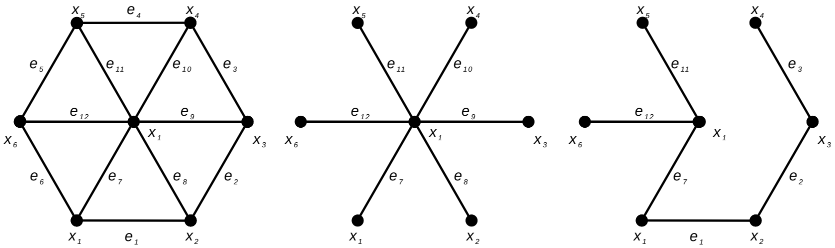
Graph and Network Theory

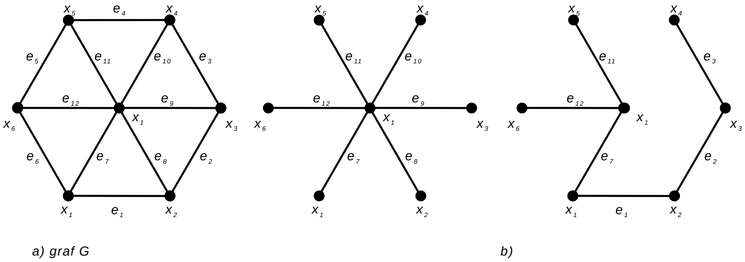
Definition of spanning tree

A tree T is said to be a **spanning tree** of a connected graph G, if it is a subgraph of G ($T \subset G$) and includes all the vertices of G i.e. $V_G = V_T$. If a graph G is unconnected then the set of spanning trees (found for each component of G) is called a spanning forest of G.



a) graf G

b)



G has 320 spanning trees.

Theorem

Every connected graph has a spanning tree.

Spanning tree algorithm.

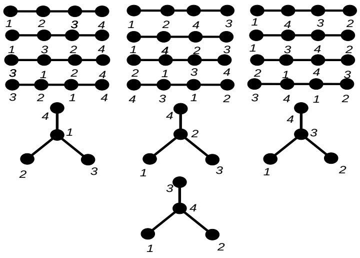
1 $T \leftarrow G$
2 choose a cycle in T
3 delete an arbitrary edge e from this cycle e
4 $T \leftarrow T - \{e\}$
5 if T has a cycle go to step 1
6 if not T is a spanning tree G

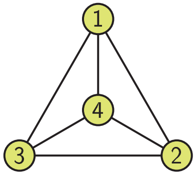
Property

Any spanning tree and any spanning forest of can be obtained by removing a certain number of edges of the graph G and leaving all its vertices.

Theorem

A graph K_n has n^{n-2} spanning trees.





Theorem

A graph $K_{2,s}$ has $s \cdot 2^{s-1}$ spanning trees.

Theorem

Let G be a connected simple graph with n vertices and let $M = [m_{ij}]$ be a $n \times n$ matrix where $m_{ii} = deg(v_i)$, $m_{ij} = -1$, if v_i i v_j are adjacent and $m_{ij} = 0$ otherwise. Then the number of spanning trees in G is equal to a cofactor of any entry of M.

$$M = \begin{bmatrix} 2 & -1 & 0 & -1 \\ -1 & 3 & -1 & -1 \\ 0 & -1 & 2 & -1 \\ -1 & -1 & -1 & 3 \end{bmatrix}$$

The cofactor m_{44} equals to

$$(-1)^{4+4} \det \begin{bmatrix} 2 & -1 & 0 \\ -1 & 3 & -1 \\ 0 & -1 & 2 \end{bmatrix} = 12 - (2+2) = 8$$

Theorem

Let G be a connected simple graph with n vertices and let $M = [m_{ij}]$ be a $n \times n$ matrix where $m_{ii} = deg(v_i)$, $m_{ij} = -1$, if v_i i v_j are adjacent and $m_{ij} = 0$ otherwise. Then the number of spanning trees in G is equal to a cofactor of any entry of M.

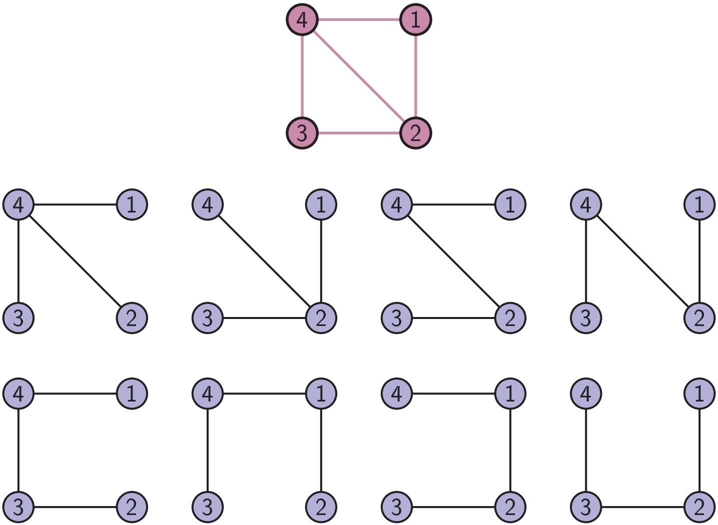
$$M = \begin{bmatrix} 2 & -1 & 0 & -1 \\ -1 & 3 & -1 & -1 \\ 0 & -1 & 2 & -1 \\ -1 & -1 & -1 & 3 \end{bmatrix}$$

We call the matrix of **L** the Laplace's matrix.

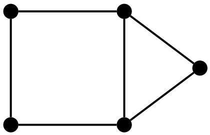
There are 8 spanning trees.

The cofactor m_{44} equals to

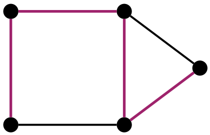
$$(-1)^{4+4} \det \begin{bmatrix} 2 & -1 & 0 \\ -1 & 3 & -1 \\ 0 & -1 & 2 \end{bmatrix} = 12 - (2+2) = 8$$


4
1
3
2
4
1
3
2
4
1
3
2
4
1
3
2
4
1
3
2
4
1
3
2
4
1
3
2
4
1
3
2
4
1
3
2

Let T be the spinning tree of the graph $G = (V, E)$. If e is an edge that **does not belong** to the T, then adding it to the T tree, creates exactly one cycle $C_T(e)$. Such a cycle is called a **cycle fundamental to the T tree**.

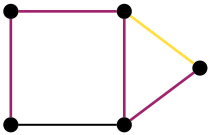


Let T be the spinning tree of the graph $G = (V, E)$. If e is an edge that **does not belong** to the T, then adding it to the T tree, creates exactly one cycle $C_T(e)$. Such a cycle is called a **cycle fundamental to the T tree**.



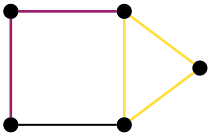
- the spanning tree

Let T be the spinning tree of the graph $G = (V, E)$. If e is an edge that **does not belong** to the T, then adding it to the T tree, creates exactly one cycle $C_T(e)$. Such a cycle is called a **cycle fundamental to the T tree**.



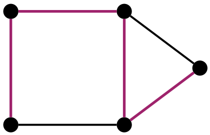
- fundamental cycle -(1)

Let T be the spinning tree of the graph $G = (V, E)$. If e is an edge that **does not belong** to the T, then adding it to the T tree, creates exactly one cycle $C_T(e)$. Such a cycle is called a **cycle fundamental to the T tree**.

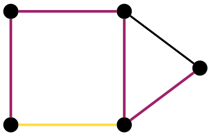


- fundamental cycle -(1)

Let T be the spinning tree of the graph $G = (V, E)$. If e is an edge that **does not belong** to the T, then adding it to the T tree, creates exactly one cycle $C_T(e)$. Such a cycle is called a **cycle fundamental to the T tree**.

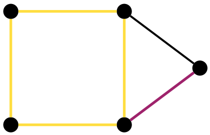


Let T be the spinning tree of the graph $G = (V, E)$. If e is an edge that **does not belong** to the T, then adding it to the T tree, creates exactly one cycle $C_T(e)$. Such a cycle is called a **cycle fundamental to the T tree**.



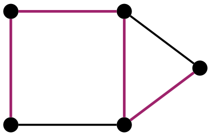
- fundamental cycle -(2)

Let T be the spinning tree of the graph $G = (V, E)$. If e is an edge that **does not belong** to the T, then adding it to the T tree, creates exactly one cycle $C_T(e)$. Such a cycle is called a **cycle fundamental to the T tree**.



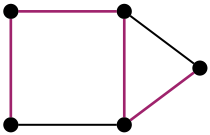
- fundamental cycle -(2)

Let T be the spinning tree of the graph $G = (V, E)$. If e is an edge that **does not belong** to the T, then adding it to the T tree, creates exactly one cycle $C_T(e)$. Such a cycle is called a **cycle fundamental to the T tree**.



- for G, we have $n = 5$, $m = 6$ and $\lambda(G) = 6 - 5 + 1 = 2$

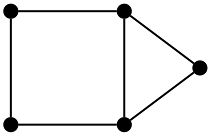
Let T be the spinning tree of the graph $G = (V, E)$. If e is an edge that **does not belong** to the T, then adding it to the T tree, creates exactly one cycle $C_T(e)$. Such a cycle is called a **cycle fundamental to the T tree**.



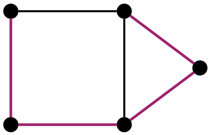
- for G, we have $n = 5$, $m = 6$ and $\lambda(G) = 6 - 5 + 1 = 2$

In the connected graph G we have $\lambda(G) = m - n + 1$ fundamental cycles.

Other spinning tree generates other fundamental cycles.

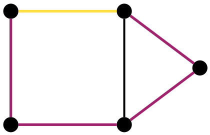


Other spinning tree generates other fundamental cycles.



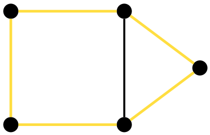
- the spanning tree

Other spinning tree generates other fundamental cycles.



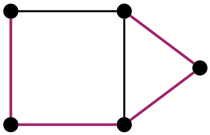
- fundamental cycle -(1)

Other spinning tree generates other fundamental cycles.

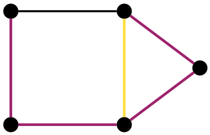


- fundamental cycle -(1)

Other spinning tree generates other fundamental cycles.

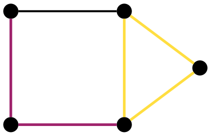


Other spinning tree generates other fundamental cycles.



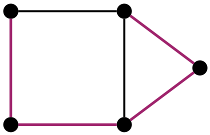
- fundamental cycle -(2)

Other spinning tree generates other fundamental cycles.



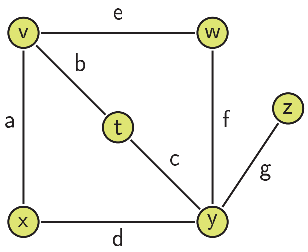
- fundamental cycle -(2)

Other spinning tree generates other fundamental cycles.

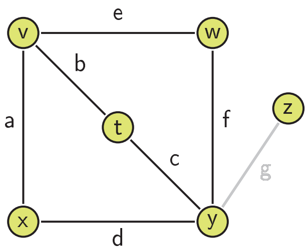


- for G, we have $n = 5$, $m = 6$ and $\lambda(G) = 6 - 5 + 1 = 2$

The **cut** P in a connected graph $G = (V, E)$ we call any set of of edges for which the graph $G \backslash P$ is disconnected, and at the same time for any of the proper subsets A of the set P, $A \subset P$, the graph $G \backslash A$ is conected.

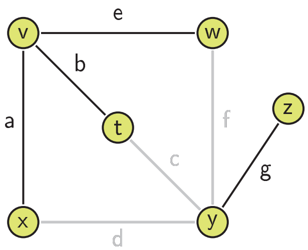


The **cut** P in a connected graph $G = (V, E)$ we call any set of of edges for which the graph $G \backslash P$ is disconnected, and at the same time for any of the proper subsets A of the set P, $A \subset P$, the graph $G \backslash A$ is conected.



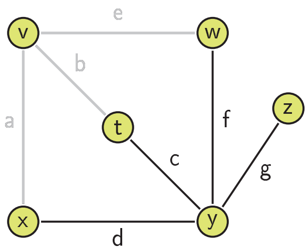
- each of the sets of edges $\{g\}$, $\{d, c, f\}$, $\{a, b, e\}$, $\{e, b, d\}$ is a cut;

The **cut** P in a connected graph $G = (V, E)$ we call any set of of edges for which the graph $G \backslash P$ is disconnected, and at the same time for any of the proper subsets A of the set P, $A \subset P$, the graph $G \backslash A$ is conected.



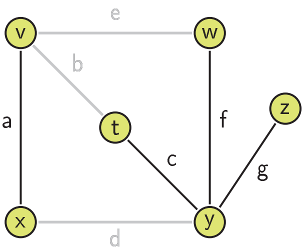
- each of the sets of edges $\{g\}$, $\{d, c, f\}$, $\{a, b, e\}$, $\{e, b, d\}$ is a cut;

The **cut** P in a connected graph $G = (V, E)$ we call any set of of edges for which the graph $G \backslash P$ is disconnected, and at the same time for any of the proper subsets A of the set P, $A \subset P$, the graph $G \backslash A$ is conected.



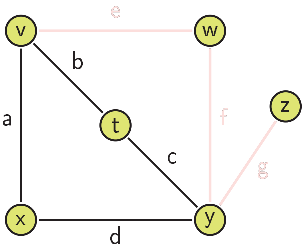
- each of the sets of edges $\{g\}$, $\{d, c, f\}$, $\{a, b, e\}$, $\{e, b, d\}$ is a cut;

The **cut** P in a connected graph $G = (V, E)$ we call any set of of edges for which the graph $G \backslash P$ is disconnected, and at the same time for any of the proper subsets A of the set P, $A \subset P$, the graph $G \backslash A$ is conected.



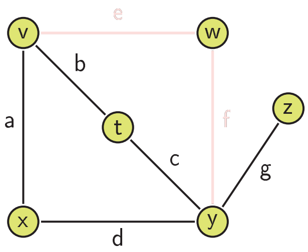
- each of the sets of edges $\{g\}$, $\{d, c, f\}$, $\{a, b, e\}$, $\{e, b, d\}$ is a cut;

The **cut** P in a connected graph $G = (V, E)$ we call any set of of edges for which the graph $G \backslash P$ is disconnected, and at the same time for any of the proper subsets A of the set P, $A \subset P$, the graph $G \backslash A$ is conected.



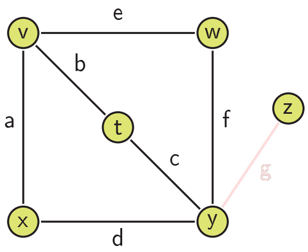
- each of the sets of edges $\{g\}$, $\{d, c, f\}$, $\{a, b, e\}$, $\{e, b, d\}$ is a cut;
- the set $\{e, f, g\}$ is not a cut, because proper subsets: $\{e, f\}$ and $\{g\}$ are cuts.

The **cut** P in a connected graph $G = (V, E)$ we call any set of of edges for which the graph $G \backslash P$ is disconnected, and at the same time for any of the proper subsets A of the set P, $A \subset P$, the graph $G \backslash A$ is conected.



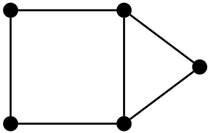
- each of the sets of edges $\{g\}$, $\{d, c, f\}$, $\{a, b, e\}$, $\{e, b, d\}$ is a cut;
- the set $\{e, f, g\}$ is not a cut, because proper subsets: $\{e, f\}$ and $\{g\}$ are cuts.

The **cut** P in a connected graph $G = (V, E)$ we call any set of of edges for which the graph $G \backslash P$ is disconnected, and at the same time for any of the proper subsets A of the set P, $A \subset P$, the graph $G \backslash A$ is conected.

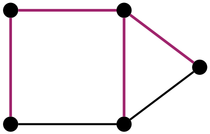


- each of the sets of edges $\{g\}$, $\{d, c, f\}$, $\{a, b, e\}$, $\{e, b, d\}$ is a cut;
- the set $\{e, f, g\}$ is not a cut, because proper subsets: $\{e, f\}$ and $\{g\}$ are cuts.

Let T be the spanning tree in graph $G = (V, E)$. For every edge $e \in E_T$ there exists exactly one cut $S_T(e)$ in G, such that e is a unique tree-edge in $E(S_T(e))$. Such cut is called **fundamental cut of $S_T(e)$ relative to the tree T**.

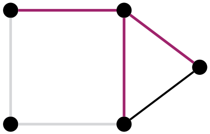


Let T be the spanning tree in graph $G = (V, E)$. For every edge $e \in E_T$ there exists exactly one cut $S_T(e)$ in G, such that e is a unique tree-edge in $E(S_T(e))$. Such cut is called **fundamental cut of $S_T(e)$ relative to the tree T**.



In G we have 4 fundamental cuts.

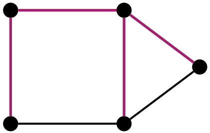
Let T be the spanning tree in graph $G = (V, E)$. For every edge $e \in E_T$ there exists exactly one cut $S_T(e)$ in G, such that e is a unique tree-edge in $E(S_T(e))$. Such cut is called **fundamental cut of $S_T(e)$ relative to the tree T**.



In G we have 4 fundamental cuts.

- the fundamental cut (1)

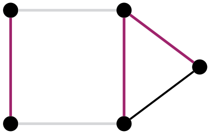
Let T be the spanning tree in graph $G = (V, E)$. For every edge $e \in E_T$ there exists exactly one cut $S_T(e)$ in G, such that e is a unique tree-edge in $E(S_T(e))$. Such cut is called **fundamental cut of $S_T(e)$ relative to the tree T**.



In G we have 4 fundamental cuts.

- the fundamental cut (1)

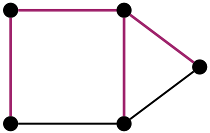
Let T be the spanning tree in graph $G = (V, E)$. For every edge $e \in E_T$ there exists exactly one cut $S_T(e)$ in G, such that e is a unique tree-edge in $E(S_T(e))$. Such cut is called **fundamental cut of $S_T(e)$ relative to the tree T**.



In G we have 4 fundamental cuts.

- the fundamental cut(2)

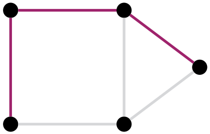
Let T be the spanning tree in graph $G = (V, E)$. For every edge $e \in E_T$ there exists exactly one cut $S_T(e)$ in G, such that e is a unique tree-edge in $E(S_T(e))$. Such cut is called **fundamental cut of $S_T(e)$ relative to the tree T.**



In G we have 4 fundamental cuts.

- the fundamental cut(2)

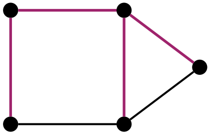
Let T be the spanning tree in graph $G = (V, E)$. For every edge $e \in E_T$ there exists exactly one cut $S_T(e)$ in G, such that e is a unique tree-edge in $E(S_T(e))$. Such cut is called **fundamental cut of $S_T(e)$ relative to the tree T**.



In G we have 4 fundamental cuts.

- the fundamental cut (3)

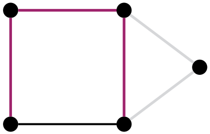
Let T be the spanning tree in graph $G = (V, E)$. For every edge $e \in E_T$ there exists exactly one cut $S_T(e)$ in G, such that e is a unique tree-edge in $E(S_T(e))$. Such cut is called **fundamental cut of $S_T(e)$ relative to the tree T**.



In G we have 4 fundamental cuts.

- the fundamental cut (3)

Let T be the spanning tree in graph $G = (V, E)$. For every edge $e \in E_T$ there exists exactly one cut $S_T(e)$ in G, such that e is a unique tree-edge in $E(S_T(e))$. Such cut is called **fundamental cut of $S_T(e)$ relative to the tree T**.

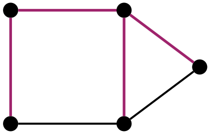


In G we have 4 fundamental cuts.

- the fundamental cut (4)

Let T be the spanning tree in graph $G = (V, E)$. For every edge $e \in E_T$ there exists exactly one cut $S_T(e)$ in G, such that e is a unique tree-edge in $E(S_T(e))$. Such cut is called **fundamental cut of $S_T(e)$ relative to the tree T**.

In G we have 4 fundamental cuts.



In connected graph G with n vertices we have $n - 1$ fundamental cuts.

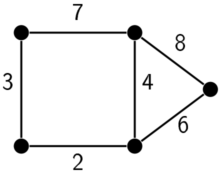
Let $G = (V, E)$ and let $T = (V, E_T)$ be a a spanning tree of G.

A **weight** of a tree T is the number

$$w(T) := \sum_{e \in E_T} w(e)$$

A spanning tree T is said to be **minimum spanning tree – MST** of a graph G, if T is minimal in the set of spanning trees of G:

$$w(T) \to min$$



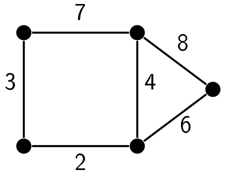
Let $G = (V, E)$ and let $T = (V, E_T)$ be a a spanning tree of G.

A **weight** of a tree T is the number

$$w(T) := \sum_{e \in E_T} w(e)$$

A spanning tree T is said to be **minimum spanning tree – MST** of a graph G, if T is minimal in the set of spanning trees of G:

$$w(T) \to min$$



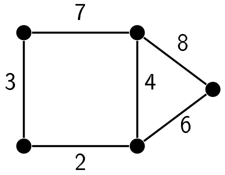
Let $G = (V, E)$ and let $T = (V, E_T)$ be a a spanning tree of G.

A **weight** of a tree T is the number

$$w(T) := \sum_{e \in E_T} w(e)$$

A spanning tree T is said to be **minimum spanning tree – MST** of a graph G, if T is minimal in the set of spanning trees of G:

$$w(T) \to min$$



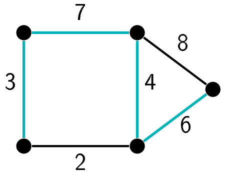
Let $G = (V, E)$ and let $T = (V, E_T)$ be a a spanning tree of G.

A **weight** of a tree T is the number

$$w(T) := \sum_{e \in E_T} w(e)$$

A spanning tree T is said to be **minimum spanning tree – MST** of a graph G, if T is minimal in the set of spanning trees of G:

$$w(T) \to min$$



$w(T) = 20$

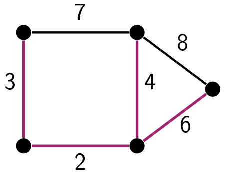
Let $G = (V, E)$ and let $T = (V, E_T)$ be a a spanning tree of G.

A **weight** of a tree T is the number

$$w(T) := \sum_{e \in E_T} w(e)$$

A spanning tree T is said to be **minimum spanning tree – MST** of a graph G, if T is minimal in the set of spanning trees of G:

$$w(T) \to min$$



$w(T) = 20$

$w(T) = 15$

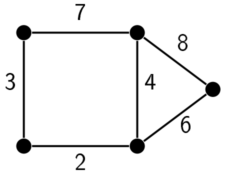
Let $G = (V, E)$ and let $T = (V, E_T)$ be a a spanning tree of G.

A **weight** of a tree T is the number

$$w(T) := \sum_{e \in E_T} w(e)$$

A spanning tree T is said to be **minimum spanning tree – MST** of a graph G, if T is minimal in the set of spanning trees of G:

$$w(T) \to min$$



$w(T) = 20$

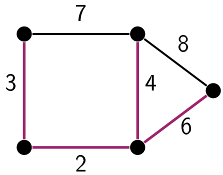
$w(T) = 15$

The spinning tree T is minimal if and only if, for each edge $e \in G \setminus T$

$$w(e) \geq w(f) \text{ for any edge } f \in C_T(e)$$

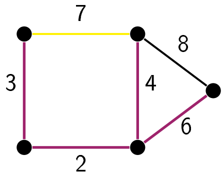
The spinning tree T is minimal if and only if, for each edge $e \in G \setminus T$

$$w(e) \geq w(f) \text{ for any edge } f \in C_T(e)$$



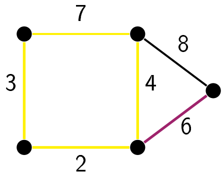
The spinning tree T is minimal if and only if, for each edge $e \in G \setminus T$

$$w(e) \geq w(f) \text{ for any edge } f \in C_T(e)$$



The spinning tree T is minimal if and only if, for each edge $e \in G \setminus T$

$$w(e) \geq w(f) \text{ for any edge } f \in C_T(e)$$

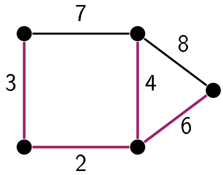


The spinning tree T is minimal if and only if for each edge $e \in T$

$$w(e) \leq w(f) \text{ for any edge } f \in E(S_T(e))$$

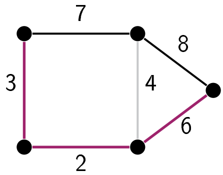
The spinning tree T is minimal if and only if for each edge $e \in T$

$$w(e) \leq w(f) \text{ for any edge } f \in E(S_T(e))$$



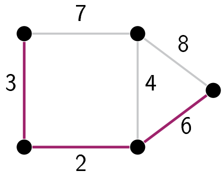
The spinning tree T is minimal if and only if for each edge $e \in T$

$$w(e) \leq w(f) \text{ for any edge } f \in E(S_T(e))$$



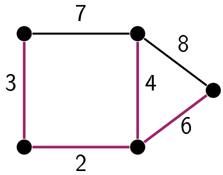
The spinning tree T is minimal if and only if for each edge $e \in T$

$$w(e) \leq w(f) \text{ for any edge } f \in E(S_T(e))$$



The spinning tree T is minimal if and only if for each edge $e \in T$

$$w(e) \leq w(f) \text{ for any edge } f \in E(S_T(e))$$

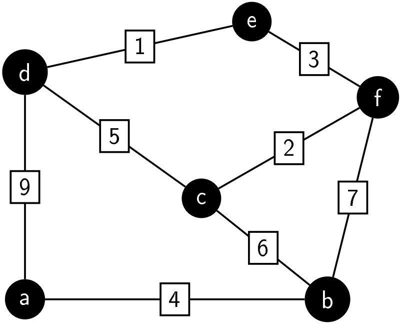


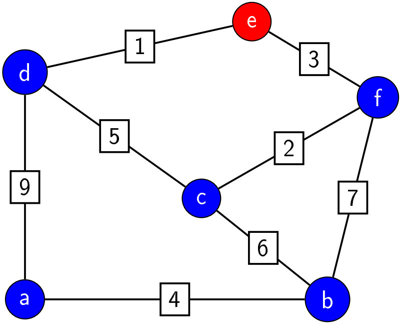
- **1 strategy: Prim's Algorithm**
- **2 strategy: Kruskal's Algorithm**
- **3 strategy: Boruvka's Algorithm**

Boruvka's Algorithm

Boruvka MST(G)

$A \leftarrow \varnothing$
for $v \in V$
    do
        make one-vertex trees and add to L
while $|A| < n - 1$
    do
        for every tree $T \in L$
            do
                find e_T with minimum weight s.t. it connects T with $G \setminus T$
                let T' – a tree from L connected by e_T with T
                $A \leftarrow A \cup \{e_T\}$
        for every $T \in L$ connect T i T'

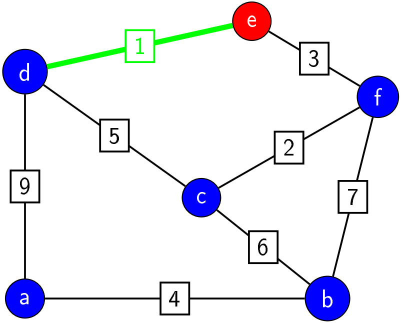

e
1
3
d
f
5
2
9
c
7
6
a
4
b



Cut:

$S = \{a, b, c, d, f\}$ $V \setminus S = \{e\}$

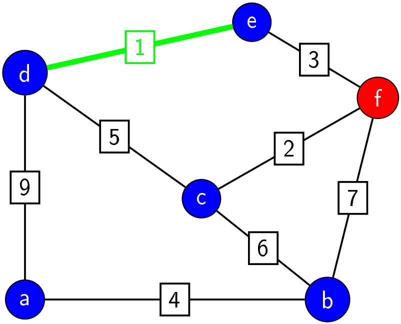
$A = \emptyset$



Cut:

$S = \{a, b, c, d, f\}$ $V \setminus S = \{e\}$

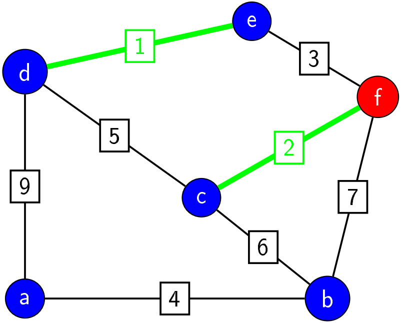
$A = \{\{d, e\}\}$



Cut:

$S = \{a, b, c, d, e\}$ $V \setminus S = \{f\}$

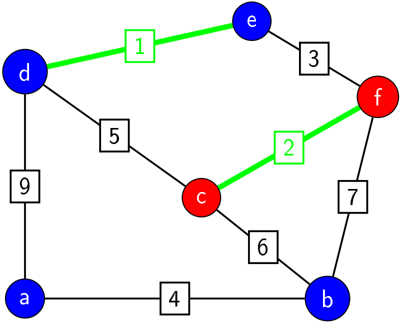
$A = \{\{d, e\}\}$



Cut:

$S = \{a, b, c, d, e\}$ $V \setminus S = \{f\}$

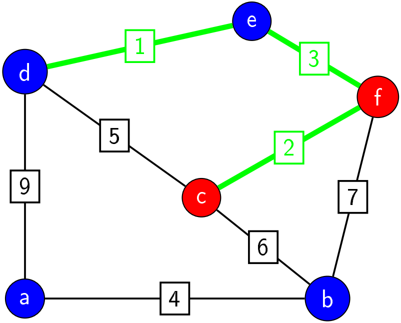
$A = \{\{d, e\}, \{c, f\}\}$



Cut:

$S = \{a, b, d, e\}$ $\quad V \setminus S = \{f, c\}$

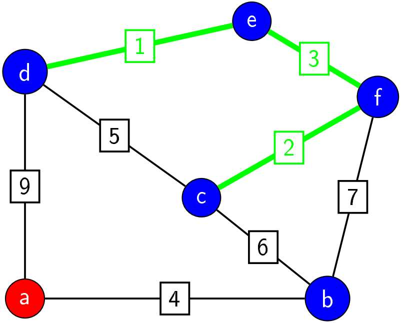
$A = \{\{d, e\}, \{c, f\}\}$



Cut:

$S = \{a, b, d, e\}$ $\quad$ $V \setminus S = \{f, c\}$

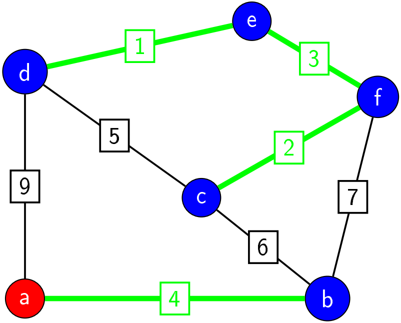
$A = \{\{d, e\}, \{c, f\}, \{e, f\}\}$



Cut:

$S = \{b, c, d, e, f\}$ $\quad V \setminus S = \{a\}$

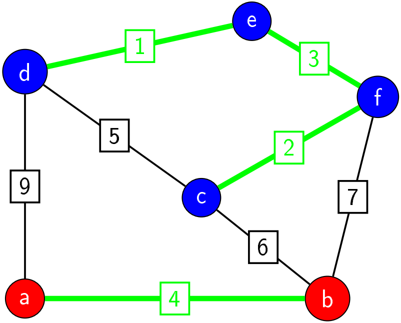
$A = \{\{d, e\}, \{c, f\}, \{e, f\}\}$



Cut:

$S = \{b, c, d, e, f\}$ $\quad V \setminus S = \{a\}$

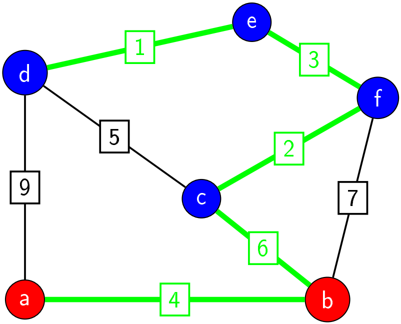
$A = \{\{d, e\}, \{c, f\}, \{e, f\}, \{a, b\}\}$



Cut:

$S = \{a, b, c, d, e, f\}$ $V \setminus S = \{a, b\}$

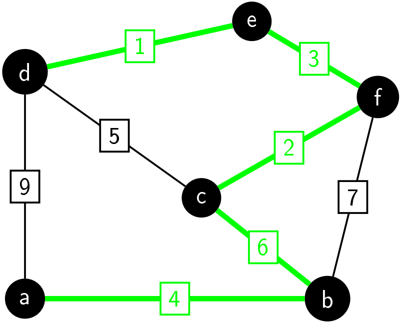
$A = \{\{d, e\}, \{c, f\}, \{e, f\}, \{a, b\}\}$



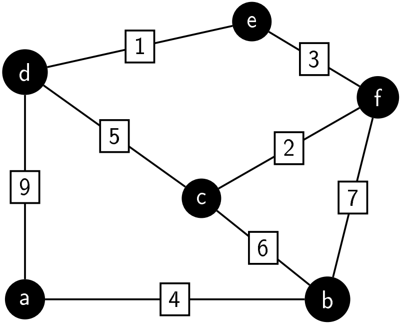
Cut:

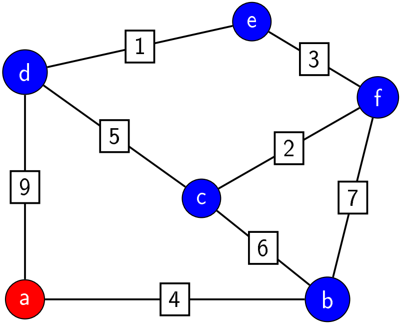
$S = \{a, b, c, d, e, f\}$ $V \setminus S = \{a, b\}$

$A = \{\{d, e\}, \{c, f\}, \{e, f\}, \{a, b\}, \{c, b\}\}$



$A = \{\{d, e\}, \{c, f\}, \{e, f\}, \{a, b\}, \{c, b\}\}$

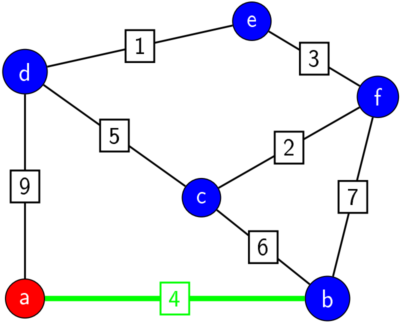

e
1
3
d
f
5
2
9
c
7
6
a
4
b



Cut:

$S = \{b, c, d, e, f\}$ $\quad$ $V \setminus S = \{a\}$

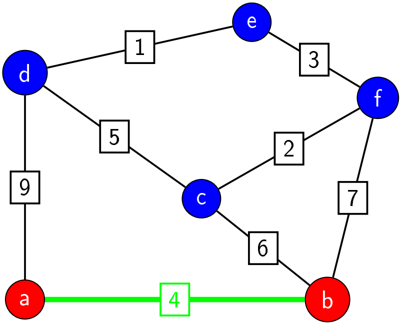
$A = \emptyset$



Cut:

$S = \{b, c, d, e, f\}$ $V \setminus S = \{a\}$

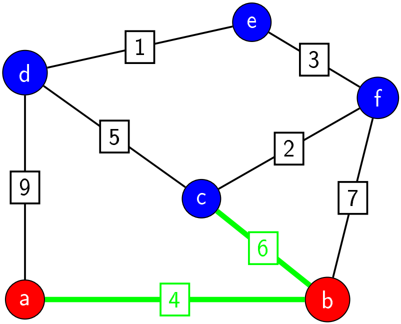
$A = \{\{a, b\}\}$



Cut:

$S = \{c, d, e, f\}$ $\quad$ $V \setminus S = \{a, b\}$

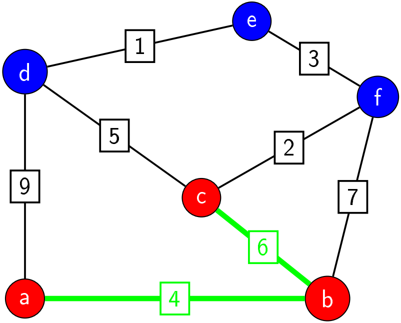
$A = \{\{a, b\}\}$



Cut:

$S = \{c, d, e, f\}$ $V \setminus S = \{a, b\}$

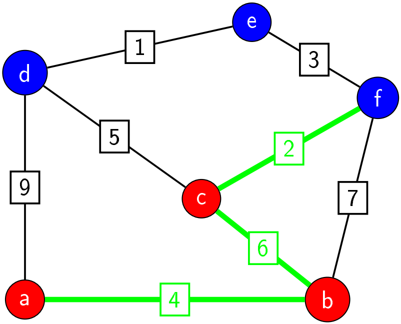
$A = \{\{a, b\}, \{b, c\}\}$



Cut:

$S = \{d, e, f\}$ $\quad$ $V \setminus S = \{a, b, c\}$

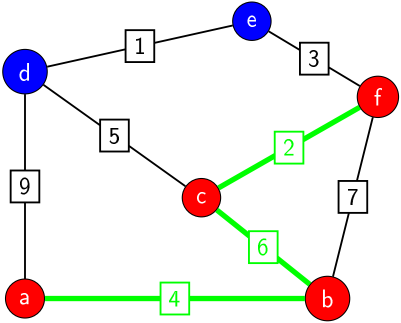
$A = \{\{a, b\}, \{b, c\}\}$



Cut:

$S = \{d, e, f\}$ $\quad$ $V \setminus S = \{a, b, c\}$

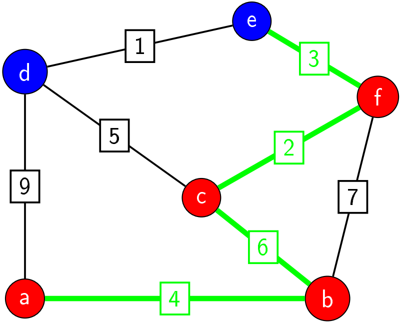
$A = \{\{a, b\}, \{b, c\}, \{c, f\}\}$



Cut:

$S = \{d, e\}$ $\quad V \setminus S = \{a, b, c, f\}$

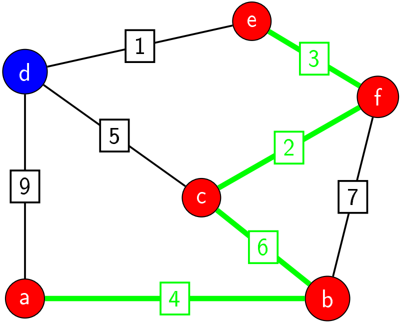
$A = \{\{a, b\}, \{b, c\}, \{c, f\}\}$



Cut:

$S = \{d, e\}$ $\quad V \setminus S = \{a, b, c, f\}$

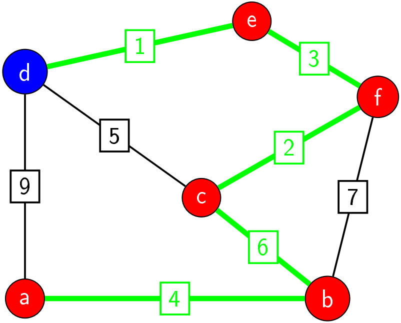
$A = \{\{a, b\}, \{b, c\}, \{c, f\}, \{e, f\}\}$



Cut:

$S = \{d\}$ $\quad V \setminus S = \{a, b, c, e, f\}$

$A = \{\{a, b\}, \{b, c\}, \{c, f\}, \{e, f\}\}$



Cut:

$S = \{d\}$ $\quad V \setminus S = \{a, b, c, e, f\}$

$A = \{\{a, b\}, \{b, c\}, \{c, f\}, \{e, f\}, \{d, e\}\}$

Maximum Spanning tree is such a spanning tree whose weight is the highest among all spanning trees of a graph $G = (V, E, w)$.

$$w(T) \to max$$

Maximum Spanning tree is such a spanning tree whose weight is the highest among all spanning trees of a graph $G = (V, E, w)$.

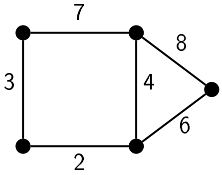
$$w(T) \to max$$

A spanning tree T is maximal for $G = (V, E, w)$ if and only if it is minimal for the graph $G = (V, E, -w)$.

Maximum Spanning tree is such a spanning tree whose weight is the highest among all spanning trees of a graph $G = (V, E, w)$.

$$w(T) \to max$$

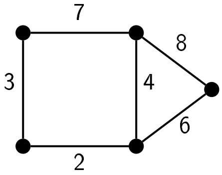
A spanning tree T is maximal for $G = (V, E, w)$ if and only if it is minimal for the graph $G = (V, E, -w)$.

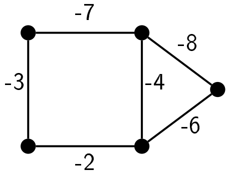


Maximum Spanning tree is such a spanning tree whose weight is the highest among all spanning trees of a graph $G = (V, E, w)$.

$$w(T) \to max$$

A spanning tree T is maximal for $G = (V, E, w)$ if and only if it is minimal for the graph $G = (V, E, -w)$.

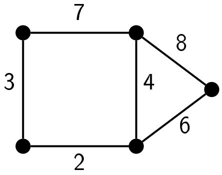


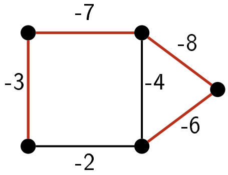


Maximum Spanning tree is such a spanning tree whose weight is the highest among all spanning trees of a graph $G = (V, E, w)$.

$$w(T) \to max$$

A spanning tree T is maximal for $G = (V, E, w)$ if and only if it is minimal for the graph $G = (V, E, -w)$.

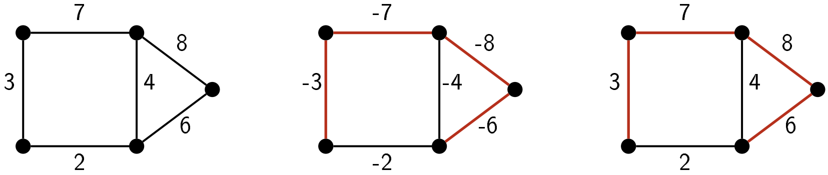




Maximum Spanning tree is such a spanning tree whose weight is the highest among all spanning trees of a graph $G = (V, E, w)$.

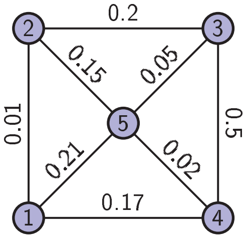
$$w(T) \to max$$

A spanning tree T is maximal for $G = (V, E, w)$ if and only if it is minimal for the graph $G = (V, E, -w)$.

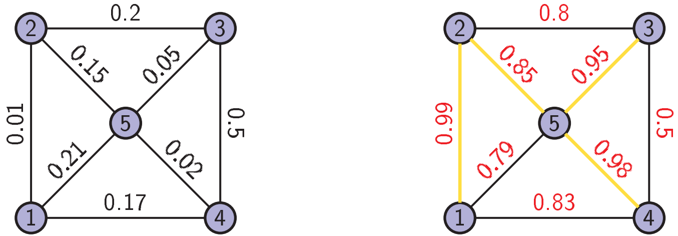


Let us define a graph, whose vertices V correspond to the nodes of some communication network and edges E are connections between these nodes. Let p_{ij} determines the probability that the connection between the nodes i and j fails. Then $q_{ij} = 1 - p_{ij}$ determines the probability of the correct working. **The maximum spanning tree of (V, E, q) determines the highest probability of failure-free functioning of the network.**

Let us define a graph, whose vertices V correspond to the nodes of some communication network and edges E are connections between these nodes. Let p_{ij} determines the probability that the connection between the nodes i and j fails. Then $q_{ij} = 1 - p_{ij}$ determines the probability of the correct working. **The maximum spanning tree of (V, E, q) determines the highest probability of failure-free functioning of the network.**



Let us define a graph, whose vertices V correspond to the nodes of some communication network and edges E are connections between these nodes. Let p_{ij} determines the probability that the connection between the nodes i and j fails. Then $q_{ij} = 1 - p_{ij}$ determines the probability of the correct working. **The maximum spanning tree of (V, E, q) determines the highest probability of failure-free functioning of the network.**



Let $G = (V, E, w)$ be a connected network and let

$$W = v_0 \overset{e_1}{-} v_1 \overset{e_2}{-} v_2 \ldots v_{n-1} \overset{e_n}{-} v_n$$

be any path with different vertices. Then

$$c(W) = \min\{w(e_i); i = 1, ..., n\}$$

is called a **capacity of path W**.

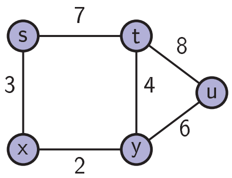
Let $G = (V, E, w)$ be a connected network and let

$$W = v_0 \overset{e_1}{-} v_1 \overset{e_2}{-} v_2 \ldots v_{n-1} \overset{e_n}{-} v_n$$

be any path with different vertices. Then

$$c(W) = \min\{w(e_i); i = 1, ..., n\}$$

is called a **capacity of path** W.



path $W = (t, u, y)$

$$c(W) = \min\{w(t, u), w(u, y)\} = 6$$

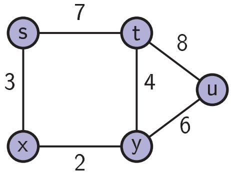
Let $G = (V, E, w)$ be a connected network and let

$$W = v_0 \overset{e_1}{-} v_1 \overset{e_2}{-} v_2 \dots v_{n-1} \overset{e_n}{-} v_n$$

be any path with different vertices. Then

$$c(W) = \min\{w(e_i); i = 1, ..., n\}$$

is called a **capacity of path W**.



path $W = (t, u, y)$

$$c(W) = \min\{w(t,u), w(u,y)\} = 6$$

path $W' = (t, s, x, y)$

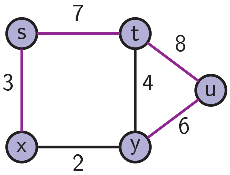
$$c(W') = \min\{7, 3, 2\} = 2$$

Theorem

Let $G = (V, E, w)$ be a connected network and let T be a maximum spanning tree in G. Then for any vertices u and v, uniquely defined path in the tree T, has the largest capacity in the graph G.

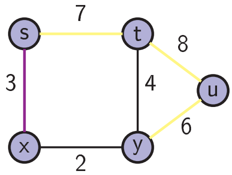
Theorem

Let $G = (V, E, w)$ be a connected network and let T be a maximum spanning tree in G. Then for any vertices u and v, uniquely defined path in the tree T, has the largest capacity in the graph G.



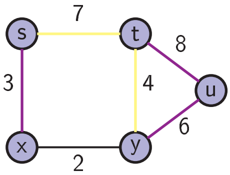
Theorem

Let $G = (V, E, w)$ be a connected network and let T be a maximum spanning tree in G. Then for any vertices u and v, uniquely defined path in the tree T, has the largest capacity in the graph G.



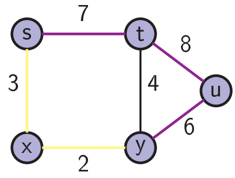
Theorem

Let $G = (V, E, w)$ be a connected network and let T be a maximum spanning tree in G. Then for any vertices u and v, uniquely defined path in the tree T, has the largest capacity in the graph G.

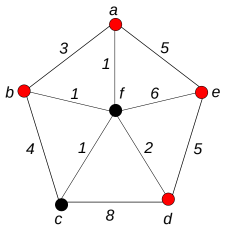


Theorem

Let $G = (V, E, w)$ be a connected network and let T be a maximum spanning tree in G. Then for any vertices u and v, uniquely defined path in the tree T, has the largest capacity in the graph G.



Let $G = (V, E, w)$ be a network with the weight function $w : E \to \mathbb{R}_+$ and let $N \subset V$ be a given subset of vertices, whose members are called **terminals**.

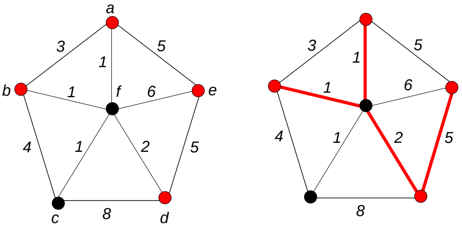


A tree $T = (V_T, E_T)$ being a subgraph of G is called the **Steiner tree** for G and a given set of terminals N, if $N \subset V_T$. Vertices from the set $V_T \setminus N$ are called **Steiner's points**.

Minimum Steiner tree is such a tree that

$$\sum_{(i,j)\in E_T} w_{ij} \to min$$

(shortly: SMT)



Theorem

The SMT problem is NP-complete.

Remark

We have two special cases of Steiner tree:

1. if $N = V$, then the Steiner tree us just a MST (we use Kruskal's or Prim's)
2. if $|N| = 2$ (the set of terminals consists of two vertices, then the solution to the problem is reduced to determining of the shortest path between the vertices (eg. Dijkstra's algorithm)

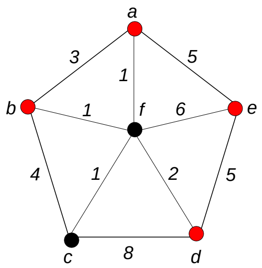
The Exact Algorithm for SMT

Remark

If N is a set of terminals, and T is maximum Steiner tree for a graph G, then T is a minimum spanning tree for a subgraph of G which induced by the set N and the Steiner points of T.

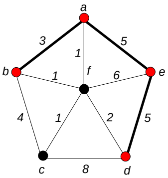
Inputs: a graph G, a set of terminals N
Otputs: minimum Steiner tree T with a sum of weights equal to w

```
SMT(G, N)
w ← 0
for every subset A ⊂ V \ N
      do
           find minimum spanning tree T_1 for the induced grap G[A ∪ N]
           w_1 - the sum of weights of edges of T_1
           w ← min(w, w_1)
```

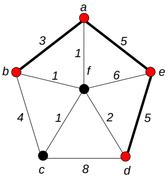


- for the graph from the picture, we will find all subsets of the set of **Steiner points**
- $A \subset V \setminus N$

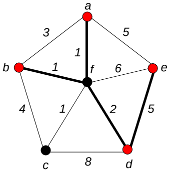
If $A = \emptyset$, then the graph is induced on the set of terminals $G[N]$. **Sum of weights** $w = 13$.



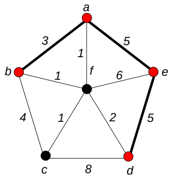
If $A = \emptyset$, then the graph is induced on the set of terminals $G[N]$. **Sum of weights** $w = 13$.



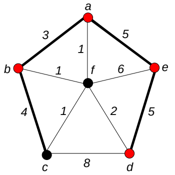
If $A = \{f\}$, then the graph is induced on the set of terminals $G[N \cup \{f\}]$. **Sum of weights** $w = 9$.



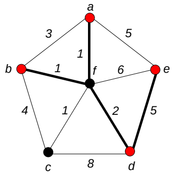
If $A = \emptyset$, then the graph is induced on the set of terminals $G[N]$. **Sum of weights $w = 13$.**



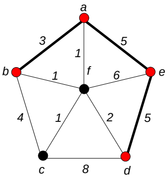
If $A = \{c\}$, then the graph is induced on the set of terminals $G[N \cup \{c\}]$. **Sum of weights $w = 17$.**



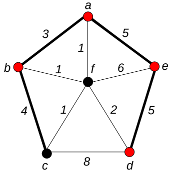
If $A = \{f\}$, then the graph is induced on the set of terminals $G[N \cup \{f\}]$. **Sum of weights $w = 9$.**



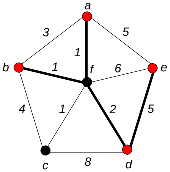
If $A = \emptyset$, then the graph is induced on the set of terminals $G[N]$. **Sum of weights $w = 13$.**



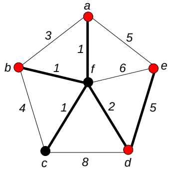
If $A = \{c\}$, then the graph is induced on the set of terminals $G[N \cup \{c\}]$. **Sum of weights $w = 17$.**



If $A = \{f\}$, then the graph is induced on the set of terminals $G[N \cup \{f\}]$. **Sum of weights $w = 9$.**



If $A = \{c, f\}$, then the graph is induced on the set of terminals $G[N \cup \{c, f\}]$. **Sum of weights $w = 10$.**



Unicast, multicast, and broadcast are three transmission modes of IP packets.

Multicast

- **Unicast** is a one-to-one communication mode between hosts. In this mode, each intermediate device selects a transmission path according to the destination address included in each received packet, and forwards the packet accordingly, without copying the packet. Unicast ensures that each host is responded in time.
- **Broadcast** is a one-to-all communication mode among hosts. In this mode, each device copies received broadcast packets and forwards them to all possible receivers on the network through all interfaces except the inbound interface.
- **Multicast** is a one-to-many communication mode among hosts. Multicast allows one or more multicast sources to send the same packet to multiple receivers. A multicast source sends a packet to a specific multicast address. Different from unicast addresses, each multicast address belongs to a group of hosts, not to a specific host. All hosts that need to receive packets from the multicast source must join the group.

- the sender and receivers are treated as terminal nodes.
- the goal is to construct an efficient communication subgraph connecting them.
- a Steiner Tree provides an optimal or near-optimal solution by minimizing:
  - total communication cost (e.g., bandwidth);
  - delay or hop count;

In network design or multicast routing, computing a Steiner tree helps determine the most efficient way to distribute information from one sender to many receivers.

Thank you for your attention!!!