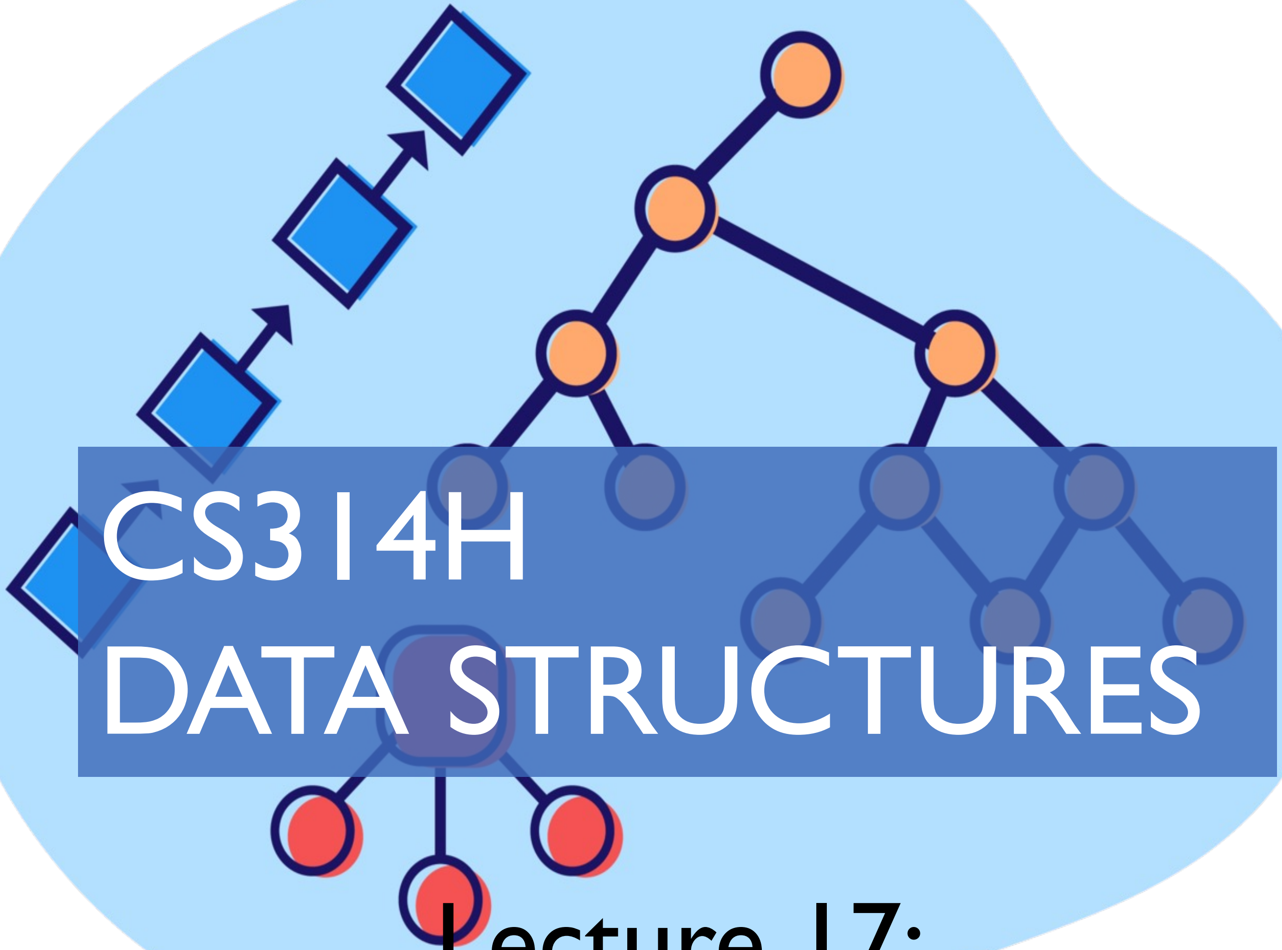


CS314H DATA STRUCTURES

Lecture 17: AVL Analaysis

Mikyung Han

Please, interrupt and ask questions AT ANY TIME !

Outline

1. AVL implementation

Source code

- https://github.com/mikiehan/CS314H-CSB-AVL-Solution

Outline

1. AVL implementation
2. AVL analysis

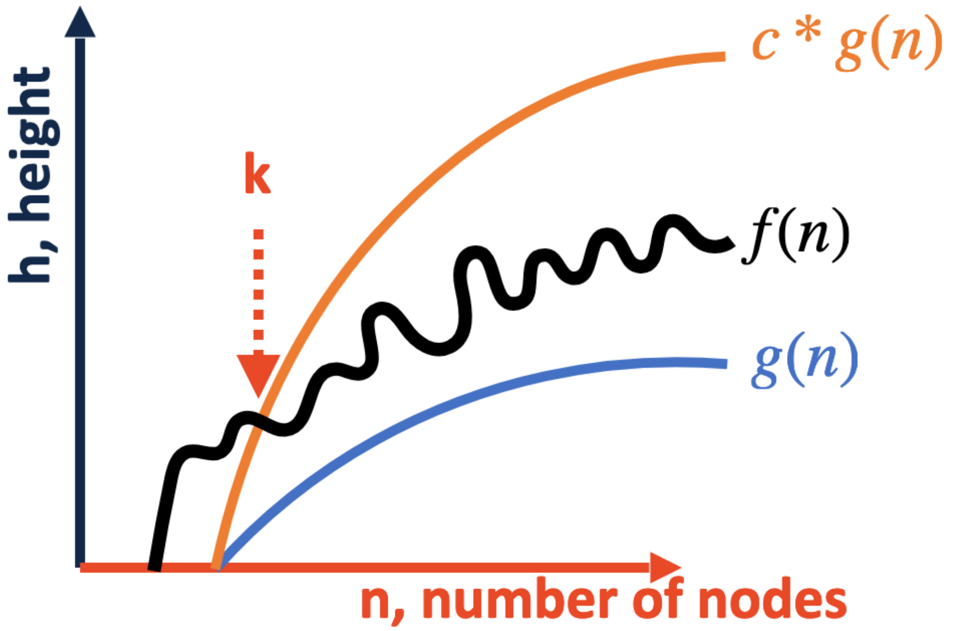
Time complexity of AVL

- Find
- Insert
- Delete

- Rotation cost is O(1)
- What is the dominant cost?

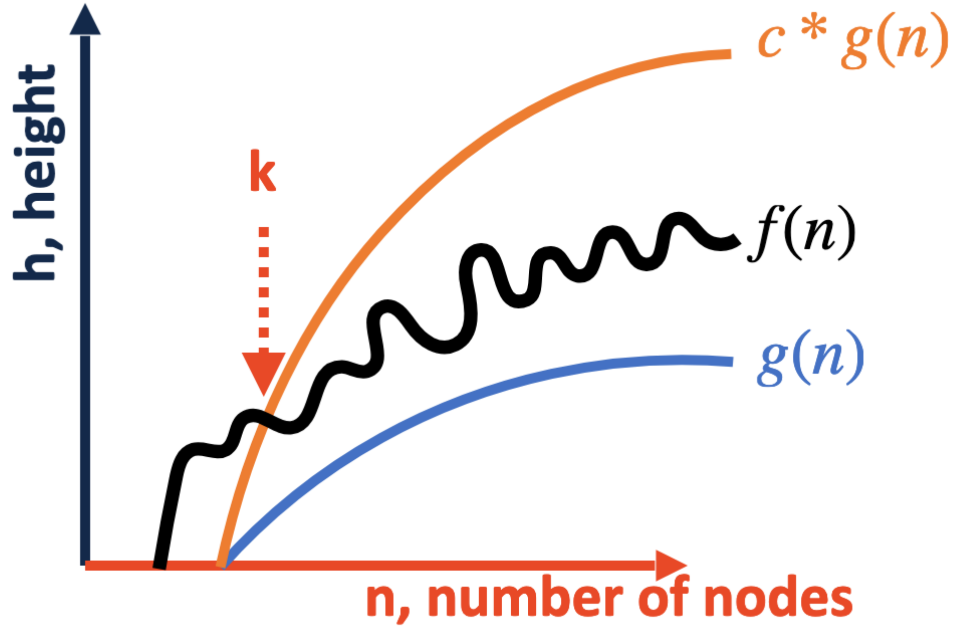
Walking down the tree to its deepest leaf! O(Tree height)

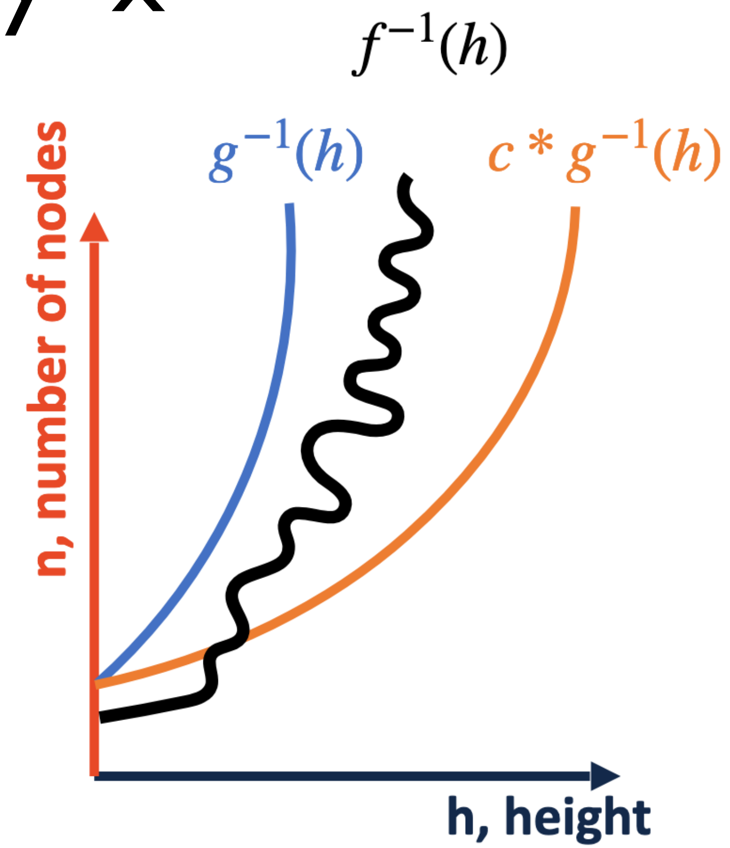
Let f(n) be the tree height given number of nodes n




```
f(n)= height h given n
f(n)≤ c*g(n) for n ≥ k
```

Flip the graph over the line y=x





$f(n) =$ height h given n

$f(n) \leq c * g(n)$ for $n \geq k$

$f^{-1}(h) =$ n given height h

$f^{-1}(h) \geq c * g^{-1}(h)$ for $h \geq k'$

Let’s find the lower bound of n given height h!

N(h) be the min num of nodes in AVL tree with height h

- What is the base case?

- How to construct min num of nodes of height h?
  - Need a root
  - Need one subtree that has min num of nodes with height h-1
  - Need another subtree that has min num of nodes with height ~~h-1~~ h-2
- Recurrence
  - $N(h) = 1 + N(h-1) + N(h-2)$
  - Base case: $N(0) = 1, N(-1) = 0$

Theorem 1: An AVL tree with height h has at least $2^{h/2}$ nodes

Proof by induction

- What is the base case?
  - Tree with height 0 has at least 1 node (true!)
- Induction hypothesis
  - Let's say for all $i < h$, AVL tree with height i has at least $2^{i/2}$ nodes
- $N(h) = 1 + N(h-1) + N(h-2)$

By theorem 1, num of nodes $n > 2^{h/2}$

- Given above, we can find the upper bound of h
- Take log base 2 on both sides
  - $\log_2 n > h/2$
  - $2 \log_2 n > h$ //what does this mean?
- $h = O(\log_2 n)$

Time complexity of find, insert, delete of AVL Tree is $O(\log_2 n)$

Alternatively, we can also use Fibonacci sequence

- $N(h) = 1 + N(h-1) + N(h-2)$, where $N(-1) = 0$, $N(0) = 1$
- $F(k) = F(k-1) + F(k-2)$, where $F(0) = 0$, $F(1) = 1$

h	AVL's N	Fibonacci number F
0		
1		
2		
3		
4		
5		
6		
7		

Alternatively, we can also use Fibonacci sequence

- $N(h) = 1 + N(h-1) + N(h-2)$, where $N(-1) = 0$, $N(0) = 1$
- $F(k) = F(k-1) + F(k-2)$, where $F(0) = 0$, $F(1) = 1$

h	AVL's N	Fibonacci number F
0	1	0
1	2	1
2	4	1
3	7	2
4	12	3
5	20	5
6	33	8
7	54	13

$N(h) = F(h+3) - 1$. Follows the same growth rate as Fibonacci

Fibonacci's growth rate is exponential

- F(k) $\approx \frac{\varphi}{\sqrt{5}}^{k}$, where φ = $\frac{1+\sqrt{5}}{2}$ $\approx$ 1.618 (smaller than 2)
- F(k) $\approx$ C φ^{k}

Fibonacci's growth rate is exponential

- F(k) ≈ $\frac{\varphi^{k}}{\sqrt{5}}$, where φ = $\frac{1+\sqrt{5}}{2}$ ≈ 1.618 (smaller than 2)
- F(k) ≈ C φ^{k}
- F(h+3) ≈ C φ^{h+3}
- F(h+3) - 1 ≈ C φ^{h+3} - 1
- N(h) ≈ C φ^{h+3} - 1
- h+3 ≈ $log_{\varphi}(\frac{N+1}{C})$ = log_{φ}(N+1) + $log_{\varphi}(\frac{1}{C})$
- h ≈ O(log_{φ} N)

Slightly larger than $\log_2$ (but by a constant factor)