CHAPTER 7

TREES

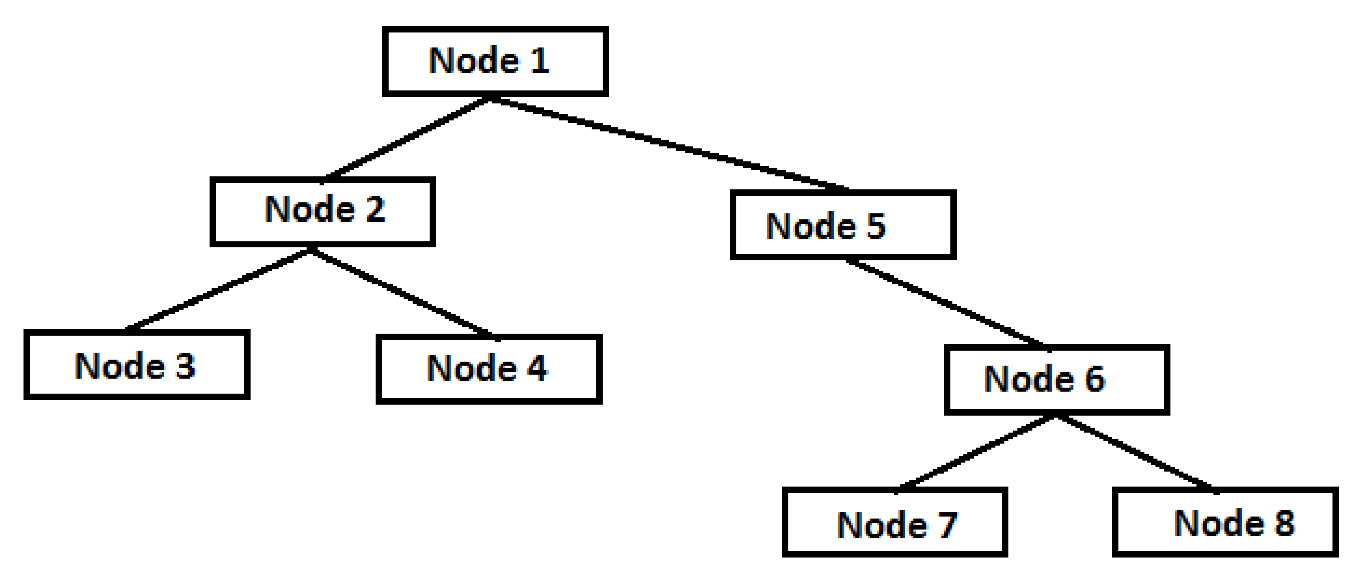
HOMEWORK

- **Again, all homework is from the Exercises**
  - **No problems are from the Review Exercises**
- **Section 7.1, (p. 330), #1-4, 14, 15, 18, 19, 24-25**
- **Section 7.3, (p. 342), #2, 5, 7-9**
- **Section 7.5, (p. 354), #1, 5-6**
- **Section 7.6, (p. 360), #1-3, 6-8, 11-15, 16-17**
- **Section 7.7, (p. 366), #1-4, 9**
- **Section 7.8, (p. 375). #1-3, 7-12**

TREES

- **A tree T is a graph where**
  - **there is a single path from any vertex to any other**
- **Usually a tree has a root**
  - **A root is just a special vertex**
  - **A tree with a root is called a rooted tree**

VOCABULARY



- Node, edge, leaf (terminal vertex), interior node
- Height of the tree, level of a vertex
- Most trees have roots, and have this additional vocabulary
  - Sibling, parent, child, ancestor, descendant

THREE QUICK NOTES

- Trees are always connected
- If a tree has n nodes (vertices), it has n-1 edges
- A tree has no cycles

USES OF TREES

- Lots of types of data can be stored naturally in a tree
  - A company's structure (organization chart)
  - File hierarchy on a disk
  - Family trees (Example: p. 331, Fig. 7.2.1)
  - Tournaments (A very common use)
- A tree can be used for making decisions
  - A game tree is one example of a decision tree

HUFFMAN CODES

- Sometimes we compress data
  - One reason to do that is to transmit it across the internet
- One method of compression is Huffman codes
- These can be pictured using a tree
- First, we use a Huffman code to decode data
  - p. 327, Figure 7.1.10
- Notice that different characters might have different numbers of bits

CONSTRUCTING A HUFFMAN CODE TREE FROM DATA-THE RAW DATA

- p. 330, #24

Letter	Frequency
A	5
B	6
C	6
D	11
E	20

CONSTRUCTING A HUFFMAN CODE TREE FROM DATA-THE ALGORITHM (PART 1)

- Start with the smallest two frequencies
- Add them
- Find the next smallest frequency
- Add it to the sum
- Continue the process until you have only two numbers
  - 5,6,6,11,20 >>> 11, 6,11,20 >>> 11,17,20 >>> 28,20

CONSTRUCTING A HUFFMAN CODE TREE FROM DATA-THE ALGORITHM (PART 2)

- Draw a tree with two edges
- Label the left edge 1 and the right edge 0
- Make the two numbers the leaves
- Repeat this process, undoing the summing process you just did

VOCABULARY FOR PROOF

- An acyclic graph
  - A graph with no cycles
- "The following are equivalent"
  - Common abbreviation: TFAE
  - What does it mean?
    - It means that if any one of the parts is true, so are all the others
  - How do we usually prove it?

A USEFUL THEOREM (T7.2.3, P.333)

- Let T be a graph with n vertices. The following are equivalent
  a) T is a tree
  b) T is connected and acyclic
  c) T is connected and has n-1 edges
  d) T is acyclic and has n-1 edges

PROOF OF T7.2.3 (A→B)

- a→b: A tree is connected and acyclic by definition

PROOF OF T7.2.3 (B→C)

- b→c: Suppose T is connected and acyclic
  - Proof by induction on the number of vertices
  - The Base Step: 1 vertex
    - This graph is a tree

PROOF OF T7.2.3 (B→C)

- The Inductive Step
  - Assume every connected, acyclic graph with n vertices has n-1 edges
  - Show that if T is a connected, acyclic graph with n+1 vertices, then T has n edges
- Find a simple path of maximum length
  - It can't be a cycle because graph is acyclic
- So the path has a "starting point": a vertex of degree 1
- Remove that vertex and its incident edge
- This new graph has n vertices
- By the induction hypothesis, this new graph is a tree with n-1 edges
- Put the vertex and edge back in to get the result

PROOF OF T7.2.3 (C→D)

- We must show graph to be acyclic
- If the graph has no cycles, we are done
- Otherwise, (Proof by contradiction)
  - if the graph has a cycle, remove edges, one at a time, until no more cycles exist
  - Notice that this doesn't disconnect the graph and it doesn't remove vertices
  - If there is a cycle, we have removed at least one edge
  - The new graph is connected and acyclic
  - So, by b→c, the new graph has n-1 edges
  - Put back the removed edges to get the original graph
  - The original graph now has more than n-1 edges
  - This contradicts our assumption

PROOF OF T7.2.3 (D→A)

- The graph has no loops because a loop is a cycle
- It also cannot have multiple edges
  - These are also cycles
- So there is a unique path between any pair of vertices
  - The graph may be disconnected, though
- If it is disconnected, (Proof by contradiction)
  - The proof continues on the next slide

PROOF OF T7.2.3 (D→A, CONTINUED)

- If it is disconnected, (Proof by contradiction)
  - Suppose the pieces are $G_1, \ldots, G_n$, where $n >= 2$
  - The total number of vertices in each G_i is n_i
  - The total number of edges in each G_i is $n_i - 1$, by $b \to c$
    - $n - 1 = (n_1 - 1) + (n_2 - 1) + \ldots + (n_k - 1)$ counting edges
    - $< n_1 + n_2 + \ldots + (n_k - 1)$
    - $= n - 1$ counting vertices
  - This is impossible, and so T is connected.

SPANNING TREES

- **A spanning tree is a subgraph of a graph that contains all vertices**
- **A graph may have more than one spanning tree**

GRAPHS AND SPANNING TREES

- A graph with a spanning tree must be connected
  - Proof by contradiction
  - Suppose you have a graph with a spanning tree that is not connected
  - Find two vertices in different components
  - Since there is a spanning tree, there is a path between these two vertices
  - This is a contradiction, and the graph must be connected

HOMEWORK

- Again, all homework is from the Exercises
  - No problems are from the Review Exercises
- Section 7.1, (p. 330), #1-4, 14, 15, 18, 19, 24-25
- Section 7.3, (p. 342), #2, 5, 7-9
- Section 7.5, (p. 354), #1, 5-6
- Section 7.6, (p. 360), #1-3, 6-8, 11-15, 16-17
- Section 7.7, (p. 366), #1-4, 9
- Section 7.8, (p. 375). #1-3, 7-12