CSE 632: Analysis of Algorithms II: Combinatorial Optimization and Linear Programming (Fall 2020)
Matroid and Submodular Optimization

Lecturer: Shi Li
Department of Computer Science and Engineering
University at Buffalo
1 Motivation: Maximum Weight Spanning Tree

2 Introduction to Matroid
   - Analysis of Greedy Algorithm

3 Examples of Matroids
Maximum Weight Spanning Tree

**Input:** connected graph with edge weights (weights = profits)

**Output:** the maximum weight spanning tree (or a sub-graph without cycles)
Maximum Weight Spanning Tree

**Input:** connected graph with edge weights (weights = profits)

**Output:** the maximum weight spanning tree (or a sub-graph without cycles)
Kruskal’s Algorithm for Maximum Weight Spanning Tree

1: $F \leftarrow \emptyset$
2: while $F$ is not a spanning tree do
3: find the most profitable edge $e \in E \setminus F$ such that $F \cup \{e\}$ does not contain a cycle
4: $F \leftarrow F \cup \{e\}$
Kruskal’s Algorithm for Maximum Weight Spanning Tree

1: $F \leftarrow \emptyset$
2: while $F$ is not a spanning tree do
3: find the most profitable edge $e \in E \setminus F$ such that $F \cup \{e\}$ does not contain a cycle
4: $F \leftarrow F \cup \{e\}$
Kruskal’s Algorithm for Maximum Weight Spanning Tree

1: \( F \leftarrow \emptyset \)
2: \textbf{while} \( F \) is not a spanning tree \textbf{do}
3: \hspace{1em} find the most profitable edge \( e \in E \setminus F \) such that \( F \cup \{e\} \) does not contain a cycle
4: \hspace{1em} \( F \leftarrow F \cup \{e\} \)

![Graph](image-url)
Kruskal’s Algorithm for Maximum Weight Spanning Tree

1: \( F \leftarrow \emptyset \)
2: **while** \( F \) is not a spanning tree **do**
3: \( \) find the most profitable edge \( e \in E \setminus F \) such that \( F \cup \{e\} \) does not contain a cycle
4: \( F \leftarrow F \cup \{e\} \)

![Graph with labeled edges and nodes](image-url)
Kruskal’s Algorithm for Maximum Weight Spanning Tree

1: \( F \leftarrow \emptyset \)
2: \textbf{while} \( F \) is not a spanning tree \textbf{do} \\
3: \hspace{1em} find the most profitable edge \( e \in E \setminus F \) such that \\
\hspace{2em} \( F \cup \{e\} \) does not contain a cycle \\
4: \hspace{1em} \( F \leftarrow F \cup \{e\} \)
Kruskal’s Algorithm for Maximum Weight Spanning Tree

1:  \( F \leftarrow \emptyset \)
2:  while \( F \) is not a spanning tree do
3:      find the most profitable edge \( e \in E \setminus F \) such that \( F \cup \{e\} \) does not contain a cycle
4:      \( F \leftarrow F \cup \{e\} \)
Kruskal’s Algorithm for Maximum Weight Spanning Tree

1: \( F \leftarrow \emptyset \)
2: \textbf{while} \( F \) is not a spanning tree \textbf{do}
3: \hspace{1em} find the most profitable edge \( e \in E \setminus F \) such that \( F \cup \{e\} \) does not contain a cycle
4: \hspace{1em} \( F \leftarrow F \cup \{e\} \)
Kruskal’s Algorithm for Maximum Weight Spanning Tree

1: $F \leftarrow \emptyset$
2: while $F$ is not a spanning tree do
3: find the most profitable edge $e \in E \setminus F$ such that $F \cup \{e\}$ does not contain a cycle
4: $F \leftarrow F \cup \{e\}$
Kruskal’s Algorithm for Maximum Weight Spanning Tree

1: $F \leftarrow \emptyset$
2: while $F$ is not a spanning tree do
3: find the most profitable edge $e \in E \setminus F$ such that $F \cup \{e\}$ does not contain a cycle
4: $F \leftarrow F \cup \{e\}$
Proof of Correctness of Kruskal’s Algorithm

**Def.** We say a set $F \subseteq E$ of edges is a **failure** if there is no optimum solution $S$ such that $F \subseteq S$. That is, $F$ is a failure if it is not a subset of any optimum solution.
Proof of Correctness of Kruskal’s Algorithm

Def. We say a set $F \subseteq E$ of edges is a failure if there is no optimum solution $S$ such that $F \subseteq S$. That is, $F$ is a failure if it is not a subset of any optimum solution.

Obs. If $F$ becomes a failure during the algorithm, then the algorithm does not give the optimum solution.
**Def.** We say a set $F \subseteq E$ of edges is a **failure** if there is no optimum solution $S$ such that $F \subseteq S$. That is, $F$ is a failure if it is not a subset of any optimum solution.

**Obs.** If $F$ becomes a failure during the algorithm, then algorithm does not give the optimum solution.

**Obs.** If algorithm does not give an optimum solution, then there is a first iteration in which the algorithm constructed a failure $F$. 
Def. We say a set $F \subseteq E$ of edges is a failure if there is no optimum solution $S$ such that $F \subseteq S$. That is, $F$ is a failure if it is not a subset of any optimum solution.
Def. We say a set $F \subseteq E$ of edges is a failure if there is no optimum solution $S$ such that $F \subseteq S$. That is, $F$ is a failure if it is not a subset of any optimum solution.

- Assume towards contradiction, algorithm does not produce optimum solution
Def. We say a set $F \subseteq E$ of edges is a failure if there is no optimum solution $S$ such that $F \subseteq S$. That is, $F$ is a failure if it is not a subset of any optimum solution.

- assume towards contradiction, algorithm does not produce optimum solution
- consider first iteration $i^*$ which constructed a failure
Proof of Correctness of Kruskal’s Algorithm

**Def.** We say a set $F \subseteq E$ of edges is a **failure** if there is no optimum solution $S$ such that $F \subseteq S$. That is, $F$ is a failure if it is not a subset of any optimum solution.

- Assume towards contradiction, algorithm does not produce optimum solution.
- Consider first iteration $i^*$ which constructed a failure.
- $F$: chosen edges before iteration $i^*$ (So, $F$ is not a failure.)
Proof of Correctness of Kruskal’s Algorithm

Def. We say a set $F \subseteq E$ of edges is a failure if there is no optimum solution $S$ such that $F \subseteq S$. That is, $F$ is a failure if it is not a subset of any optimum solution.

- $e^*$: the edge algorithm chooses in iteration $i^*$
**Def.** We say a set $F \subseteq E$ of edges is a **failure** if there is no optimum solution $S$ such that $F \subseteq S$. That is, $F$ is a failure if it is not a subset of any optimum solution.

- $e^*$: the edge algorithm chooses in iteration $i^*$
- red edges: the optimum solution $S$ containing $F$.
Proof of Correctness of Kruskal’s Algorithm

**Def.** We say a set $F \subseteq E$ of edges is a failure if there is no optimum solution $S$ such that $F \subseteq S$. That is, $F$ is a failure if it is not a subset of any optimum solution.

- $e^*$: the edge algorithm chooses in iteration $i^*$
- red edges: the optimum solution $S$ containing $F$.
- $S \cup \{e^*\}$ contains a cycle

![Graph Diagram](image-url)
Def. We say a set $F \subseteq E$ of edges is a failure if there is no optimum solution $S$ such that $F \subseteq S$. That is, $F$ is a failure if it is not a subset of any optimum solution.

- $e'$: another edge on cycle
Def. We say a set $F \subseteq E$ of edges is a **failure** if there is no optimum solution $S$ such that $F \subseteq S$. That is, $F$ is a failure if it is not a subset of any optimum solution.

- $e'$: another edge on cycle
- swapping $e^*$ and $e'$ gives another optimum solution
Def. We say a set $F \subseteq E$ of edges is a failure if there is no optimum solution $S$ such that $F \subseteq S$. That is, $F$ is a failure if it is not a subset of any optimum solution.

- $e'$: another edge on cycle
- swapping $e^*$ and $e'$ gives another optimum solution
- contradiction with that $F \cup \{e^*\}$ is a failure
A Generic Problem

**Input:** \( E \): ground set, non-negative weights \( w \) on \( E \)
\( \mathcal{I} \): an implicitly given downward-closed family of subsets of \( E \)

**Output:** a maximum weight subset \( F \in \mathcal{I} \)
A Generic Problem

**Input:** \( E \): ground set, non-negative weights \( w \) on \( E \)
\[ \mathcal{I} \]: an implicitly given downward-closed family of subsets of \( E \)

**Output:** a maximum weight subset \( F \in \mathcal{I} \)

**Def.** \( \mathcal{I} \) is downward-closed if for every \( S \in \mathcal{I} \) and \( S' \subseteq S \), we have \( S' \in \mathcal{I} \).
A Generic Problem

**Input:** \( E \): ground set, non-negative weights \( w \) on \( E \)
\( \mathcal{I} \): an implicitly given downward-closed family of subsets of \( E \)

**Output:** a maximum weight subset \( F \in \mathcal{I} \)

**Def.** \( \mathcal{I} \) is downward-closed if for every \( S \in \mathcal{I} \) and \( S' \subseteq S \), we have \( S' \in \mathcal{I} \).

- \( \mathcal{I} \): family of valid solutions.
A Generic Problem

**Input:** \( E \): ground set, non-negative weights \( w \) on \( E \)
\( \mathcal{I} \): an implicitly given **downward-closed** family of subsets of \( E \)

**Output:** a maximum weight subset \( F \in \mathcal{I} \)

**Def.** \( \mathcal{I} \) is downward-closed if for every \( S \in \mathcal{I} \) and \( S' \subseteq S \), we have \( S' \in \mathcal{I} \).

- \( \mathcal{I} \): family of valid solutions.
- \( \mathcal{I} \) is downward-closed: a subset of a valid solution is also valid.
A Generic Problem

**Input:**  
\( E \): ground set, non-negative weights \( w \) on \( E \)  
\( \mathcal{I} \): an implicitly given downward-closed family of subsets of \( E \)

**Output:** a maximum weight subset \( F \in \mathcal{I} \)

**Def.** \( \mathcal{I} \) is downward-closed if for every \( S \in \mathcal{I} \) and \( S' \subseteq S \), we have \( S' \in \mathcal{I} \).

- \( \mathcal{I} \): family of valid solutions.
- \( \mathcal{I} \) is downward-closed: a subset of a valid solution is also valid.
- typical assumption for maximization problems.
A Generic Problem

**Input:** $E$: ground set, non-negative weights $w$ on $E$

$I$: an **implicitly** given downward-closed family of subsets of $E$

**Output:** a maximum weight subset $F \in I$

**Def.** $I$ is downward-closed if for every $S \in I$ and $S' \subseteq S$, we have $S' \in I$.

- $I$: family of valid solutions.
- $I$ is downward-closed: a subset of a valid solution is also valid.
- Typical assumption for maximization problems.

- **implicitly-given:** we do not list all the sets in $I$ in the input. Instead, there is an efficient oracle which, given $S \subseteq E$, decides if $S \in I$. 
A Generic Problem

**Input:** $E$: ground set, non-negative weights $w$ on $E$

$I$: an implicitly given downward-closed family of subsets of $E$

**Output:** a maximum weight subset $F \in I$

Maximum Weight Spanning Tree
A Generic Problem

**Input:** $E$: ground set, non-negative weights $w$ on $E$

$I$: an implicitly given downward-closed family of subsets of $E$

**Output:** a maximum weight subset $F \in I$

Maximum Weight Spanning Tree

- $E$: set of edges in a graph $G = (V, E)$
A Generic Problem

**Input:** $E$: ground set, non-negative weights $w$ on $E$

$I$: an implicitly given downward-closed family of subsets of $E$

**Output:** a maximum weight subset $F \in I$

---

Maximum Weight Spanning Tree

- $E$: set of edges in a graph $G = (V, E)$
- $I$: family of forests in $G$. That is, a set $S \subseteq E$ is in $S$ if and only if $S$ does not contain a cycle of $G$
A Generic Problem

**Input:** $E$: ground set, non-negative weights $w$ on $E$

$I$: an implicitly given downward-closed family of subsets of $E$

**Output:** a maximum weight subset $F \in I$

Maximum Weight Spanning Tree

- $E$: set of edges in a graph $G = (V, E)$
- $I$: family of forests in $G$. That is, a set $S \subseteq E$ is in $I$ if and only if $S$ does not contain a cycle of $G$
- $I$ is downward-closed: if $S$ does not contain a cycle, then removing edges from $S$ can not create a cycle
A Generic Problem

**Input:** \( E \): ground set, non-negative weights \( w \) on \( E \)

\( \mathcal{I} \): an implicitly given downward-closed family of subsets of \( E \)

**Output:** a maximum weight subset \( F \in \mathcal{I} \)

Maximum Weight Spanning Tree

- \( E \): set of edges in a graph \( G = (V, E) \)

- \( \mathcal{I} \): family of **forests** in \( G \). That is, a set \( S \subseteq E \) is in \( \mathcal{I} \) if and only if \( S \) does not contain a cycle of \( G \)

- \( \mathcal{I} \) is downward-closed: if \( S \) does not contain a cycle, then removing edges from \( S \) can not create a cycle

- we do not list all forests; instead, there is an efficient oracle to check if \( S \) is a forest or not
A Generic Problem

**Input:** $E$: ground set, non-negative weights $w$ on $E$

$I$: an implicitly given downward-closed family of subsets of $E$

**Output:** a maximum weight subset $F \in I$

A Natural Generic Greedy Algorithm

1. $F \leftarrow \emptyset$
2. **while** $\exists e \in E \setminus F$ such that $F \cup \{e\} \in I$ **do**
3. find the $e^* \in E \setminus F$, $F \cup \{e^*\} \in I$ with maximum $w_{e^*}$
4. $F \leftarrow F \cup \{e^*\}$

- For maximum-weight spanning tree, the generic algorithm becomes Kruskal’s algorithm.
Q: When does the greedy algorithm give an optimum solution?

- when the problem is maximum-weight spanning tree, algorithm is optimum
- there are cases where algorithm is not optimum

Example:

\[ E = \{a, b, c\}, w_a = 10, w_b = 9, w_c = 9, I = \{\emptyset, \{a\}, \{b\}, \{c\}, \{b, c\}\} \]

Greedy algorithm will choose \(a\), which has weight 10.

Optimum solution \(\{b, c\}\) has weight 18.
Q: When does the greedy algorithm give an optimum solution?

- When the problem is maximum-weight spanning tree, algorithm is optimum
- There are cases where algorithm is not optimum

Example:

\[ E = \{a, b, c\}, w_a = 10, w_b = 9, w_c = 9, \]
\[ I = \{\emptyset, \{a\}, \{b\}, \{c\}, \{b, c\}\} \]
Q: When does the greedy algorithm give an optimum solution?

- When the problem is maximum-weight spanning tree, the algorithm is optimum.
- There are cases where the algorithm is not optimum.

Example:

- \( E = \{a, b, c\}, w_a = 10, w_b = 9, w_c = 9 \),
- \( I = \{\emptyset, \{a\}, \{b\}, \{c\}, \{b, c\}\} \),
- Greedy algorithm will choose \( a \), which has weight 10.
Q: When does the greedy algorithm gives an optimum solution?

- when the problem is maximum-weight spanning tree, algorithm is optimum
- there are cases where algorithm is not optimum

Example:

- \( E = \{a, b, c\}, w_a = 10, w_b = 9, w_c = 9, \)
- \( I = \{\emptyset, \{a\}, \{b\}, \{c\}, \{b, c\}\} \)
- greedy algorithm will choose \( a \), which has weight 10
- optimum solution \( \{b, c\} \) has weight 18.
Q: When does the greedy algorithm give an optimum solution?
Q: When does the greedy algorithm gives an optimum solution?

A: when the valid solutions form a matroid.
1 Motivation: Maximum Weight Spanning Tree

2 Introduction to Matroid
   - Analysis of Greedy Algorithm

3 Examples of Matroids
Def. A matroid $M$ is a pair $(E, \mathcal{I})$, where $E$ is a finite set, and $\mathcal{I}$ is a family of subsets of $E$ such that

- $\emptyset \in \mathcal{I}$.
- $\mathcal{I}$ is downward-closed: if $A \in \mathcal{I}$ and $A' \subseteq A$, then $A' \in \mathcal{I}$.
- (exchange property) If $A, B \in \mathcal{I}$ and $|A| > |B|$, then there exists $x \in A \setminus B$ such that $B \cup \{x\} \in \mathcal{I}$.

$E$ is called the ground set of $M$. Every set in $\mathcal{I}$ is called an independent set of $M$. So, $\mathcal{I}$ is the set of independent sets of $M$. 
Def. A matroid $M$ is a pair $(E, \mathcal{I})$, where $E$ is a finite set, and $\mathcal{I}$ is a family of subsets of $E$ such that

- $\emptyset \in \mathcal{I}$.
Def. A matroid $M$ is a pair $(E, \mathcal{I})$, where $E$ is a finite set, and $\mathcal{I}$ is a family of subsets of $E$ such that

- $\emptyset \in \mathcal{I}$.
- $\mathcal{I}$ is downward-closed: if $A \in \mathcal{I}$ and $A' \subseteq A$, then $A' \in \mathcal{I}$.

$E$ is called the ground set of $M$. Every set in $\mathcal{I}$ is called an independent set of $M$. So, $\mathcal{I}$ is the set of independent sets of $M$. 

(exchange property) If $A, B \in \mathcal{I}$ and $|A| > |B|$, then there exists $x \in A \setminus B$ such that $B \cup \{x\} \in \mathcal{I}$. 


Def. A matroid $M$ is a pair $(E, I)$, where $E$ is a finite set, and $I$ is a family of subsets of $E$ such that

- $\emptyset \in I$.
- $I$ is downward-closed: if $A \in I$ and $A' \subseteq A$, then $A' \in I$.

- $E$ is called the ground set of $M$. 

So, $I$ is the set of independent sets of $M$. 

Def. A matroid $M$ is a pair $(E, \mathcal{I})$, where $E$ is a finite set, and $\mathcal{I}$ is a family of subsets of $E$ such that

- $\emptyset \in \mathcal{I}$.
- $\mathcal{I}$ is downward-closed: if $A \in \mathcal{I}$ and $A' \subseteq A$, then $A' \in \mathcal{I}$.

- $E$ is called the ground set of $M$.
- Every set in $\mathcal{I}$ is called an independent set of $M$.
- So, $\mathcal{I}$ is the set of independent sets of $M$.
**Def.** A matroid $M$ is a pair $(E, \mathcal{I})$, where $E$ is a finite set, and $\mathcal{I}$ is a family of subsets of $E$ such that

- $\emptyset \in \mathcal{I}$.
- $\mathcal{I}$ is downward-closed: if $A \in \mathcal{I}$ and $A' \subseteq A$, then $A' \in \mathcal{I}$.
- (exchange property) If $A, B \in \mathcal{I}$ and $|A| > |B|$, then there exists $x \in A \setminus B$ such that $B \cup \{x\} \in \mathcal{I}$.

- $E$ is called the **ground set** of $M$.
- Every set in $\mathcal{I}$ is called an **independent set** of $M$.
- So, $\mathcal{I}$ is the set of independent sets of $M$. 


Def. A matroid $M$ is a pair $(E, \mathcal{I})$, where $E$ is a finite set, and $\mathcal{I}$ is a family of subsets of $E$ such that

- $\emptyset \in \mathcal{I}$.
- $\mathcal{I}$ is downward-closed: if $A \in \mathcal{I}$ and $A' \subseteq A$, then $A' \in \mathcal{I}$.
- (exchange property) If $A, B \in \mathcal{I}$ and $|A| > |B|$, then there exists $x \in A \setminus B$ such that $B \cup \{x\} \in \mathcal{I}$. 
Def. A matroid $M$ is a pair $(E, \mathcal{I})$, where $E$ is a finite set, and $\mathcal{I}$ is a family of subsets of $E$ such that

- $\emptyset \in \mathcal{I}$.
- $\mathcal{I}$ is downward-closed: if $A \in \mathcal{I}$ and $A' \subseteq A$, then $A' \in \mathcal{I}$.
- (exchange property) If $A, B \in \mathcal{I}$ and $|A| > |B|$, then there exists $x \in A \setminus B$ such that $B \cup \{x\} \in \mathcal{I}$.
Def. A matroid $M$ is a pair $(E, \mathcal{I})$, where $E$ is a finite set, and $\mathcal{I}$ is a family of subsets of $E$ such that

- $\emptyset \in \mathcal{I}$.
- $\mathcal{I}$ is downward-closed: if $A \in \mathcal{I}$ and $A' \subseteq A$, then $A' \in \mathcal{I}$.
- (exchange property) If $A, B \in \mathcal{I}$ and $|A| > |B|$, then there exists $x \in A \setminus B$ such that $B \cup \{x\} \in \mathcal{I}$. 

\begin{figure}[h]
\centering
\begin{tikzpicture}
  \node (A) at (0,0) {\small $A$};
  \node (B) at (1.5,0) {\small $B$};
  \node (x) at (0.75,-1.5) {\small $x$};
  \draw (A) circle (1cm);
  \draw (B) circle (1cm);
  \fill (0,0) circle (2pt);
  \fill (1.5,0) circle (2pt);
  \fill (0.75,-1.5) circle (2pt);
\end{tikzpicture}
\end{figure}
Def. A matroid $M$ is a pair $(E, \mathcal{I})$, where $E$ is a finite set, and $\mathcal{I}$ is a family of subsets of $E$ such that

- $\emptyset \in \mathcal{I}$.
- $\mathcal{I}$ is downward-closed: if $A \in \mathcal{I}$ and $A' \subseteq A$, then $A' \in \mathcal{I}$.
- **(exchange property)** If $A, B \in \mathcal{I}$ and $|A| > |B|$, then there exists $x \in A \setminus B$ such that $B \cup \{x\} \in \mathcal{I}$. 

![Diagram showing the exchange property](https://via.placeholder.com/150)
**Def.** Let $G = (V, E)$ be a connected undirected graph. Let $\mathcal{I}$ be the family of subsets of edges that form a forest in $G$. Then, $(E, \mathcal{I})$ is called a **graphic matroid**.

\[ E = \{e_1, e_2, e_3, e_4, e_5\} \]

\[ \mathcal{I} = \{\emptyset, \{e_1\}, \{e_2\}, \{e_3\}, \{e_4\}, \{e_5\}, \]
\[ \{e_1, e_2\}, \{e_1, e_3\}, \{e_1, e_4\}, \{e_1, e_5\}, \{e_2, e_3\}, \]
\[ \{e_2, e_4\}, \{e_2, e_5\}, \{e_3, e_4\}, \{e_3, e_5\}, \{e_4, e_5\}, \]
\[ \{e_1, e_2, e_4\}, \{e_1, e_2, e_5\}, \{e_1, e_3, e_4\}, \{e_1, e_3, e_5\}, \]
\[ \{e_1, e_4, e_5\}, \{e_2, e_3, e_4\}, \{e_2, e_3, e_5\}\]
A Graphic Matroid is Indeed a Matroid

- $G = (V, E)$
- $\mathcal{I}$ is the family of forests in $G$

3 Properties to Check

- $\emptyset \in \mathcal{I}$.
- $\mathcal{I}$ is downward-closed: if $A \in \mathcal{I}$ and $A' \subseteq A$, then $A' \in \mathcal{I}$.
- (exchange property) If $A, B \in \mathcal{I}$ and $|A| > |B|$, then there exists $x \in A \setminus B$ such that $B \cup \{x\} \in \mathcal{I}$. 

First two properties are trivial.

Forest $B$ has $n - |B|$ connected components

Forest $A$ has $n - |A| < n - |B|$ connected components

some $e \in A$ must connect two different components of $B$

$e \not\in B$ and $B \cup \{e\}$ is also a forest
A Graphic Matroid is Indeed a Matroid

- \( G = (V, E) \)
- \( \mathcal{I} \) is the family of forests in \( G \)

### 3 Properties to Check

- \( \emptyset \in \mathcal{I} \).
- \( \mathcal{I} \) is downward-closed: if \( A \in \mathcal{I} \) and \( A' \subseteq A \), then \( A' \in \mathcal{I} \).
- **(exchange property)** If \( A, B \in \mathcal{I} \) and \( |A| > |B| \), then there exists \( x \in A \setminus B \) such that \( B \cup \{x\} \in \mathcal{I} \).

- First two properties are trivial.
A Graphic Matroid is Indeed a Matroid

- $G = (V, E)$
- $\mathcal{I}$ is the family of forests in $G$

### 3 Properties to Check

- $\emptyset \in \mathcal{I}$.
- $\mathcal{I}$ is downward-closed: if $A \in \mathcal{I}$ and $A' \subseteq A$, then $A' \in \mathcal{I}$.
- **(exchange property)** If $A, B \in \mathcal{I}$ and $|A| > |B|$, then there exists $x \in A \setminus B$ such that $B \cup \{x\} \in \mathcal{I}$.

- First two properties are trivial.
A Graphic Matroid is Indeed a Matroid

- $G = (V, E)$
- $\mathcal{I}$ is the family of forests in $G$

3 Properties to Check

- $\emptyset \in \mathcal{I}$.
- $\mathcal{I}$ is downward-closed: if $A \in \mathcal{I}$ and $A' \subseteq A$, then $A' \in \mathcal{I}$.
- (exchange property) If $A, B \in \mathcal{I}$ and $|A| > |B|$, then there exists $x \in A \setminus B$ such that $B \cup \{x\} \in \mathcal{I}$.

- First two properties are trivial.
- Forest $B$ has $n - |B|$ connected components
A Graphic Matroid is Indeed a Matroid

- $G = (V, E)$
- $\mathcal{I}$ is the family of forests in $G$

### 3 Properties to Check

- $\emptyset \in \mathcal{I}$.
- $\mathcal{I}$ is downward-closed: if $A \in \mathcal{I}$ and $A' \subseteq A$, then $A' \in \mathcal{I}$.
- **(exchange property)** If $A, B \in \mathcal{I}$ and $|A| > |B|$, then there exists $x \in A \setminus B$ such that $B \cup \{x\} \in \mathcal{I}$.

- First two properties are trivial.
- Forest $B$ has $n - |B|$ connected components
- Forest $A$ has $n - |A| < n - |B|$ connected components
A Graphic Matroid is Indeed a Matroid

- $G = (V, E)$
- $\mathcal{I}$ is the family of forests in $G$

3 Properties to Check

- $\emptyset \in \mathcal{I}$.
- $\mathcal{I}$ is downward-closed: if $A \in \mathcal{I}$ and $A' \subseteq A$, then $A' \in \mathcal{I}$.
- (exchange property) If $A, B \in \mathcal{I}$ and $|A| > |B|$, then there exists $x \in A \setminus B$ such that $B \cup \{x\} \in \mathcal{I}$.

- First two properties are trivial.
- Forest $B$ has $n - |B|$ connected components
- Forest $A$ has $n - |A| < n - |B|$ connected components
- some $e \in A$ must connect two different components of $B$
A Graphic Matroid is Indeed a Matroid

- $G = (V, E)$
- $\mathcal{I}$ is the family of forests in $G$

### 3 Properties to Check

- $\emptyset \in \mathcal{I}$.
- $\mathcal{I}$ is downward-closed: if $A \in \mathcal{I}$ and $A' \subseteq A$, then $A' \in \mathcal{I}$.
- **(exchange property)** If $A, B \in \mathcal{I}$ and $|A| > |B|$, then there exists $x \in A \setminus B$ such that $B \cup \{x\} \in \mathcal{I}$.

- First two properties are trivial.
- Forest $B$ has $n - |B|$ connected components
- Forest $A$ has $n - |A| < n - |B|$ connected components
- some $e \in A$ must connect two different components of $B$
- $e \notin B$ and $B \cup \{e\}$ is also a forest
Now go back to the counter example.

**Example:**

- $E = \{a, b, c\}$, $w_a = 10$, $w_b = 9$, $w_c = 9$,
- $\mathcal{I} = \{\emptyset, \{a\}, \{b\}, \{c\}, \{b, c\}\}$
- greedy algorithm will choose $a$, which has weight 10
- optimum solution $\{b, c\}$ has weight 18.

$(E, \mathcal{I})$ is not a matroid since it does not satisfy the exchange property:
- $\{a\} \in \mathcal{I}$, $\{b, c\} \in \mathcal{I}$, but $\{a, b\} \notin \mathcal{I}$, $\{a, c\} \notin \mathcal{I}$. 
Outline

1. Motivation: Maximum Weight Spanning Tree

2. Introduction to Matroid
   - Analysis of Greedy Algorithm

3. Examples of Matroids
Maximum Weighted Independent Set of a Matroid

**Input:** A matroid \((E, \mathcal{I})\), weights \(w \in \mathbb{R}^E_{\geq 0}\)

**Output:** A set \(S \in \mathcal{I}\) (i.e, an independent set of the matroid) with the maximum \(\sum_{e \in S} w_e\)

### Greedy Algorithm

1. \(F \leftarrow \emptyset\)
2. **while** \(\exists e \in E \setminus F\) such that \(F \cup \{e\} \in \mathcal{I}\) **do**
3. \(\text{find the } e^* \in E \setminus F, F \cup \{e^*\} \in \mathcal{I}\) with maximum \(w_{e^*}\)
4. \(F \leftarrow F \cup \{e^*\}\)

### Theorem
The greedy algorithm gives an optimum solution to the maximum weight independent set problem in a matroid.
exchange property: If $A, B \in \mathcal{I}$ and $|A| > |B|$, then there exists $x \in A \setminus B$ such that $B \cup \{x\} \in \mathcal{I}$.
exchange property: If $A, B \in \mathcal{I}$ and $|A| > |B|$, then there exists $x \in A \setminus B$ such that $B \cup \{x\} \in \mathcal{I}$.

Lemma Let $F \subsetneq S \in \mathcal{I}$, $e^* \notin S$ and $F \cup \{e^*\} \in \mathcal{I}$. Then there exists some $e' \in S \setminus F$ such that $S \setminus \{e'\} \cup \{e^*\} \in \mathcal{I}$.
**Lemma** Let $F \subsetneq S \in \mathcal{I}$, $e^* \notin S$ and $F \cup \{e^*\} \in \mathcal{I}$. Then there exists some $e' \in S \setminus F$ such that $S \setminus \{e'\} \cup \{e^*\} \in \mathcal{I}$.

- **exchange property:** If $A, B \in \mathcal{I}$ and $|A| > |B|$, then there exists $x \in A \setminus B$ such that $B \cup \{x\} \in \mathcal{I}$. 

\[
\begin{array}{c}
F \cup \{e^*\} \in \mathcal{I} \\
S \in \mathcal{I}
\end{array}
\]
exchange property: If $A, B \in \mathcal{I}$ and $|A| > |B|$, then there exists $x \in A \setminus B$ such that $B \cup \{x\} \in \mathcal{I}$.

**Lemma** Let $F \subsetneq S \in \mathcal{I}$, $e^* \notin S$ and $F \cup \{e^*\} \in \mathcal{I}$. Then there exists some $e' \in S \setminus F$ such that $S \setminus \{e'\} \cup \{e^*\} \in \mathcal{I}$. 

![Diagram](https://via.placeholder.com/150)
exchange property: If $A, B \in \mathcal{I}$ and $|A| > |B|$, then there exists $x \in A \setminus B$ such that $B \cup \{x\} \in \mathcal{I}$.

**Lemma** Let $F \subseteq S \in \mathcal{I}$, $e^* \notin S$ and $F \cup \{e^*\} \in \mathcal{I}$. Then there exists some $e' \in S \setminus F$ such that $S \setminus \{e'\} \cup \{e^*\} \in \mathcal{I}$.
• exchange property: If $A, B \in \mathcal{I}$ and $|A| > |B|$, then there exists $x \in A \setminus B$ such that $B \cup \{x\} \in \mathcal{I}$.

**Lemma** Let $F \subsetneq S \in \mathcal{I}$, $e^* \notin S$ and $F \cup \{e^*\} \in \mathcal{I}$. Then there exists some $e' \in S \setminus F$ such that $S \setminus \{e'\} \cup \{e^*\} \in \mathcal{I}$.

• indeed, lemma $\Leftrightarrow$ exchange property
- exchange property: If $A, B \in \mathcal{I}$ and $|A| > |B|$, then there exists $x \in A \setminus B$ such that $B \cup \{x\} \in \mathcal{I}$.

**Lemma** Let $F \subset S \in \mathcal{I}$, $e^* \notin S$ and $F \cup \{e^*\} \in \mathcal{I}$. Then there exists some $e' \in S \setminus F$ such that $S \setminus \{e'\} \cup \{e^*\} \in \mathcal{I}$.

- indeed, lemma $\Leftrightarrow$ exchange property
- name “exchange property” is more suitable for the property in the lemma: when two sets in $\mathcal{I}$ cross, we can “exchange” two elements to make the resulting set in $\mathcal{I}$
exchange property: If $A, B \in \mathcal{I}$ and $|A| > |B|$, then there exists $x \in A \setminus B$ such that $B \cup \{x\} \in \mathcal{I}$.

**Lemma** Let $F \subsetneq S \in \mathcal{I}$, $e^* \notin S$ and $F \cup \{e^*\} \in \mathcal{I}$. Then there exists some $e' \in S \setminus F$ such that $S \setminus \{e'\} \cup \{e^*\} \in \mathcal{I}$.
**exchange property:** If $A, B \in \mathcal{I}$ and $|A| > |B|$, then there exists $x \in A \setminus B$ such that $B \cup \{x\} \in \mathcal{I}$.

**Lemma** Let $F \subsetneq S \in \mathcal{I}$, $e^* \notin S$ and $F \cup \{e^*\} \in \mathcal{I}$. Then there exists some $e' \in S \setminus F$ such that $S \setminus \{e'\} \cup \{e^*\} \in \mathcal{I}$.

**Algorithm for Proof of Lemma Using Exchange Property**

1: $B \leftarrow F \cup \{e^*\}$

2: **while** $|B| < |S|$ **do**

3: by exchange property, there is some $x \in S \setminus B$ such that $B \cup \{x\} \in \mathcal{I}$

4: $B \leftarrow B \cup \{x\}$

In the end, $|B| = |S|$ and they differ by exactly 1 element. Thus, $B = S \setminus \{e'\} \cup \{e^*\}$ for some $e' \in S \setminus F$. 
- exchange property: If \( A, B \in \mathcal{I} \) and \( |A| > |B| \), then there exists \( x \in A \setminus B \) such that \( B \cup \{x\} \in \mathcal{I} \).

**Lemma** Let \( F \subsetneq S \in \mathcal{I} \), \( e^* \notin S \) and \( F \cup \{e^*\} \in \mathcal{I} \). Then there exists some \( e' \in S \setminus F \) such that \( S \setminus \{e'\} \cup \{e^*\} \in \mathcal{I} \).

**Algorithm for Proof of Lemma Using Exchange Property**

1. \( B \leftarrow F \cup \{e^*\} \)
2. while \( |B| < |S| \) do
3. by exchange property, there is some \( x \in S \setminus B \) such that \( B \cup \{x\} \in \mathcal{I} \)
4. \( B \leftarrow B \cup \{x\} \)

In the end, \( |B| = |S| \) and they differ by exactly 1 element.

Thus, \( B = S \setminus \{e'\} \cup \{e^*\} \) for some \( e' \in S \setminus F \).
• exchange property: If $A, B \in \mathcal{I}$ and $|A| > |B|$, then there exists $x \in A \setminus B$ such that $B \cup \{x\} \in \mathcal{I}$.

**Lemma** Let $F \subsetneq S \in \mathcal{I}$, $e^* \notin S$ and $F \cup \{e^*\} \in \mathcal{I}$. Then there exists some $e' \in S \setminus F$ such that $S \setminus \{e'\} \cup \{e^*\} \in \mathcal{I}$.

**Algorithm for Proof of Lemma Using Exchange Property**

1: $B \leftarrow F \cup \{e^*\}$
2: while $|B| < |S|$ do
3: by exchange property, there is some $x \in S \setminus B$ such that $B \cup \{x\} \in \mathcal{I}$
4: $B \leftarrow B \cup \{x\}$
- exchange property: If $A, B \in \mathcal{I}$ and $|A| > |B|$, then there exists $x \in A \setminus B$ such that $B \cup \{x\} \in \mathcal{I}$.

**Lemma**  Let $F \varsubsetneq S \in \mathcal{I}$, $e^* \notin S$ and $F \cup \{e^*\} \in \mathcal{I}$. Then there exists some $e' \in S \setminus F$ such that $S \setminus \{e'\} \cup \{e^*\} \in \mathcal{I}$.

**Algorithm for Proof of Lemma Using Exchange Property**

1. $B \leftarrow F \cup \{e^*\}$
2. **while** $|B| < |S|$ **do**
   3. by exchange property, there is some $x \in S \setminus B$ such that $B \cup \{x\} \in \mathcal{I}$
4. $B \leftarrow B \cup \{x\}$

In the end, $|B| = |S|$ and they differ by exactly 1 element. Thus, $B = S \setminus \{e'\} \cup \{e^*\}$ for some $e' \in S \setminus F$. 

![Diagram](attachment://diagram.png)
exchange property: If $A, B \in \mathcal{I}$ and $|A| > |B|$, then there exists $x \in A \setminus B$ such that $B \cup \{x\} \in \mathcal{I}$.

**Lemma** Let $F \subsetneq S \in \mathcal{I}$, $e^* \not\in S$ and $F \cup \{e^*\} \in \mathcal{I}$. Then there exists some $e' \in S \setminus F$ such that $S \setminus \{e'\} \cup \{e^*\} \in \mathcal{I}$.

**Algorithm for Proof of Lemma Using Exchange Property**

1: $B \leftarrow F \cup \{e^*\}$
2: **while** $|B| < |S|$ **do**
3: by exchange property, there is some $x \in S \setminus B$ such that $B \cup \{x\} \in \mathcal{I}$
4: $B \leftarrow B \cup \{x\}$

In the end, $|B| = |S|$ and they differ by exactly 1 element. Thus, $B = S \setminus \{e'\} \cup \{e^*\}$ for some $e' \in S \setminus F$. 
exchange property: If $A, B \in \mathcal{I}$ and $|A| > |B|$, then there exists $x \in A \setminus B$ such that $B \cup \{x\} \in \mathcal{I}$.

**Lemma** Let $F \subsetneq S \in \mathcal{I}$, $e^* \notin S$ and $F \cup \{e^*\} \in \mathcal{I}$. Then there exists some $e' \in S \setminus F$ such that $S \setminus \{e'\} \cup \{e^*\} \in \mathcal{I}$.

**Algorithm for Proof of Lemma Using Exchange Property**

1: $B \leftarrow F \cup \{e^*\}$
2: while $|B| < |S|$ do
3: by exchange property, there is some $x \in S \setminus B$ such that $B \cup \{x\} \in \mathcal{I}$
4: $B \leftarrow B \cup \{x\}$

In the end, $|B| = |S|$ and they differ by exactly 1 element. Thus, $B = S \setminus \{e'\} \cup \{e^*\}$ for some $e' \in S \setminus F$. 
exchange property: If $A, B \in \mathcal{I}$ and $|A| > |B|$, then there exists $x \in A \setminus B$ such that $B \cup \{x\} \in \mathcal{I}$.

**Lemma** Let $F \subsetneq S \in \mathcal{I}$, $e^* \notin S$ and $F \cup \{e^*\} \in \mathcal{I}$. Then there exists some $e' \in S \setminus F$ such that $S \setminus \{e'\} \cup \{e^*\} \in \mathcal{I}$.

**Algorithm for Proof of Lemma Using Exchange Property**

1: $B \leftarrow F \cup \{e^*\}$
2: while $|B| < |S|$ do
3: by exchange property, there is some $x \in S \setminus B$ such that $B \cup \{x\} \in \mathcal{I}$
4: $B \leftarrow B \cup \{x\}$

In the end, $|B| = |S|$ and they differ by exactly 1 element.
exchange property: If $A, B \in \mathcal{I}$ and $|A| > |B|$, then there exists $x \in A \setminus B$ such that $B \cup \{x\} \in \mathcal{I}$.

**Lemma** Let $F \subsetneq S \in \mathcal{I}$, $e^* \notin S$ and $F \cup \{e^*\} \in \mathcal{I}$. Then there exists some $e' \in S \setminus F$ such that $S \setminus \{e'\} \cup \{e^*\} \in \mathcal{I}$.

Algorithm for Proof of Lemma Using Exchange Property

1: $B \leftarrow F \cup \{e^*\}$
2: **while** $|B| < |S|$ **do**
3: by exchange property, there is some $x \in S \setminus B$ such that $B \cup \{x\} \in \mathcal{I}$
4: $B \leftarrow B \cup \{x\}$

In the end, $|B| = |S|$ and they differ by exactly 1 element.

Thus, $B = S \setminus \{e'\} \cup \{e^*\}$ for some $e' \in S \setminus F$. 
Lemma  Let $F \subsetneq S \in \mathcal{I}$, $e^* \notin S$ and $F \cup \{e^*\} \in \mathcal{I}$. Then there exists some $e' \in S \setminus F$ such that $S \setminus \{e'\} \cup \{e^*\} \in \mathcal{I}$. 

Greedy Algorithm

1: $F \leftarrow \emptyset$
2: while $\exists e \in E \setminus F$ such that $F \cup \{e\} \in \mathcal{I}$ do
3: find the $e^* \in E \setminus F, F \cup \{e^*\} \in \mathcal{I}$ with maximum $w_{e^*}$
4: $F \leftarrow F \cup \{e^*\}$

Analysis of Greedy Algorithm

show the algorithm will never encounter a failure $F$ (recall $F$ is a failure if it is not a subset of any optimum solution), $F = \emptyset$ is not a failure initially assume $F$ is not a failure at the beginning of some iteration. i.e, there is an optimum solution $S$ such that $F \subseteq S$, $e^*$: the element chosen in the iteration
**Lemma** Let $F \subsetneq S \in \mathcal{I}$, $e^* \notin S$ and $F \cup \{e^*\} \in \mathcal{I}$. Then there exists some $e' \in S \setminus F$ such that $S \setminus \{e'\} \cup \{e^*\} \in \mathcal{I}$.

**Greedy Algorithm**

1: $F \leftarrow \emptyset$
2: **while** $\exists e \in E \setminus F$ such that $F \cup \{e\} \in \mathcal{I}$ **do**
3: find the $e^* \in E \setminus F, F \cup \{e^*\} \in \mathcal{I}$ with maximum $w_{e^*}$
4: $F \leftarrow F \cup \{e^*\}$
Lemma  Let $F \subsetneq S \in \mathcal{I}$, $e^* \notin S$ and $F \cup \{e^*\} \in \mathcal{I}$. Then there exists some $e' \in S \setminus F$ such that $S \setminus \{e'\} \cup \{e^*\} \in \mathcal{I}$.

Greedy Algorithm

1: $F \leftarrow \emptyset$
2: while $\exists e \in E \setminus F$ such that $F \cup \{e\} \in \mathcal{I}$ do
3: find the $e^* \in E \setminus F$, $F \cup \{e^*\} \in \mathcal{I}$ with maximum $w_{e^*}$
4: $F \leftarrow F \cup \{e^*\}$

Analysis of Greedy Algorithm
Lemma Let $F \subsetneq S \in \mathcal{I}$, $e^* \notin S$ and $F \cup \{e^*\} \in \mathcal{I}$. Then there exists some $e' \in S \setminus F$ such that $S \setminus \{e'\} \cup \{e^*\} \in \mathcal{I}$.

Greedy Algorithm

1: $F \leftarrow \emptyset$
2: while $\exists e \in E \setminus F$ such that $F \cup \{e\} \in \mathcal{I}$ do
3: find the $e^* \in E \setminus F$, $F \cup \{e^*\} \in \mathcal{I}$ with maximum $w_{e^*}$
4: $F \leftarrow F \cup \{e^*\}$

Analysis of Greedy Algorithm

- show the algorithm will never encounter a failure $F$ (recall $F$ is a failure if it is not a subset of any optimum solution),
Lemma  Let $F \not\subset S \in \mathcal{I}$, $e^* \not\in S$ and $F \cup \{e^*\} \in \mathcal{I}$. Then there exists some $e' \in S \setminus F$ such that $S \setminus \{e'\} \cup \{e^*\} \in \mathcal{I}$.

Greedy Algorithm

1: $F \leftarrow \emptyset$
2: while $\exists e \in E \setminus F$ such that $F \cup \{e\} \in \mathcal{I}$ do
3: find the $e^* \in E \setminus F, F \cup \{e^*\} \in \mathcal{I}$ with maximum $w_{e^*}$
4: $F \leftarrow F \cup \{e^*\}$

Analysis of Greedy Algorithm

- show the algorithm will never encounter a failure $F$ (recall $F$ is a failure if it is not a subset of any optimum solution),
- $F = \emptyset$ is not a failure initially
Lemma  Let $F \subsetneq S \in \mathcal{I}$, $e^* \notin S$ and $F \cup \{e^*\} \in \mathcal{I}$. Then there exists some $e' \in S \setminus F$ such that $S \setminus \{e'\} \cup \{e^*\} \in \mathcal{I}$.

Greedy Algorithm

1: $F \leftarrow \emptyset$
2: while $\exists e \in E \setminus F$ such that $F \cup \{e\} \in \mathcal{I}$ do
3: find the $e^* \in E \setminus F$, $F \cup \{e^*\} \in \mathcal{I}$ with maximum $w_{e^*}$
4: $F \leftarrow F \cup \{e^*\}$

Analysis of Greedy Algorithm

- show the algorithm will never encounter a failure $F$ (recall $F$ is a failure if it is not a subset of any optimum solution),
- $F = \emptyset$ is not a failure initially
- assume $F$ is not a failure at the beginning of some iteration, i.e, there is an optimum solution $S$ such that $F \subseteq S$, 
Lemma  Let $F \varsubsetneq S \in \mathcal{I}$, $e^* \notin S$ and $F \cup \{e^*\} \in \mathcal{I}$. Then there exists some $e' \in S \setminus F$ such that $S \setminus \{e'\} \cup \{e^*\} \in \mathcal{I}$.

Greedy Algorithm

1: $F \leftarrow \emptyset$
2: while $\exists e \in E \setminus F$ such that $F \cup \{e\} \in \mathcal{I}$ do
3: find the $e^* \in E \setminus F, F \cup \{e^*\} \in \mathcal{I}$ with maximum $w_{e^*}$
4: $F \leftarrow F \cup \{e^*\}$

Analysis of Greedy Algorithm

- show the algorithm will never encounter a failure $F$ (recall $F$ is a failure if it is not a subset of any optimum solution),
- $F = \emptyset$ is not a failure initially
- assume $F$ is not a failure at the beginning of some iteration. i.e, there is an optimum solution $S$ such that $F \subseteq S$,
- $e^*$: the element chosen in the iteration
Lemma Let $F \subsetneq S \in \mathcal{I}$, $e^* \notin S$ and $F \cup \{e^*\} \in \mathcal{I}$. Then there exists some $e' \in S \setminus F$ such that $S \setminus \{e'\} \cup \{e^*\} \in \mathcal{I}$.

Greedy Algorithm

1: $F \leftarrow \emptyset$
2: while $\exists e \in E \setminus F$ such that $F \cup \{e\} \in \mathcal{I}$ do
3: find the $e^* \in E \setminus F$, $F \cup \{e^*\} \in \mathcal{I}$ with maximum $w_{e^*}$
4: $F \leftarrow F \cup \{e^*\}$

Analysis of Greedy Algorithm

- if $e^* \in S$, then $F \cup \{e^*\}$ is not a failure
**Lemma** Let $F \subsetneq S \in \mathcal{I}$, $e^* \notin S$ and $F \cup \{e^*\} \in \mathcal{I}$. Then there exists some $e' \in S \setminus F$ such that $S \setminus \{e'\} \cup \{e^*\} \in \mathcal{I}$.

**Greedy Algorithm**

1: $F \leftarrow \emptyset$
2: while $\exists e \in E \setminus F$ such that $F \cup \{e\} \in \mathcal{I}$ do
3: find the $e^* \in E \setminus F$, $F \cup \{e^*\} \in \mathcal{I}$ with maximum $w_{e^*}$
4: $F \leftarrow F \cup \{e^*\}$

**Analysis of Greedy Algorithm**

- if $e^* \in S$, then $F \cup \{e^*\}$ is not a failure
- if $e^* \notin S$, there is $e' \in S \setminus F$ such that $S \setminus \{e'\} \cup \{e^*\} \in \mathcal{I}$
Lemma Let $F \subsetneq S \in \mathcal{I}$, $e^* \notin S$ and $F \cup \{e^*\} \in \mathcal{I}$. Then there exists some $e' \in S \setminus F$ such that $S \setminus \{e'\} \cup \{e^*\} \in \mathcal{I}$.

Greedy Algorithm

1: $F \leftarrow \emptyset$
2: while $\exists e \in E \setminus F$ such that $F \cup \{e\} \in \mathcal{I}$ do
3: find the $e^* \in E \setminus F$, $F \cup \{e^*\} \in \mathcal{I}$ with maximum $w_{e^*}$
4: $F \leftarrow F \cup \{e^*\}$

Analysis of Greedy Algorithm

- if $e^* \in S$, then $F \cup \{e^*\}$ is not a failure
- if $e^* \notin S$, there is $e' \in S \setminus F$ such that $S \setminus \{e'\} \cup \{e^*\} \in \mathcal{I}$
- $S$ is valid $\rightarrow F \cup \{e'\}$ is valid
Lemma  Let $F \subsetneq S \in \mathcal{I}$, $e^* \not\in S$ and $F \cup \{e^*\} \in \mathcal{I}$. Then there exists some $e' \in S \setminus F$ such that $S \setminus \{e'\} \cup \{e^*\} \in \mathcal{I}$.

Greedy Algorithm

1: $F \leftarrow \emptyset$
2: while $\exists e \in E \setminus F$ such that $F \cup \{e\} \in \mathcal{I}$ do
3: find the $e^* \in E \setminus F$, $F \cup \{e^*\} \in \mathcal{I}$ with maximum $w_{e^*}$
4: $F \leftarrow F \cup \{e^*\}$

Analysis of Greedy Algorithm

- if $e^* \in S$, then $F \cup \{e^*\}$ is not a failure
- if $e^* \not\in S$, there is $e' \in S \setminus F$ such that $S \setminus \{e'\} \cup \{e^*\} \in \mathcal{I}$
- $S$ is valid $\rightarrow F \cup \{e'\}$ is valid
- by the way we choose $e^*$: $w_{e'} \leq w_{e^*}$
Lemma  Let \( F \subseteq S \in \mathcal{I}, e^* \notin S \) and \( F \cup \{e^*\} \in \mathcal{I} \). Then there exists some \( e' \in S \setminus F \) such that \( S \setminus \{e'\} \cup \{e^*\} \in \mathcal{I} \).

Greedy Algorithm

1: \( F \leftarrow \emptyset \)
2: \( \textbf{while} \ \exists e \in E \setminus F \text{ such that } F \cup \{e\} \in \mathcal{I} \ \textbf{do} \)
3: \( \text{find the } e^* \in E \setminus F, F \cup \{e^*\} \in \mathcal{I} \text{ with maximum } w_{e^*} \)
4: \( F \leftarrow F \cup \{e^*\} \)

Analysis of Greedy Algorithm

- if \( e^* \in S \), then \( F \cup \{e^*\} \) is not a failure
- if \( e^* \notin S \), there is \( e' \in S \setminus F \) such that \( S \setminus \{e'\} \cup \{e^*\} \in \mathcal{I} \)
- \( S \) is valid \( \rightarrow F \cup \{e'\} \) is valid
- by the way we choose \( e^* \): \( w_{e'} \leq w_{e^*} \)
- \( S \setminus \{e'\} \cup \{e^*\} \) is also optimum
**Lemma**  Let $F \subsetneq S \in \mathcal{I}$, $e^* \notin S$ and $F \cup \{e^*\} \in \mathcal{I}$. Then there exists some $e' \in S \setminus F$ such that $S \setminus \{e'\} \cup \{e^*\} \in \mathcal{I}$.

**Greedy Algorithm**

1: $F \leftarrow \emptyset$
2: while $\exists e \in E \setminus F$ such that $F \cup \{e\} \in \mathcal{I}$ do
3: find the $e^* \in E \setminus F$, $F \cup \{e^*\} \in \mathcal{I}$ with maximum $w_{e^*}$
4: $F \leftarrow F \cup \{e^*\}$

**Analysis of Greedy Algorithm**

- $S \setminus \{e'\} \cup \{e^*\}$ is also optimum
Lemma  Let $F \subsetneq S \in \mathcal{I}$, $e^* \notin S$ and $F \cup \{e^*\} \in \mathcal{I}$. Then there exists some $e' \in S \setminus F$ such that $S \setminus \{e'\} \cup \{e^*\} \in \mathcal{I}$.

Greedy Algorithm

1: $F \leftarrow \emptyset$
2: while $\exists e \in E \setminus F$ such that $F \cup \{e\} \in \mathcal{I}$ do
3: find the $e^* \in E \setminus F, F \cup \{e^*\} \in \mathcal{I}$ with maximum $w_{e^*}$
4: $F \leftarrow F \cup \{e^*\}$

Analysis of Greedy Algorithm

- $S \setminus \{e'\} \cup \{e^*\}$ is also optimum
- thus $F \cup \{e^*\}$ is not a failure.
Outline

1. Motivation: Maximum Weight Spanning Tree

2. Introduction to Matroid
   - Analysis of Greedy Algorithm

3. Examples of Matroids
Uniform Matroid

\[ \mathcal{I} = \{ X \subseteq E : |X| \leq k \}, \text{ where } k \geq 1 \text{ is an integer}. \]
Uniform Matroid

\[ I = \{ X \subseteq E : |X| \leq k \}, \text{ where } k \geq 1 \text{ is an integer.} \]

Example:

\[ E = \{ a, b, c, d \}, \quad k = 2 \]
\[ I = \{ \emptyset, \{ a \}, \{ b \}, \{ c \}, \{ d \}, \{ a, b \}, \{ a, c \}, \{ a, d \}, \{ b, c \}, \{ b, d \}, \{ c, d \} \} \]
Uniform Matroid

\[ \mathcal{I} = \{ X \subseteq E : |X| \leq k \}, \text{ where } k \geq 1 \text{ is an integer.} \]

Example:

\[ E = \{ a, b, c, d \}, \ k = 2 \]
\[ \mathcal{I} = \{ \emptyset, \{ a \}, \{ b \}, \{ c \}, \{ d \}, \{ a, b \}, \{ a, c \}, \{ a, d \}, \{ b, c \}, \{ b, d \}, \{ c, d \} \} \]

- Exchange property holds trivially.
**Uniform Matroid**

\[ \mathcal{I} = \{ X \subseteq E : |X| \leq k \}, \text{ where } k \geq 1 \text{ is an integer.} \]

**Example:**

\[ E = \{ a, b, c, d \}, \ k = 2 \]

\[ \mathcal{I} = \{ \emptyset, \{ a \}, \{ b \}, \{ c \}, \{ d \}, \]
\[ \{ a, b \}, \{ a, c \}, \{ a, d \}, \{ b, c \}, \{ b, d \}, \{ c, d \} \} \]

- Exchange property holds trivially.
- Greedy algorithm is optimum trivially.
Partition Matroid

- $E$: ground set
- $E$: ground set
- $E$ is partitioned into disjoint sets $E_1, E_2, \cdots, E_\ell$
$E$: ground set
$E$ is partitioned into disjoint sets $E_1, E_2, \cdots, E_\ell$
$k_1, k_2, \cdots, k_\ell$ are non-negative integers.
Partition Matroid

- $E$: ground set
- $E$ is partitioned into disjoint sets $E_1, E_2, \cdots, E_\ell$
- $k_1, k_2, \cdots, k_\ell$ are non-negative integers.
- $\mathcal{I} = \{X \subseteq E : |X \cap E_i| \leq k_i, \forall i = 1, 2, \cdots, \ell\}$
Partition Matroid

- $E$: ground set
- $E$ is partitioned into disjoint sets $E_1, E_2, \cdots, E_\ell$
- $k_1, k_2, \cdots, k_\ell$ are non-negative integers.
- $\mathcal{I} = \{X \subseteq E : |X \cap E_i| \leq k_i, \forall i = 1, 2, \cdots, \ell\}$
- That is, $X \subseteq E$ is independent if it contains at most $k_i$ elements in $E_i$, for every $i \in \{1, 2, \cdots, \ell\}$.
Example

- $E = \{1, 2, 3, 4, 5\}$ is partitioned into $E_1 = \{1, 2\}$ and $E_2 = \{3, 4, 5\}$
- $k_1 = 1$ and $k_2 = 2$

$I = \{\emptyset, \{3\}, \{4\}, \{5\}, \{3, 4\}, \{3, 5\}, \{4, 5\},$

$\{1\}, \{1, 3\}, \{1, 4\}, \{1, 5\}, \{1, 3, 4\}, \{1, 3, 5\}, \{1, 4, 5\},$

$\{2\}, \{2, 3\}, \{2, 4\}, \{2, 5\}, \{2, 3, 4\}, \{2, 3, 5\}, \{2, 4, 5\}\}$

- $(E, I)$ is a partition matroid.
A partition matroid is indeed a matroid

Proof of Exchange property

Assume $A \in I$, $B \in I$ and $|A| > |B|$. Then there must be some $i$ such that $|A \cap E_i| > |B \cap E_i|$. Let $x \in E_i$ be an item in $A$ but not in $B$. $B \cup \{x\} \in I$. 
A partition matroid is indeed a matroid

Proof of Exchange property

- Assume $A \in \mathcal{I}$, $B \in \mathcal{I}$ and $|A| > |B|$
A partition matroid is indeed a matroid

Proof of Exchange property

- Assume $A \in \mathcal{I}$, $B \in \mathcal{I}$ and $|A| > |B|$
- Then there must be some $i$ such that $|A \cap E_i| > |B \cap E_i|$. 
A partition matroid is indeed a matroid

Proof of Exchange property

- Assume $A \in \mathcal{I}$, $B \in \mathcal{I}$ and $|A| > |B|$.
- Then there must be some $i$ such that $|A \cap E_i| > |B \cap E_i|$.
- Let $x \in E_i$ be an item in $A$ but not in $B$. 


Proof of Exchange property

- Assume $A \in \mathcal{I}$, $B \in \mathcal{I}$ and $|A| > |B|$
- Then there must be some $i$ such that $|A \cap E_i| > |B \cap E_i|$.
- Let $x \in E_i$ be an item in $A$ but not in $B$
- $B \cup \{x\} \in \mathcal{I}$
Q: What is the next generalization?
Q: What is the next generalization?

A: A laminar matroid.
Def. Given a ground set \( E \), a family \( \mathcal{E} \) of subsets of \( E \) is called a **laminar family** if for every two distinct subsets \( X, Y \in \mathcal{E} \), we have either \( X \subsetneq Y \), or \( Y \subsetneq X \), or \( X \cap Y = \emptyset \).
Def. Given a ground set $E$, a family $\mathcal{E}$ of subsets of $E$ is called a **laminar family** if for every two distinct subsets $X, Y \in \mathcal{E}$, we have either $X \subsetneq Y$, or $Y \subsetneq X$, or $X \cap Y = \emptyset$. 

![Diagram of laminar family](image)
Def. Given a ground set $E$, a family $\mathcal{E}$ of subsets of $E$ is called a laminar family if for every two distinct subsets $X, Y \in \mathcal{E}$, we have either $X \subseteq Y$, or $Y \subseteq X$, or $X \cap Y = \emptyset$.

$\mathcal{E}$ is a laminar family if no two circles cross each other.
A laminar family of subsets can be organized into nodes of many rooted trees.
A laminar family of subsets can be organized into nodes of many rooted trees.
A laminar family of subsets can be organized into nodes of many rooted trees.

A set $X \in \mathcal{E}$ is a parent of $Y \in \mathcal{E}$ if $Y \subsetneq X$ and there is no $Z \in \mathcal{E}$ with $Y \subsetneq Z \subsetneq X$. 
Def. (Laminar Matroid)
Def. (Laminar Matroid)

- $E$: ground set
**Def.** (Laminar Matroid)

- $E$: ground set
- $\mathcal{E}$: a laminar family of subsets of $E$

$(E, I)$ is called a laminar matroid.

**Example:**

$E = \{1, 2, 3, 4, 5, 6\}$  
$\mathcal{E} = \{\{1, 2\}, \{3, 4, 5\}, \{1, 2, 3, 4, 5, 6\}\}  
\{1, 2\} = 1, \{3, 4, 5\} = 2, \{1, 2, 3, 4, 5, 6\} = 3$

Then,  
$\{1, 3, 6\} \in I$ since it contains 1 element from $\{1, 2\}$, 1 element from $\{3, 4, 5\}$ and 3 elements in total.  
$\{1, 2, 6\} /\in I$ since it contains 2 elements in $\{1, 2\}$.  
$\{3, 4, 5\} /\in I$ since it contains 3 elements in $\{3, 4, 5\}$.  

Def. (Laminar Matroid)

- $E$: ground set
- $\mathcal{E}$: a laminar family of subsets of $E$
- $k_A : A \in \mathcal{E}$: an positive integer.

$$\mathcal{I} = \{X \subseteq E : |X \cap A| \leq k_A, \forall A \in \mathcal{E}\}$$
Def. (Laminar Matroid)

- \(E\): ground set
- \(\mathcal{E}\): a laminar family of subsets of \(E\)
- \(k_A: A \in \mathcal{E}\): an positive integer.

\[\mathcal{I} = \{X \subseteq E : |X \cap A| \leq k_A, \forall A \in \mathcal{E}\}\]

- \((E, \mathcal{I})\) is called a laminar matroid.
Def. (Laminar Matroid)

- $E$: ground set
- $\mathcal{E}$: a laminar family of subsets of $E$
- $k_A : A \in \mathcal{E}$: an positive integer.

$$\mathcal{I} = \{ X \subseteq E : |X \cap A| \leq k_A, \forall A \in \mathcal{E} \}$$

$(E, \mathcal{I})$ is called a laminar matroid.

Example:
Def. (Laminar Matroid)

- $E$: ground set
- $\mathcal{E}$: a laminar family of subsets of $E$
- $k_A : A \in \mathcal{E}$: an positive integer.
- $(E, \mathcal{I})$ is called a laminar matroid.

\[
\mathcal{I} = \{ X \subseteq E : |X \cap A| \leq k_A, \forall A \in \mathcal{E} \}
\]

Example:

- $E = \{1, 2, 3, 4, 5, 6\}$
Def. (Laminar Matroid)

- \( E \): ground set
- \( \mathcal{E} \): a laminar family of subsets of \( E \)
- \( k_A : A \in \mathcal{E} \): an positive integer.
- \( (E, \mathcal{I}) \) is called a laminar matroid.

\[
\mathcal{I} = \{ X \subseteq E : |X \cap A| \leq k_A, \forall A \in \mathcal{E} \}
\]

Example:

- \( E = \{1, 2, 3, 4, 5, 6\} \)
- \( \mathcal{E} = \{ \{1, 2\}, \{3, 4, 5\}, \{1, 2, 3, 4, 5, 6\} \} \)
**Def. (Laminar Matroid)**

- $E$: ground set
- $\mathcal{E}$: a laminar family of subsets of $E$
- $k_A : A \in \mathcal{E}$: an positive integer.
- $\mathcal{I} = \{ X \subseteq E : |X \cap A| \leq k_A, \forall A \in \mathcal{E} \}$
- $(E, \mathcal{I})$ is called a laminar matroid.

**Example:**

- $E = \{1, 2, 3, 4, 5, 6\}$
- $\mathcal{E} = \{\{1, 2\}, \{3, 4, 5\}, \{1, 2, 3, 4, 5, 6\}\}$
- $k_{\{1,2\}} = 1, k_{\{3,4,5\}} = 2, k_{\{1,2,3,4,5,6\}} = 3$
Def. (Laminar Matroid)

- $E$: ground set
- $\mathcal{E}$: a laminar family of subsets of $E$
- $k_A: A \in \mathcal{E}$: an positive integer.
  \[ \mathcal{I} = \{ X \subseteq E : |X \cap A| \leq k_A, \forall A \in \mathcal{E} \} \]
- $(E, \mathcal{I})$ is called a laminar matroid.

Example:

- $E = \{1, 2, 3, 4, 5, 6\}$
- $\mathcal{E} = \{\{1, 2\}, \{3, 4, 5\}, \{1, 2, 3, 4, 5, 6\}\}$
- $k_{\{1,2\}} = 1, k_{\{3,4,5\}} = 2, k_{\{1,2,3,4,5,6\}} = 3$
- Then, $\{1, 3, 6\} \in \mathcal{I}$ since it contains 1 elements from $\{1, 2\}$, 1 $\leq$ 2 elements from $\{3, 4, 5\}$ and 3 $\leq$ elements in total.
Def. (Laminar Matroid)

- $E$: ground set
- $\mathcal{E}$: a laminar family of subsets of $E$
- $k_A : A \in \mathcal{E}$: an positive integer.
  
  $$\mathcal{I} = \{X \subseteq E : |X \cap A| \leq k_A, \forall A \in \mathcal{E}\}$$

- $(E, \mathcal{I})$ is called a laminar matroid.

Example:

- $E = \{1, 2, 3, 4, 5, 6\}$
- $\mathcal{E} = \{\{1, 2\}, \{3, 4, 5\}, \{1, 2, 3, 4, 5, 6\}\}$
- $k_{\{1,2\}} = 1, k_{\{3,4,5\}} = 2, k_{\{1,2,3,4,5,6\}} = 3$

- Then, $\{1, 3, 6\} \in \mathcal{I}$ since it contains 1 elements from $\{1, 2\}$, 1 $\leq$ 2 elements from $\{3, 4, 5\}$ and 3 $\leq$ elements in total.
- $\{1, 2, 6\} \notin \mathcal{I}$ since it contains 2 elements in $\{1, 2\}$. 
Def. (Laminar Matroid)
- \( E \): ground set
- \( \mathcal{E} \): a laminar family of subsets of \( E \)
- \( k_A : A \in \mathcal{E} \): an positive integer.
  \[ \mathcal{I} = \{ X \subseteq E : |X \cap A| \leq k_A, \forall A \in \mathcal{E} \} \]
- \((E, \mathcal{I})\) is called a laminar matroid.

Example:
- \( E = \{1, 2, 3, 4, 5, 6\} \)
- \( \mathcal{E} = \{\{1, 2\}, \{3, 4, 5\}, \{1, 2, 3, 4, 5, 6\}\} \)
- \( k_{\{1,2\}} = 1, k_{\{3,4,5\}} = 2, k_{\{1,2,3,4,5,6\}} = 3 \)
- Then, \( \{1, 3, 6\} \in \mathcal{I} \) since it contains 1 elements from \( \{1, 2\} \), 1 \( \leq 2 \) elements from \( \{3, 4, 5\} \) and 3 \( \leq \) elements in total.
- \( \{1, 2, 6\} \notin \mathcal{I} \) since it contains 2 elements in \( \{1, 2\} \).
- \( \{3, 4, 5\} \notin \mathcal{I} \) since it contains 3 elements in \( \{3, 4, 5\} \).
Note: some constraints may be redundant.
Note: some constraints may be redundant.

- If $k_{\{2,3,4,5\}} = 3$ but $k_{\{1,2,3,4,5\}} = 2$, then the constraint that $|X \cap \{2,3,4,5\}| \leq 3$ is redundant.
Note: some constraints may be redundant.

- If $k_{\{2,3,4,5\}} = 3$ but $k_{\{1,2,3,4,5\}} = 2$, then the constraint that $|X \cap \{2,3,4,5\}| \leq 3$ is redundant.

- If $k_{\{1,2,3\}} = 2$ and $k_{\{4,5,6\}} = 2$ and $k_{\{1,2,3,4,5,6\}} = 4$, then the constraint that $|X \cap \{1,2,3,4,5,6\}| \leq 4$ is redundant.
For simplicity, we assume the laminar family $\mathcal{E}$ is complete:
For simplicity, we assume the laminar family $\mathcal{E}$ is complete:

- The whole set $E$ is in the laminar family
A Laminar Matroid is Indeed a Matroid

- For simplicity, we assume the laminar family $\mathcal{E}$ is complete:
  - The whole set $E$ is in the laminar family
  - Every singleton set $\{e\}$ is in the laminar family.
A Laminar Matroid is Indeed a Matroid

- For simplicity, we assume the laminar family $\mathcal{E}$ is complete:
  - The whole set $E$ is in the laminar family
  - Every singleton set $\{e\}$ is in the laminar family.
A Laminar Matroid is Indeed a Matroid

- For simplicity, we assume the laminar family $\mathcal{E}$ is complete:
  - The whole set $E$ is in the laminar family
  - Every singleton set $\{e\}$ is in the laminar family.
A Laminar Matroid is Indeed a Matroid

- For simplicity, we assume the laminar family $\mathcal{E}$ is complete:
  - The whole set $E$ is in the laminar family
  - Every singleton set $\{e\}$ is in the laminar family.
We maintain a set $C$ in the laminar tree and the invariant that $|C \cap A| > |C \cap B|$.

Maintain: $|C \cap A| > |C \cap B|$.
A Laminar Matroid is Indeed a Matroid

- We maintain a set $C$ in the laminar tree and the invariant that $|C \cap A| > |C \cap B|$.
- Initially $C = E$ and $|E \cap A| > |E \cap B|$ holds.

Maintain: $|C \cap A| > |C \cap B|$
A Laminar Matroid is Indeed a Matroid

- We maintain a set $C$ in the laminar tree and the invariant that $|C \cap A| > |C \cap B|$.  
- Initially $C = E$ and $|E \cap A| > |E \cap B|$ holds.  
- While $C$ is not a singleton set, repeat the following:

Maintain: $|C \cap A| > |C \cap B|$
A Laminar Matroid is Indeed a Matroid

- We maintain a set $C$ in the laminar tree and the invariant that $|C \cap A| > |C \cap B|$.
- Initially $C = E$ and $|E \cap A| > |E \cap B|$ holds.
- While $C$ is not a singleton set, repeat the following:
  - Consider the children of $C$ in the laminar tree; they form a partition of $C$. Maintain: $|C \cap A| > |C \cap B|$
A Laminar Matroid is Indeed a Matroid

- We maintain a set $C$ in the laminar tree and the invariant that $|C \cap A| > |C \cap B|$. 

- Initially $C = E$ and $|E \cap A| > |E \cap B|$ holds.

- While $C$ is not a singleton set, repeat the following:
  - Consider the children of $C$ in the laminar tree; they form a partition of $C$.
  - There must be one child $C'$ such that $|C' \cap A| > |C' \cap B|$.

Maintain: $|C \cap A| > |C \cap B|$.
A Laminar Matroid is Indeed a Matroid

- We maintain a set $C$ in the laminar tree and the invariant that $|C \cap A| > |C \cap B|$.
- Initially $C = E$ and $|E \cap A| > |E \cap B|$ holds.
- While $C$ is not a singleton set, repeat the following:
  - Consider the children of $C$ in the laminar tree; they form a partition of $C$.
  - There must be one child $C'$ such that $|C' \cap A| > |C' \cap B|$.
  - Let $C = C'$

Maintain: $|C \cap A| > |C \cap B|$
A Laminar Matroid is Indeed a Matroid

- We maintain a set $C$ in the laminar tree and the invariant that $|C \cap A| > |C \cap B|$.
- Initially $C = E$ and $|E \cap A| > |E \cap B|$ holds.
- While $C$ is not a singleton set, repeat the following:
  - Consider the children of $C$ in the laminar tree; they form a partition of $C$.
  - There must be one child $C'$ such that $|C' \cap A| > |C' \cap B|$.
  - Let $C = C'$.
A Laminar Matroid is Indeed a Matroid

- We maintain a set $C$ in the laminar tree and the invariant that $|C \cap A| > |C \cap B|$.
- Initially $C = E$ and $|E \cap A| > |E \cap B|$ holds.
- While $C$ is not a singleton set, repeat the following:
  - Consider the children of $C$ in the laminar tree; they form a partition of $C$.
  - There must be one child $C'$ such that $|C' \cap A| > |C' \cap B|$.
  - Let $C = C'$
Eventually, we have a path of sets

\[ E = C_0 \supseteq C_1 \supseteq C_2 \supseteq C_3 \cdots \supseteq C_\ell = \{x\} \]

in the laminar tree, such that for every \( C_i \) in the path, \(|C_i \cap A| > |C_i \cap B|\)
Maintain: $|C \cap A| > |C \cap B|$

Eventually, we have a path of sets

$E = C_0 \supseteq C_1 \supseteq C_2 \supseteq C_3 \cdots \supseteq C_\ell = \{x\}$ in the laminar tree, such that for every $C_i$ in the path, $|C_i \cap A| > |C_i \cap B|$

$B \cup \{x\}$ satisfies all the cardinality constraints since for every $C \in \mathcal{E}$ that contains $x$, we have $|B \cap C| < |A \cap C| \leq k_C$, which implies $|(B \cup \{x\}) \cap C| \leq k_C$
The constraint that $\mathcal{E}$ is a laminar family is needed.
• The constraint that \( \mathcal{E} \) is a laminar family is needed.
• The following example is not a matroid:

\[
\mathcal{E} = \{1, 2, 3\}.
\]

Then \( \{1, 3\} \in \mathcal{I} \) and \( \{2\} \in \mathcal{I} \), but \( \{1, 2\} \not\in \mathcal{I} \) and \( \{2, 3\} \not\in \mathcal{I} \).

So the exchange property does not hold.
Thus, laminar matroids are the most general matroids based on cardinality constraints on subsets.
The constraint that $\mathcal{E}$ is a laminar family is needed.

The following example is not a matroid:

- $E = \{1, 2, 3\}$. 
The constraint that $E$ is a laminar family is needed.

The following example is not a matroid:

- $E = \{1, 2, 3\}$.
- $X \subseteq E$ is in $\mathcal{I}$ if and only if $|X \cap \{1, 2\}| \leq 1$ and $|X \cap \{2, 3\}| \leq 1$.

Then $\{1, 3\} \in \mathcal{I}$ and $\{2\} \in \mathcal{I}$, but $\{1, 2\} \not\in \mathcal{I}$ and $\{2, 3\} \not\in \mathcal{I}$.

So the exchange property does not hold.

Thus, laminar matroids are the most general matroids based on cardinality constraints on subsets.
The constraint that $E$ is a laminar family is needed.

The following example is not a matroid:

- $E = \{1, 2, 3\}$.
- $X \subseteq E$ is in $\mathcal{I}$ if and only if $|X \cap \{1, 2\}| \leq 1$ and $|X \cap \{2, 3\}| \leq 1$.
- Then $\{1, 3\} \in \mathcal{I}$ and $\{2\} \in \mathcal{I}$, but $\{1, 2\} \notin \mathcal{I}$ and $\{2, 3\} \notin \mathcal{I}$.

So the exchange property does not hold.
• The constraint that $\mathcal{E}$ is a laminar family is needed.

• The following example is not a matroid:

  • $E = \{1, 2, 3\}$.
  • $X \subseteq E$ is in $\mathcal{I}$ if and only if $|X \cap \{1, 2\}| \leq 1$ and $|X \cap \{2, 3\}| \leq 1$.
  • Then $\{1, 3\} \in \mathcal{I}$ and $\{2\} \in \mathcal{I}$, but $\{1, 2\} \not\in \mathcal{I}$ and $\{2, 3\} \not\in \mathcal{I}$.
    So the exchange property does not hold.

• Thus, laminar matroids are the most general matroids based on cardinality constraints on subsets.
Def. Linear Matroid
Def. Linear Matroid

- \( E = \{v_1, v_2, \cdots, v_n\} \): a set of vectors in \( \mathbb{R}^d \)
**Def. Linear Matroid**

- $E = \{v_1, v_2, \cdots, v_n\}$: a set of vectors in $\mathbb{R}^d$
- A set $X \subseteq E$ is in $\mathcal{I}$, iff the vectors in $X$ are linearly independent.

Let $X = \{u_1, u_2, \cdots, u_k\}$ be a set of vectors. $X$ is linearly independent iff for every $k$ real numbers $\gamma_1, \gamma_2, \cdots, \gamma_k$ that are not all 0's, we have

$$\gamma_1 u_1 + \gamma_2 u_2 + \cdots + \gamma_k u_k \neq 0.$$
Def. Linear Matroid

- $E = \{v_1, v_2, \cdots, v_n\}$: a set of vectors in $\mathbb{R}^d$
- A set $X \subseteq E$ is in $\mathcal{I}$, iff the vectors in $X$ are linearly independent.
- $(E, \mathcal{I})$ is called a linear matroid.
Def. Linear Matroid

- \( E = \{v_1, v_2, \cdots, v_n\} \): a set of vectors in \( \mathbb{R}^d \)
- A set \( X \subseteq E \) is in \( \mathcal{I} \), iff the vectors in \( X \) are linearly independent.
- \( (E, \mathcal{I}) \) is called a \text{linear matroid}.

Recall: \( X = \{u_1, u_2, \cdots, u_k\} \) is linearly independent iff for every \( k \) real numbers \( \gamma_1, \gamma_2, \cdots, \gamma_k \) that are not all 0’s, we have \( \gamma_1 u_1 + \gamma_2 u_2 + \gamma_3 u_3 + \cdots + \gamma_k u_k \neq 0 \).
Def. Linear Matroid

- $E = \{v_1, v_2, \cdots, v_n\}$: a set of vectors in $\mathbb{R}^d$
- A set $X \subseteq E$ is in $\mathcal{I}$, iff the vectors in $X$ are linearly independent.
- $(E, \mathcal{I})$ is called a linear matroid.

Recall: $X = \{u_1, u_2, \cdots, u_k\}$ is linearly independent iff for every $k$ real numbers $\gamma_1, \gamma_2, \cdots, \gamma_k$ that are not all 0’s, we have $\gamma_1 u_1 + \gamma_2 u_2 + \gamma_3 u_3 + \cdots + \gamma_k u_k \neq 0$.

Also, $X = \{u_1, u_2, \cdots, u_k\}$ is linearly independent iff $\text{rank}((u_1, u_2, \cdots, u_k)) = k$. 
A Linear Matroid is Indeed a Matroid

- $E = \{v_1, v_2, \ldots, v_n\}$. 

- $A, B \in I$, i.e., the vectors in $A$ are linearly independent, and the vectors in $B$ are linearly independent.

- $|A| > |B|$, but $\text{span}(B)$ has dimension $|B|$.

- There is at least one vector $v_i \in A$ that is not in $\text{span}(B)$.

- Vectors in $B \cup \{v_i\}$ are also linearly independent.
A Linear Matroid is Indeed a Matroid

- $E = \{v_1, v_2, \ldots, v_n\}$.
- $A, B \in \mathcal{I}$, i.e, the vectors in $A$ are linearly independently, and the vectors in $B$ are linearly independent.
A Linear Matroid is Indeed a Matroid

- \( E = \{ v_1, v_2, \cdots, v_n \} \).
- \( A, B \in \mathcal{I} \), i.e, the vectors in \( A \) are linearly independently, and the vectors in \( B \) are linearly independent.
- \( |A| > |B| \)
A Linear Matroid is Indeed a Matroid

- $E = \{v_1, v_2, \cdots, v_n\}$.
- $A, B \in \mathcal{I}$, i.e, the vectors in $A$ are linearly independently, and the vectors in $B$ are linearly independent.
- $|A| > |B|$
- $\text{span}(B)$ has dimension $|B|$. 
A Linear Matroid is Indeed a Matroid

- \( E = \{v_1, v_2, \ldots, v_n\} \).
- \( A, B \in I \), i.e., the vectors in \( A \) are linearly independently, and the vectors in \( B \) are linearly independent.
- \( |A| > |B| \)
- \( \text{span}(B) \) has dimension \( |B| \).
- There is at least one vector \( v_i \in A \) that is not in \( \text{span}(B) \).
A Linear Matroid is Indeed a Matroid

- $E = \{v_1, v_2, \cdots, v_n\}$.
- $A, B \in \mathcal{I}$, i.e, the vectors in $A$ are linearly independently, and the vectors in $B$ are linearly independent.
- $|A| > |B|$
- span$(B)$ has dimension $|B|$.
- there is at least one vector $v_i \in A$ that is not in span$(B)$.
- vectors in $B \cup \{v_i\}$ are also linearly independent.
Recall: Graphic Matroid

Def.

\[ G = (V, E) \]: an undirected graph. 

\( E \) is the ground set of the matroid. 

\( F \subseteq E \) is in \( I \) iff \((V, F)\) is a forest, i.e., \( F \) does not contain a cycle. 

\((E, I)\) is called a graphic matroid.
Recall: Graphic Matroid

Def.

- \( G = (V, E) \): an undirected graph. \( E \) is the ground set of the matroid.
Recall: Graphic Matroid

**Def.**
- $G = (V, E)$: an undirected graph. $E$ is the ground set of the matroid.
- $F \subseteq E$ is in $\mathcal{I}$ iff $(V, F)$ is a forest, i.e., $F$ does not contain a cycle.
Recall: Graphic Matroid

\[ G = (V, E) \]: an undirected graph. \( E \) is the ground set of the matroid.

\( F \subseteq E \) is in \( \mathcal{I} \) iff \((V, F)\) is a forest, i.e, \( F \) does not contain a cycle.

\((E, \mathcal{I})\) is called a graphic matroid.
Def.

- $G = (U \cup V, E)$: a bipartite graph.
- $U$: ground set of the matroid
- $A \subseteq U$ is in $\mathcal{I}$ iff there is a matching in $G$ that covers $A$. 
Transversal Matroid

Def.

- $G = (U \cup V, E)$: a bipartite graph.
- $U$: ground set of the matroid
- $A \subseteq U$ is in $\mathcal{I}$ iff there is a matching in $G$ that covers $A$. 

\[ \{3, 4, 5\} \in \mathcal{I} \] since there is a matching covering them.

\[ \{1, 2, 3\} \not\in \mathcal{I} \] since no matching can cover them.
Def.

- $G = (U \cup V, E)$: a bipartite graph.
- $U$: ground set of the matroid
- $A \subseteq U$ is in $\mathcal{I}$ iff there is a matching in $G$ that covers $A$.

\[ \{3, 4, 5\} \in \mathcal{I} \text{ since there is a matching covering them.} \]
Transversal Matroid

**Def.**
- $G = (U \cup V, E)$: a bipartite graph.
- $U$: ground set of the matroid
- $A \subseteq U$ is in $\mathcal{I}$ iff there is a matching in $G$ that covers $A$.

- $\{3, 4, 5\} \in \mathcal{I}$ since there is a matching covering them.
- $\{1, 2, 3\} \notin \mathcal{I}$ since no matching can cover them.
A Transversal Matroid is Indeed a Matroid

- $G = (U \uplus V, E)$: a bipartite graph.
A Transversal Matroid is Indeed a Matroid

- $G = (U \uplus V, E)$: a bipartite graph.
- $U$: ground set of the matroid
A Transversal Matroid is Indeed a Matroid

- \( G = (U \cup V, E) \): a bipartite graph.
- \( U \): ground set of the matroid
- \( A \subseteq U \) is in \( \mathcal{I} \) iff there is a matching in \( G \) that covers \( A \).
A Transversal Matroid is Indeed a Matroid

- \( A, B \in \mathcal{I}, |A| > |B| \).
A Transversal Matroid is Indeed a Matroid

- $A, B \in \mathcal{I}$, $|A| > |B|$.
- Red edges: matching covering $A$.
A Transversal Matroid is Indeed a Matroid

- $A, B \in \mathcal{I}, |A| > |B|.$
- Red edges: matching covering $A.$
- Blue edges: matching covering $B.$
A Transversal Matroid is Indeed a Matroid

- \( A, B \in \mathcal{I}, \ |A| > |B| \).
- Red edges: matching covering \( A \).
- Blue edges: matching covering \( B \).
- Consider the graph formed by red and blue edges.
A Transversal Matroid is Indeed a Matroid

- \( A, B \in \mathcal{I}, |A| > |B| \).
- Red edges: matching covering \( A \).
- Blue edges: matching covering \( B \).
- Consider the graph formed by red and blue edges.
- Each connected component is either a cycle, with alternating red and blue edges, or a path, with alternating red and blue edges. One path must have 1 more red edge than the blue edge.
- Augmenting using the path will give a matching that covers \( B \cup \{x\} \), for some \( x \in A \setminus B \).
A Transversal Matroid is Indeed a Matroid

- \( A, B \in \mathcal{I}, |A| > |B| \).
- Red edges: matching covering \( A \).
- Blue edges: matching covering \( B \).
- Consider the graph formed by red and blue edges.
- Each connected component is
  - either a cycle, with alternating red and blue edges.
A Transversal Matroid is Indeed a Matroid

- $A, B \in I$, $|A| > |B|$.
- Red edges: matching covering $A$.
- Blue edges: matching covering $B$.
- Consider the graph formed by red and blue edges.
- Each connected component is
  - either a cycle, with alternating red and blue edges.
  - or a path, with alternating red and blue edges.
A Transversal Matroid is Indeed a Matroid

- $A, B \in \mathcal{I}$, $|A| > |B|$.
- Red edges: matching covering $A$.
- Blue edges: matching covering $B$.
- Consider the graph formed by red and blue edges.
- Each connected component is either a cycle, with alternating red and blue edges.
- or a path, with alternating red and blue edges.
- $|A| > |B|$: one path must have 1 more red edge than the blue edge.
A Transversal Matroid is Indeed a Matroid

- $A, B \in \mathcal{I}$, $|A| > |B|$.
- Red edges: matching covering $A$.
- Blue edges: matching covering $B$.
- Consider the graph formed by red and blue edges.
- Each connected component is
  - either a cycle, with alternating red and blue edges.
  - or a path, with alternating red and blue edges.
- $|A| > |B|$: one path must have 1 more red edge than the blue edge.
- Augmenting using the path will give a matching that covers $B \cup \{x\}$, for some $x \in A \setminus B$. 