매트로이드와 그리디 알고리즘

Matroids and Greedy Algorithms

Jongseo Lee

KAIST 수학문제연구회 $\mathcal{M}^2$
KAIST School of Freshman
November 18, 2019

Today's Topic

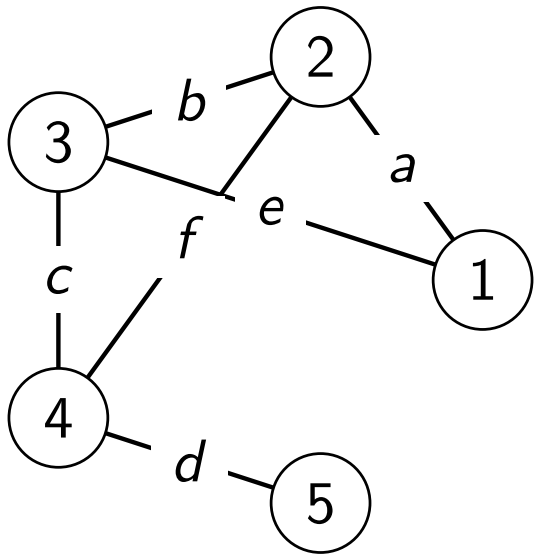
- What is a matroid?

Today's Topic

- What is a matroid?
- Greedy algorithms

Preliminaries

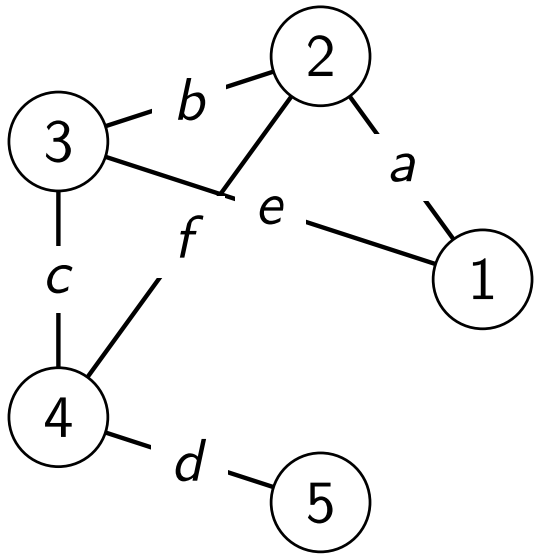
Let $G = (V, E)$ be a graph.



Preliminaries

Let $G = (V, E)$ be a graph.

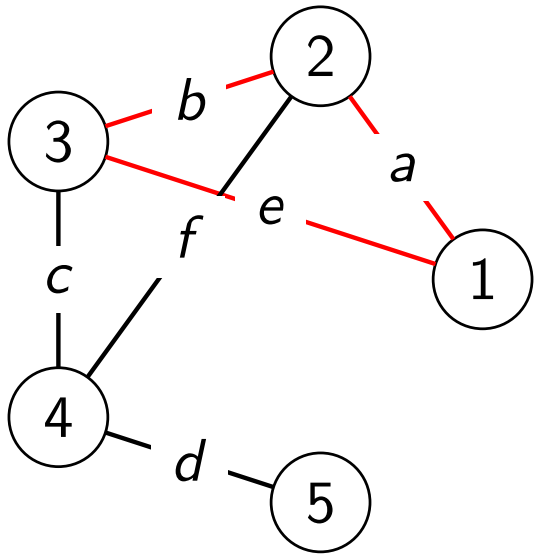
- Cycle



Preliminaries

Let $G = (V, E)$ be a graph.

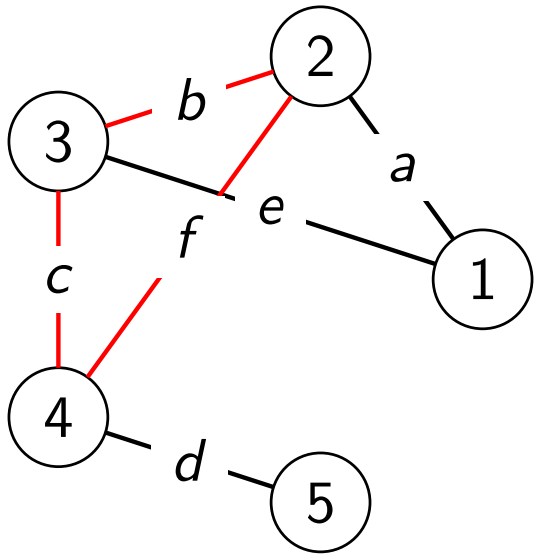
- Cycle



Preliminaries

Let $G = (V, E)$ be a graph.

- Cycle



Preliminaries

Let $G = (V, E)$ be a graph.

- Tree: connected graph without cycle

Preliminaries

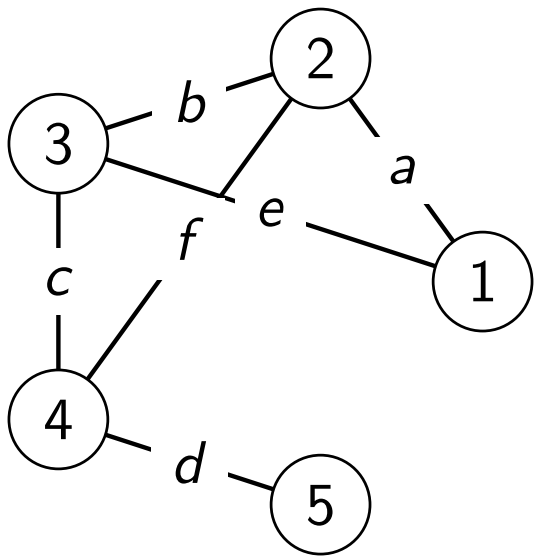
Let $G = (V, E)$ be a graph.

- Tree: connected graph without cycle
- Spanning tree

Preliminaries

Let $G = (V, E)$ be a graph.

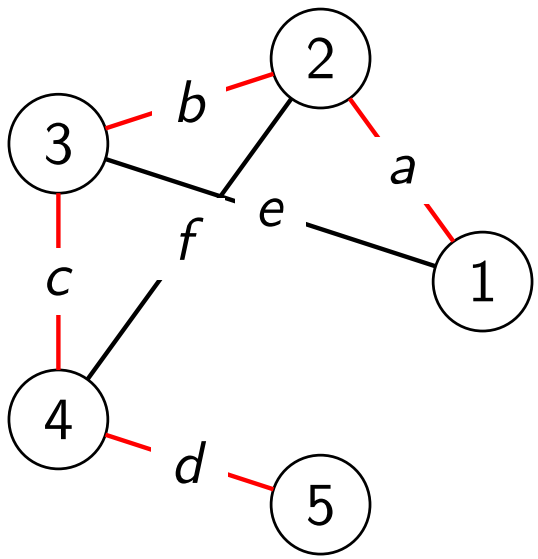
- Tree: connected graph without cycle
- Spanning tree



Preliminaries

Let $G = (V, E)$ be a graph.

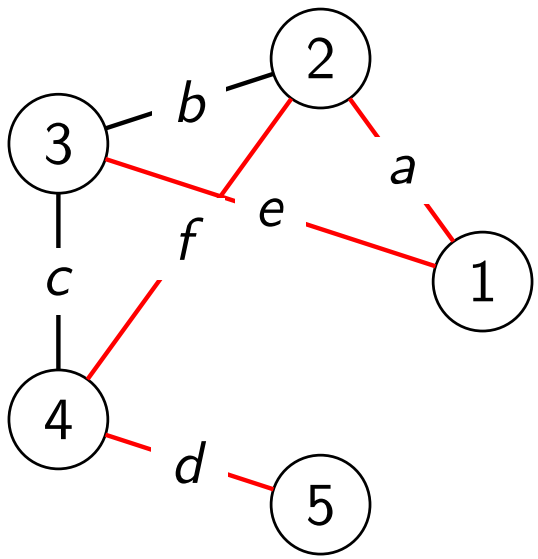
- Tree: connected graph without cycle
- Spanning tree



Preliminaries

Let $G = (V, E)$ be a graph.

- Tree: connected graph without cycle
- Spanning tree



Section 1

What is a matroid?

Example from matrix

Consider:

- $A = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 3 & 0 & 3 \\ 2 & 4 & 2 & 6 \end{bmatrix}$ over the field $\mathbb{R}$.
- $C = \{1, 2, 3, 4\}$: the set of column vectors

Example from matrix

Consider:

- $A = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 3 & 0 & 3 \\ 2 & 4 & 2 & 6 \end{bmatrix}$ over the field $\mathbb{R}$.
- $C = \{1, 2, 3, 4\}$: the set of column vectors

Then,

- $\emptyset$, $\{1, 2\}$, $\{1, 2, 3\}$, $\{3, 4\}$, $\{2, 3\}$, $\cdots$: linearly independent
- $\{1, 2, 4\}$, $\{1, 2, 3, 4\}$, $\cdots$: linearly dependent

Example from matrix

Consider:

- $A = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 3 & 0 & 3 \\ 2 & 4 & 2 & 6 \end{bmatrix}$ over the field $\mathbb{R}$.
- $C = \{1, 2, 3, 4\}$: the set of column vectors

Then,

- $\emptyset$, $\{1, 2\}$, $\{1, 2, 3\}$, $\{3, 4\}$, $\{2, 3\}$, $\cdots$: linearly independent
- $\{1, 2, 4\}$, $\{1, 2, 3, 4\}$, $\cdots$: linearly dependent

There is a strange fact:

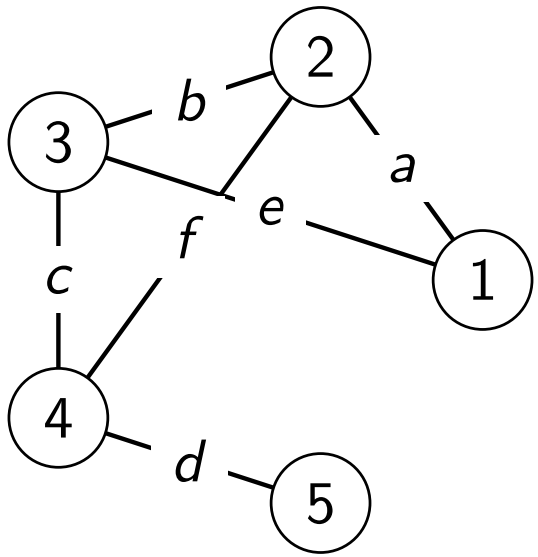
- All the bases(=maximal independent sets) have the same size.

Example from graph

Let $G = (V, E)$ be a graph.

Example from graph

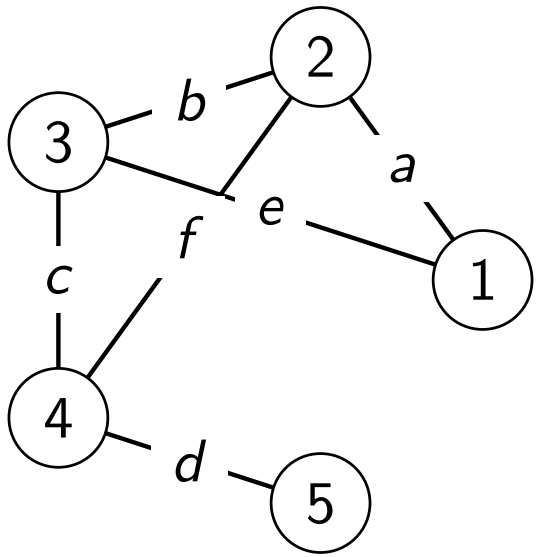
Let $G = (V, E)$ be a graph.



- $\{a\}, \{a, b\}, \{a, b, c\}, \{a, b, c, d\}, \{a, c, d, f\}$: acyclic
- $\{a, b, c, e\}$, $\{a, b, e\}$, $\{a, b, c, f\}$: contains cycle

Example from graph

Let $G = (V, E)$ be a graph.



There is a strange fact:

- All the maximal acyclic subset(i.e. spanning tree) of E have the same size.

What is a matroid?

Can we find a common generalization of those examples?

What is a matroid?

Can we find a common generalization of those examples?

Matroid [Whitney, 1935]

A matroid is a pair $M = (S, \mathcal{I})$ of a finite set S and a set $\mathcal{I} \subseteq 2^S$ satisfying:

(I1) $\emptyset \in \mathcal{I}$

(I2) If $X \subseteq Y$ and $Y \in \mathcal{I}$, then $X \in \mathcal{I}$

(I3) If $X, Y \in \mathcal{I}$ and $|X| < |Y|$, then $X \cup \{e\} \in \mathcal{I}$ for some $e \in Y - X$.

What is a matroid?

Can we find a common generalization of those examples?

Matroid [Whitney, 1935]

A matroid is a pair $M = (S, \mathcal{I})$ of a finite set S and a set $\mathcal{I} \subseteq 2^S$ satisfying:

(I1) $\emptyset \in \mathcal{I}$

(I2) If $X \subseteq Y$ and $Y \in \mathcal{I}$, then $X \in \mathcal{I}$

(I3) If $X, Y \in \mathcal{I}$ and $|X| < |Y|$, then $X \cup \{e\} \in \mathcal{I}$ for some $e \in Y - X$.

(I3) looks somewhat similar to following theorem from linear algebra.

(a) If S is a linearly independent set, and if $\mathbf{v} \in V$ but $\mathbf{v} \notin \text{span}(S)$, then the set $S \cup \{\mathbf{v}\}$ is linearly independent.

What is a matroid?

Can we find a common generalization of those examples?

Matroid [Whitney, 1935]

A matroid is a pair $M = (S, \mathcal{I})$ of a finite set S and a set $\mathcal{I} \subseteq 2^S$ satisfying:

(I1) $\emptyset \in \mathcal{I}$

(I2) If $X \subseteq Y$ and $Y \in \mathcal{I}$, then $X \in \mathcal{I}$

(I3) If $X, Y \in \mathcal{I}$ and $|X| < |Y|$, then $X \cup \{e\} \in \mathcal{I}$ for some $e \in Y - X$.

We can easily obtain following property from (I3).

Property

All the maximal *independent* set have the same size, and they are called the *bases*. We denote the set of bases as $\mathcal{B}$.

Example of matroids

Uniform matroids

For $n \geq r \geq 0$, $U_{r,n}$ is a matroid on $[n] = \{1, 2, \cdots, n\}$ such that X is independent if and only if $|X| \leq r$.

Example of matroids

Uniform matroids

For $n \geq r \geq 0$, $U_{r,n}$ is a matroid on $[n] = \{1, 2, \cdots, n\}$ such that X is independent if and only if $|X| \leq r$.

Vector matroid

For a matrix A over the field $\mathbb{F}$, the *vector matroid* $M[A]$ is the matroid on C, the column set of A, where $X \subseteq C$ is independent if and only if $A[X]$ is linearly independent.

Example of matroids

Uniform matroids

For $n \geq r \geq 0$, $U_{r,n}$ is a matroid on $[n] = \{1, 2, \cdots, n\}$ such that X is independent if and only if $|X| \leq r$.

Vector matroid

For a matrix A over the field $\mathbb{F}$, the *vector matroid* $M[A]$ is the matroid on C, the column set of A, where $X \subseteq C$ is independent if and only if $A[X]$ is linearly independent.

Note that:

- If $A = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 3 & 0 & 3 \\ 2 & 4 & 2 & 4 \end{bmatrix}$, then $A[\{1, 2, 4\}] = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 3 & 3 \\ 2 & 4 & 4 \end{bmatrix}$

Example of matroids

Uniform matroids

For $n \geq r \geq 0$, $U_{r,n}$ is a matroid on $[n] = \{1, 2, \cdots, n\}$ such that X is independent if and only if $|X| \leq r$.

Vector matroid

For a matrix A over the field $\mathbb{F}$, the *vector matroid* $M[A]$ is the matroid on C, the column set of A, where $X \subseteq C$ is independent if and only if $A[X]$ is linearly independent.

We can easily check that the base of $M[A]$ corresponds to the column base of A.

Example of matroids

Uniform matroids

For $n \geq r \geq 0$, $U_{r,n}$ is a matroid on $[n] = \{1, 2, \cdots, n\}$ such that X is independent if and only if $|X| \leq r$.

Vector matroid

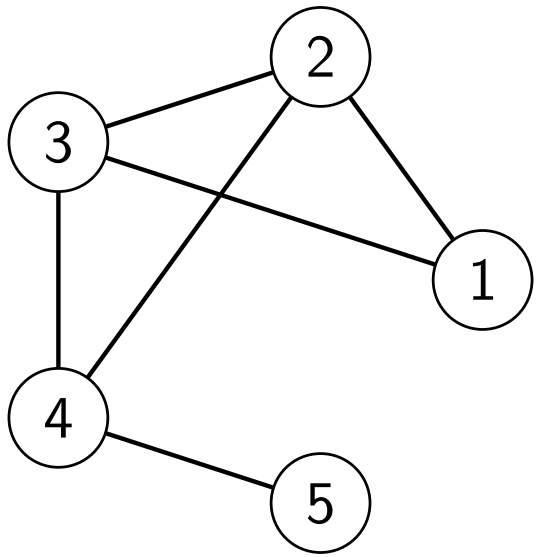
For a matrix A over the field $\mathbb{F}$, the *vector matroid* $M[A]$ is the matroid on C, the column set of A, where $X \subseteq C$ is independent if and only if $A[X]$ is linearly independent.

Cycle matroid

For a graph $G = (V, E)$, the *Cycle matroid* $M(G)$ is a matroid on E s.t. $X \subseteq E$ is independent if X contains no cycle.

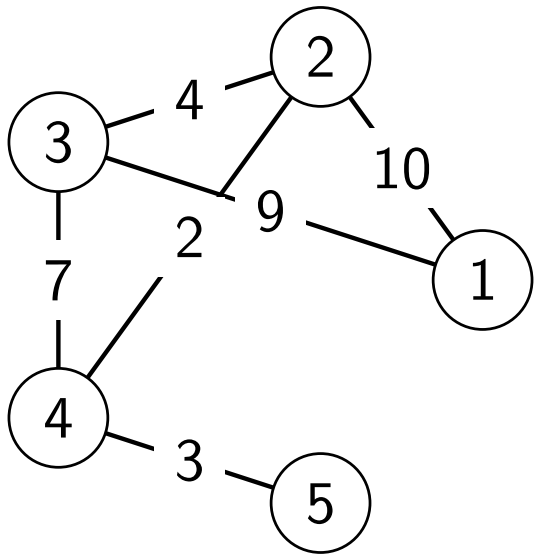
Graph vs. Weighted Graph

Graph $G = (V, E)$



Graph vs. Weighted Graph

Weighted graph $G = (V, E, w)$, $w : E \to \mathbb{R}$



Weighted matroids

Weighted matroids

- $M = (S, \mathcal{I}, w)$ is a weighted matroid if $(S, \mathcal{I})$ is a matroid and $w : S \to \mathbb{R}_{\geq 0}$

Weighted matroids

Weighted matroids

- $M = (S, \mathcal{I}, w)$ is a weighted matroid if $(S, \mathcal{I})$ is a matroid and $w : S \to \mathbb{R}_{\geq 0}$
- For $A \subseteq S$, the *weight* of A is the sum of its elements, that is, $w(A) = \sum_{e \in A} w(e)$.

Weighted matroids

Weighted matroids

- $M = (S, \mathcal{I}, w)$ is a weighted matroid if $(S, \mathcal{I})$ is a matroid and $w : S \to \mathbb{R}_{\geq 0}$
- For $A \subseteq S$, the *weight* of A is the sum of its elements, that is, $w(A) = \sum_{e \in A} w(e)$.
- $\mathcal{I}^\star := \{A \in \mathcal{I} : w(A) \text{ is maximum}\}$

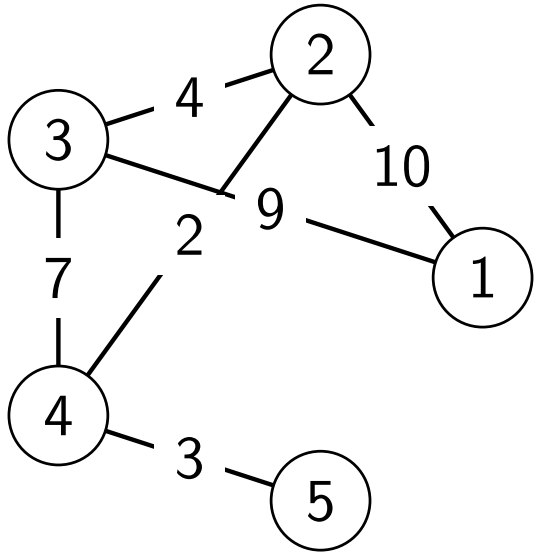
Weighted matroids

Weighted matroids

- $M = (S, \mathcal{I}, w)$ is a weighted matroid if $(S, \mathcal{I})$ is a matroid and $w : S \to \mathbb{R}_{\geq 0}$
- For $A \subseteq S$, the *weight* of A is the sum of its elements, that is, $w(A) = \sum_{e \in A} w(e)$.
- $\mathcal{I}^\star := \{A \in \mathcal{I} : w(A) \text{ is maximum}\}$
- $w(\mathcal{I}^\star) := \max_{A \in \mathcal{I}} w(A)$

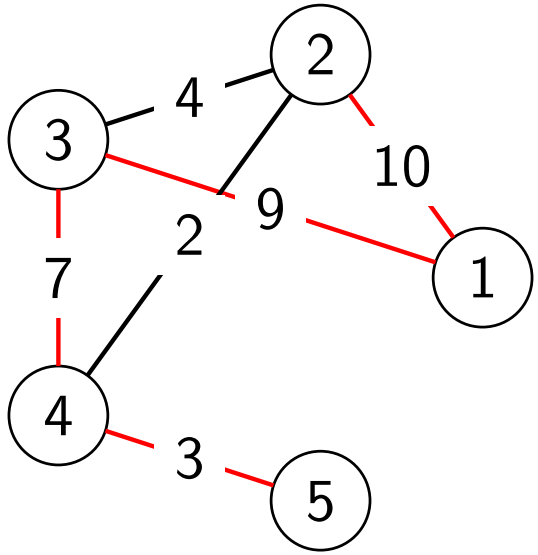
Example of weighted matroids

Consider following graph:



Example of weighted matroids

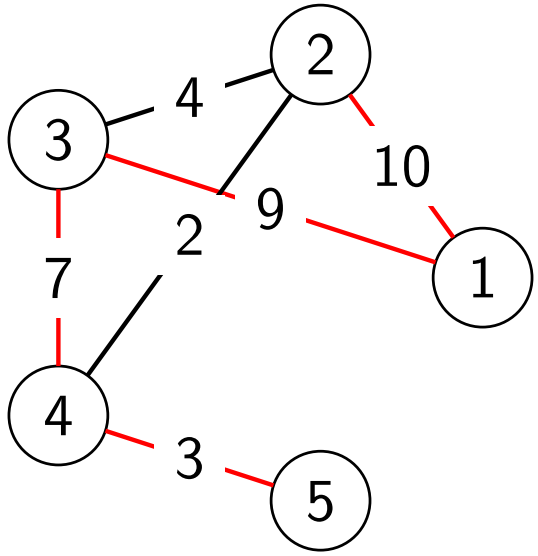
Consider following graph:



- $\mathcal{I}^\star$ is the set of maximum-weight spanning trees

Example of weighted matroids

Consider following graph:



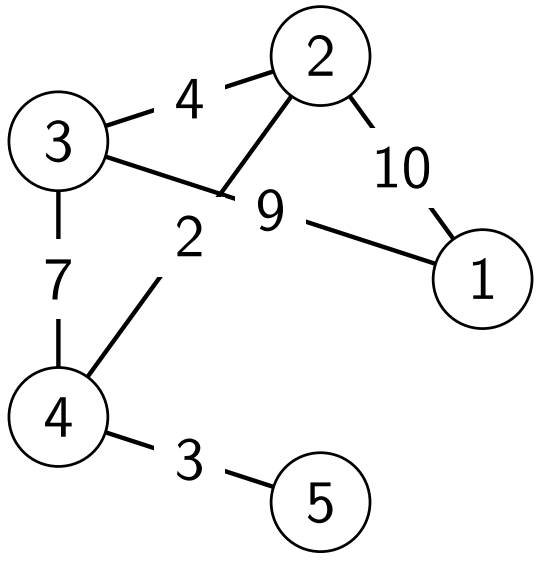
- $\mathcal{I}^\star$ is the set of maximum-weight spanning trees
- $w(\mathcal{I}^\star)$ is the weight of maximum spanning tree

Section 2

Greedy algorithms

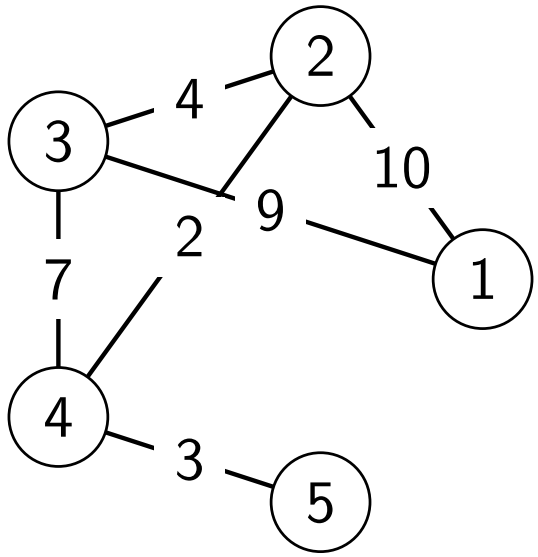
Spanning trees and greedy algorithms

Recall the previous example.



Spanning trees and greedy algorithms

Recall the previous example.



- How can we find maximum/minimum weighted spanning trees?

Spanning trees and greedy algorithms

You might heard of it if you have taken CS300 (Introduction to Algorithms) or etc.

Spanning trees and greedy algorithms

You might heard of it if you have taken CS300 (Introduction to Algorithms) or etc.

Kruskal's MST algorithm

SpanningTree($G = (V, E, w)$)

Spanning trees and greedy algorithms

You might heard of it if you have taken CS300 (Introduction to Algorithms) or etc.

Kruskal's MST algorithm

SpanningTree($G = (V, E, w)$)

- $T := \emptyset$

Spanning trees and greedy algorithms

You might heard of it if you have taken CS300 (Introduction to Algorithms) or etc.

Kruskal's MST algorithm

SpanningTree($G = (V, E, w)$)

- $T := \emptyset$
- Sort $E = \{e_1, e_2, \cdots, e_n\}$ so that $w(e_1) \geq w(e_2) \geq \cdots w(e_n)$.

Spanning trees and greedy algorithms

You might heard of it if you have taken CS300 (Introduction to Algorithms) or etc.

Kruskal's MST algorithm

SpanningTree($G = (V, E, w)$)

- $T := \emptyset$
- Sort $E = \{e_1, e_2, \cdots, e_n\}$ so that $w(e_1) \geq w(e_2) \geq \cdots w(e_n)$.
- for ($i := 1$ to n) do

Spanning trees and greedy algorithms

You might heard of it if you have taken CS300 (Introduction to Algorithms) or etc.

Kruskal's MST algorithm

SpanningTree($G = (V, E, w)$)

- $T := \emptyset$
- Sort $E = \{e_1, e_2, \cdots, e_n\}$ so that $w(e_1) \geq w(e_2) \geq \cdots w(e_n)$.
- for ($i := 1$ to n) do
  - if $T \cup \{e_i\}$ contains no cycle

Spanning trees and greedy algorithms

You might heard of it if you have taken CS300 (Introduction to Algorithms) or etc.

Kruskal's MST algorithm

SpanningTree($G = (V, E, w)$)

- $T := \emptyset$
- Sort $E = \{e_1, e_2, \cdots, e_n\}$ so that $w(e_1) \geq w(e_2) \geq \cdots w(e_n)$.
- for ($i := 1$ to n) do
  - if $T \cup \{e_i\}$ contains no cycle
    - $T := T \cup \{e_i\}$

Spanning trees and greedy algorithms

You might heard of it if you have taken CS300 (Introduction to Algorithms) or etc.

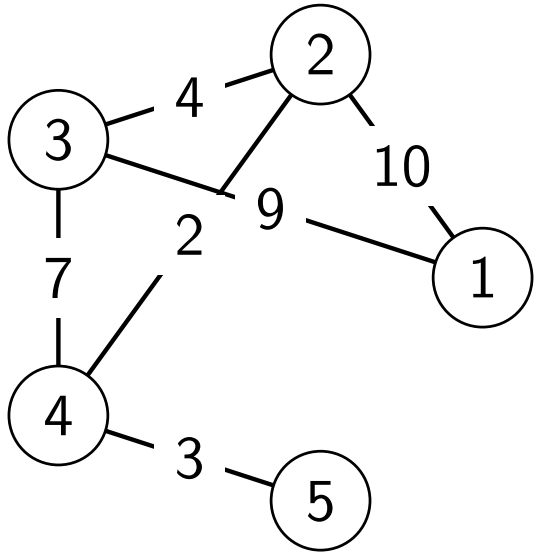
Kruskal's MST algorithm

SpanningTree($G = (V, E, w)$)

- $T := \emptyset$
- Sort $E = \{e_1, e_2, \cdots, e_n\}$ so that $w(e_1) \geq w(e_2) \geq \cdots w(e_n)$.
- for ($i := 1$ to n) do
  - if $T \cup \{e_i\}$ contains no cycle
    - $T := T \cup \{e_i\}$
- return T

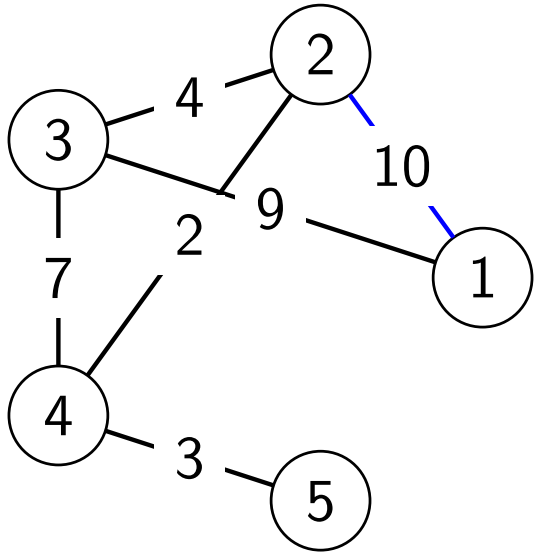
Kruskal's MST algorithm

Demo of Kruskal's algorithm



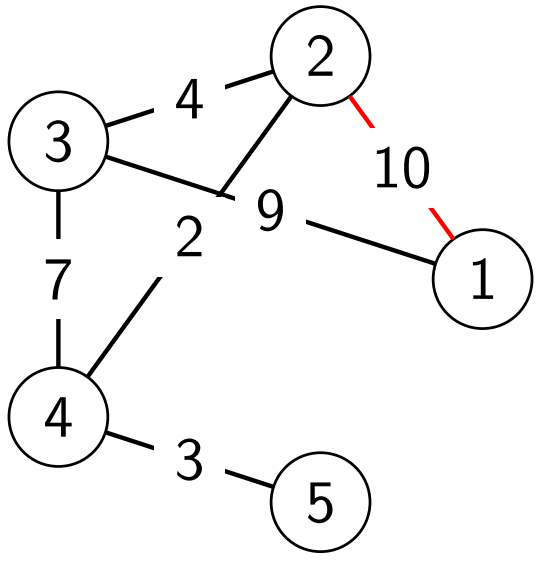
Kruskal's MST algorithm

Demo of Kruskal's algorithm



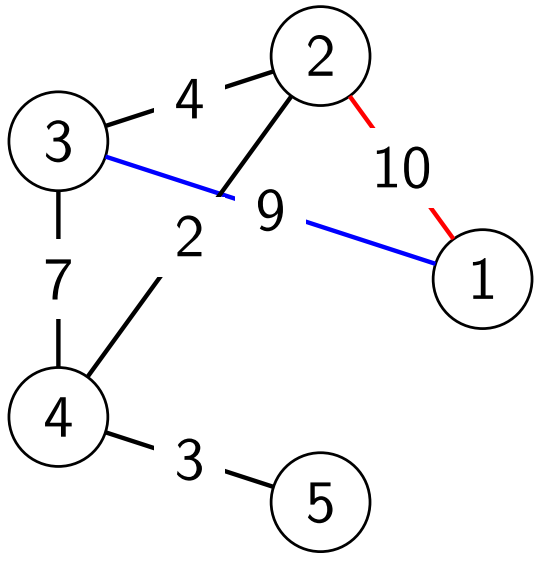
Kruskal's MST algorithm

Demo of Kruskal's algorithm



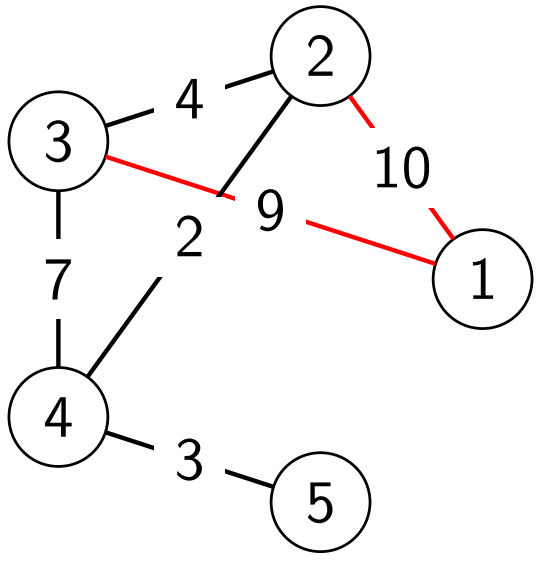
Kruskal's MST algorithm

Demo of Kruskal's algorithm



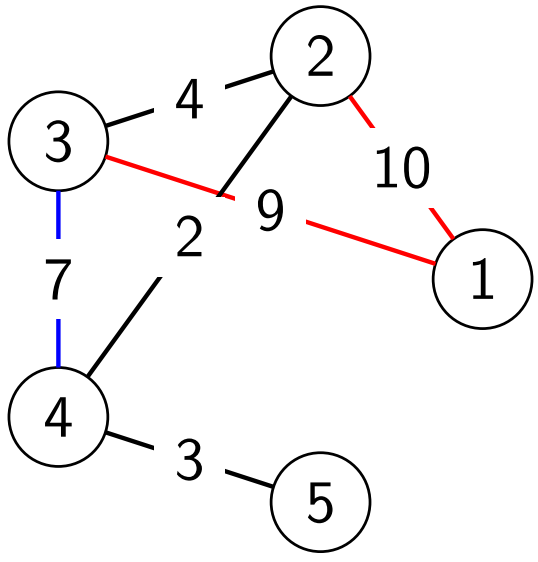
Kruskal's MST algorithm

Demo of Kruskal's algorithm



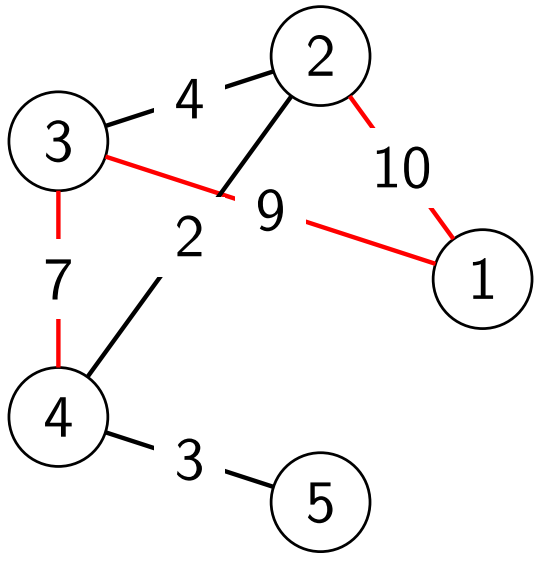
Kruskal's MST algorithm

Demo of Kruskal's algorithm



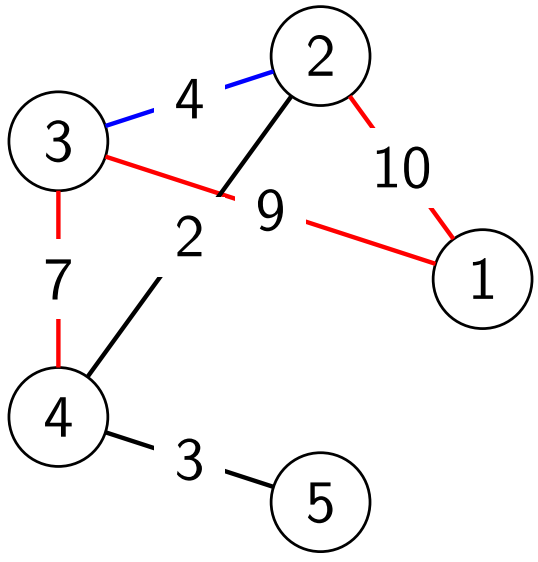
Kruskal's MST algorithm

Demo of Kruskal's algorithm



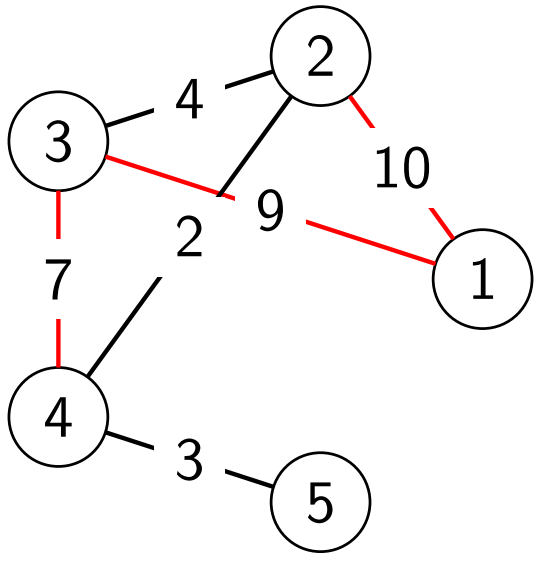
Kruskal's MST algorithm

Demo of Kruskal's algorithm



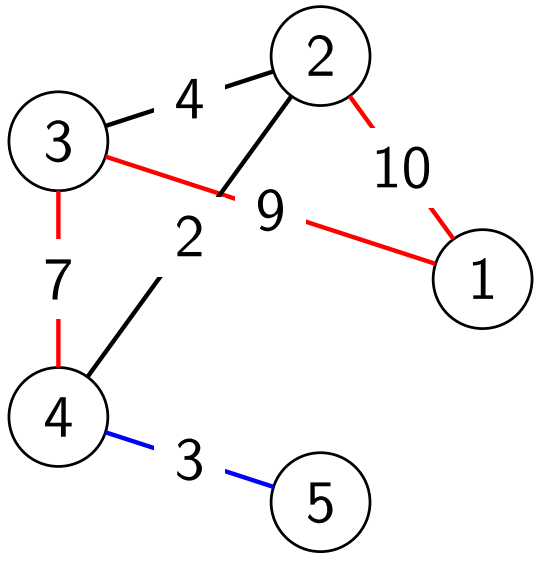
Kruskal's MST algorithm

Demo of Kruskal's algorithm



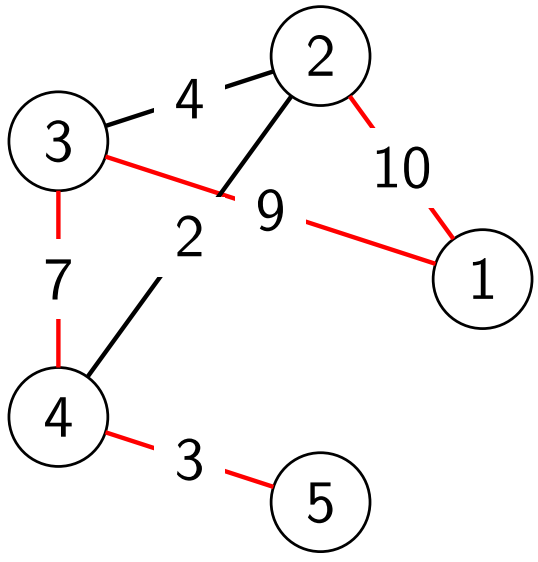
Kruskal's MST algorithm

Demo of Kruskal's algorithm



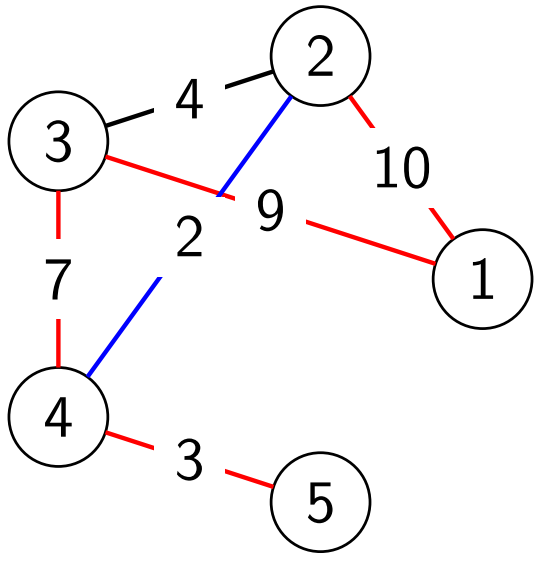
Kruskal's MST algorithm

Demo of Kruskal's algorithm



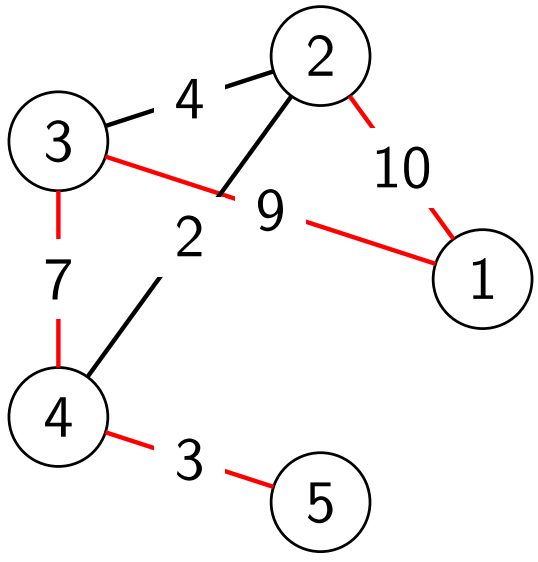
Kruskal's MST algorithm

Demo of Kruskal's algorithm



Kruskal's MST algorithm

Demo of Kruskal's algorithm



Spanning trees and greedy algorithms

You might heard of it if you have taken CS300 (Introduction to Algorithms) or etc.

Kruskal's MST algorithm

SpanningTree($G = (V, E, w)$)

- $T := \emptyset$
- Sort $E = \{e_1, e_2, \cdots, e_n\}$ so that $w(e_1) \geq w(e_2) \geq \cdots w(e_n)$.
- for ($i := 1$ to n) do
  - if $T \cup \{e_i\}$ contains no cycle
    - $T := T \cup \{e_i\}$
- return T

Spanning trees and greedy algorithms

You might heard of it if you have taken CS300 (Introduction to Algorithms) or etc.

Kruskal's MST algorithm

SpanningTree($G = (V, E, w)$)

- $T := \emptyset$
- Sort $E = \{e_1, e_2, \cdots, e_n\}$ so that $w(e_1) \geq w(e_2) \geq \cdots w(e_n)$.
- for ($i := 1$ to n) do
  - if $T \cup \{e_i\}$ contains no cycle
    - $T := T \cup \{e_i\}$
- return T

Above algorithm finds an element of $\mathcal{I}^\star$ of a cycle matroid. Can we generalize it more general?

Generalized algorithm

Generalized algorithm

Generalized greedy algorithm

Greedy($M = (S, \mathcal{I}, w)$) // Goal: to find some $A \in \mathcal{I}^\star$

Generalized algorithm

Generalized greedy algorithm

Greedy($M = (S, \mathcal{I}, w)$) `// Goal: to find some` $A \in \mathcal{I}^\star$

- $A := \emptyset$

Generalized algorithm

Generalized greedy algorithm

Greedy($M = (S, \mathcal{I}, w)$) // Goal: to find some $A \in \mathcal{I}^\star$

- $A := \emptyset$
- Sort $S = \{x_1, x_2, \cdots, x_n\}$ so that $w(x_1) \geq w(x_2) \geq \cdots \geq w(x_n)$.

Generalized algorithm

Generalized greedy algorithm

Greedy($M = (S, \mathcal{I}, w)$) // Goal: to find some $A \in \mathcal{I}^\star$

- $A := \emptyset$
- Sort $S = \{x_1, x_2, \cdots, x_n\}$ so that $w(x_1) \geq w(x_2) \geq \cdots \geq w(x_n)$.
- for ($i := 1$ to n) do

Generalized algorithm

Generalized greedy algorithm

Greedy($M = (S, \mathcal{I}, w)$) // Goal: to find some $A \in \mathcal{I}^\star$

- $A := \emptyset$
- Sort $S = \{x_1, x_2, \cdots, x_n\}$ so that $w(x_1) \geq w(x_2) \geq \cdots \geq w(x_n)$.
- for ($i := 1$ to n) do
  - if $A \cup \{x_i\} \in \mathcal{I}$

Generalized algorithm

Generalized greedy algorithm

Greedy($M = (S, \mathcal{I}, w)$) // Goal: to find some $A \in \mathcal{I}^\star$

- $A := \emptyset$
- Sort $S = \{x_1, x_2, \cdots, x_n\}$ so that $w(x_1) \geq w(x_2) \geq \cdots \geq w(x_n)$.
- for ($i := 1$ to n) do
  - if $A \cup \{x_i\} \in \mathcal{I}$
    - $A := A \cup \{x_i\}$

Generalized algorithm

Generalized greedy algorithm

Greedy($M = (S, \mathcal{I}, w)$) // Goal: to find some $A \in \mathcal{I}^\star$

- $A := \emptyset$
- Sort $S = \{x_1, x_2, \cdots, x_n\}$ so that $w(x_1) \geq w(x_2) \geq \cdots \geq w(x_n)$.
- for ($i := 1$ to n) do
  - if $A \cup \{x_i\} \in \mathcal{I}$
    - $A := A \cup \{x_i\}$
- return A // $A \in \mathcal{I}^\star$?

Generalized algorithm

Generalized greedy algorithm

Greedy($M = (S, \mathcal{I}, w)$) // Goal: to find some $A \in \mathcal{I}^\star$

- $A := \emptyset$
- Sort $S = \{x_1, x_2, \cdots, x_n\}$ so that $w(x_1) \geq w(x_2) \geq \cdots \geq w(x_n)$.
- for ($i := 1$ to n) do
  - if $A \cup \{x_i\} \in \mathcal{I}$
    - $A := A \cup \{x_i\}$
- return A // $A \in \mathcal{I}^\star$?

So, is this algorithm *actually* works (correctly) for any matroid?

Generalized algorithm

Generalized greedy algorithm

Greedy($M = (S, \mathcal{I}, w)$) // Goal: to find some $A \in \mathcal{I}^\star$

- $A := \emptyset$
- Sort $S = \{x_1, x_2, \cdots, x_n\}$ so that $w(x_1) \geq w(x_2) \geq \cdots \geq w(x_n)$.
- for ($i := 1$ to n) do
  - if $A \cup \{x_i\} \in \mathcal{I}$
    - $A := A \cup \{x_i\}$
- return A // $A \in \mathcal{I}^\star$?

So, is this algorithm *actually* works (correctly) for any matroid?

- Let's prove it!

Correctness Proof

For sake of convenience, sort $S = \{x_1, \cdots, x_n\}$ so that $w(x_1) \geq \cdots \geq w(x_n)$.

Correctness Proof

For sake of convenience, sort $S = \{x_1, \cdots, x_n\}$ so that $w(x_1) \geq \cdots \geq w(x_n)$.

Lemma 1

Let $\{x_i\}$ be the *first* element that $\{x_i\} \in \mathcal{I}$. Then, some $A \in \mathcal{I}^\star$ contains x_i.

Correctness Proof

Lemma 1

Let $\{x_i\}$ be the *first* element that $\{x_i\} \in \mathcal{I}$. Then, some $A \in \mathcal{I}^\star$ contains x_i.

- To prove it, take any $B \in \mathcal{I}^\star$.

Correctness Proof

Lemma 1

Let $\{x_i\}$ be the *first* element that $\{x_i\} \in \mathcal{I}$. Then, some $A \in \mathcal{I}^\star$ contains x_i.

- To prove it, take any $B \in \mathcal{I}^\star$.
- If $x_i \in B$, done. Otherwise, construct A inductively.

Correctness Proof

Lemma 1

Let $\{x_i\}$ be the *first* element that $\{x_i\} \in \mathcal{I}$. Then, some $A \in \mathcal{I}^\star$ contains x_i.

- To prove it, take any $B \in \mathcal{I}^\star$.
- If $x_i \in B$, done. Otherwise, construct A inductively.

Algorithm to construct A

- $A := \{x_i\}$ initially
- while $(|A| < |B|)$ do
  - Take $x' \in B - A$ such that $A \cup \{x'\} \in \mathcal{I}$ //(I3)
  - $A := A \cup \{x'\}$

Correctness Proof

Lemma 1

Let $\{x_i\}$ be the *first* element that $\{x_i\} \in \mathcal{I}$. Then, some $A \in \mathcal{I}^\star$ contains x_i.

- To prove it, take any $B \in \mathcal{I}^\star$.
- If $x_i \in B$, done. Otherwise, construct A inductively.

Algorithm to construct A

- $A := \{x_i\}$ initially
- while $(|A| < |B|)$ do
  - Take $x' \in B - A$ such that $A \cup \{x'\} \in \mathcal{I}$ //(I3)
  - $A := A \cup \{x'\}$

- Then, $A = B \cup \{x_i\} - \{x_j\}$ for some $j > i$ so that

$$w(I^\star) \geq w(A) = w(B) - w(x_j) + w(x_i) \geq w(B) = w(I^\star).$$

Correctness Proof

Lemma 2

For $x \in S$, if $\{x\} \notin \mathcal{I}$, then no element of $\mathcal{I}$ contains x.

Correctness Proof

Lemma 2

For $x \in S$, if $\{x\} \notin \mathcal{I}$, then no element of $\mathcal{I}$ contains x.

Its proof is clear from (I2).

Correctness Proof

Definition 1

For a weighted matroid $M = (S, \mathcal{I}, w)$ and $x \in S$, we define new weighted matroid $M_x = (S_x, \mathcal{I}_x, w_x)$ as:

- $S_x = \{y \in S - \{x\} : \{x, y\} \in \mathcal{I}\}$
- $\mathcal{I}_x = \{B \subseteq S - \{x\} : B \cup \{x\} \in \mathcal{I}\}$
- $w_x = w|_{S_x}$

if $S_x \neq \emptyset$, it is *acutally* a matroid.

Correctness Proof

Definition 1

For a weighted matroid $M = (S, \mathcal{I}, w)$ and $x \in S$, we define new weighted matroid $M_x = (S_x, \mathcal{I}_x, w_x)$ as:

- $S_x = \{y \in S - \{x\} : \{x, y\} \in \mathcal{I}\}$
- $\mathcal{I}_x = \{B \subseteq S - \{x\} : B \cup \{x\} \in \mathcal{I}\}$
- $w_x = w|_{S_x}$

if $S_x \neq \emptyset$, it is *acutally* a matroid.

Lemma 3 - Optimal substructure property

Let $M = (S, \mathcal{I}, w)$ be a weighted matroid and suppose $x \in S$ satisfies:

- x is contained by some element of $\mathcal{I}^\star$.

Then, for each $B \in \mathcal{I}_x^\star$, $B \cup \{x\} \in \mathcal{I}^\star$.

Correctness Proof

Lemma 3 - Optimal substructure property

Let $M = (S, \mathcal{I}, w)$ be a weighted matroid and suppose $x \in S$ satisfies:

- x is contained by some element of $\mathcal{I}^\star$.

Then, for each $B \in \mathcal{I}_x^\star$, $B \cup \{x\} \in \mathcal{I}^\star$.

Correctness Proof

Lemma 3 - Optimal substructure property

Let $M = (S, \mathcal{I}, w)$ be a weighted matroid and suppose $x \in S$ satisfies:

- x is contained by some element of $\mathcal{I}^\star$.

Then, for each $B \in \mathcal{I}_x^\star$, $B \cup \{x\} \in \mathcal{I}^\star$.

- Show $w(\mathcal{I}^\star) = w(\mathcal{I}_x^\star) + w(x)$.

Correctness Proof

Lemma 3 - Optimal substructure property

Let $M = (S, \mathcal{I}, w)$ be a weighted matroid and suppose $x \in S$ satisfies:

- x is contained by some element of $\mathcal{I}^\star$.

Then, for each $B \in \mathcal{I}^\star_x$, $B \cup \{x\} \in \mathcal{I}^\star$.

- Show $w(\mathcal{I}^\star) = w(\mathcal{I}^\star_x) + w(x)$.
- So, for $B \in \mathcal{I}^\star_x$,
  $w(B \cup \{x\}) = w(B) + w(x) = w(\mathcal{I}^\star_x) + w(x) = w(\mathcal{I}^\star)$.

Correctness Proof

Lemma 3 - Optimal substructure property

Let $M = (S, \mathcal{I}, w)$ be a weighted matroid and suppose $x \in S$ satisfies:

- x is contained by some element of $\mathcal{I}^\star$.

Then, for each $B \in \mathcal{I}_x^\star$, $B \cup \{x\} \in \mathcal{I}^\star$.

- Show $w(\mathcal{I}^\star) = w(\mathcal{I}_x^\star) + w(x)$.
- So, for $B \in \mathcal{I}_x^\star$,
  $w(B \cup \{x\}) = w(B) + w(x) = w(\mathcal{I}_x^\star) + w(x) = w(\mathcal{I}^\star)$.

Combining those things all together, we can show the algorithm works correctly.

Generalized algorithm

We have shown that below algorithm works correctly for every (weighted) matroid.

Generalized algorithm

We have shown that below algorithm works correctly for every (weighted) matroid.

Generalized greedy algorithm

Greedy($M = (S, \mathcal{I}, w)$) // `Goal: to find some` $A \in \mathcal{I}^\star$

- $A := \emptyset$
- Sort $S = \{x_1, x_2, \cdots, x_n\}$ so that $w(x_1) \geq w(x_2) \geq \cdots \geq w(x_n)$.
- for ($i := 1$ to n) do
  - if $A \cup \{x_i\} \in \mathcal{I}$ //$(\star)$
    - $A := A \cup \{x_i\}$
- return A // $A \in \mathcal{I}^\star$

Generalized algorithm

We have shown that below algorithm works correctly for every (weighted) matroid.

Generalized greedy algorithm

Greedy($M = (S, \mathcal{I}, w)$) // `Goal: to find some` $A \in \mathcal{I}^\star$

- $A := \emptyset$
- Sort $S = \{x_1, x_2, \cdots, x_n\}$ so that $w(x_1) \geq w(x_2) \geq \cdots \geq w(x_n)$.
- for ($i := 1$ to n) do
  - if $A \cup \{x_i\} \in \mathcal{I}$ //$(\star)$
    - $A := A \cup \{x_i\}$
- return A // $A \in \mathcal{I}^\star$

- Furthermore, it is efficient since it only takes $\mathcal{O}(n \log n + nf(n))$ time where $f(n)$ is running time of $(\star)$.

Generalized algorithm

We have shown that below algorithm works correctly for every (weighted) matroid.

Generalized greedy algorithm

Greedy($M = (S, \mathcal{I}, w)$) // `Goal: to find some` $A \in \mathcal{I}^\star$

- $A := \emptyset$
- Sort $S = \{x_1, x_2, \cdots, x_n\}$ so that $w(x_1) \geq w(x_2) \geq \cdots \geq w(x_n)$.
- for ($i := 1$ to n) do
  - if $A \cup \{x_i\} \in \mathcal{I}$
    - $A := A \cup \{x_i\}$
- return A // $A \in \mathcal{I}^\star$

- Q. Can we use matroid to formulate problems that *generic greedy algorithm* works correctly?

Defining matroid from greedy algorithms

Theorem

Suppose that we are given a finite set S and $\mathcal{I} \subseteq 2^S$ satisfying (I1) $\emptyset \in \mathcal{I}$ and (I2) Subset of independent set is independent. If (G) holds, $M = (S, \mathcal{I})$ is a matroid.

(G) Generic greedy algorithm works correctly for every $w : S \to \mathbb{R}$.

Defining matroid from greedy algorithms

Theorem

Suppose that we are given a finite set S and $\mathcal{I} \subseteq 2^S$ satisfying (I1) and (I2). If (G) holds, $M = (S, \mathcal{I})$ is a matroid.

(G) Generic greedy algorithm works correctly for every $w : S \to \mathbb{R}$.

- Suppose that (I3) does not holds for some $X, Y \in \mathcal{I}$.

Defining matroid from greedy algorithms

Theorem

Suppose that we are given a finite set S and $\mathcal{I} \subseteq 2^S$ satisfying (I1) and (I2). If (G) holds, $M = (S, \mathcal{I})$ is a matroid.

(G) Generic greedy algorithm works correctly for every $w : S \to \mathbb{R}$.

- Suppose that (I3) does not holds for some $X, Y \in \mathcal{I}$.
  - That is, $X \cup \{e\} \notin \mathcal{I}$ for every $e \in Y - X$.

Defining matroid from greedy algorithms

Theorem

Suppose that we are given a finite set S and $\mathcal{I} \subseteq 2^S$ satisfying (I1) and (I2). If (G) holds, $M = (S, \mathcal{I})$ is a matroid.

(G) Generic greedy algorithm works correctly for every $w : S \to \mathbb{R}$.

- Suppose that (I3) does not holds for some $X, Y \in \mathcal{I}$.
- For sufficient $0 < \epsilon < (|Y| - |X|)/|X - Y|$, define

$$w(x) = \begin{cases} 1 + \epsilon & \text{if } x \in X \\ 1 & \text{if } x \in Y - X \\ 0 & \text{otherwise} \end{cases}.$$

Defining matroid from greedy algorithms

Theorem

Suppose that we are given a finite set S and $\mathcal{I} \subseteq 2^S$ satisfying (I1) and (I2). If (G) holds, $M = (S, \mathcal{I})$ is a matroid.

(G) Generic greedy algorithm works correctly for every $w : S \to \mathbb{R}$.

- Suppose that (I3) does not holds for some $X, Y \in \mathcal{I}$.
- For sufficient $0 < \epsilon < (|Y| - |X|)/|X - Y|$, define

$$w(x) = \begin{cases} 1+\epsilon & \text{if } x \in X \\ 1 & \text{if } x \in Y - X \\ 0 & \text{otherwise} \end{cases}.$$

- Then, X', result of greedy algorithm contains X so that $|Y| + \epsilon|X \cap Y| = w(Y) \leq w(X') = (1+\epsilon)|X|$, which implies $\epsilon|X - Y| \geq |Y| - |X|$.

Results

Interestingly, although matroid is the concept that defines (linear) *independence* combinatorially, we checked that it also characterizes generic greedy algorithm.

Generalized greedy algorithm

Greedy($M = (S, \mathcal{I}, w)$) // `Goal: to find some` $A \in \mathcal{I}^\star$

- $A := \emptyset$
- Sort $S = \{x_1, x_2, \cdots, x_n\}$ so that $w(x_1) \geq w(x_2) \geq \cdots \geq w(x_n)$.
- for ($i := 1$ to n) do
  - if $A \cup \{x_i\} \in \mathcal{I}$
    - $A := A \cup \{x_i\}$
- return A // $A \in \mathcal{I}^\star$

References

- James Oxley, *Matroid theory*, second ed., Oxford, 2011
- James Oxley, *What is a matroid?*, `https://www.math.lsu.edu/~oxley/survey4.pdf`