

Lesson 24

TRIGS

A graph $G = (V, E)$ is connected if for every pair of vertices $x, y \in V$, there is a path from x to y . A graph that is not connected can be partitioned into components (subgraphs) that are connected and there are no edges between the components.

Ex. Let $G = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$

There is an edge between $x, y \in G$ if and only if they have the same parity. So the components are

$$G_1 = \{1, 3, 5, 7, 9\} \text{ and } G_2 = \{2, 4, 6, 8, 10\}$$

2.

Trees

Let $G = (V, E)$ be a connected graph, with $|V| = n$. The following are equivalent

1. The number of edges is $n - 1$

2. There are no cycles in G (recall cycle = closed path)

3. Every edge is a bridge

Where a bridge is an edge in a connected graph where, if it is removed, then the graph is no longer connected



No bridges



(a, b) and (b, c) are bridges



(c, d) is a bridge

3 Proof: $1 \rightarrow 2$. (we show $\sim_2 \rightarrow \sim_1$,

Suppose $\gamma: x_0 - x_1 - \dots - x_k = x_0$ is

a cycle. There are k edges

and k vertices in this cycle

Each $y \in G$, not on γ , is connected to γ using at least one edge. There are $n-k$ vertices not on γ , so there are $n-k$ edges used to connect them to γ . (at least)

So there are $(n-k)+k=n$ edges

at least. $\rightarrow \sim_2 \rightarrow \sim_1$ and $1 \rightarrow 2$.

$2 \rightarrow 3$. We show $\sim 3 \rightarrow \sim 2$.

Suppose $\alpha: u-v$ is an edge that is not a bridge. Removal of α leaves a subgraph, G_1 , that is connected. Hence

There is a path

$$v: v-x_1 - \dots - x_k = u \text{ in } G_1$$

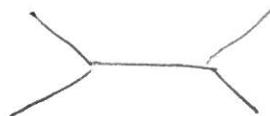
Then $v-x_1 - \dots - x_k = u-v$ is a cycle in G . So ~ 2 holds and $2 \rightarrow 3$.

$3 \rightarrow 1$. Proceed by induction on n . If $n=2$, both 3 and 1 hold.

Assume $3 \rightarrow 1$ for all $|G| < n$ and suppose $|G|=n$. Let $u-v$ be an edge, hence it is a bridge by assumption. The subgraphs on each side of the bridge, call G_1 and G_2 , $|G_1| \leq k$ and $|G_2| = n-k$.

5) Each edge is a bridge in G_1 ,
and also in G_2 . Hence
the number of edges in G_1 is
 $k-1$ and in G_2 is ~~$n-(k-1)$~~ $n-k-1$
by induction. The total number
of edges in G is then
 $(k-1) + (n-k-1) + 1 = n-k-1$ where
the last 1 is from $u-v$. Hence
1 holds and all parts are
equivalent.

Def A tree is a connected graph
which satisfies any (hence all)
of the conditions 1, 2, 3



is a tree. It has
6 vertices and 5 edges,
each edge is a bridge and
there are no cycles.



is not a tree
(What can you say about the 3
conditions?)

When $n=4$, there are 2 trees



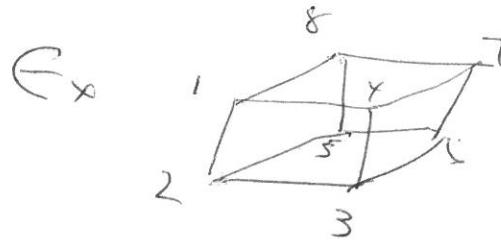
A subgraph of G which is a tree and contains all the vertices of G is called a spanning tree for G .

Given G , we can construct ~~the~~ a spanning tree (there usually are many spanning trees)

1. Start with any $v \in V$.
2. Take all vertices not equal to v that are connected to v by an edge and take all those edges. Call them (V_2, E_2)
3. For each vertex v_{inst} in V_2 that is connected to a vertex v_i in V_2 by an edge, take v_i and $v_i - v_{inst}$. For each such v_i only take one v_i . Call this graph (V_2, E_3)

It contains (V_1, E_1) and all the new vertices and edges.

Continue this process until there are no ~~edges~~ vertices left



1. Start with $V = 1$

2. Take 3, 4, 8 and $(1, 2), (1, 4), (1, 8)$

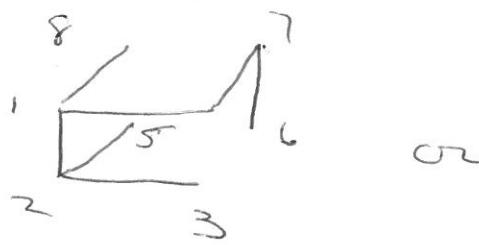
3. We can use $(2, 3), (2, 5), (4, 7)$

so we have $V_2 = \{1, 2, 4, 8, 3, 5, 7\}$

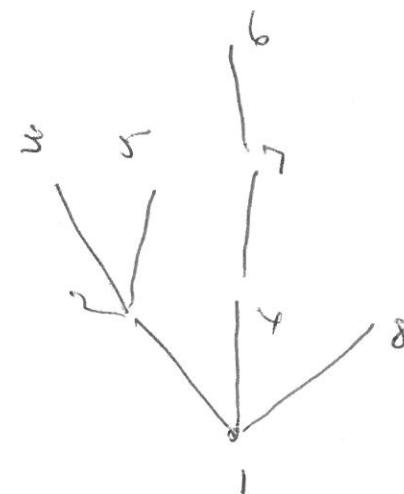
$E_2 = \{(1, 2), (1, 4), (1, 8), (2, 3), (2, 5), (4, 7)\}$

4. Use $(7, 6)$ and add $(7, 6)$ to E_2 to set E_3 and 6 to V_2 to set V_3

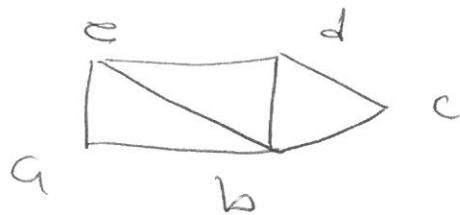
5. Tree



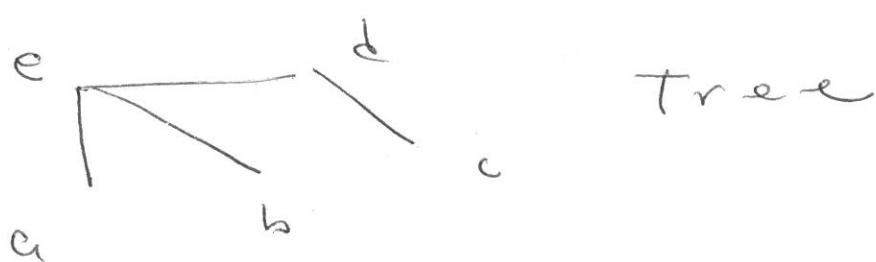
or



8 Ex



1. Pick e
2. Take e-b, e-a, e-d
3. Take d-c



The algorithm we just described is called the breadth first algorithm for a spanning tree. Once a beginning vertex, v is picked, every other vertex's path to v is as short as possible.

9

For applications, at times a non-negative number is associated with each vertex, notation $c(x,y)$. $c(x,y)$ is called the weight of the edge.

Algorithm for a distance tree

Let G be a weighted graph

1. Pick $u \in V$ Put

$$U = \{u\} \quad D(u) = 0 \quad F = \emptyset \quad T = (U, F)$$

Here T is a subgraph, U is its vertex set, F is its edge set

2. Pick $x \in V$ such that $x-u \in E$ and $D(u) + c(u, x)$ is as small as possible

$$\text{Put } U = \{u, x\} \quad F = \{(u, x)\}$$

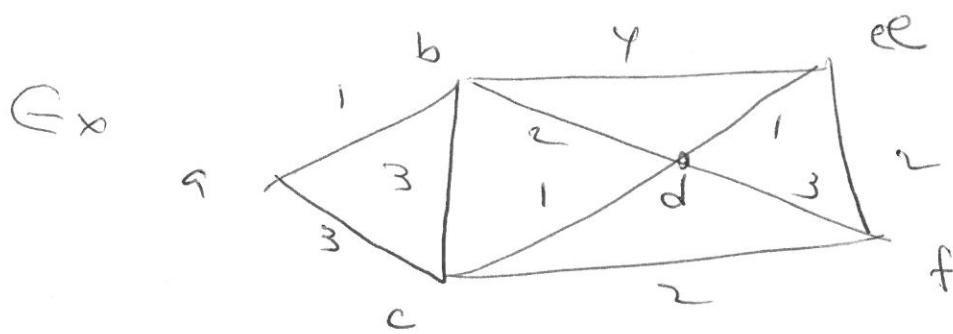
$$D(x) = D(u) + c(u, x).$$

3. Repeat this process until all $x \in V$ are used

16

The result is a spanning tree

and the weights from u to each x in the spanning tree and in G are the same



Let $u = a$

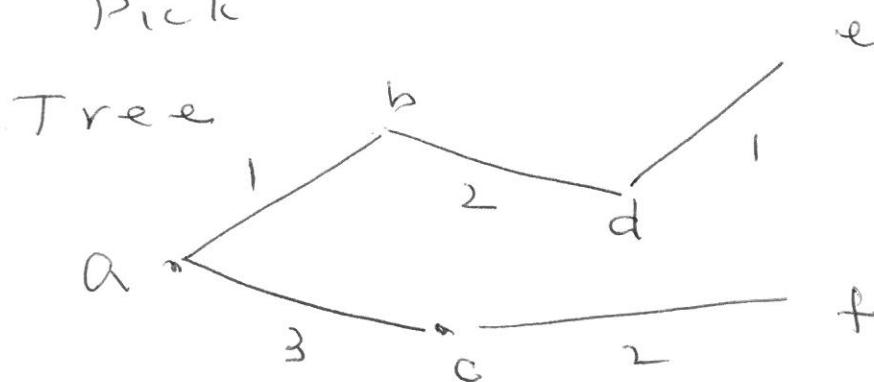
Pick $a - b$ 1

Pick $b - d$ 2

Pick $a - c$ 3

Pick $d - e$ 1

Pick $c - f$ 2



11

Sometimes the aim is to construct a tree where the total weights in the tree is minimum.

Algorithm (Prim)

1. Pick u , a vertex in V

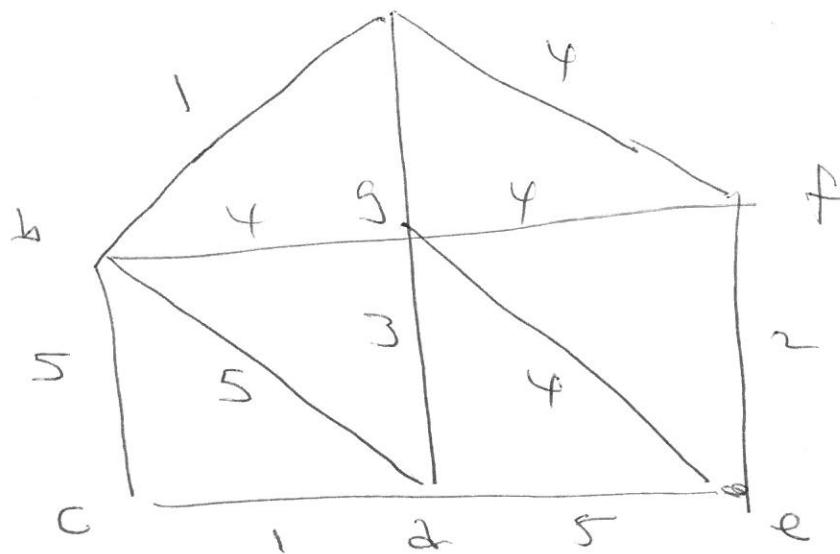
Set $U = \{u\}$ $F = \emptyset$ $T = (U, F)$

2. Pick edge $\{x, y\}$ of smallest weight such that $x \in U$, $y \notin U$

Then $U = U \cup \{y\}$ $F = F \cup \{(x, y)\}$

$T = (U, F)$

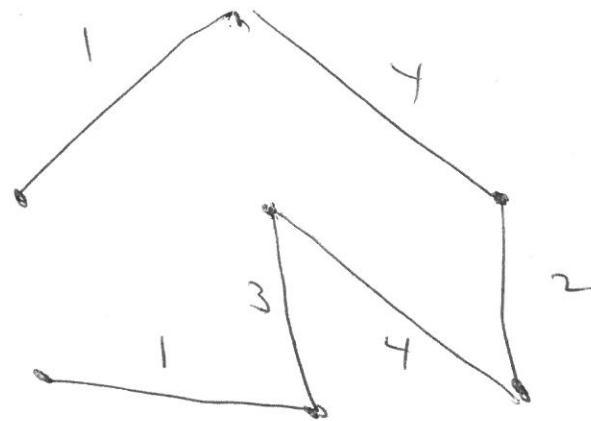
Repeat until all vertices are used. 9



Start with a

12 Add vertices and edges as follows

(a, b), (a, f), (f, e), (e, g), (g, d), (d, c)



Spanning Tree

$$\text{Total} = 15$$

Ex. A company located in town a needs to service towns g, b, c, d, e, f, g. The cost of going between towns is listed in the graph on the last page. What is the best route and how much will the trip cost?