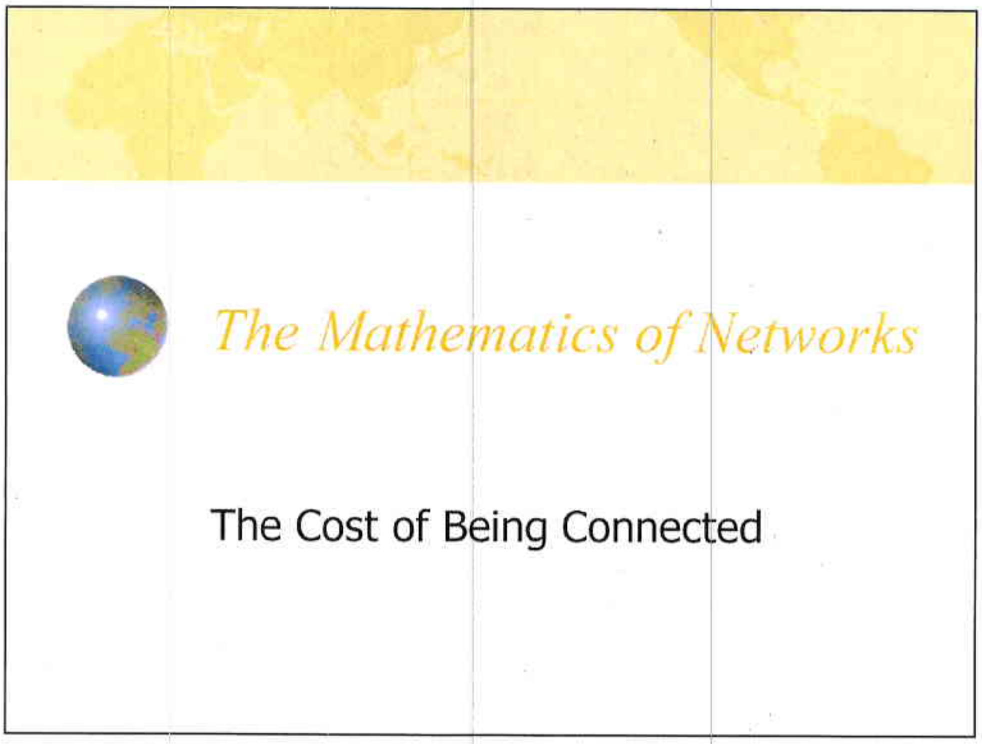


Chapter 71.ppt

7.1 – 7.3

Networks: on the cutting edge of modern mathematics:
Transportation networks, communication networks, social networks, the internet, etc.

There is no good definition of network.

Definitions

- Network
  - A graph that is connected
- Edges
  - Also called *links*
- Vertices
  - Also called *nodes* or *terminals*

Our definitions are simple, however.

- A network is simply a connected graph.

- Edges are called links

- Vertices are also called nodes or terminals.

-

Evolutionary Network

- No predetermined master plan, structure, or centralized direction
  - Example: the World Wide Web

⌘ Evolutionary networks evolve on their own. Other examples: networks of friends, business associates

⌘

Designed Network

- Carefully planned
  - Examples: highways, fiber-optic cable lines, rail lines, etc.
  - Cost is roughly proportional to its length (x dollars per mile), which means the cheapest networks are the shortest
  - But not always: freeway bridges and overpasses

⌘ Designed to meet specific goals and objectives. Often expensive to build.

Bridges are very expensive, so it is cheaper to build a freeway around a lake than it is to build a bridge.

⌘

Why Bother?

- An example:
  - The Amazonia Telephone Company
  - Network of underground fiber optic cables joining seven small towns in the Amazon jungle.
  - Cables should be buried along already existing roads.

Theme of this section is finding optimal networks connecting a set of points. *Optimal* means shortest or cheapest.

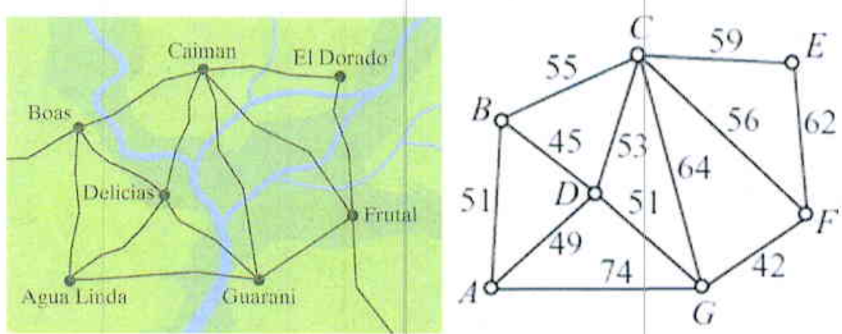
The general goals are

1. To connect all the terminals
2. Make the total cost as low as possible

Here is an example we'll be working with.

- The Amazonia Telephone Company
- Network of underground fiber optic cables joining seven small towns in the Amazon jungle.
- Cables should be buried along already existing roads.

Amazonian Cable Network



Here's the map of existing roads, and the corresponding weighted graph showing the cost, in millions of dollars, to lay the fiber-optic cable along that road. Our solution must (I) connect all the cities, and (ii) minimize cost.

Rephrasing in the language of graphs:

1. Network must be a subgraph of the original (no new edges)
2. Network must span the original graph (include all vertices)
3. Network must be minimal (inexpensive as possible)

Trees

- Network
  - another name for a connected graph
- Tree
  - Network with no circuits
- Spanning Tree of a network
  - A subgraph that connects *all* the vertices of a network and has no circuits

The backbone of a minimum network is a special type of graph called a tree. We'll look at the properties of trees and minimum spanning trees, and finally, we'll look at Kruskal's algorithm.

Some formal definitions:

- Again, a network is a connected graph
- A Tree is a network with *no circuits.*
- A Soanning Tree of a network is a tree (no circuits) that connects all the vertices of a network.
-

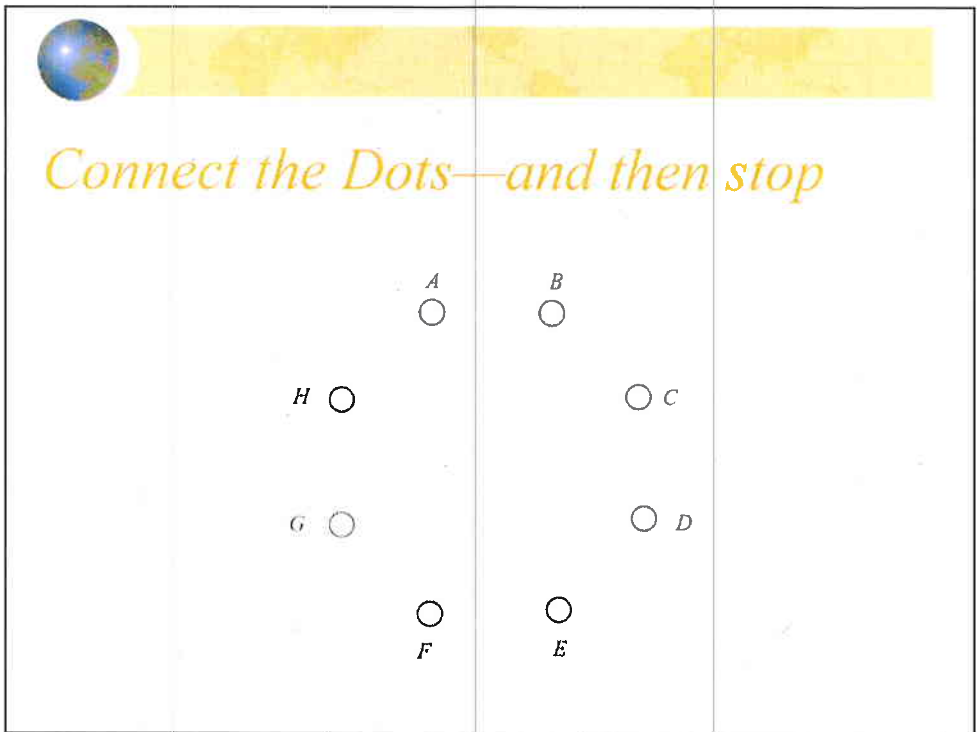
Ok, which of these are, by definition, trees?

a) Not connected, not even a network
b) Not connected, so not even a network
c) Network with circuits, so not tree
d) Network with circuits, so not tree
e) tree
f) tree
g) tree
h) tree

Trees

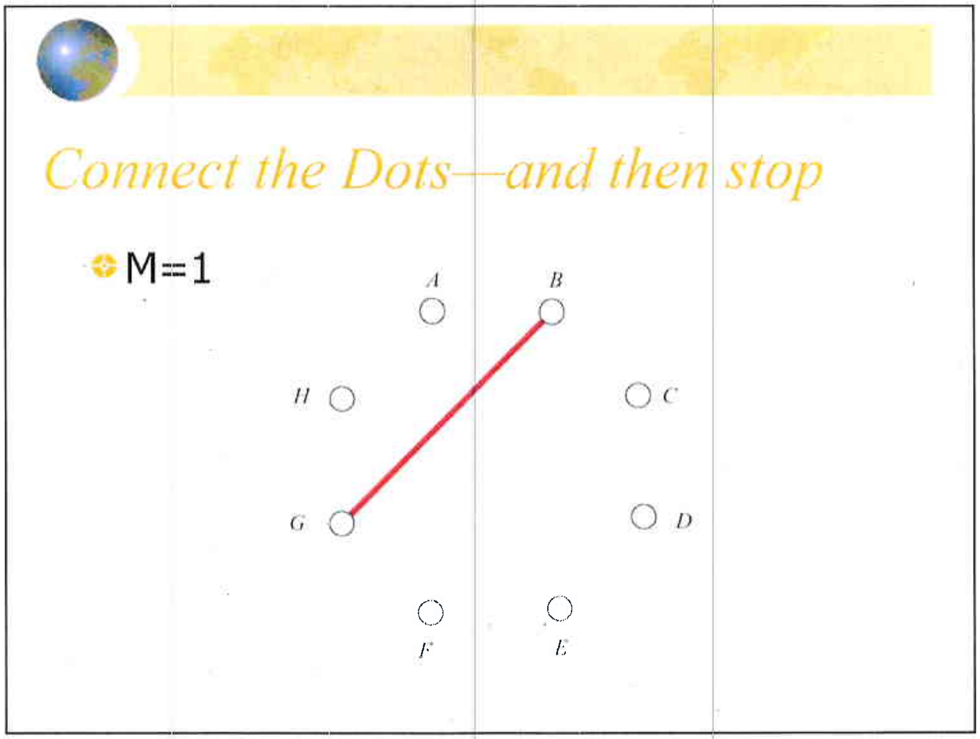
- Minimum Spanning Tree (MST)
  - The spanning tree of a weighted network with the least total weight

Our goal, to find the MMST, the minimum spanning tree. Let;'s begin to sort this out…



Here's a game. Connect the dots. Connect the vertices to make a network. For each bridge you create, you get $10.00. For each circuit you create, you pay a $20.00 penalty.

OK? Let's go.

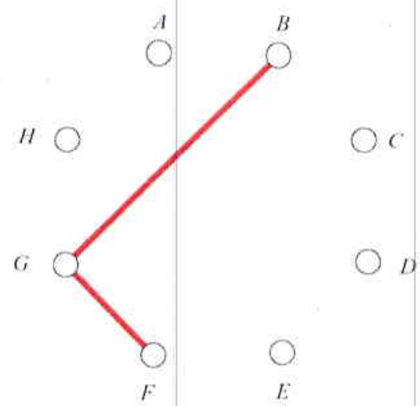


$10

⌘

Connect the Dots—and then stop

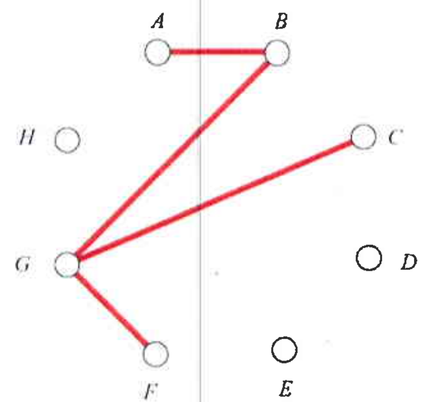
- M=2



$20

Connect the Dots—and then stop

- M=4

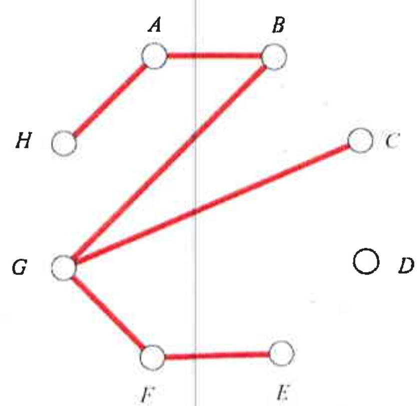


$30 and $40

⌘

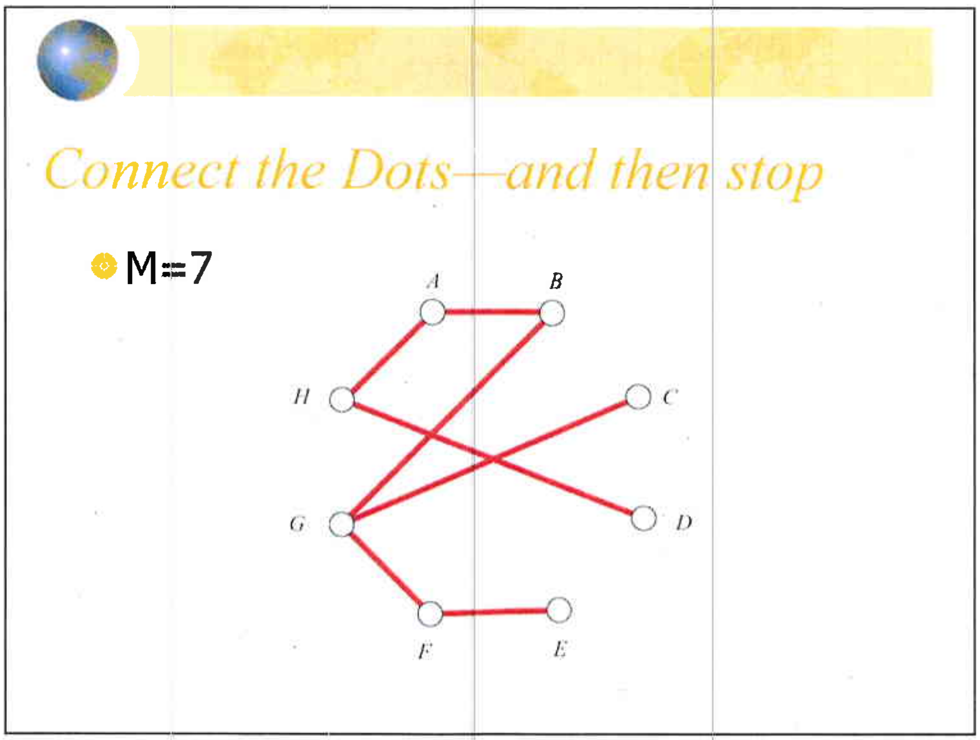
Connect the Dots—and then stop

- M=6



$50 and $60

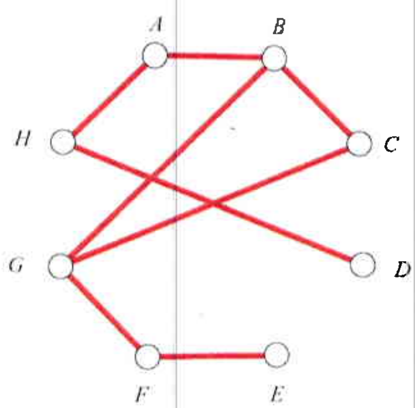
⌘



$ 70.00

Connect the Dots—and then stop

- M=8



Now you've gotten greedy. You drop back to $50 because you've created a circuit.

M=7 is as good as it gets. Add any other edge and we'll have a circuit.

As M increases, the number of circuits goes up rather quickly, and the number of bridges goes down.

So let's summarize…

Tree Property 1

- In a tree, there is one and only one path joining any two vertices.
- If there is one and only one path joining any two vertices of a graph, the the graph must be a tree.

Trees are barely connected. That means several things.

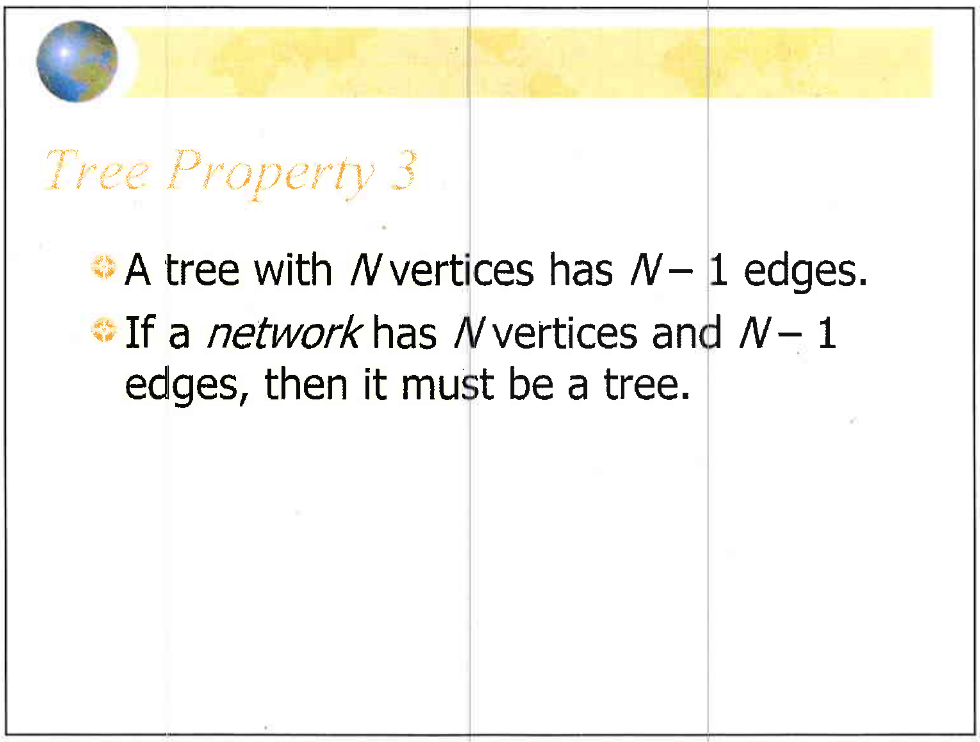
⌘

Tree Property 2

- In a tree, every edge is a bridge.
- If every edge of a graph is a bridge, then the graph must be a tree.

In a tree, every edge is a bridge.

If every edge of a graph is a bridge, then the graph must be a tree.



A tree with N vertices has $N-1$ edges.

If a *network* has N vertices and $N-1$ edges, then it must be a tree.

The word *network* here is important. A disconnected graph with N vertices can have $N-1$ edges and not be a tree.

Tree Property 4

- If a network has N vertices and M edges, then $M > N - 1$.
  - We refer to the difference $R = M - (N - 1)$ as the *redundancy* of the network.
- If $R = 0$, the network is a tree. If $R > 0$, the network has circuits and is not a tree.

In the real world, redundancy may be important for reliability, but that's a different discussion.

So: a minimum spanning tree with N vertices must have N-1 edges, or redundancy = 0.