Discrete Mathematics

Trees

H. Turgut Uyar   Ayşegül Gençata Yayımlı   Emre Harmancı

2001-2016

License

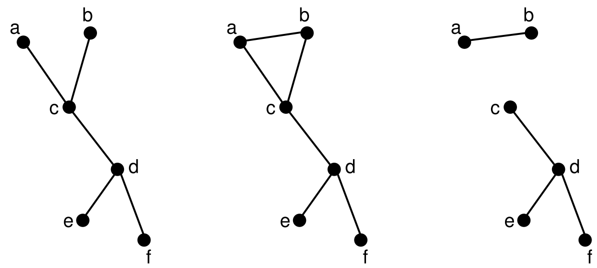
Topics

Tree

Definition

tree: connected graph with no cycle

examples



Tree Theorems

Theorem

T is a tree (T is connected and contains no cycle).

$\Leftrightarrow$

There is one and only one path between any two distinct nodes in T.

$\Leftrightarrow$

T is connected, but if any edge is removed it will no longer be connected.

$\Leftrightarrow$

T contains no cycle, but if an edge is added between any pair of nodes one and only one cycle will be formed.

Tree Theorems

Theorem

$$|E| = |V| - 1$$

- proof method: induction on the number of edges

Tree Theorems

Proof: Base step.

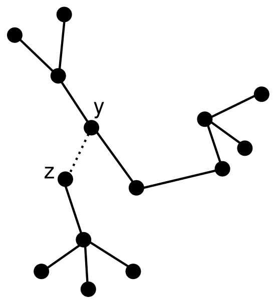
- $|E| = 0 \Rightarrow |V| = 1$
- $|E| = 1 \Rightarrow |V| = 2$
- $|E| = 2 \Rightarrow |V| = 3$

- assume that $|E| = |V| - 1$ for $|E| \leq k$

Tree Theorems

Proof: Induction step.

- $|E| = k + 1$



- remove edge (y, z):
  $T_1 = (V_1, E_1)$, $T_2 = (V_2, E_2)$

$$\begin{aligned} |V| &= |V_1| + |V_2| \\ &= |E_1| + 1 + |E_2| + 1 \\ &= (|E_1| + |E_2| + 1) + 1 \\ &= |E| + 1 \end{aligned}$$

$\square$

Tree Theorems

Theorem

$$T \text{ is a tree } (T \text{ is connected and contains no cycle}).$$
$$\Leftrightarrow$$
$$T \text{ is connected and } |E| = |V| - 1.$$
$$\Leftrightarrow$$
$$T \text{ contains no cycle and } |E| = |V| - 1.$$

Tree Theorems

Theorem

In a tree, there are at least two nodes with degree 1.

Proof.

- $2|E| = \sum_{v \in V} d_v$
- assume: only 1 node with degree 1:
  $\Rightarrow 2|E| \geq 2(|V| - 1) + 1$
  $\Rightarrow 2|E| \geq 2|V| - 1$
  $\Rightarrow |E| \geq |V| - \frac{1}{2} > |V| - 1$

$\square$

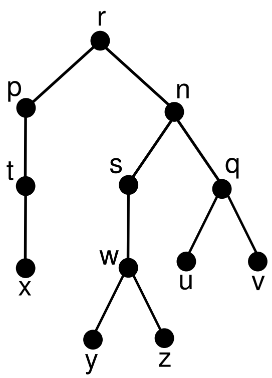
Rooted Tree

- hierarchy between nodes
- creates implicit direction on edges: in and out degrees
- in-degree 0: root (only 1 such node)
- out-degree 0: leaf
- not a leaf: internal node

Node Level

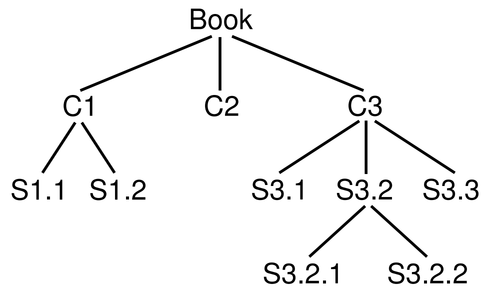
- level of node: distance from root
- *parent*: adjacent node closer to root (only 1 such node)
- *child*: adjacent nodes further from root
- *sibling*: nodes with same parent
- depth of tree: maximum level in tree

Rooted Tree Example



- root: r
- leaves: $x\ y\ z\ u\ v$
- internal nodes: $r\ p\ n\ t\ s\ q\ w$
- parent of y: w
  children of w: y and z
- y and z are siblings

Rooted Tree Example



Book
- C1
  - S1.1
  - S1.2
- C2
- C3
  - S3.1
  - S3.2
    - S3.2.1
    - S3.2.2
  - S3.3

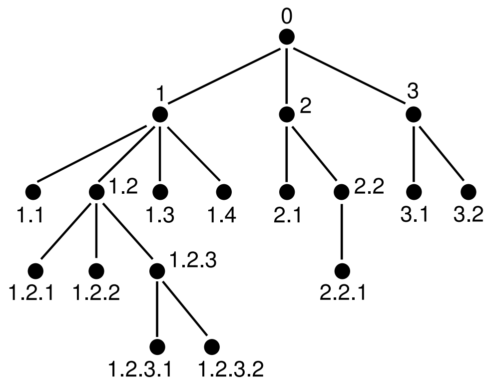
Ordered Rooted Tree

- siblings ordered from left to right
- universal address system
- root: 0
- children of root: $1, 2, 3, \ldots$
- v: internal node with address a
  children of v: $a.1, a.2, a.3, \ldots$

Lexicographic Order

- address A comes before address B if one of:
- $A = x_1 x_2 \ldots x_i x_j \ldots$
  $B = x_1 x_2 \ldots x_i x_k \ldots$
  x_j comes before x_k
- $A = x_1 x_2 \ldots x_i$
  $B = x_1 x_2 \ldots x_i x_k \ldots$

Lexicographic Order Example



- 0 - 1 - 1.1 - 1.2
  - 1.2.1 - 1.2.2 - 1.2.3
  - 1.2.3.1 - 1.2.3.2
  - 1.3 - 1.4 - 2
  - 2.1 - 2.2 - 2.2.1
  - 3 - 3.1 - 3.2

Binary Trees

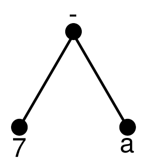
- $T = (V, E)$ is a binary tree:
  $\forall v \in V \ [d_v{}^o \in \{0, 1, 2\}]$
- $T = (V, E)$ is a *complete* binary tree:
  $\forall v \in V \ [d_v{}^o \in \{0, 2\}]$

Expression Tree

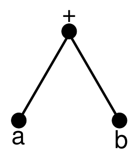
- binary operations can be represented as binary trees
- root: operator, children: operands
- mathematical expression can be represented as trees
- internal nodes: operators, leaves: variables and values

Expression Tree Examples

$7 - a$

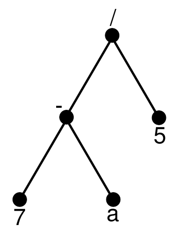


$a + b$

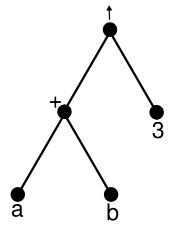


Expression Tree Examples

$$\frac{7-a}{5}$$

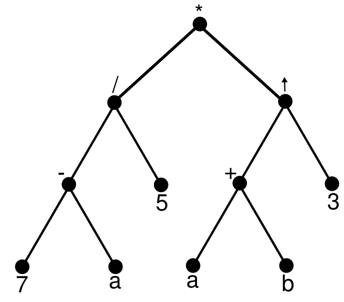


$$(a+b)^3$$



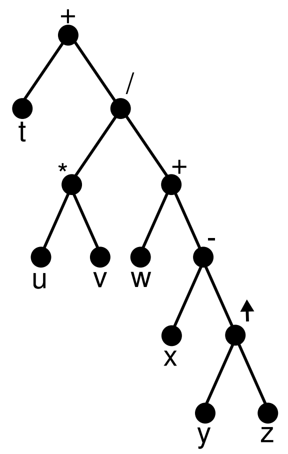
Expression Tree Examples

$$\frac{7-a}{5} \cdot (a+b)^3$$



Expression Tree Examples

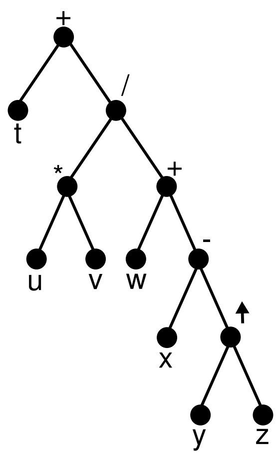
$$t + \frac{u * v}{w + x - y^z}$$



Expression Tree Traversals

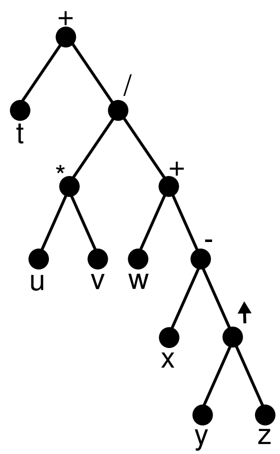
1. inorder traversal:
   traverse left subtree, visit root, traverse right subtree
2. preorder traversal:
   visit root, traverse left subtree, traverse right subtree
3. postorder traversal (reverse Polish notation):
   traverse left subtree, traverse right subtree, visit root

Inorder Traversal Example



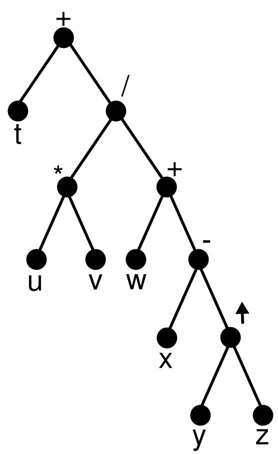
$t + u * v / w + x - y \uparrow z$

Preorder Traversal Example



$+ \; t \; / * u \; v + w - x \uparrow y \; z$

Postorder Traversal Example



$t \; u \; v * w \; x \; y \; z \uparrow - + / +$

Expression Tree Evaluation

- inorder traversal requires parantheses for precedence
- preorder and postorder traversals do not require parantheses

Postorder Evaluation Example

$t\ u\ v\ *\ w\ x\ y\ z\ \uparrow\ -\ +\ /\ +$
$4\ 2\ 3\ *\ 1\ 9\ 2\ 3\ \uparrow\ -\ +\ /\ +$

4	2	3	$*$			
4	6	1	9	2	3	$\uparrow$
4	6	1	9	8	$-$	
4	6	1	1	$+$		
4	6	2	$/$			
4	3	$+$				
7						

Regular Trees

- $T = (V, E)$ is an m-ary tree:
  $\forall v \in V\ [d_v{}^o \leq m]$
- $T = (V, E)$ is a complete m-ary tree:
  $\forall v \in V\ [d_v{}^o \in \{0, m\}]$

Regular Tree Theorem

Theorem
$T = (V, E)$: complete m-ary tree

- *n: number of nodes*
- *l: number of leaves*
- *i: number of internal nodes*

- $n = m \cdot i + 1$
- $l = n - i = m \cdot i + 1 - i = (m - 1) \cdot i + 1$

$$i = \frac{l - 1}{m - 1}$$

Regular Tree Examples

- how many matches are played in a tennis tournament with 27 players?

- every player is a leaf: $l = 27$
- every match is an internal node: $m = 2$
- number of matches: $i = \frac{l-1}{m-1} = \frac{27-1}{2-1} = 26$

Regular Tree Examples

- how many extension cords with 4 outlets are required to connect 25 computers to a wall socket?

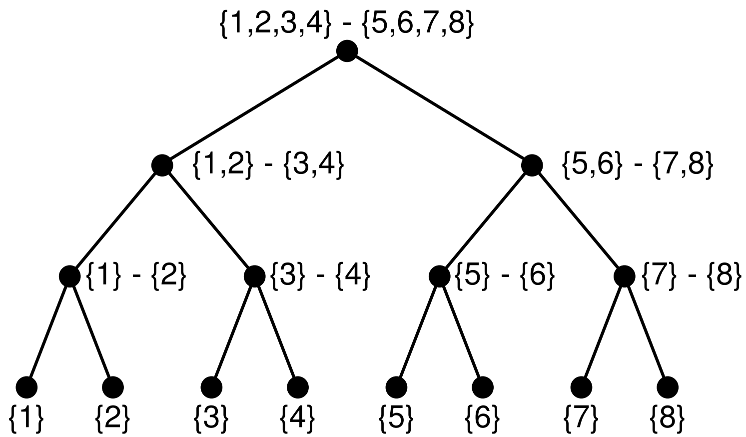
- every computer is a leaf: $l = 25$
- every extension cord is an internal node: $m = 4$
- number of cords: $i = \frac{l-1}{m-1} = \frac{25-1}{4-1} = 8$

Decision Trees

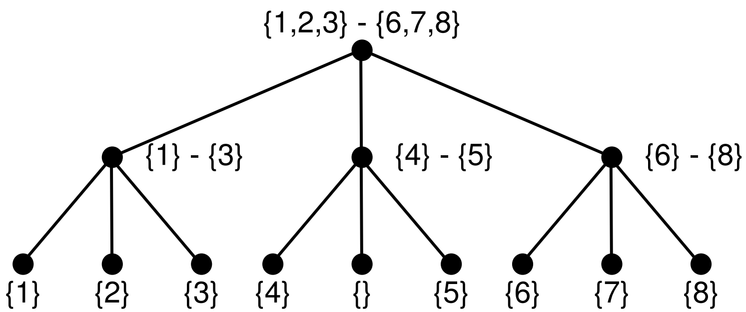
- one of 8 coins is counterfeit (heavier)
- find counterfeit coin using a beam balance

- depth of tree: number of weighings

Decision Trees



Decision Trees



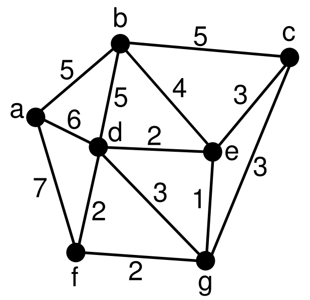
Spanning Tree

- $T = (V', E')$ is a spanning tree of $G(V, E)$:
  T is a subgraph of G
  T is a tree
  $V' = V$
- minimum spanning tree:
  total weight of edges in E' is minimal

Kruskal's Algorithm

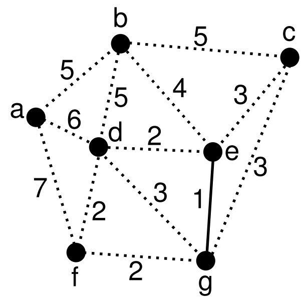
1. $G' = (V', E')$, $V' = \emptyset$, $E' = \emptyset$
2. select $e = (v_1, v_2) \in E - E'$ such that:
   $E' \cup \{e\}$ contains no cycle, and $wt(e)$ is minimal
3. $E' = E \cup \{e\}$, $V' = V' \cup \{v_1, v_2\}$
4. if $|E'| = |V| - 1$: result is G'
5. go to step 2

Kruskal's Algorithm Example



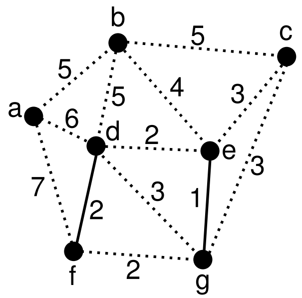
- minimum weight: 1
  (e, g)
- $E' = \{(e, g)\}$
- $|E'| = 1$

Kruskal's Algorithm Example



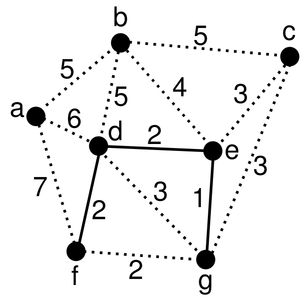
- minimum weight: 2
  $(d, e), (d, f), (f, g)$
- $E' = \{(e, g), (d, f)\}$
- $|E'| = 2$

Kruskal's Algorithm Example



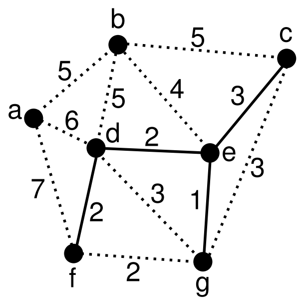
- minimum weight: 2
  $(d, e), (f, g)$
- $E' = \{(e, g), (d, f), (d, e)\}$
- $|E'| = 3$

Kruskal's Algorithm Example



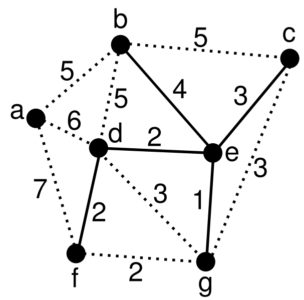
- minimum weight: 2
  (f, g) forms a cycle
- minimum weight: 3
  $(c, e), (c, g), (d, g)$
  (d, g) forms a cycle
- $E' = \{(e, g), (d, f), (d, e), (c, e)\}$
- $|E'| = 4$

Kruskal's Algorithm Example



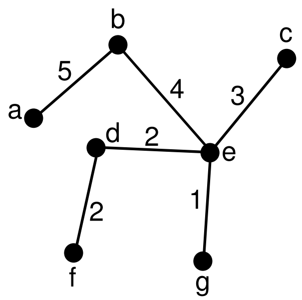
- $E' = \{$
  $(e, g), (d, f), (d, e),$
  $(c, e), (b, e)$
  $\}$
- $|E'| = 5$

Kruskal's Algorithm Example



- $E' = \{$
  $(e, g), (d, f), (d, e),$
  $(c, e), (b, e), (a, b)$
  $\}$
- $|E'| = 6$

Kruskal's Algorithm Example

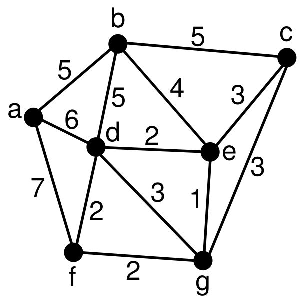


- total weight: 17

Prim's Algorithm

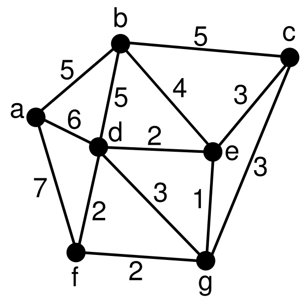
1. $T' = (V', E'), E' = \emptyset, v_0 \in V, V' = \{v_0\}$
2. select $v \in V - V'$ such that for a node $x \in V'$ $e = (x, v), E' \cup \{e\}$ contains no cycle, and $wt(e)$ is minimal
3. $E' = E \cup \{e\}, V' = V' \cup \{x\}$
4. if $|V'| = |V|$: result is T'
5. go to step 2

Prim's Algorithm Example



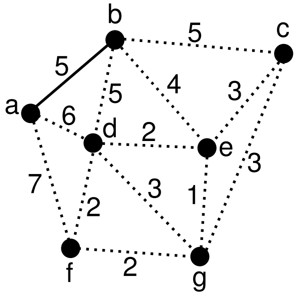
- $E' = \emptyset$
- $V' = \{a\}$
- $|V'| = 1$

Prim's Algorithm Example



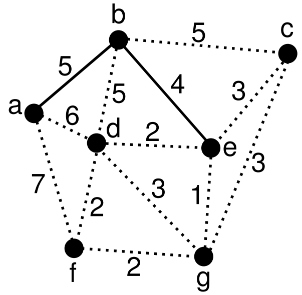
- $E' = \{(a, b)\}$
- $V' = \{a, b\}$
- $|V'| = 2$

Prim's Algorithm Example



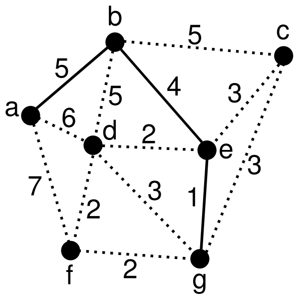
- $E' = \{(a, b), (b, e)\}$
- $V' = \{a, b, e\}$
- $|V'| = 3$

Prim's Algorithm Example



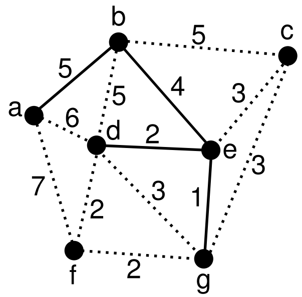
- $E' = \{(a, b), (b, e), (e, g)\}$
- $V' = \{a, b, e, g\}$
- $|V'| = 4$

Prim's Algorithm Example



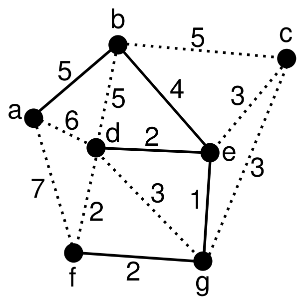
- $E' = \{(a, b), (b, e), (e, g), (d, e)\}$
- $V' = \{a, b, e, g, d\}$
- $|V'| = 5$

Prim's Algorithm Example



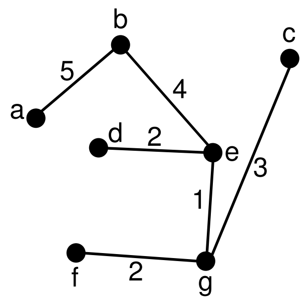
- $E' = \{$
  $(a, b), (b, e), (e, g),$
  $(d, e), (f, g)$
  $\}$
- $V' = \{a, b, e, g, d, f\}$
- $|V'| = 6$

Prim's Algorithm Example



- $E' = \{$
  $(a, b), (b, e), (e, g),$
  $(d, e), (f, g), (c, g)$
  $\}$
- $V' = \{a, b, e, g, d, f, c\}$
- $|V'| = 7$

Prim's Algorithm Example



- total weight: 17

References

Required Reading: Grimaldi

- Chapter 12: Trees
  - 12.1. Definitions and Examples
  - 12.2. Rooted Trees
- Chapter 13: Optimization and Matching
  - 13.2. Minimal Spanning Trees:
    The Algorithms of Kruskal and Prim