1. RED BLACK TREE

1.1 Introduction:

A red-black tree is a self-balancing binary search tree with one extra bit of storage per node: its color, which can be either *RED* or *BLACK* **.**

1.2 Properties

1. Every node is either red or black.
2. The root is black.
3. Every NULL leaf is black.
4. If a node is red, then both its children are black.
5. For each node, all simple paths from the node to descendant leaves contain the same number of black nodes.

Black Height of a Red-Black Tree **:**

Black height is number of black nodes on a path from root to a leaf. Leaf nodes are also counted black nodes.

1. A Red-Black Tree of height h has **black-height >= h/2 .**
2. Every Red Black Tree with n nodes has **height <= 2Log$_2$(n+1) .**

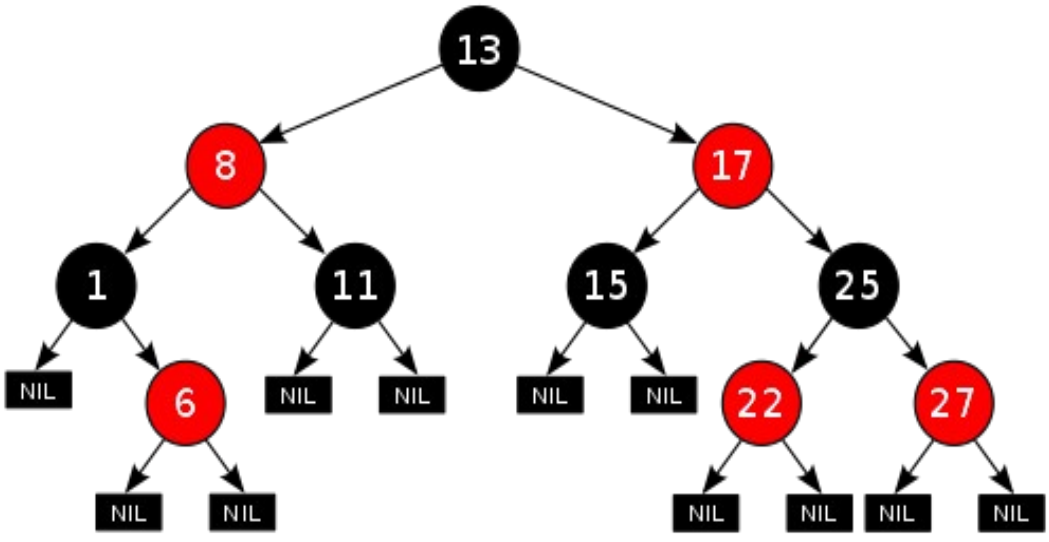

Figure 1 – Structure of Red Black tree

1.5 Applications

- Java: java.util.TreeMap , java.util.TreeSet .
- C++ STL: map, multimap, multiset.
- Used in K-mean clustering algorithm for reducing time complexity.
- Linux kernel: completely fair scheduler, linux/rbtree.h