Week 4: Graph Theory and Structural Induction

CSC 236:Introduction to the Theory of Computation

Summer 2024

Instructor: Lily

csc236-2024-05@cs.toronto.edu

Announcement

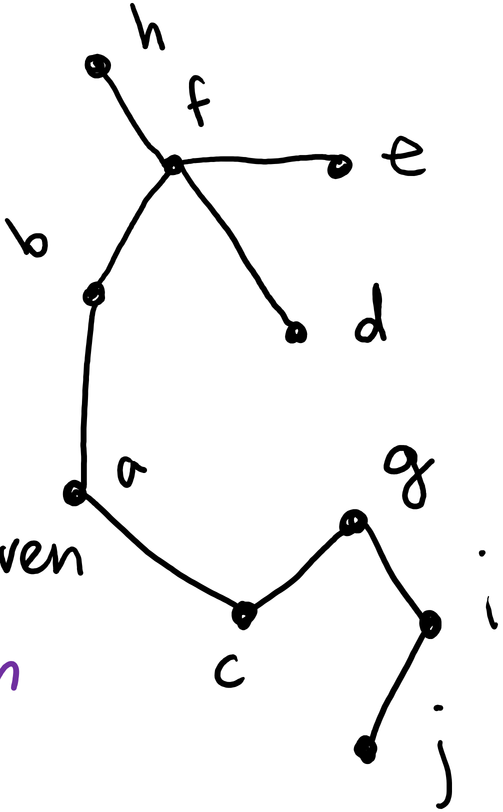
- Peer review (5 points)
  - [1 point] Complete all your assigned reviews
  - [1 point] For each review accuracy of marking (for now: if you give mark a, and the TA gives mark t
    - $|a - t| \leq 1$: full marks
    - $|a - t| \in (1, 2]$: $+0.75$
    - $|a - t| \in (2, 3]$: $+0.5$
    - $|a - t| \in (3,4]$: $+0.25$
    - $|a - t| > 4$: no marks
- A2 Q1 (a) now unmarked, A2 Q1 (d) question modified, A2 Q2 (a)-(c) hints modified, Q2 (d) removed.

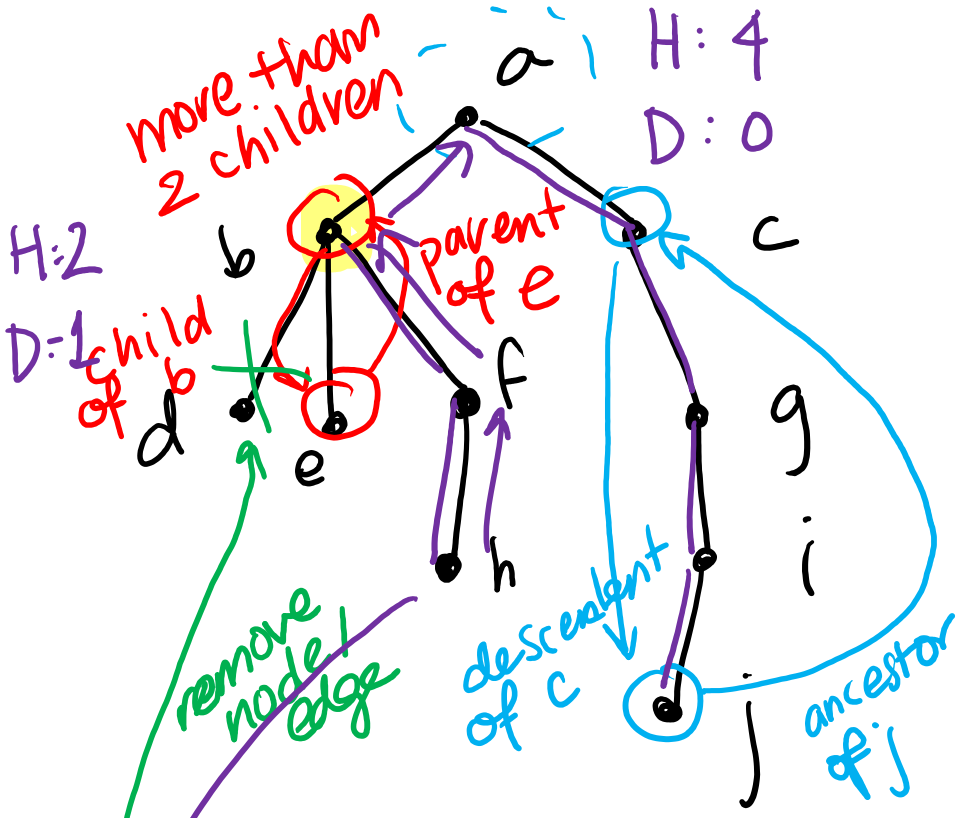
Trees

- Root : node in tree without a parent
- Binary tree : every node has ≤ 2 children
- Height : (of u) length of longest path from u down to leaf node

of tree = height of root

VS DEPTH : length of path to root





IS NOT BINARY TREE

IS BINARY TREE

H:0 D:3

Recursively Defined Sets

alphabet $\{1, +\}$

standard $0, 1, 2, 3, 4, \dots$

- $\mathbb{N}$

my way $\underbrace{\ }_{\text{empty}}, 1, 1+1, 1+1+1, \dots$

emptyset

- $\varepsilon \in \overline{\mathbb{N}}$
- $1 \in \overline{\mathbb{N}}$
- $a, b \in \overline{\mathbb{N}} \Rightarrow a+b \in \overline{\mathbb{N}}$

smallest set satisfying this is $\mathbb{N}$

$1 \in \overline{\mathbb{N}}$, $\underbrace{1+1}_{=2} \in \overline{\mathbb{N}}$

- Sequence of balanced brackets

alphabet = $\{ (,) \}$ YES $()()$ ✓ $((()))$ ✓ NO $(()$

$\varepsilon \in S$

if $a, b \in S \Rightarrow ab \in S$

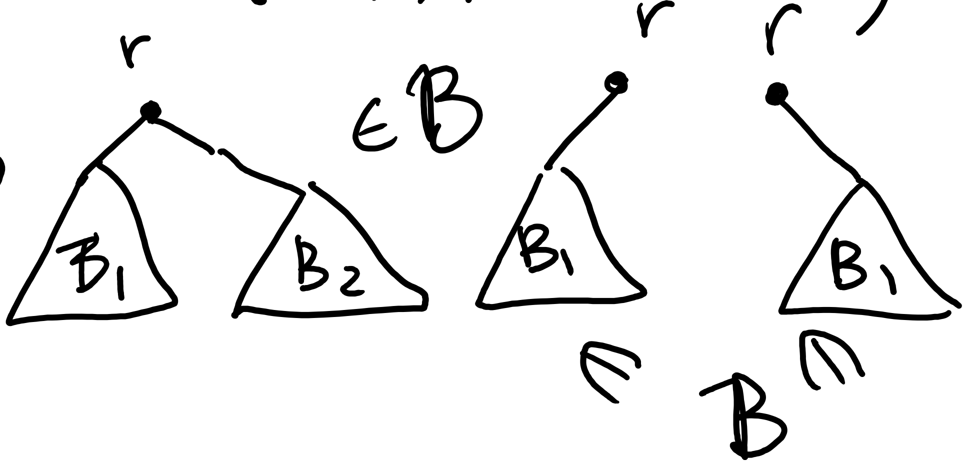
if $a \in S \Rightarrow (a) \in S$

$(((\varepsilon)))$

- Binary trees $\mathcal{B}$

single node $\overset{r}{\bullet} \in \mathcal{B}$

if $B_1, B_2 \in \mathcal{B} \Rightarrow$



Structural Induction

k-ary tree := max number of children is k

Prove: every non-empty binary tree has one more node than edge.

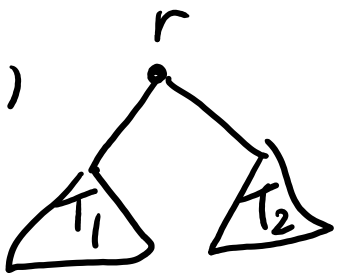
to show for all trees, replicate arguement for k-ary tree where k is max deg

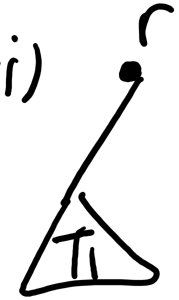
$P(T)$: for binary tree T, T has one more node than edge.

Base case: let $T =$ 

$\underbrace{|V(T)|}_{\#\text{vertices of } T} = 1$, $\underbrace{|E(T)|}_{\#\text{edges of } T} = 0$

Inductive step: if $T_1, T_2 \in B$

(i) 

(ii) 

(i) $|V(T)| = 1 + n_1 + n_2$

$|E(T)| = 2 + (n_1 - 1) + (n_2 - 1)$
$= n_1 + n_2$

(ii) $|V(T)| = 1 + n_1$

$|E(T)| = 1 + (n_1 - 1) = n_1$

(IH) $|V(T_1)| = n_1$, $|E(T_1)| = n_1 - 1$

$|V(T_2)| = n_2$, $|E(T_2)| = n_2 - 1$

Recursively define set $S \subseteq \mathbb{N} \times \mathbb{N}$.

- $(0,0) \in S$
- If $(a, b) \in S$, then both $(a+1, b+1) \in S$ and $(a+3, b) \in S$

Define $S' = \{(x, y) \in \mathbb{N} \times \mathbb{N} : (x \geq y) \wedge (3 | x - y)\}$. Prove that $S = S'$. $\Rightarrow$ (1) $S \subseteq S'$ (2) $S' \subseteq S$

(1) $P(x,y)$: if $(x,y) \in S$, then $x \geq y \wedge 3 | x-y$

Base case : $(0,0) \in S$, $0 \geq 0 \wedge 3 | 0-0$ ✓ $\Rightarrow (0,0) \in S'$

Inductive step : $(a,b) \in S$ WTS $(a+1, b+1)$ and $(a+3, b) \in S'$

BY IH $a \geq b$ and $3 | a-b$ $\Rightarrow$ $a+1 \geq b+1$ and $3 | (a+1)-(b+1)$

$\Rightarrow$ $a+3 \geq b$ and $3 | (a+3) - b$

(2) $P(n)$: $\forall (x,y) \in S'$, $x+y=n$ then $(x,y) \in S$ (Base case. $n=0 \Rightarrow x=y=0$) $\Rightarrow (0,0) \in S$

Let $k \geq 0$, $P(0) \wedge \cdots \wedge P(k)$ is true, WTS $P(k+1)$: $x+y=k+1$

(i) if $y=0$, since $x+y=x=k+1 \geq 1$ and $3 | x-y \Rightarrow 3 | x$ we have $x \geq 3$ so $x-3 \geq 0$

By IH we have $(x-3, y) \in S$ so this implies $((x-3)+3, y) = (x,y) \in S$

(ii) if $y > 0$, since $x \geq y$ so $x > 0$ and $x-1, y-1 \geq 0$. By IH $(x-1, y-1) \in S$

so by recursive def of S, $((x-1)+1, (y-1)+1) = (x,y) \in S$.

Now You Try!

1. Give a recursive definition over the alphabet $\{+, -, (,)\} \cup \mathbb{N}$ of *well-formed* expressions involving addition and subtraction on the natural numbers.

2. A *ternary tree* can have *at most* three children. Prove using structural induction, that for every $n \geq 1$, every non-empty ternary tree of height n has at most $(3^n - 1)/2$ nodes.

Q1. Give a recursive definition over the alphabet $\{+, -, (,)\} \cup \mathbb{N}$ of *well-formed* expressions involving addition and subtraction on the natural numbers.

i $\mathbb{N} \in S$ ii $a, b \in S \Rightarrow (a+b) \in S, (a-b) \in S$

use ii

IN S $(((1+0)-(4+6))+2)$

use i

use ii

NOT IN S $1+0+3$

use i

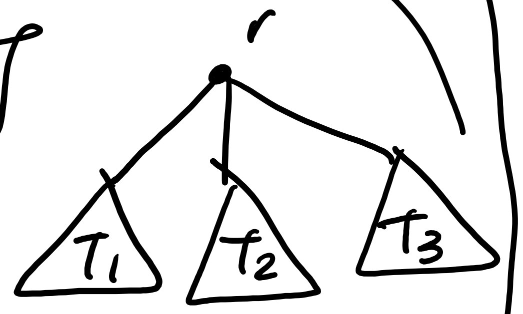
$h(T) = h(T_1)+1$ $\;= 1 + \left(\dfrac{3^{h(T_1)+2}-1}{2}\right) - 1$

Q2. A *ternary tree* can have *at most* three children. Prove using structural induction, that for every ~~$n \geq 1$~~ $n \geq 0$, every non-empty ternary tree of height n has at most $(3^{n+1} - 1)/2$ nodes.

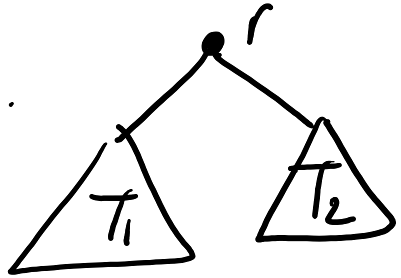
all in $\mathcal{T}$

recursive def of ternary tree: $\overset{r}{\bullet} \in \mathcal{T}$

$T_1, T_2, T_3 \in \mathcal{T}$



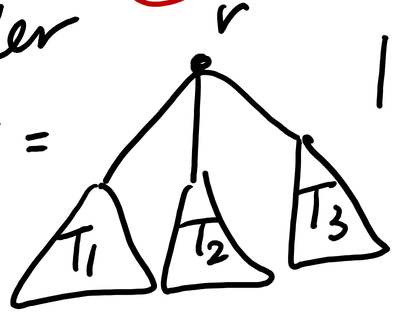
$P(T)$ = T of height n has $\leq (3^{n+1}-1)/2$ nodes.



Base case: $T = \overset{r}{\bullet}$ $\quad 1 \leq (3^{n+1}-1)/2 = 1.$

Inductive step: $T_1, T_2, T_3 \in \mathcal{T}$ $\longrightarrow$ has greatest height h_1.

consider

$T =$ 

$|V(T)| = 1 + |V(T_1)| + |V(T_2)| + |V(T_3)|$

IH $\leq 1 + (3^{h(T_1)+1}-1)/2 + (3^{h(T_2)+1}-1)/2 + \left(\dfrac{3^{h(T_3)+1}-1}{2}\right)$

$\leq 1 + 3(3^{h(T_1)+1}-1)/2$

Minimum Spanning Tree (MST)

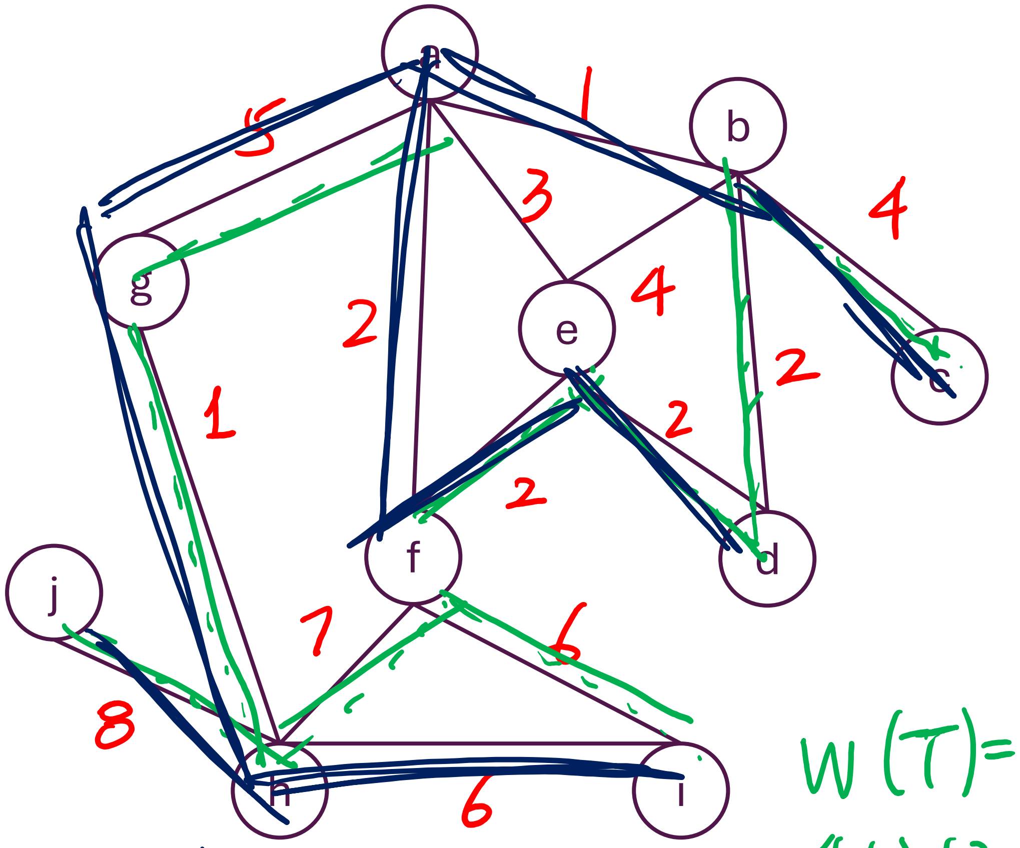
- Weighted graph: $G = (V, E)$ and weight function $w: E \to \mathbb{R}$
- Spanning tree: subgraph $T = (V', E')$ where $V' \subseteq V$ and $E' \subseteq E$ which is a tree

$V = V'$

- Weight of subgraph T:

$$w(T) = \sum_{e \in E'} w(e)$$

- MST: for connected weighted graph G, spanning tree T with minimum weight



$W(T) = 31 <$

$W(T) = 37$

$4+2+2+2+6$
$+7+8+1+5$

Prim's Algorithm

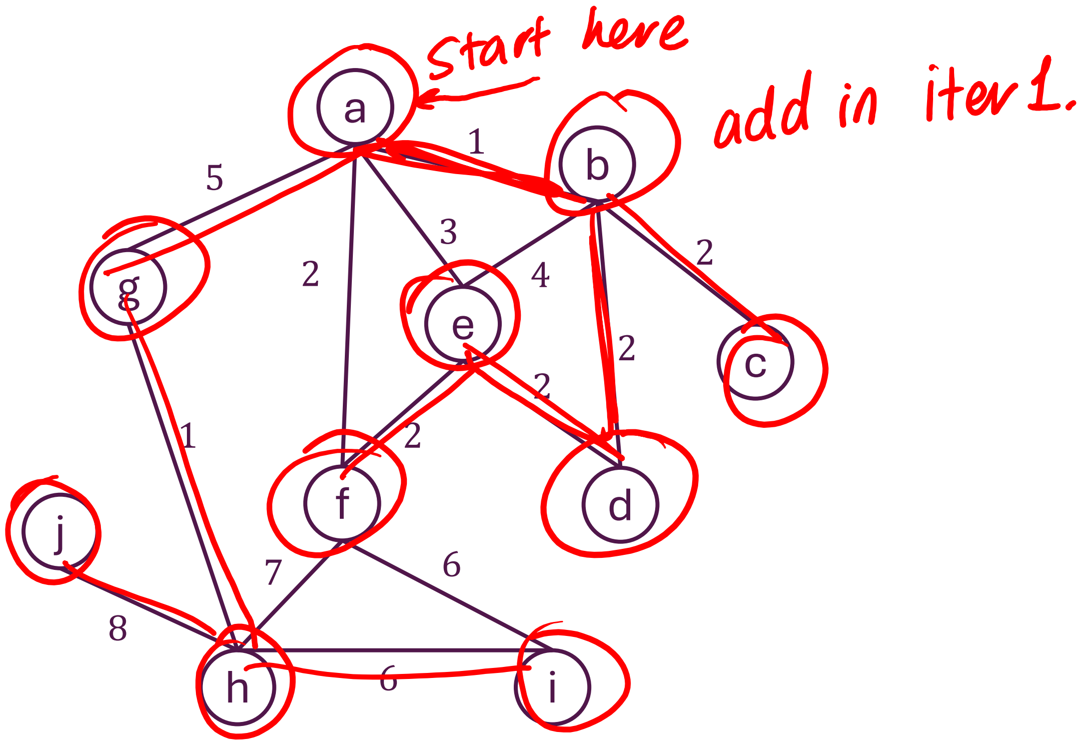
```python
def mst_prim(V, E, w)-> list[edges]:
  # Pre: G = (V,E) connected
  # Post: output MST
  T = []
  visited = {a}
  while visited != V:
    (u,v) = min weight edge
    T = T.append((u,v))
    visited.add(v)
  return T
```

← arbitrary

(u)

u ∈ visited

v ∈ V \ u



ITER	CONSIDERED EDGE	VISITED
0	[ ]	a
1	+ (a,b)	+ b
2	+ (b,d)	+ d
3 → 9	(d,e), (e,f), (b,c), (a,g), (g,h) (h,i) (h,j)	

Program Correctness (Iterative)

- Preconditions: properties of the input
- Postconditions: properties of the output

Program Correctness. Let f be a function with a set of preconditions and post conditions. Then f is *correct* (with respect to the pre- and postconditions) if for every input $I\ to\ f$, if I satisfies the preconditions, then $f(I)$ terminates and all the postconditions hold after termination.

Correctness of Prim’s Algorithm

```
def mst_prim(V, E, w)-> list[edges]:
  # Pre: G = (V,E) connect
  # Post: output MST
  T = []
  visited = {a}
  while visited != V:
    (u,v) = min weight edge
    T = T.append((u,v))
    visited.add(v)
  return T
```

Asymptotic Analysis

```
def mst_prim(V, E, w)-> list[edges]:
  # Pre: G = (V,E) connect
  # Post: output MST
  T = []
  visited = {a}
  while visited != V:
    (u,v) = min weight edge
    T = T.append((u,v))
    visited.add(v)
  return T
```

Recap

- Graph terminology: trees
- Structural induction
    - Recursive definition
- Introduction to proof-of-correctness
- More thorough asymptotic analysis recap

Next time… many more examples of proofs-of-correctness