Kruskal's Greedy Algorithm

Jonas Hofmann
Kantonsschule Uster 5th year
Mentor: Kaloyan Slavov
Primes-Switzerland

23.Juni 2018

Problem

You have a graph G and a function $\omega : E(G) \to \mathbb{R}^+$.
Find a spanning-tree T such that the sum of the weight of all edges in T is minimal.

Problem

You have a graph G and a function $\omega : E(G) \to \mathbb{R}^+$.
Find a spanning-tree T such that the sum of the weight of all edges in T is minimal.

- What is a Graph?

Problem

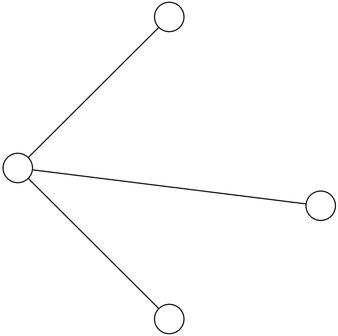
You have a graph G and a function $\omega : E(G) \to \mathbb{R}^+$.
Find a spanning-tree T such that the sum of the weight of all edges in T is minimal.

- What is a Graph?
- What is a Tree?

What is a graph?

What is a graph?

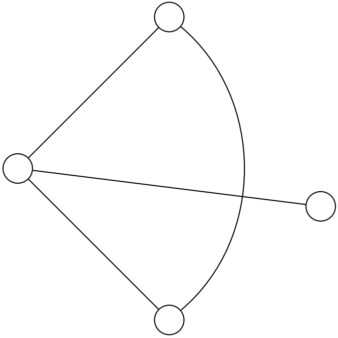
A set of vertices connected by edges



The edges don't have to be straight, and they're allowed to intersect

What is a graph?

A set of vertices connected by edges

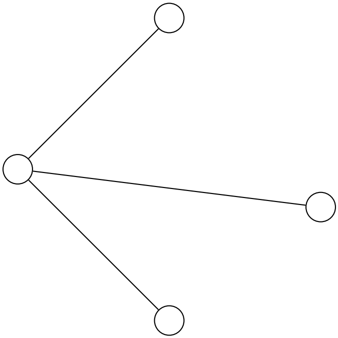


The edges don't have to be straight, and they're allowed to intersect

What is a tree?

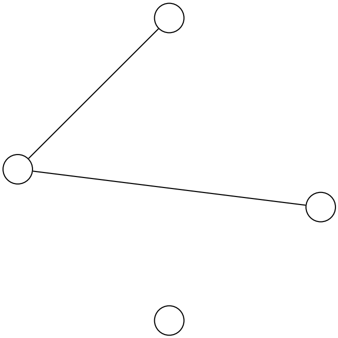
What is a tree?

A connected graph without any cycles



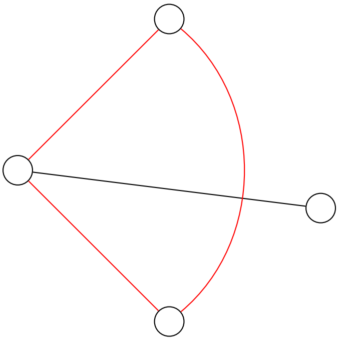
What is a tree?

A connected graph without any cycles



What is a tree?

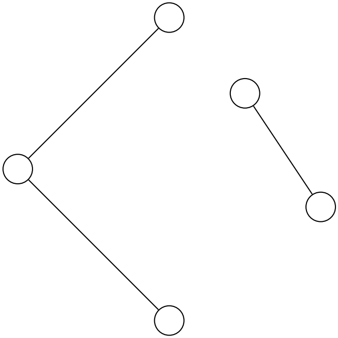
A connected graph without any cycles



What is a forest?

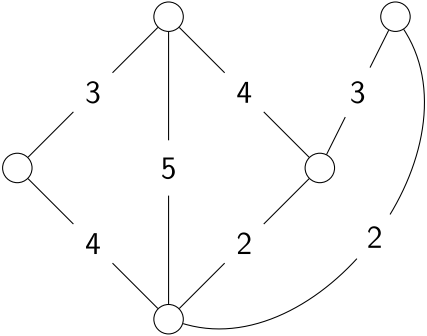
What is a forest?

Like a tree, but it doesn't need to be connected.
$\Rightarrow$ A graph without a cycle



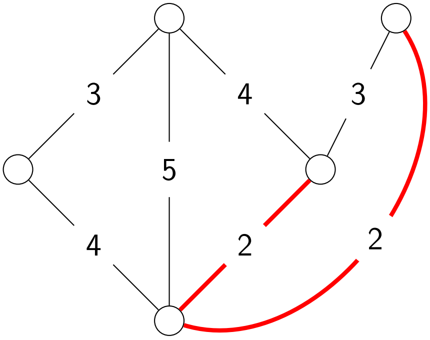
Problem

You have a graph G and a function $\omega : E(G) \to \mathbb{R}^+$.
Find a tree T such that the sum of the weight of all edges in T is minimal.



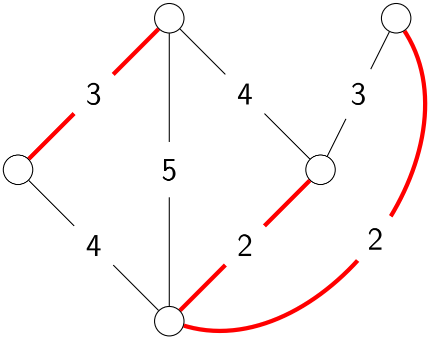
Problem

You have a graph G and a function $\omega : E(G) \to \mathbb{R}^+$.
Find a tree T such that the sum of the weight of all edges in T is minimal.



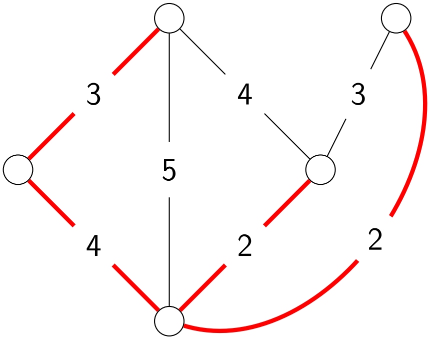
Problem

You have a graph G and a function $\omega : E(G) \to \mathbb{R}^+$.
Find a tree T such that the sum of the weight of all edges in T is minimal.



Problem

You have a graph G and a function $\omega : E(G) \to \mathbb{R}^+$.
Find a tree T such that the sum of the weight of all edges in T is minimal.



That is Kruskal's algorithm

Always add a edge with the lowest weight, which isn't already in the forest and doesn't create a cycle.

Lemma 1

A graph is a tree on n vertices $\Rightarrow$ it has $n-1$ edges

Induction on the number of vertices.

Lemma 1

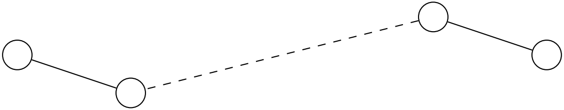
A graph is a tree on n vertices $\Rightarrow$ it has $n-1$ edges

Induction on the number of vertices.
Trivial case: 1 vertex

Lemma 1

A graph is a tree on n vertices $\Rightarrow$ it has $n-1$ edges

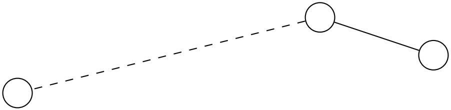
Induction on the number of vertices.
Suppose it holds for n vertices. Let T be a tree on $n+1$ vertices.
Find a longest path.



Lemma 1

A graph is a tree on n vertices $\Rightarrow$ it has $n-1$ edges

Induction on the number of vertices.
Suppose it holds for n vertices. Let T be a graph on $n+1$ vertices.
Find a longest path. Remove one of the vertices where it ends, you get a new tree T'

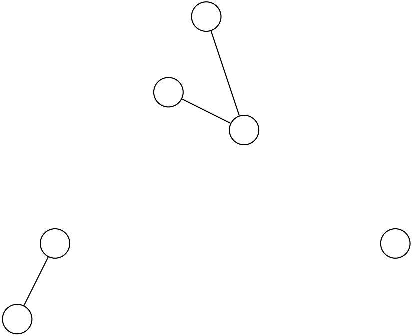


T' is a tree on n vertices $\Rightarrow$ (inductive hypothesis) it has $n-1$ edges. $\Rightarrow T$ has n edges.
$\square$

Lemma 2

A graph is a forest on n vertices with k components $\Rightarrow$ it has $n-k$ edges.

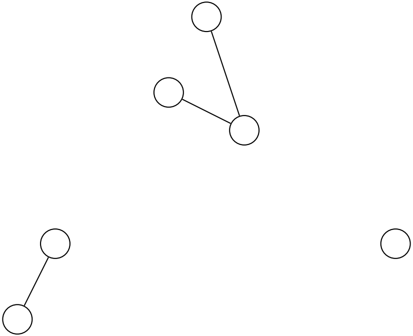
Apply Lemma 1 on all components



Lemma 2

A graph is a forest on n vertices with k components $\Rightarrow$ it has $n-k$ edges.

Apply Lemma 1 on all components

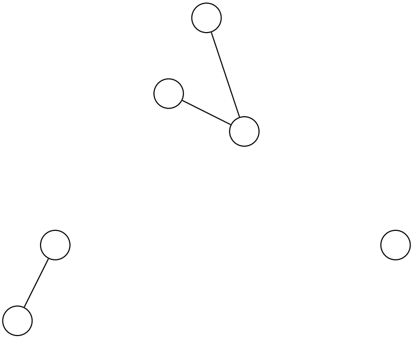


by Lemma 1 : in every component # edges = # vertices - 1

Lemma 2

A graph is a forest on n vertices with k components $\Rightarrow$ it has $n - k$ edges.

Apply Lemma 1 on all components



by Lemma 1 : in every component $\#$ edges $=$ $\#$ vertices - 1

$\Rightarrow$ $\#$ all edges $=$ $\#$ all vertices - k

$\square$

Lemma 3

F, F' are forests on the same n vertices, with $|E(F)| < |E(F')|$.
$\Rightarrow$ There exists $e \in F'$ such that $F \cup e$ is still a forest.

Lemma 3

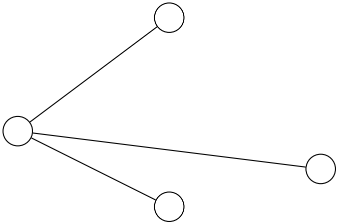
F, F' are forests on the same n vertices, with $|E(F)| < |E(F')|$.
$\Rightarrow$ There exists $e \in F'$ such that $F \cup e$ is still a forest.

If that wasn't the case, adding any $e \in F'$ to F would create a cycle.

Lemma 3

F, F' are forests on the same n vertices, with $|E(F)| < |E(F')|$.
$\Rightarrow$ There exists $e \in F'$ such that $F \cup e$ is still a forest.

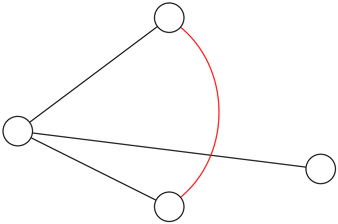
If that wasn't the case, adding any $e \in F'$ to F would create a cycle.
$\Rightarrow$ All $e \in F'$ connect two vertices in a component in F



Lemma 3

F, F' are forests on the same n vertices, with $|E(F)| < |E(F')|$.
$\Rightarrow$ There exists $e \in F'$ such that $F \cup e$ is still a forest.

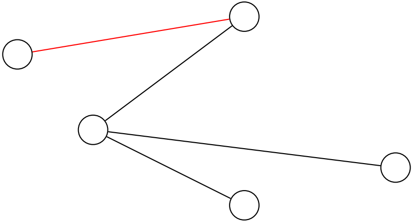
If that wasn't the case, adding any $e \in F'$ to F would create a cycle.
$\Rightarrow$ All $e \in F'$ connect two vertices in a component in F



Lemma 3

F, F' are forests on the same n vertices, with $|E(F)| < |E(F')|$.
$\Rightarrow$ There exists $e \in F'$ such that $F \cup e$ is still a forest.

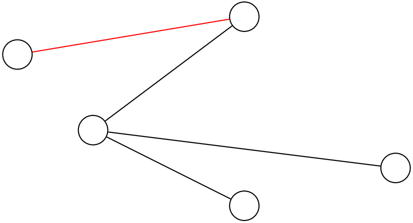
If that wasn't the case, adding any $e \in F'$ to F would create a cycle.
$\Rightarrow$ All $e \in F'$ connect two vertices in a component in F



Lemma 3

F, F' are forests on the same n vertices, with $|E(F)| < |E(F')|$.
$\Rightarrow$ There exists $e \in F'$ such that $F \cup e$ is still a forest.

If that wasn't the case, adding any $e \in F'$ to F would create a cycle.
$\Rightarrow$ All $e \in F'$ connect two vertices in a component in F

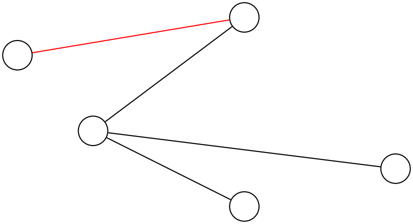


$\Rightarrow$ $\#$ components in $F \leq$ $\#$components in F'.

Lemma 3

F, F' are forests on the same n vertices, with $|E(F)| < |E(F')|$.
$\Rightarrow$ There exists $e \in F'$ such that $F \cup e$ is still a forest.

If that wasn't the case, adding any $e \in F'$ to F would create a cycle.
$\Rightarrow$ All $e \in F'$ connect two vertices in a component in F

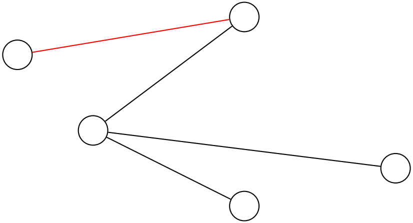


$$\Rightarrow |C(F)| \leq |C(F')|$$

Lemma 3

F, F' are forests on the same n vertices, with $|E(F)| < |E(F')|$.
$\Rightarrow$ There exists $e \in F'$ such that $F \cup e$ is still a forest.

If that wasn't the case, adding any $e \in F'$ to F would create a cycle.
$\Rightarrow$ All $e \in F'$ connect two vertices in a component in F

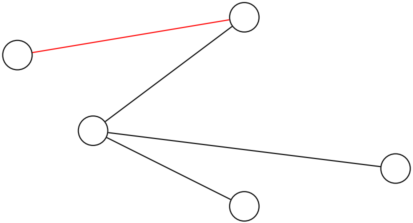


$$\Rightarrow |C(F)| \leq |C(F')|$$
$$|E(F)| < |E(F')|$$

Lemma 3

F, F' are forests on the same n vertices, with $|E(F)| < |E(F')|$.
$\Rightarrow$ There exists $e \in F'$ such that $F \cup e$ is still a forest.

If that wasn't the case, adding any $e \in F'$ to F would create a cycle.
$\Rightarrow$ All $e \in F'$ connect two vertices in a component in F

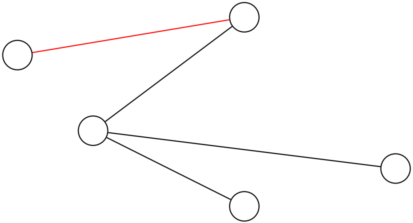


$$\Rightarrow |C(F)| \leq |C(F')|$$
$$|E(F)| < |E(F')|$$
$$\Rightarrow n - |C(F)| < n - |C(F')|$$

Lemma 3

F, F' are forests on the same n vertices, with $|E(F)| < |E(F')|$.
$\Rightarrow$ There exists $e \in F'$ such that $F \cup e$ is still a forest.

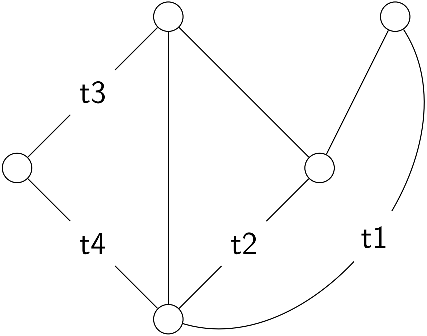
If that wasn't the case, adding any $e \in F'$ to F would create a cycle.
$\Rightarrow$ All $e \in F'$ connect two vertices in a component in F



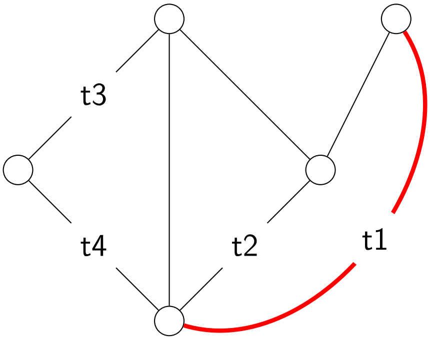
$$\Rightarrow |C(F)| \leq |C(F')|$$
$$|E(F)| < |E(F')|$$
$$\Rightarrow n - |C(F)| < n - |C(F')|$$

contradiction! $\square$

Graph G, trees T and H, with $\omega(T) > \omega(H)$
Sort the edges in H and T by weight. $\Rightarrow h_1, ..., h_n$ and $t_1, ..., t_n$
H_i and T_i are the forests made by the first i edges of H and T

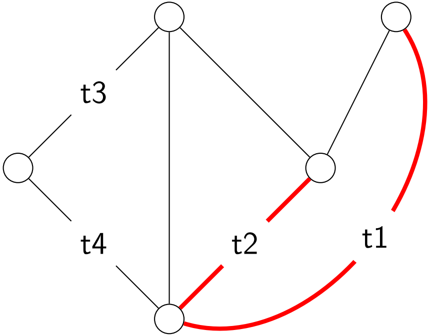


Graph G, trees T and H, with $\omega(T) > \omega(H)$
Sort the edges in H and T by weight. $\Rightarrow h_1, ..., h_n$ and $t_1, ..., t_n$
H_i and T_i are the forests made by the first i edges of H and T



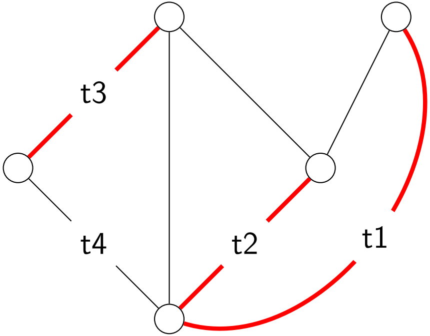
Let k be the first step $\omega(H_k)$ is smaller than $\omega(T_k)$
i.e $\omega(H_k) < \omega(T_k)$ but $\omega(H_{k-1}) \geq \omega(T_{k-1})$

Graph G, trees T and H, with $\omega(T) > \omega(H)$
Sort the edges in H and T by weight. $\Rightarrow h_1, ..., h_n$ and $t_1, ..., t_n$
H_i and T_i are the forests made by the first i edges of H and T



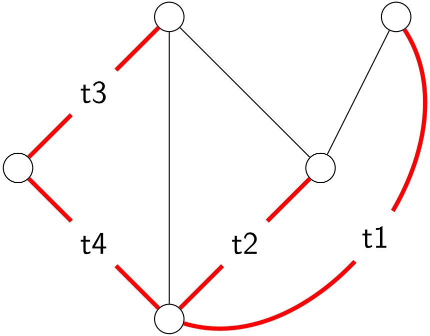
Let k be the first step $\omega(H_k)$ is smaller than $\omega(T_k)$
i.e $\omega(H_k) < \omega(T_k)$ but $\omega(H_{k-1}) \geq \omega(T_{k-1}) \Rightarrow \omega(h_k) < \omega(t_k)$

Graph G, trees T and H, with $\omega(T) > \omega(H)$
Sort the edges in H and T by weight. $\Rightarrow h_1, ..., h_n$ and $t_1, ..., t_n$
H_i and T_i are the forests made by the first i edges of H and T



Let k be the first step $\omega(H_k)$ is smaller than $\omega(T_k)$
i.e $\omega(H_k) < \omega(T_k)$ but $\omega(H_{k-1}) \geq \omega(T_{k-1}) \Rightarrow \omega(h_k) < \omega(t_k)$

Graph G, trees T and H, with $\omega(T) > \omega(H)$
Sort the edges in H and T by weight. $\Rightarrow h_1, ..., h_n$ and $t_1, ..., t_n$
H_i and T_i are the forests made by the first i edges of H and T



Let k be the first step $\omega(H_k)$ is smaller than $\omega(T_k)$
i.e $\omega(H_k) < \omega(T_k)$ but $\omega(H_{k-1}) \geq \omega(T_{k-1}) \Rightarrow \omega(h_k) < \omega(t_k)$

$$\omega(h_k) < \omega(t_k)$$

$\omega(h_k) < \omega(t_k)$

Lemma 3 $\Rightarrow$ there is a $h_j \in H_k$ such that $T_{k-1} \cup h_j$ is a forest.

$\omega(h_k) < \omega(t_k)$
Lemma 3 $\Rightarrow$ there is a $h_j \in H_k$ such that $T_{k-1} \cup h_j$ is a forest.
We know $\omega(h_j) \leq \omega(h_k)$ because h is sorted.

$\omega(h_k) < \omega(t_k)$
Lemma 3 $\Rightarrow$ there is a $h_j \in H_k$ such that $T_{k-1} \cup h_j$ is a forest.
We know $\omega(h_j) \leq \omega(h_k)$ because h is sorted.
$\Rightarrow \omega(h_j) \leq \omega(h_k) < \omega(t_k)$

$\omega(h_k) < \omega(t_k)$
Lemma 3 $\Rightarrow$ there is a $h_j \in H_k$ such that $T_{k-1} \cup h_j$ is a forest.
We know $\omega(h_j) \leq \omega(h_k)$ because h is sorted.
$\Rightarrow \omega(h_j) \leq \omega(h_k) < \omega(t_k)$
Kruskal's algorithm wouldn't chose t_k at step k.

$\omega(h_k) < \omega(t_k)$
Lemma 3 $\Rightarrow$ there is a $h_j \in H_k$ such that $T_{k-1} \cup h_j$ is a forest.
We know $\omega(h_j) \leq \omega(h_k)$ because h is sorted.
$\Rightarrow \omega(h_j) \leq \omega(h_k) < \omega(t_k)$
Kruskal's algorithm wouldn't chose t_k at step k.
$\Rightarrow$ contradiction $\Rightarrow$ Kruskal's Algorithm finds a minimum weight spanning-tree!