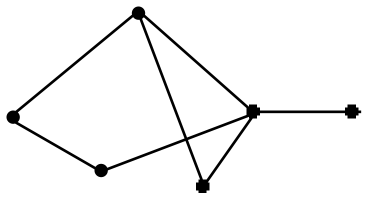
ECE 130 - TREES - INVESTIGATION 22
SPANNING TREES

WINTER 2004 **A.P. FELZER**

To do "well" on this investigation you must not only get the right answers but must also do neat, complete and concise writeups that make obvious what each problem is, how you're solving the problem and what your answer is. You also need to include appropriate graphs and tables.

The main objective of this Investigation is spanning trees. We also introduce tree traversals since they turn out to be particularly useful in applications involving calculations.

1. Given a simple graph like the following



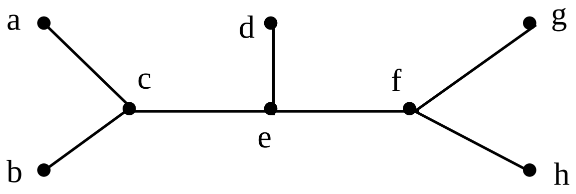
a **spanning tree** is a tree that is a subgraph of G containing all the vertices of G together with a subset of the edges

a. Find a spanning tree of G
b. Find another spanning tree of G

2. There are several ways to find spanning trees of simple graphs. One way is **depth-first search** as follows

(1) Arbitrarily choose a vertex to be the root
(2) Starting from the root add edges and vertices to form a path until you reach a leaf or no more edges can be added without forming a circuit
(3) If all the vertices are in the tree then you're done. Otherwise move back one vertex and repeat until all the vertices are included

a. Use depth-first search to find the spanning tree of the following simple graph

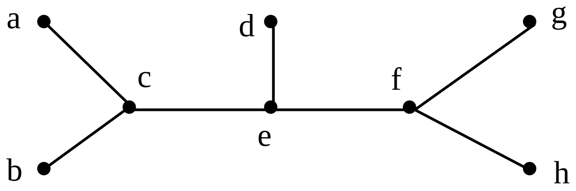


b. Find the spanning tree of a graph of your own containing at least 8 vertices

3. Another way to find a spanning tree is by breadth first search as follows

(1) Arbitrarily choose a vertex to be the root
(2) Add all the vertices adjacent to the root
(3) For each vertex in (2) add all adjacent vertices as long as they don't form a circuit
(4) Continue until all vertices are included

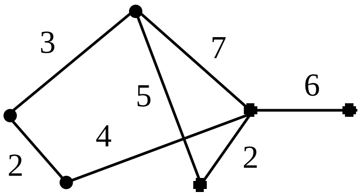
a. Use depth-first search to find the spanning tree of the following simple graph



b. Find the spanning tree of a graph of your own containing at least 8 vertices

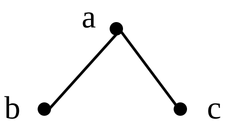
4. How are spanning trees useful for web spiders (crawlers)

5. Given a weighted graph as follows



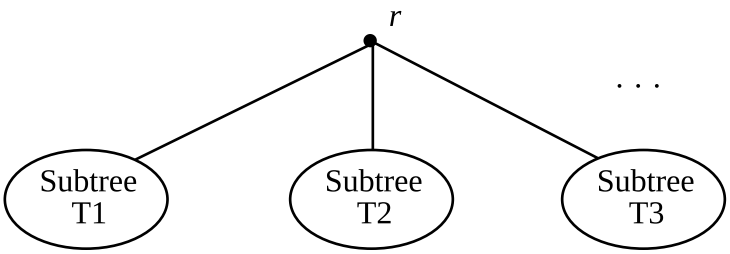
a. What would you expect is meant by a **minimum spanning tree** of a weighted graph
b. Find, on your own, an algorithm for a minimum spanning tree of a weighted graph
c. Try out your algorithm on the above tree. How did it work
d. Try out your algorithm on a graph of your own

6. The objective of this and the following problems is to intrtoduce **tree traversals** - paths that visit every vertex of a tree. Some examples for the following tree are



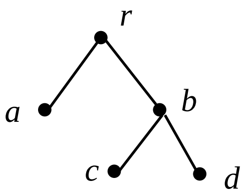
a - *b* - *c*, *b* - *a* - *c*. Note that in general the path taken by a tree traversal will retrace its steps but that each vertex is listed only once - the first time it's visited
a. Find another tree traversal for this graph
b. Make up your own tree and then find at least three tree traversals

7. There are a number of algorithms for doing tree traversals. This problem introduces the **preorder traversal** for a tree T as follows



(1) Start with the root *r*
(2) Go to an adjacent vertex of subtree T1 on the far left
(3) Proceed with the preorder algorithm until all vertices in T1 have been visited and then move on to subtree T2 and the rest of the subtrees

a. Find the preorder traversal for the following tree



b. Make up and do your own example of a preorder traversal

c. Describe how the preorder algorithm is recursive