Trees

P. Danziger

1 Trees

Definition 1

1. *A graph is called circuit free if it contains no circuits.*
2. *A graph which is circuit free and connected is called a tree.*
3. *A graph which is circuit free is called a forest*
   (Note that a forest is a collection of trees.)
4. *A trivial tree is a graph consisting of a single vertex.*
5. *The empty tree is the graph consisting of no vertices or edges.*

Note that a tree must be simple (no loops or parallel edges).
Trees are useful in sorting and searching problems.

Lemma 2 *Any non trivial tree has at least one vertex of degree 1.*

Proof: Let $G = (V, E)$ be a non trivial tree (i.e. $|V| > 1$). Pick any vertex $v \in V$.
Randomly follow any path from v without reusing any edges.
We cannot return to any vertex in our path so far (otherwise G would contain a circuit) and we must eventually terminate (since $|V|$ is finite).
The only way we can get stuck is if we hit a vertex of degree 1. □

Definition 3

1. *A vertex of degree 1 in a tree is called a leaf, or terminal vertex.*
2. *A vertex of degree greater than 1 in a tree is called an internal vertex or a branch vertex.*

Theorem 4 *For any positive integer n a tree with n vertices has $n-1$ edges*

Proof: By induction on n.
Base Case Let $n = 1$, this is the trivial tree with 0 edges. So true for $n = 1$.
Inductive Step Let $n = k$ and assume true for k. i.e. every tree with k vertices has $k - 1$ edges.
Let T be a tree with $k + 1$ vertices.
Let v be a leaf of V (exists by previous Lemma).
Consider the tree T' obtained by deleting v from T.
T' has k vertices, so by the inductive hypothesis it has $k - 1$ edges.
But v is a leaf, and so has degree 1, thus T has one more edge than T'. □

Theorem 5 *If G is a connected graph with n vertices and $n-1$ edges then G is a tree.*

Proof By contradiction.
Let G be a connected graph with n vertices and $n-1$ edges.
Suppose that G is not a tree, thus G has a circuit.
Remove an edge from this circuit to get a new graph G_1.
If G_1 still has circuits repeat this process of removing edges from circuits until we get a circuit free graph.
Note that removing edges from a circuit never disconnects the graph.
Suppose we have repeated this process k times, we a re left with a tree with n vertices and $n-1-k$ edges.
But this contradicts the previous Theorem. □

1.1 Rooted Trees

Definition 6

1. *A rooted tree is a tree with a distinguished vertex called the root.*
2. *The level of a vertex in a rooted tree is the number of edges from the vertex to the root.*
3. *the height of a rooted tree is the maximum level of any vertex in the tree.*
4. *Given an internal vertex in a rooted tree its children are those vertices adjacent to it and one level higher.*
5. *If u and v are vertices in a rooted tree, with u a child of v, then v is called the parent of u.*
6. *Two vertices which are children of the same vertex are called siblings.*
7. *Given two vertices u and v in a rooted tree, if u lies on the path from v to the root then u is called an ancestor of v, and v is called a descendent of u.*

1.2 Binary Trees

Definition 7

1. *A binary tree is a rooted tree in which each vertex has at most two children.*
2. *A full binary tree is a binary tree in which each internal vertex has exactly two children.*
3. *The children of a vertex v in a binary tree can be identified as the left child and right child of v respectively.*
4. *Given an internal vertex, v, of a binary tree the left subtree is the binary tree consisting of all the descendents of the left child of v, whose root is left child of v. Similarly for right subtree.*

Theorem 8 *For any positive integer n, if T is a full binary tree with n internal vertices, then T has $n+1$ leaves and a total of $2n+1$ vertices.*

Proof: Let T be a full binary tree with n internal vertices.
Every internal vertex of T has 2 children, so the number of vertices that have a parent is equal to twice the number of internal vertices, or $2n$.
Let m be the total number of vertices of T.

$m =$ The number of vertices which have a parent + those that don't
$=$ $2n + 1$ (the root)
$=$ The number of internal vertices + The number of leaves
$=$ $n +$ The number of leaves.

So the number of leaves $= 2k + 1 - k = k + 1$. □

Theorem 9 *If T is a binary tree with t leaves and height h then $t \leq 2^h$ (i.e. $\log_2 t \leq h$).*

Proof: By induction on h
Base Case Let $h = 0$, then T is either the empty tree, or the trivial tree.
Empty: $h = 0$, $t = 0 \leq 2^0 = 1$
Trivial: $h = 0$, $t = 1 \leq 2^1 = 1$.
Inductive Step Let $h = k$, and assume true for k. That is all trees with height k have at most 2^k terminal vertices.
Let T be a binary tree with height k, hence by the inductive hypothesis the number of leaves of T, $t \leq 2^k$.
We create a new binary tree T' by adding vertices to the leaves of T.
The largest number of leaves we can derive from T is obtained by adding exactly two vertices to each leaf of T.
Thus T' has at most $2t \leq 2 \cdot 2^k = 2^{k+1}$ leaves. □

2 Spanning Trees

Definition 10 *A spanning tree of a graph G is a subgraph of G which contains every vertex of G and is a tree.*

Theorem 11 *Every connected graph has a spanning tree. Any two spanning trees of a graph have the same number of edges.*

Proof: Let G be a connected graph.
If G is circuit free it is a tree and hence its own spanning tree.
If not G has a circuit.
Remove an edge of this circuit to get a new graph G_1.
If G_1 has a circuit remove an edge from the circuit to obtain a new graph G_2.
Continue until we reach a circuit free graph, G_k for some k.
G_k is a spanning tree for G.

Theorem 12 *Any two spanning trees of a graph have the same number of edges.*

Let n be the number of vertices of G.
If T is a spanning tree for G then T has n vertices, and hence $n - 1$ edges. □
Note that the proof of Theorem 11 gives an algorithm for finding a spanning tree of any connected graph G.

Definition 13

- *A* weighted graph *G is a graph with weights (numbers) assigned to each edge.*
- *The* total weight *of a weighted graph G is the sum of all the weights on the edges.*
- *A* minimal spanning tree *of a weighted graph G is a spanning tree of G with the least possible weight.*

Input a weighted graph $G = (V, E)$ with n vertices ($|V| = n$)

Kruskal's Algorithm

$T := (V, \phi)$; $m := 0$
While $m < n - 1$
    Find $e \in E$ with minimal weight.
    $E := E - \{e\}$
    If $T \cup \{e\}$ does not contain a circuit
        $T := T \cup \{e\}$; $m := m + 1$
Return T

Prim's Algorithm

Pick a random $v \in V$
$V := V - \{v\}$; $U := \{v\}$; $F := \phi$
$T := (U, F)$
For i from 1 to $n - 1$
    Find a minimal weight $e \in E$ so that $e = \{u, w\}$ with $u \notin V$ and $w \in V$
    $U := U \cup \{w\}$; $F := F \cup \{e\}$
    $V := V - \{w\}$; $E := E - \{e\}$
Return $T = (U, F)$.