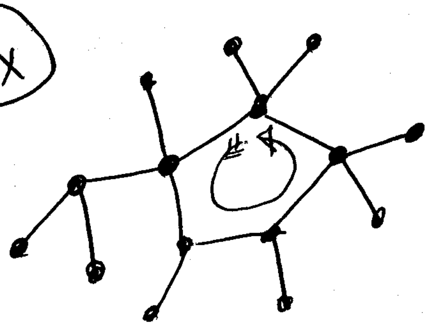
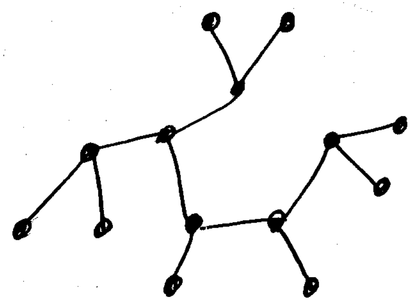
Ch-12 Trees

- Recall the definition of a tree T as a simple connected graph with no cycles

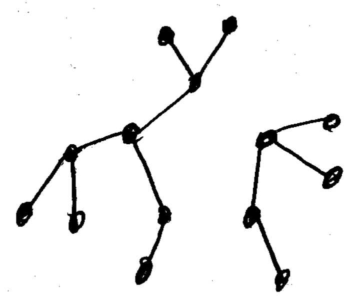
ex



not a tree
(contains a cycle!)



a tree



not a tree, but a forest!

Thm For a simple graph G the following are equivalent:

i) G is a tree

ii) Between any two vertices $u, v \in V(G)$ there is exactly one (unique!) path.

<u>proof</u>: (idea!)

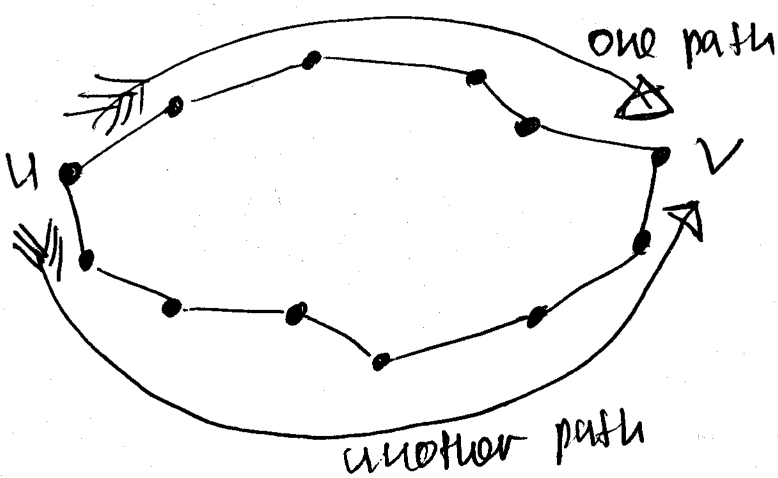
- If G is not a tree, then either it is not connected or it has a cycle C_k for some k

  - G not connected:

    $\Longrightarrow$ There are u, v with <u>no</u> path between them

  - G has a cycle

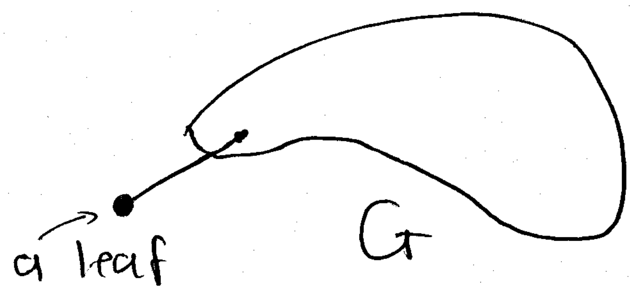
    $\Longrightarrow$ There are u, v in G (in fact in C_k) with two distinct paths between them



- If G has vertices with 0 or ≥ 2 paths between them

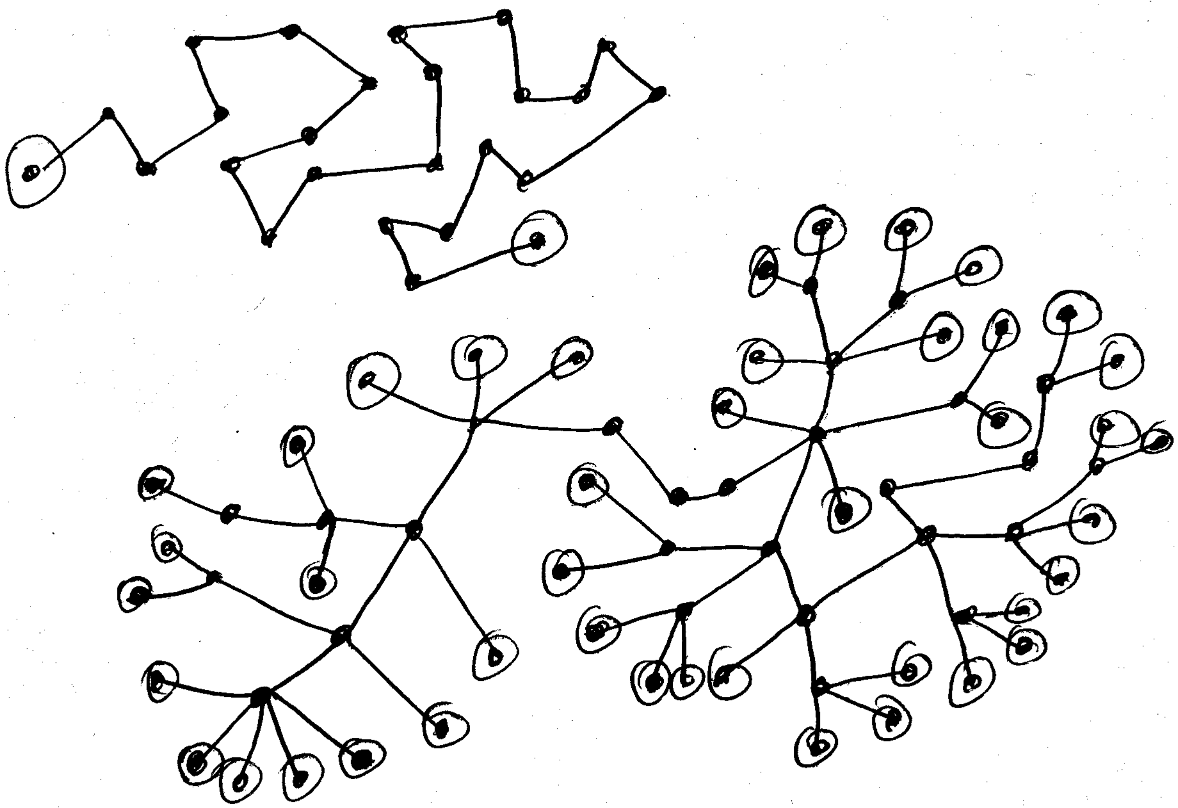
  $\Longrightarrow$ G is either disconnected or must contain a cycle (formed by 2 distinct paths!)

Def A vertex u in a graph G of degree one, $d_G(u) = 1$ is called a <u>leaf</u>.



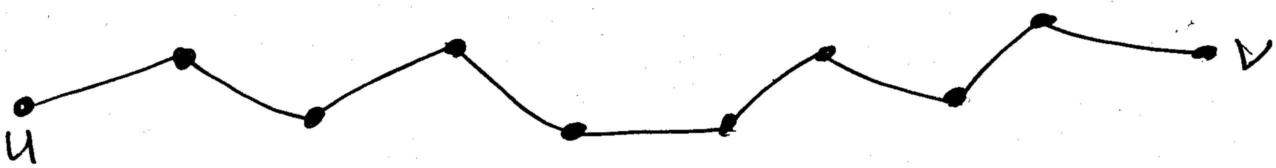
– One of the most important properties of trees the following:

Fact Any tree T on $n \geq 2$ vertices has @ least two leaves



pf (Sketch:)

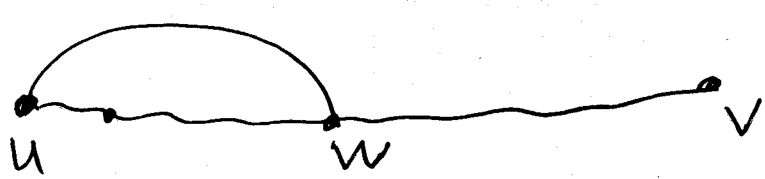
- Let P be the longest path in T



$\Longrightarrow$ Both the initial & final vertex u & v of P are leaves of T:

- u is not connected to any vertex w not in P, since then P would not be of maximum length

- u is not connected to any vertex w in P (other than its sole neighbor), since then we would have a cycle



$\Longrightarrow$ Both u & v are leaves of T !

$\blacksquare$

NB! The mere fact that every tree has a leaf is enough to prove some nice properties of trees by induction:

Fact For a graph G TFAE:

i) G is a tree on n vertices

ii) G is a connected graph on n vertices and $n-1$ edges

Pf: i) $\Longrightarrow$ ii) Let u be a leaf of G

$d_G(u) = 1$
since u = leaf!

Then $G' = G \setminus u$ is connected graph on $n-1$ vertices and $n-2$ edges

$\Longrightarrow$ By induction (on n) G is then also connected on n vertices and $n-1$ edges.

ii) $\Rightarrow$ i) We have here

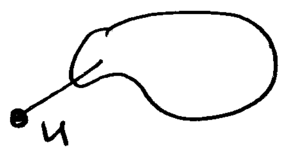
$$2(n-1) = 2|E(G)| = \sum_{u \in V(G)} d_G(u) \quad \text{-----} \quad \circledast$$

Since G is connected then $d_G(u) \geq 1$ for each $u \in V(G)$ & By $\circledast$ $d_G(u) \geq 2$ cannot hold for all u, since then

$$2(n-1) = \sum_{u \in V(G)} d_G(u) \geq \sum_{u \in V(G)} 2 = 2n$$

$\Rightarrow$ So G has a leaf (in fact two!)

call it u



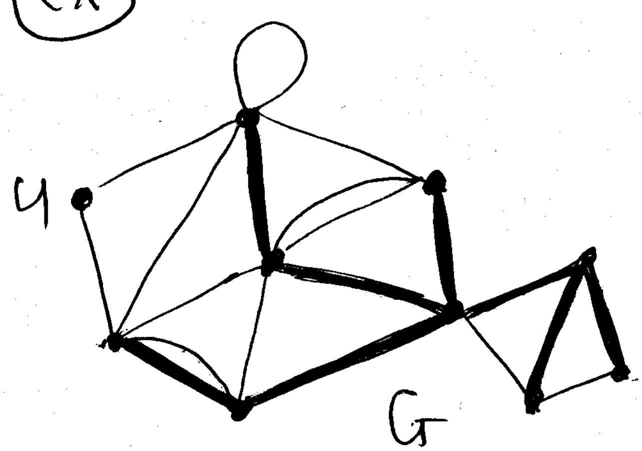
$\Rightarrow$ $G' = G \setminus u$ is connected on $n-1$ vertices and $n-2$ edges & hence a tree (by induction)

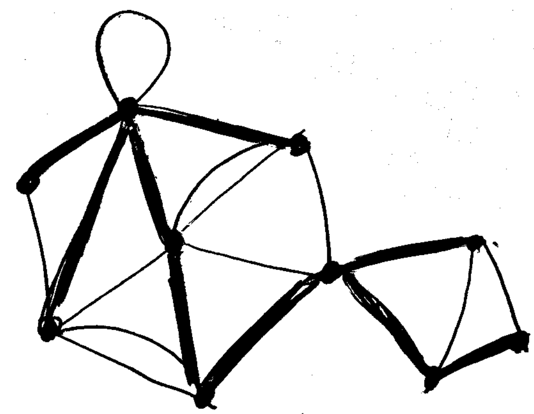
$\Rightarrow$ G = tree $\blacksquare$

Def A subgraph T of a connected graph G is a spanning tree of G if

- T is a tree &
- $V(T) = V(G)$

ex





$T =$ subgraph of G
$T =$ tree
$T \neq$ spanning tree:
$u \in V(G) \setminus V(T)$!

$T =$ subgraph of G
$T =$ tree
$V(T) = V(G)$
$\Rightarrow T$ is a spanning tree of G !

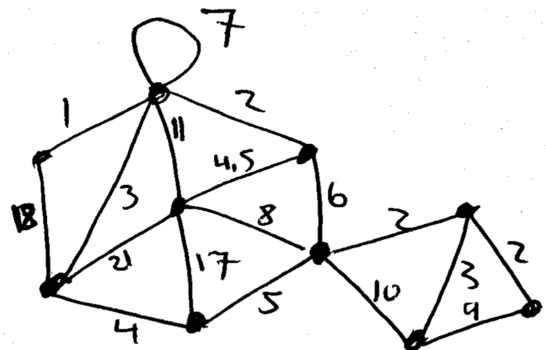
Fact Every connected graph has a spanning tree.

Def A weighted graph G is a graph with weight attached to each edge in G

— Such a weight can be given by a function

$$w : E(G) \longrightarrow \mathbb{R}$$

(ex)



A weighted graph G

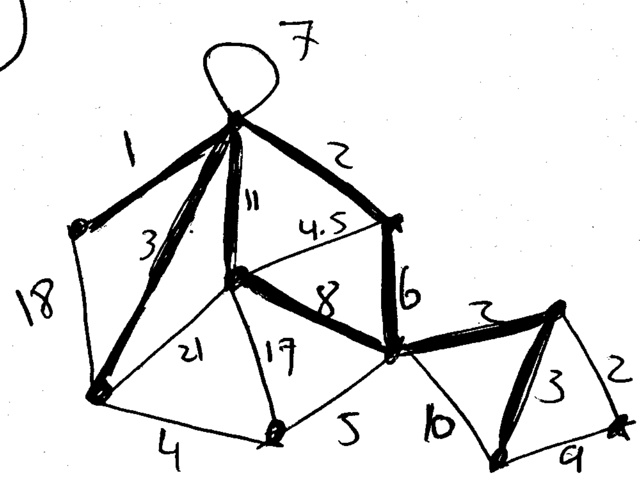
Such a weighted graph can represent cities (the vertices) and highways (edges) connecting them & the weights can denote the miles between the cities on those highways.

Def — Let G be a weighted graph and $w: E(G) \longrightarrow \mathbb{R}$ the corresponding weight function.

— For a subgraph $H \subseteq G$ we have the <u>weight of H</u> by:

$$w(H) := \sum_{e \in E(H)} w(e)$$

ex



$H \subseteq G$

thick edges

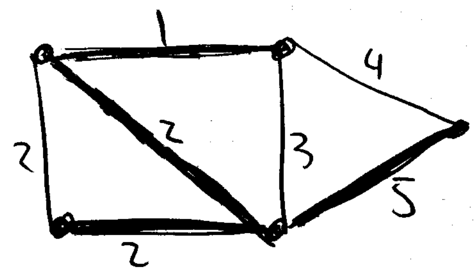
Here: $w(H) = \sum_{e \in E(H)} w(e)$

$= 1+3+11+2+8+6+2+3 = \underline{\underline{36}}$

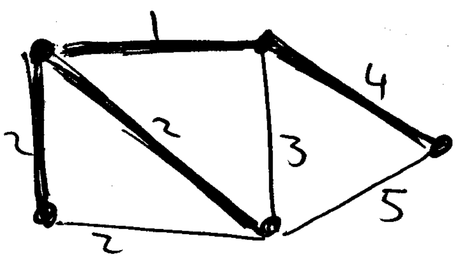
- A spanning tree in a connected graph G is the subgraph that connects all the vertices minimally

$\Rightarrow$ $\boxed{Q}$ Given a weighted graph G, how can we find a spanning tree with $w(T)$ @ minimum?

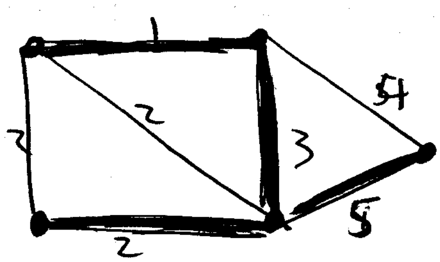
ex



$w(T_1) = 1+2+2+5$
$= 10$



$w(T_2) = 1+2+2+4 = 9$



$w(T_3) = 1+2+3+5$
$= 11$

T_1 & T_3 $\neq$ minimum weight spanning trees

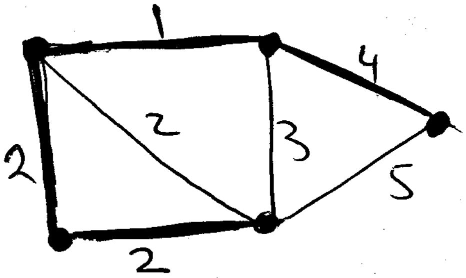
T_2 = minimum weight spanning tree (it turns out!)

Def A spanning tree T in a connected weighted graph G (with a weight function w) is called a __minimum weight spanning tree of G__, if $w(T)$ is a minimum.

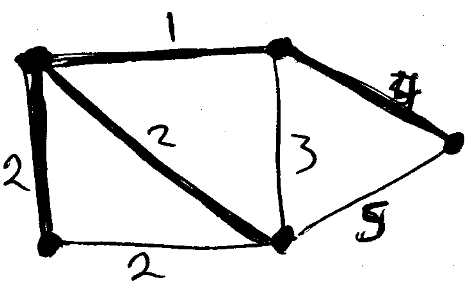
NB! A weighted graph G can have many minimum spanning trees!

ex

T_1 :



T_2 :



T_1 & T_2 are, in fact, both minimum weight spanning trees of minimum total weight $w(T_1) = w(T_2) = w_{min} =$

$$1+2+2+4 = 9 \;!$$

Kruskal's Algorithm find a

min. weight spanning tree in a weighted graph G:

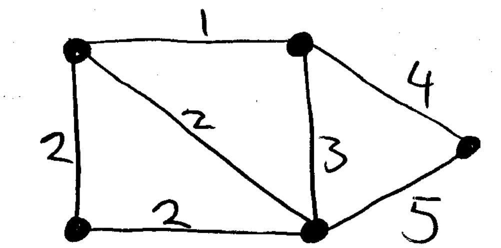
— Let G be weighted & connected with n vertices, $w: E(G) \longrightarrow \mathbb{R}$ given.

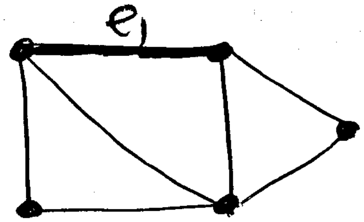
1. Pick $e_1 \in E(G)$ with $w(e_1)$ minimum.
2. While $i < n-1$ do:

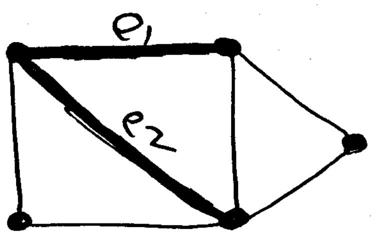
   Pick $e_{i+1} \in E(G) \setminus \{e_1, e_2, .., e_i\}$ of minimum weight such that
   $$\{e_{i+1}\} \cup \{e_1, e_2, .., e_i\}$$
   does not form a cycle.
3. Record $T = \langle \{e_1, e_2, .., e_{n-1}\} \rangle$ as a minimum weight spanning tree of G.

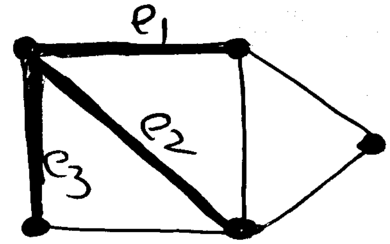
$\mathbb{Z}$.

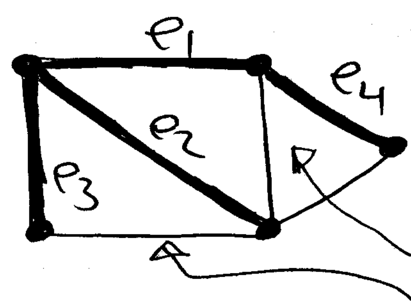
(ex) Find a min. weight spanning tree of the following graph using Kruskal's Algorithm:



1. 

2. 

3. 

4. 

- Cannot pick e_4 here since then a cycle is formed.
- $n=5 \Rightarrow n-1=4$ & we have 4 edges

$\Rightarrow$ DONE $W_{min} = W(T) = 1+2+2+4 = \underline{9}$

NB:

- In the process of Kruskal's Algorithm it is OK that the intermediate/temporary graph formed by edges $e_1, e_2, \dots, e_i$ is disconnected!
- It will connect into a <u>tree</u> when $i = n-1$!