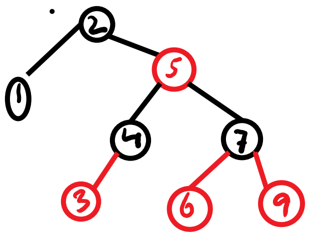


Habib University

shaping futures

CS 201 Data Structure II (L2 / L5)

Height of Red-black tree

Section: 13.1 , Introduction to Algorithms

Muhammad Qasim Pasta

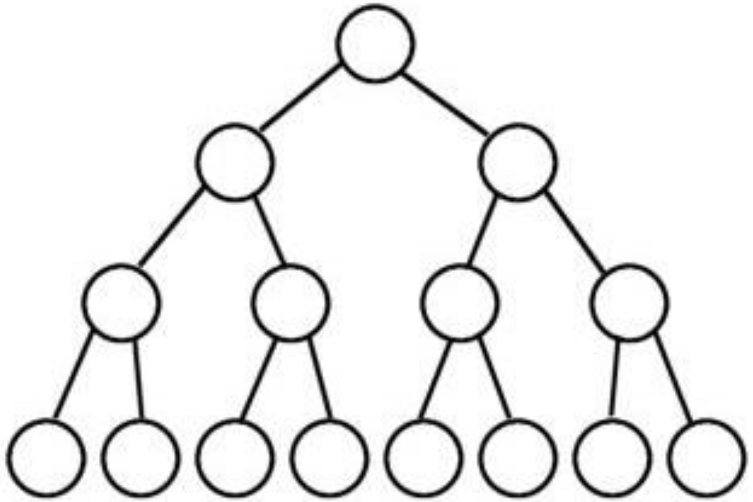
qasim.pasta@sse.habib.edu.pk

Slides are designed to be filled during the lectures. Some details are intentionally mentioned to be discussed in the class. These slides should not be used as reading.

Lemma 1: A complete binary search tree of height h has $2^{h+1} - 1$ nodes

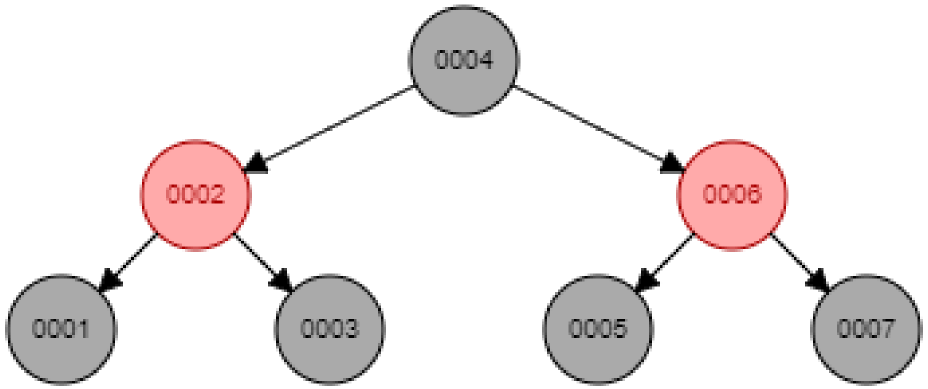
- Let n is total number of internal nodes

- $\mathrm{n} = 1 + 2 + 2^2 + \ldots + 2^{\mathrm{h}} = 2^{\mathrm{h}+1} - 1$
- $n + 1 = 2^{h+1}$
- $\log(n + 1) = h + 1$
- $h = \log(n + 1) - 1$



Lemma 2: A red-black tree of height h has at least $\frac{h}{2}$ black nodes

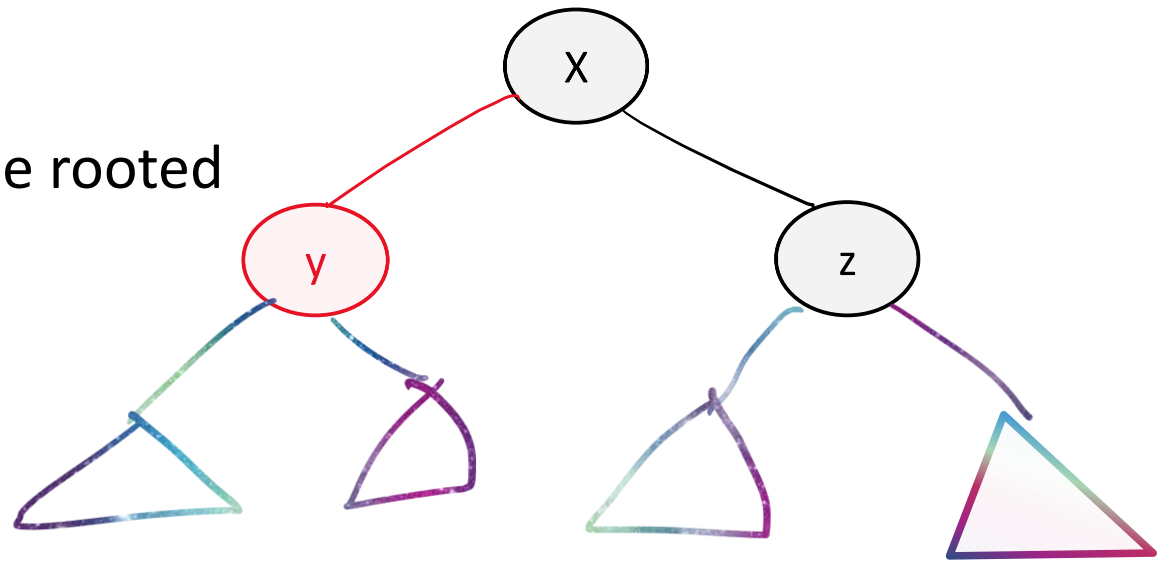
- Consider a path from root node of length h
- How many red nodes can be on this path?
- The path can contain at most $\frac{h}{2}$ red nodes. Why?



Lemma 3: A red-black sub-tree rooted at x contains at least $2^{bh(x)} - 1$ internal nodes

- $bh(x)$ = number of black nodes from x to deepest leaf, not including the node x
  - Same as the black height discussed in earlier classes but excluding the node itself.

- $bh(x)$ for an empty try?
  - $bh(x) = 2^0 - 1 = 0$
- For inductive step, consider a red-black tree rooted at x, total number of nodes can be
- $n = (2^{bh(y)} - 1) + (2^{bh(z)} - 1) + 1$
- $n \geq (2^{bh(x)-1} - 1) + (2^{bh(x)-1} - 1) + 1$
- $n \geq 2^{bh(x)} - 1$



Theorem: A red-black tree with n internal nodes has height at most $2\lg(n+1)$

- Lemma 2: the black height of a red-black tree is at least $\frac{h}{2}$
- Lemma 3: $n \geq 2^{bh(x)} - 1$

- Combining both:

$$n \geq 2^{\frac{h}{2}} - 1$$
$$n + 1 \geq 2^{\frac{h}{2}}$$
$$h \leq 2\log(n + 1)$$