Trees Chapter 7.

Definition A graph is called a tree if it is connected and contains no circuits.

A graph all of whose components are trees are called a forest.

The study of trees began with chemistry with Cayley and the enumeration of isomers of a compound.

ie. C_4H_{10} carbon atoms have bonding valency 4
hydrogen have bonding valency 1

$\bullet$ = carbon $\qquad$ $\bullet$ = hydrogen

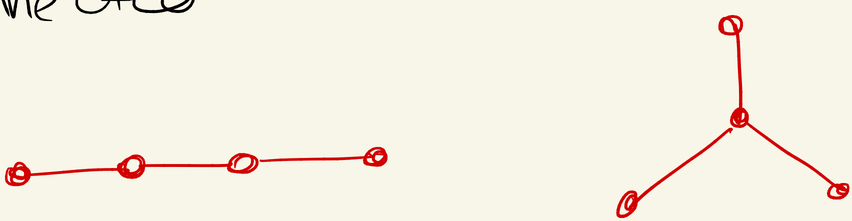
Recall $\displaystyle 2|E| = \sum_{u \in V} d(u)$

$= 4 \cdot 4 + 10 = 26 \Rightarrow |E| = 13.$

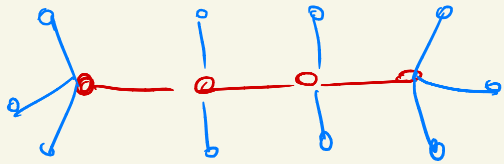
$$|V| - |E| = 4 + 10 - 13 = 1$$

$\Rightarrow$ a tree (Theorem to come)

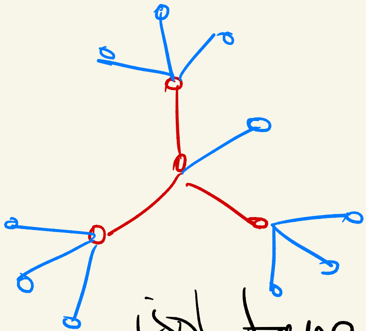
The subgraph of carbons must be connected $\Rightarrow$.



only two types of (unlabelled) trees on 4 vertices. The isomers of C_4H_{10} are butane and isobutane.



butane



isobutane.

Definition Let $G=(V,E)$ be a connected graph. A subgraph T that is a tree of order $n=|V|$ is called a *spanning tree*.

Proposition Every connected graph G possesses a spanning tree.

Proof Either G is a tree or it contains a circuit C_1. Remove an edge $e_1 \in C_1$ from G.

Now $G \setminus e_1$ is either a tree or possesses a cycle C_2. Continue removing edges from remaining cycles until we have a tree $T = G \setminus C$.

$C = \{e_1, \ldots, e_k\}$ $k = \#$ cycles of G. $\square$

Theorem 7.2 The following are equivalent:

a) $G = (V, E)$ is a tree

b) Every pair of vertices in G are joined by exactly one path

c) G is connected and $|E| = |V| - 1$.

Proof $a \Rightarrow b$ If u, v were joined by two paths we could form a circuit in G

$b \Rightarrow a$ If C is a circuit there are two paths connecting any pair of vertices in C.

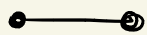
$a \Rightarrow c$ let $P = u, u_1, \ldots, u_n, v$ be a longest path in G.
Then all neighbors of u and v are in P. Otherwise extend by a neighbor and obtain a longer path, since G has no circuits. Therefore

$d(u) = d(v) = 1$. Remove the vertex u and edge uu_1 from G to get $G_1 = (V_1, E_1)$.

"Trimming the leaves of the tree"

Notice $|V|-|E| = |V_1|-|E_1|$.

Repeat the process until we are left with G_{n-2} which is a tree on two vertices

G_{n-2}  and $|V|-|E| = |V_{n-2}|-|E_{n-2}| = 1$

<u>$c \Rightarrow a$</u> If T is a spanning tree of G. Then $V(G) = V(T)$ and using the assumption on G and the above proof for T.

$$1 = |V(G)| - |E(G)| \leq |V(T)| - |E(T)| = 1$$

$$\Rightarrow |E(G)| = |E(T)| \rightarrow G = T.$$

<u>Corollary</u> If a graph consists of t components a spanning forest possesses $|V| - t$ edges.

Corollary Let G be a graph

Then

$$|V| - |E| = |\text{Components}| - \left|\begin{matrix}\text{independent}\\ \text{circuits}\end{matrix}\right|$$

Proof Assume G is connected

ie $|\text{Components}| = 1$. Then

a spanning tree T satisfies

$$|T| = |E| - \left|\begin{matrix}\text{independent}\\ \text{circuits}\end{matrix}\right|.$$

$$|V| - |E| = |V| - |T| - \left|\begin{matrix}\text{independent}\\ \text{circuits}\end{matrix}\right|$$

$$= 1 - \left|\begin{matrix}\text{independent}\\ \text{circuits}\end{matrix}\right| \quad \text{by } \underline{\text{Thm}}.$$

If G is not connected apply the above result component by component. $\square$

Corollary If T is a tree of order $|V| = n \geq 2$ and $(d_1, \ldots, d_n)$ is the degree sequence, then

$$\sum_{i=1}^{n} d_i = 2n-2$$

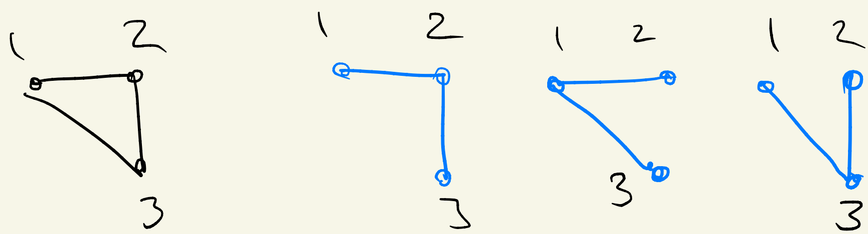
Proof $2n-2 = 2(|V|-1) = 2|E|$.

Given a graph G we can find a spanning tree. How many are there?

This is a <u>very</u> difficult question in general.

Theorem 7.4 The number of spanning trees in the complete graph K_n is given by $t(n) = n^{n-2}$.

$n=3$

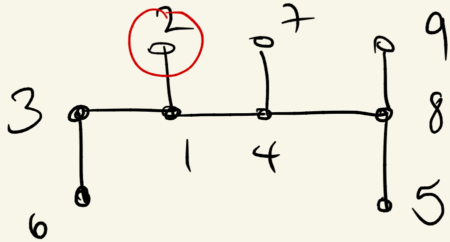


Proof Construct a bijection between spanning trees and $\{(a_1, \dots, a_{n-2})$ with $1 \le a_i \le n\}$

Spanning Trees $\longrightarrow \{(a_1, \dots, a_{n-2})\}$

Let u be minimum label of 1valent vertices. Then $a_1 = v$, v is unique label of u.

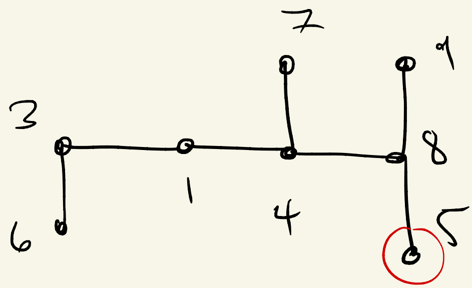
T



$a_1 = 1.$

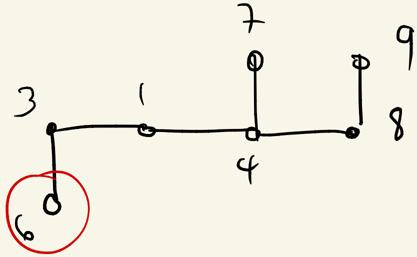
To obtain a_2, remove uv and repeat.

$T \setminus \{12\} =$



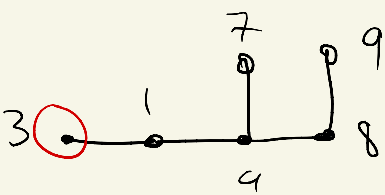
$a_2 = 8$

$T \setminus \{58\} =$



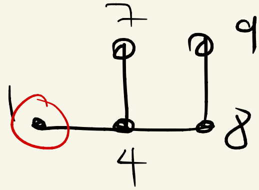
$a_3 = 3$

$T \setminus \{36\} =$

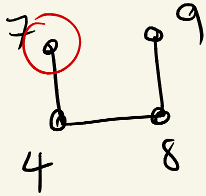


$a_4 = 1$

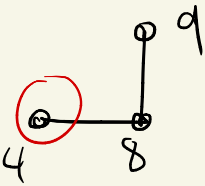
$T \setminus \{13\} =$



$a_5 = 1$

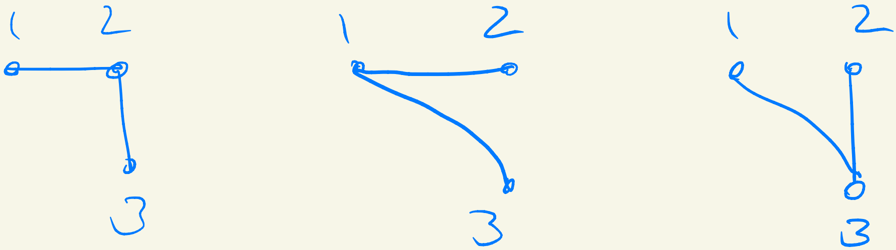
$T \setminus \{1,4\} =$ 

$a_6 = 4$

$T \setminus \{7,4\} =$ 

$a_7 = 8$

For the trivalent trees we obtain



(2) (1) (3).

The inverse mapping

$\{(a_1, \dots, a_{n-2})\} \longrightarrow$ Spanning trees

Notice firstly that

$f_i = \#$ times i appears in $(a_1, \ldots, a_{n-2})$ $= d_i - 1$.

Ie. if i does not appear it is a leaf vertex $d(i) = 1$.

Let b_1 be the minimal i not appearing in $(a_1, \ldots, a_{n-2})$.

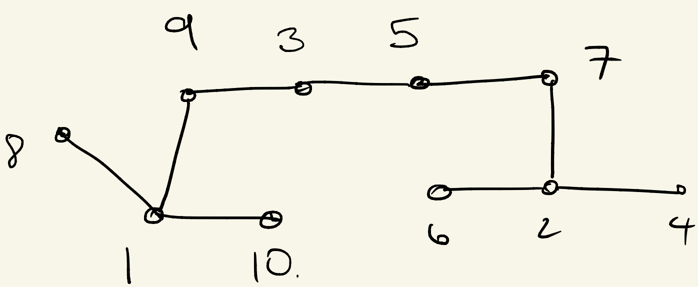
There is an edge in T between $a_1 b_1$.

Let b_2 be minimal $i \neq b_1$ $(a_2, \ldots, a_{n-2})$. There is an edge in T $b_2 a_2$. and so on. This produces the inverse mapping.

Example 7.6

$(a_1, \dots, a_{n-2}) = (2, 2, 7, 5, 3, 9, 1, 1)$.

$(b_1, \dots, b_{n-2}) = (4, 6, 2, 7, 5, 3, 8, 9, 10)$



Graph algorithms How to determine if a graph is connected (from say adjaceny matrix or neighbor list)?

A graph is connected $\iff$ $\exists$ a spanning tree.

Algorithm 7.8 Breadth-first search

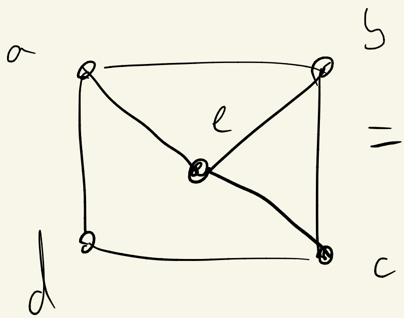
Constructs a spanning tree if G is connected

1) Start at an arbitrary vertex labelled 1.

2) Suppose current vertex is i, and labels $1, \ldots, r$ have been assigned. If $r = n$ stop! Label all unlabelled vertices adjacent to i with $r+1, \ldots, r+k$, and add edges $i(r+1), \ldots, i(r+k)$ to the tree. If $i+1$ has not been assigned stop. (G is not connected).

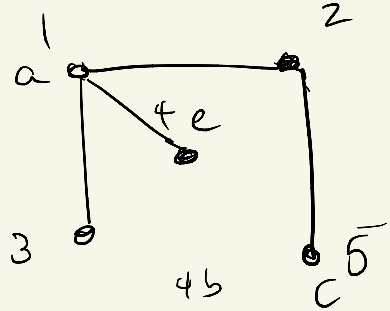
Otherwise make it the current vertex and repeat.

$$\begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array}\begin{array}{c} \begin{array}{ccccc} 1 & b & c & & \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \end{pmatrix} \end{array} = A$$

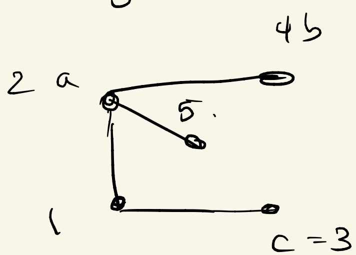


$= G$

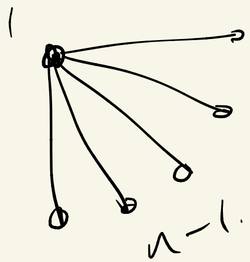
start with
$a = 1$



start with
$d = 1$



What tree does the algorithm produce for K_n?



no matter where one starts.

See text for correctness of the algorithm.

Breadth-first because breadth/width of vertices.

Dual to this is depth first search.

Algorithm 7.9 Depth first search

1) Choose starting vertex and label it 1. (this is "predecessor")

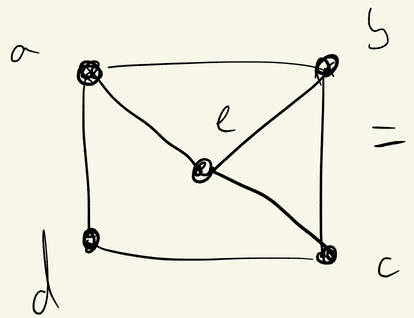
2) Suppose you are at vertex i and numbers $1, \ldots, r$ have been assigned. If $r = n$ stop. Otherwise choose an unnumbered neighbor of i and number it $r+1$ and add $i(r+1)$ to the tree. The current vertex is $r+1$ and predecessor is i. If there is no unnumbered neighbor of $r+1$ return to

predecessor vertex i. Repeat Step (2). If $i=1$ and there is no unnumbered neighbor then G is not connected.

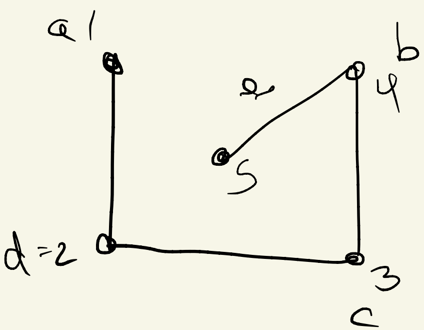
Example

$$\begin{array}{c|ccccc} & 1 & b & c & & \\ \hline a & & & & & \\ b & & & & & \\ c & & & & & \\ d & & & & & \\ e & & & & & \end{array}$$

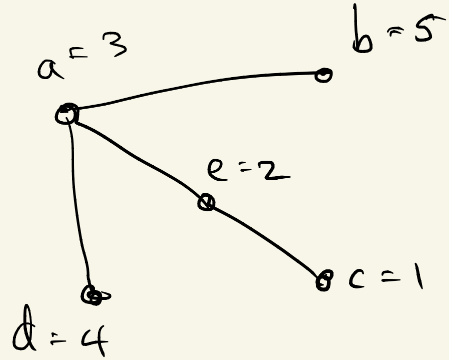
$$\begin{array}{c} a \\ b \\ c \\ d \\ e \end{array}\begin{pmatrix} 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \end{pmatrix}$$



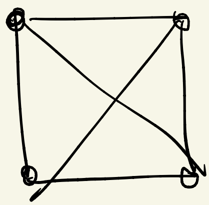
$= G$

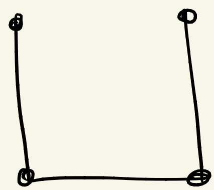


or



What does DFS return for K_n?





always a path.

Both algorithms have running time $\Theta(|E|)$.

Minimal spanning trees & Greedy algorithms