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)
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.

**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 $F$. 
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)
- \( e^* \): the edge algorithm chooses in iteration \( i^* \)
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 $F \in I$

**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

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

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

A Natural Generic Greedy Algorithm

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

- For maximum-weight spanning tree, the generic algorithm becomes Kruskal’s algorithm.
**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:**

- \( E = \{a, b, c\}, w_a = 10, w_b = 9, w_c = 9, \)
- \( \mathcal{I} = \{\emptyset, \{a\}, \{b\}, \{c\}, \{b, c\}\} \)
- 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?

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

- $E$ is called the ground set of $M$.
- Every set in $I$ is called an independent set of $M$.
- So, $I$ is the set of independent sets of $M$. 
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 $A' \in I$.
- (exchange property) If $A, B \in I$ and $|A| > |B|$, then there exists $x \in A \setminus B$ such that $B \cup \{x\} \in I$. 

![Diagram of a matroid](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.

- $E = \{e_1, e_2, e_3, e_4, e_5\}$

- $\mathcal{I} = \{\emptyset, \{e_1\}, \{e_2\}, \{e_3\}, \{e_4\}, \{e_5\},$
  $\{e_1, e_2\}, \{e_1, e_3\}, \{e_1, e_4\}, \{e_1, e_5\}, \{e_2, e_3\},$
  $\{e_2, e_4\}, \{e_2, e_5\}, \{e_3, e_4\}, \{e_3, e_5\}, \{e_4, e_5\},$
  $\{e_1, e_2, e_4\}, \{e_1, e_2, e_5\}, \{e_1, e_3, e_4\}, \{e_1, e_3, e_5\},$
  $\{e_1, e_4, e_5\}, \{e_2, e_3, e_4\}, \{e_2, e_3, e_5\}, \{e_2, e_4, e_5\}\}$

- $(E, \mathcal{I})$ is a graphic matroid.
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 \( \{a, b\} \notin \mathcal{I}, \{a, c\} \notin \mathcal{I} \).
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: 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 \( B \cup \{x\} \in \mathcal{I} \).

**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 \( \iff \) 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 $S \setminus \{e'\} \cup \{e^*\} \in \mathcal{I}$.

**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 $F \not\subseteq S \in \mathcal{I}$, $e^* \not\in S$ 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
- 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
- if $e^* \in S$, then $F \cup \{e^*\}$ is not a failure
1. Motivation: Maximum Weight Spanning Tree

2. Introduction to Matroid
   - Analysis of Greedy Algorithm

3. Examples of Matroids
Uniform Matroid

\[ \mathcal{I} = \{ X \subseteq E : |X| \leq k \} \], where \( k \geq 1 \) 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
- $E$ 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 $k_2 = 2$
  
  $\mathcal{I} = \{\emptyset, \{3\}, \{4\}, \{5\}, \{3, 4\}, \{3, 5\}, \{4, 5\}, \{1\}, \{1, 3\}, \{1, 4\}, \{1, 5\}, \{1, 3, 4\}, \{1, 3, 5\}, \{1, 4, 5\}, \{2\}, \{2, 3\}, \{2, 4\}, \{2, 5\}, \{2, 3, 4\}, \{2, 3, 5\}, \{2, 4, 5\}\}$

- $(E, \mathcal{I})$ is a partition matroid.
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 $B$.
- $B \cup \{x\} \in \mathcal{I}$.
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 $X \subseteq Y$, 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 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)**

- $E$: 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.

- 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 $E$ 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

- 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: $|C \cap A| > |C \cap B|$

- Eventually, we have a path of sets $E = C_0 \supsetneq C_1 \supsetneq C_2 \supsetneq C_3 \cdots \supsetneq C_\ell = \{x\}$ 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 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.

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

- \( E = \{v_1, v_2, \ldots, v_n\} \): 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: \( X = \{u_1, u_2, \ldots, u_k\} \) is linearly independent iff for every \( k \) real numbers \( \gamma_1, \gamma_2, \ldots, \gamma_k \) 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, \( X = \{u_1, u_2, \ldots, u_k\} \) is linearly independent iff \( \text{rank}((u_1, u_2, \ldots, u_k)) = k \).
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 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 $\mathcal{I}$ iff $(V, F)$ is a forest, i.e., $F$ does not contain a cycle.
- $(E, \mathcal{I})$ is called a graphic matroid.
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.
- $U$: ground set of the matroid
- $A \subseteq U$ is in $I$ iff there is a matching in $G$ that covers $A$. 
A Transversal Matroid is Indeed a Matroid

- $A, B \in \mathcal{I}$, $|A| > |B|$.
- 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 $x \in A \setminus B$. 