B$  maximale unabhängige Mengen sind, aber  $|A| < |B|$ . Nach der Austausch Eigenschaft gibt es ein  $x \in B$ , so daß  $A \cup \{x\}$  unabhängig ist. Das widerspricht der Maximalität von  $A$ . □

# Matroide

Wir nennen eine unabhängige Menge  $A$  **maximal**, wenn keine echte Obermenge von  $A$  unabhängig ist.

## Lemma (Lemma G1)

*Alle maximalen unabhängigen Mengen eines Matroids haben die gleiche Größe.*

## Beweis.

Angenommen, daß  $A$  und  $B$  maximale unabhängige Mengen sind, aber  $|A| < |B|$ . Nach der Austauschenschaft gibt es ein  $x \in B$ , so daß  $A \cup \{x\}$  unabhängig ist. Das widerspricht der Maximalität von  $A$ . □

# Gewichtete Matroide

- Ein **gewichtetes Matroid** ist ein Matroid  $M = (S, \mathcal{I})$  mit einer Gewichtsfunktion  $w: S \rightarrow \mathbf{Q}$ .
- Wir nennen eine Menge maximalen Gewichts unter allen maximalen unabhängigen Mengen **optimal**.
- Viele Optimierungsprobleme können durch das Finden einer optimalen Menge in einem Matroid gelöst werden.

## Beispiel

Ein minimaler Spannbaum ist eine optimale Menge im graphischen Matroid.

# Der Greedy-Algorithmus auf gewichteten Matroiden

Sei  $M = (S, \mathcal{I})$  ein Matroid mit Gewichten  $w$ .

Dieser Algorithmus findet eine optimale Menge.

## Algorithmus

**function** Greedy( $S$ ) :

$R := \text{emptyset}$ ;

sort  $S$  into  $(s[1], \dots, s[n])$  with  $w(s[i]) \geq w(s[i + 1])$ ;

**for**  $i = 1, \dots, n$  **do**

**if**  $R \cup \{s[i]\} \text{ subseteq } I$  **then**  $R := R \cup \{s[i]\}$  **fi**

**od**;

**return**  $R$

Die Laufzeit ist  $O(n \log n + nf(n))$ , wenn ein Vergleich konstante Zeit braucht, und der Unabhängigkeitstest  $O(f(n))$ .

# Korrektheit

## Lemma (G2)

*Sei  $M = (S, \mathcal{I})$  ein Matroid mit Gewichten  $w$ . Wenn  $x \in S$  maximales Gewicht unter allen  $x$  hat, für die  $\{x\}$  unabhängig ist, dann gibt eine optimale Menge, die  $x$  enthält.*

## Beweis.

Sei  $B$  eine optimale Menge mit  $x \notin B$ . Wir haben  $w(y) \leq w(x)$  für jedes  $y \in B$ , weil  $\{y\}$  nach der Vererbungseigenschaft unabhängig ist. Beginne mit  $A = \{x\}$  und füge Elemente von  $B$  zu  $A$  hinzu ( $A$  bleibt unabhängig, wegen der Austauschseigenschaft) bis  $|A| = |B|$ .

Dann ist  $A = B - \{y\} \cup \{x\}$  für ein  $y \in B$  und daher gilt  $w(A) \geq W(B)$ .



# Korrektheit

## Lemma (G2)

*Sei  $M = (S, \mathcal{I})$  ein Matroid mit Gewichten  $w$ . Wenn  $x \in S$  maximales Gewicht unter allen  $x$  hat, für die  $\{x\}$  unabhängig ist, dann gibt eine optimale Menge, die  $x$  enthält.*

## Beweis.

Sei  $B$  eine optimale Menge mit  $x \notin B$ . Wir haben  $w(y) \leq w(x)$  für jedes  $y \in B$ , weil  $\{y\}$  nach der Vererbungseigenschaft unabhängig ist. Beginne mit  $A = \{x\}$  und füge Elemente von  $B$  zu  $A$  hinzu ( $A$  bleibt unabhängig, wegen der Austauschseigenschaft) bis  $|A| = |B|$ .

Dann ist  $A = B - \{y\} \cup \{x\}$  für ein  $y \in B$  und daher gilt  $w(A) \geq W(B)$ .



# Korrektheit

## Lemma (G2)

*Sei  $M = (S, \mathcal{I})$  ein Matroid mit Gewichten  $w$ . Wenn  $x \in S$  maximales Gewicht unter allen  $x$  hat, für die  $\{x\}$  unabhängig ist, dann gibt eine optimale Menge, die  $x$  enthält.*

## Beweis.

Sei  $B$  eine optimale Menge mit  $x \notin B$ . Wir haben  $w(y) \leq w(x)$  für jedes  $y \in B$ , weil  $\{y\}$  nach der Vererbungseigenschaft unabhängig ist. Beginne mit  $A = \{x\}$  und füge Elemente von  $B$  zu  $A$  hinzu ( $A$  bleibt unabhängig, wegen der Austauschseigenschaft) bis  $|A| = |B|$ .

Dann ist  $A = B - \{y\} \cup \{x\}$  für ein  $y \in B$  und daher gilt  $w(A) \geq W(B)$ .



# Korrektheit

## Lemma (G2)

*Sei  $M = (S, \mathcal{I})$  ein Matroid mit Gewichten  $w$ . Wenn  $x \in S$  maximales Gewicht unter allen  $x$  hat, für die  $\{x\}$  unabhängig ist, dann gibt eine optimale Menge, die  $x$  enthält.*

## Beweis.

Sei  $B$  eine optimale Menge mit  $x \notin B$ . Wir haben  $w(y) \leq w(x)$  für jedes  $y \in B$ , weil  $\{y\}$  nach der Vererbungseigenschaft unabhängig ist. Beginne mit  $A = \{x\}$  und füge Elemente von  $B$  zu  $A$  hinzu ( $A$  bleibt unabhängig, wegen der Austauschseigenschaft) bis  $|A| = |B|$ .

Dann ist  $A = B - \{y\} \cup \{x\}$  für ein  $y \in B$  und daher gilt  $w(A) \geq W(B)$ .





# Kontraktion eines Matroids

## Definition

Sei  $M = (S, \mathcal{I})$  ein Matroid und  $x \in S$ .

Dann ist  $M' = (S', \mathcal{I}')$  die **Kontraktion von  $M$  um  $x$** , wobei

- $S' = \{y \in S \mid \{x, y\} \in \mathcal{I}, x \neq y\}$ ,
- $\mathcal{I}' = \{A \subseteq S - \{x\} \mid A \cup \{x\} \in \mathcal{I}\}$ .

Wir beobachten, daß der Greedy-Algorithmus auf  $M'$  arbeitet, nachdem er  $x$  gewählt hat.

# Kontraktion eines Matroids

## Definition

Sei  $M = (S, \mathcal{I})$  ein Matroid und  $x \in S$ .

Dann ist  $M' = (S', \mathcal{I}')$  die **Kontraktion von  $M$  um  $x$** , wobei

- $S' = \{y \in S \mid \{x, y\} \in \mathcal{I}, x \neq y\}$ ,
- $\mathcal{I}' = \{A \subseteq S - \{x\} \mid A \cup \{x\} \in \mathcal{I}\}$ .

Wir beobachten, daß der Greedy-Algorithmus auf  $M'$  arbeitet, nachdem er  $x$  gewählt hat.

### Lemma (G3)

*Sei  $M = (S, \mathcal{I})$  ein Matroid mit Gewichten  $w$ . Sei  $x$  das erste Element, daß durch den Greedy-Algorithmus gewählt wird.*

*Dann ist eine optimale Menge für die Kontraktion  $M'$  von  $M$  um  $x$  zusammen mit  $x$  eine optimale Menge für  $M$ .*

Beweis.

Sei  $x \in A$  und  $B = A - \{x\}$ . Dann ist  $A$  maximal unabhängig in  $M$  gdw.  $B$  maximal unabhängig in  $M'$  ist.

Da  $w(A) = w(B) + w(x)$  gilt, ist  $A$  optimal in  $M$  gdw.  $B$  optimal in  $M'$  ist. □

### Lemma (G3)

Sei  $M = (S, \mathcal{I})$  ein Matroid mit Gewichten  $w$ . Sei  $x$  das erste Element, daß durch den Greedy-Algorithmus gewählt wird.

Dann ist eine optimale Menge für die Kontraktion  $M'$  von  $M$  um  $x$  zusammen mit  $x$  eine optimale Menge für  $M$ .

### Beweis.

Sei  $x \in A$  und  $B = A - \{x\}$ . Dann ist  $A$  maximal unabhängig in  $M$  gdw.  $B$  maximal unabhängig in  $M'$  ist.

Da  $w(A) = w(B) + w(x)$  gilt, ist  $A$  optimal in  $M$  gdw.  $B$  optimal in  $M'$  ist. □

### Lemma (G3)

Sei  $M = (S, \mathcal{I})$  ein Matroid mit Gewichten  $w$ . Sei  $x$  das erste Element, daß durch den Greedy-Algorithmus gewählt wird.

Dann ist eine optimale Menge für die Kontraktion  $M'$  von  $M$  um  $x$  zusammen mit  $x$  eine optimale Menge für  $M$ .

### Beweis.

Sei  $x \in A$  und  $B = A - \{x\}$ . Dann ist  $A$  maximal unabhängig in  $M$  gdw.  $B$  maximal unabhängig in  $M'$  ist.

Da  $w(A) = w(B) + w(x)$  gilt, ist  $A$  optimal in  $M$  gdw.  $B$  optimal in  $M'$  ist. □

# Korrektheit des Greedy-Algorithmus

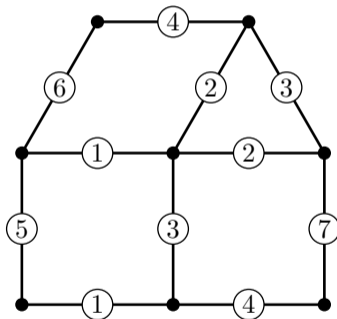
## Theorem

*Der Greedy-Algorithmus berechnet eine optimale Menge in einem gewichteten Matroid.*

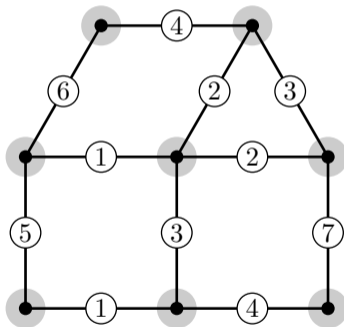
## Beweis.

Aus  $G_2$  und  $G_3$  folgt die Korrektheit mittels Induktion über die Größe der maximalen unabhängigen Menge, die nach  $G_1$  wohldefiniert ist. □

# Kruskal: Minimaler Spannbaum

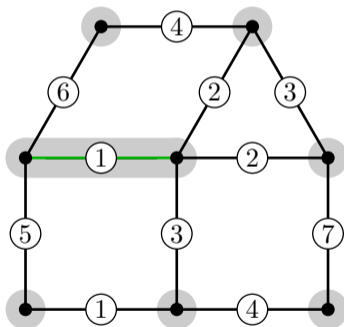


# Kruskal: Minimaler Spannbaum

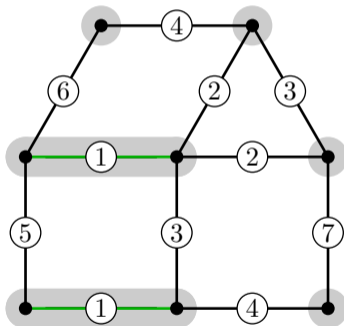




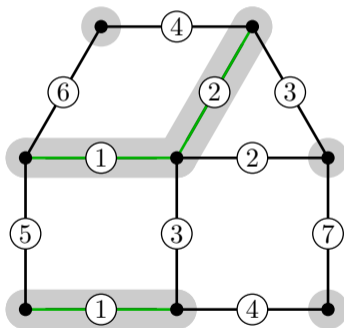
# Kruskal: Minimaler Spannbaum



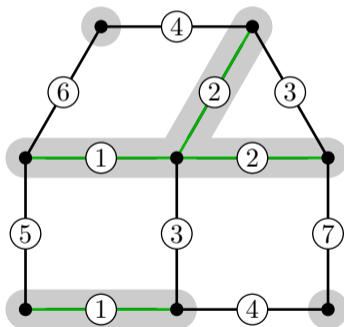
# Kruskal: Minimaler Spannbaum



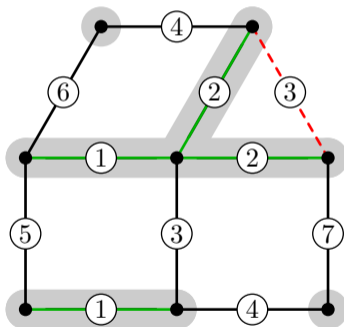
# Kruskal: Minimaler Spannbaum



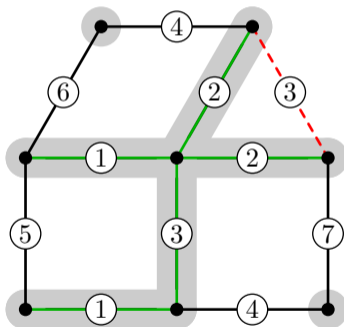
# Kruskal: Minimaler Spannbaum



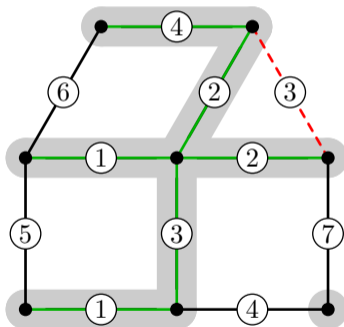
# Kruskal: Minimaler Spannbaum



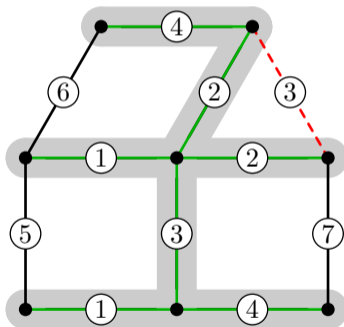
# Kruskal: Minimaler Spannbaum



# Kruskal: Minimaler Spannbaum

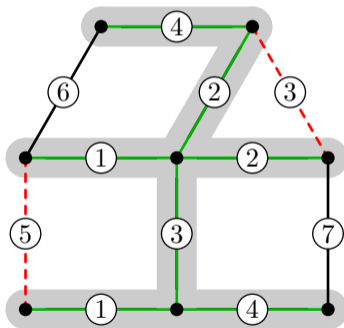


# Kruskal: Minimaler Spannbaum

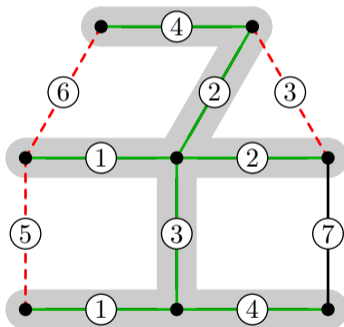




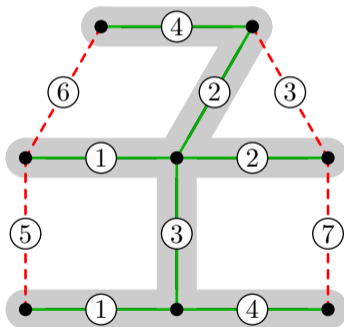
# Kruskal: Minimaler Spannbaum



# Kruskal: Minimaler Spannbaum



# Kruskal: Minimaler Spannbaum



# Kruskals Algorithmus – Implementierung

## Algorithmus

**function** Kruskal( $G, w$ ) :

$A := \text{emptyset};$

**for** each vertex  $v$  in  $V[G]$  **do**

$\text{Make\_Set}(v)$

**od**;

sort the edges of  $E$  into nondecreasing order by weight;

**for** each edge  $\{u, v\}$  in  $E$ , nondecreasingly **do**

**if**  $\text{Find\_Set}(u) \neq \text{Find\_Set}(v)$  **then**

$A := A \cup \{ \{u, v\} \};$

$\text{Union}(u, v)$

**fi**

**od**;

**return**  $A$

# Korrektheit und Laufzeit der Implementierung

Die Korrektheit folgt als Korollar aus der Korrektheit des allgemeinen Greedy-Algorithmus und der Beobachtung zum graphischen Matroid.

Laufzeit mit  $m = |E|$  und  $n = |V|$  und  $m \geq n - 1$ :

$n$  Make-Set-Operationen,

$O(m \log m)$  Zeit für das Sortieren der Kanten, und

$O(m)$  Find-Set- und Union-Operationen.

Mit einer geeigneten Union-Find-Implementierung zusammen  $O(m \log m)$ .

## Korrektheit und Laufzeit der Implementierung

Die Korrektheit folgt als Korollar aus der Korrektheit des allgemeinen Greedy-Algorithmus und der Beobachtung zum graphischen Matroid.

Laufzeit mit  $m = |E|$  und  $n = |V|$  und  $m \geq n - 1$ :

$n$  Make-Set-Operationen,

$O(m \log m)$  Zeit für das Sortieren der Kanten, und

$O(m)$  Find-Set- und Union-Operationen.

Mit einer geeigneten Union-Find-Implementierung zusammen  $O(m \log m)$ .