Suppose $G = (V, E)$ is equipped with a non-negative weight function on edges $w : E \to \mathbb{R}_+$.\

**Example**

![Diagram of a graph with labeled edges and vertices]

- The weight of a spanning tree is the sum of the weight of its edges.

**Problem**

Determine a spanning tree of minimum weight in $G$.

We examine two famous greedy algorithms which solve this problem.

In what follows let $G = (V, E), |V| = n$. 
**Prim's Algorithm** (23.2)

- Choose an initial vertex (which is a tree)
- Amongst all edges incident with the current tree whose addition would not create a cycle, choose one of minimum weight.
- Stop when no edges have been selected.

![Diagram](image)

$$W(T) = 18$$

Observe that at each stage of execution, Prim's algorithm maintains a tree since no cycles are created and only incident edges are added.

When this tree contains \( n-1 \) edges it must have \( n \) vertices (by previous theorem), hence it is a Spanning tree.

**Theorem**

This spanning tree has minimum possible weight.

*(See book or TIFR CE 177 for proof.)*
Kruskal's Algorithm (23.2)

- Choose an edge of minimum weight.
- Amongst all edges which do not create a cycle with previously selected edges, choose one of minimum weight.
- Stop when n-1 edges have been selected.

Ex.

\[
W(T) = 18
\]

Observe that at each stage of execution, Kruskal's Algorithm will create a forest (union of disjoint subgraphs) since no cycles are created.

When this forest contains n-1 edges, it must also have n vertices. (Any forest with n-1 edges has at least n vertices. This forest can contain no more than n vertices since it is a subgraph of G.)

Thus the resulting forest is connected (by previous theorem) and is a spanning tree in G.
THEOREM

The spanning tree with minimum weight among all spanning trees in G.

PROOF.
Let T be the spanning tree in G created by Kruskal's algorithm, and let S be any other spanning tree. We must show

\[ w(T) \leq w(S) \]

Let \( e_1, e_2, \ldots, e_{n-1} \) be the edges of T in the order selected by Kruskal's algorithm. Since \( S \neq T \) there is a first edge \( e_k \) which is not in \( S \), i.e.

\[ \{e_1, \ldots, e_{k-1}\} \subseteq E(S) \]
\[ e_k \notin E(S) \]

Let \( H \) be the subgraph obtained by adding \( e_k \) to \( S \) : \( H = S + e_k \). By the tree-ness theorem \( H \) contains a unique cycle which includes \( e_k \), call it \( C \). Note \( C \) must contain an edge \( e \) of \( S \) which is not in \( T \), for otherwise \( C \) is contained in the acyclic \( T \).
Now remove $e$ from $H$ to obtain a subgraph $R$, which is connected since $e$ belongs to a cycle in $H$.

$$R = H - e = S + e_k - e$$

Since $R$ is connected and has $n-1$ edges, it is another spanning tree of $G$, by Treeness Theorem.

The nature of Kruskal's algorithm guarantees that $w(e_k) \leq w(e)$.

If $e$ does not form a cycle with $e_1, \ldots, e_{k-1}$ since $e_1, \ldots, e_{k-1}, e \notin E(S)$. Thus if $w(e) < w(e_k)$, then Kruskal would have chosen $e$ on the $k$th iteration of the greedy loop instead of $e_k$.

Thus $R$ is a spanning tree of $G$ with one more edge in common with $T$ than $S$, and satisfies $w(R) \leq w(S)$.

If $R = T$ we are done, otherwise we may perform this same construction with $R$ in place of $S$. 
i.e., construct another spanning tree $R_2$ with one more edge in common with $T$ than $R$, and satisfying $W(R_1) \leq W(R)$.

Continuing in this fashion we construct a sequence of spanning trees which must eventually reach $T$:

$$W(T) \leq \ldots \leq W(R_1) \leq W(R) \leq W(S),$$

so $W(T) \leq W(S)$ as required.

### MATRICES AND THE GREEDY ALGORITHM

A matrix is an abstract mathematical structure which generalizes many examples where a greedy strategy applies.
A **matroid** is an ordered pair \( M = (S, \mathcal{I}) \) satisfying:

1. \( S \) is a finite, non-empty set, and \( \mathcal{I} \subseteq 2^S \). The members of \( \mathcal{I} \) are called the **independent subsets** of \( S \).

2. **Hereditary Property**
   - If \( B \subseteq I \) and \( A \in \mathcal{I} \), then \( A \subseteq B \).

3. **Exchange Property**
   - If \( A \in \mathcal{I} \), \( B \in \mathcal{I} \), and \( |A| < |B| \), then there exists \( x \in B \setminus A \) such that \( A \cup \{x\} \in \mathcal{I} \).

Note that (2) implies that \( \emptyset \in \mathcal{I} \) (provided \( S \) is itself non-empty).

**Example (Matrix Matroids)**

Let \( D \) be a (rectangular) matrix and let \( S = \{ \text{rows of } D \} \), considered as vectors. Let

\[ I = \{ \{ A \subseteq S \mid A \text{ is linearly independent} \} \} \]
Obviously $S$ is finite and non-empty, and $T \subseteq D(S)$. Properties (1) and (2) are elementary facts of linear algebra.

Similarly we could let $I$ be the columns of $D$.

**Example (Graphical Matroid)**

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

Let $S = E$ and

$$
I = \{ A \subseteq S \mid \text{subgraph } (V, A) \text{ is acyclic} \}.
$$

(1) is clearly satisfied. (2) holds since any subset of an acyclic set of edges is acyclic. (By removing edges we cannot create cycles.)

We prove the exchange property (2):

Let $A, B \in I$ and suppose $|A| < |B|$. Then the forest $(V, A)$ contains exactly $|V| - |A| - 1$ trees, and $(V, B)$ contains $|V| - |B| - 1$ trees.

(If: suppose $(V, A)$ contains $m$ trees:

$$
T_i = (V_i, E_i), 1 \leq i \leq m, \text{ then } |E_i| = |V_i| - 1
$$

(Only if: suppose $(V, B)$ contains $m'$ trees...)}