Red Black Tree

*Definition:*

In this Section, the Red Black Tree would be introduced which is actually the Binary Search Tree with the height of h. Of course, it can support one of any kinds of operations, such as SEARCH(搜索), PREDECESSOR(先驱), SUCCESSOR(后继), MINIMUM(最小), MAXIMUM(最大), INSERT(插入), DELETE(删除) etc, with the time complexity of O(h).

*Red Black Tree is one of Balanced Search Tree. Under the worst situation, it can ensure that the time complexity of all kinds of operation equal to O(lgn).*

*Structure:*

Red Black Tree is one of the Binary Search Tree, and extra color byte has been added to each node which equals to Red or Black. Red Black Tree is the kind of Balanced Tree which applies the constraint rule of Red or Black node in random route from root to leaf. Under this kind of constraint, we can ensure that the total length of one random route from root to leaf is never 2 times longer than the random other route, therefore we can call it Balanced Tree.

In Red Black Tree Node structure, there includes five properties from key, left, right, color and p.

*struct Node {*

*int key; Key stands for the total Black Node from root to leaf node.*

*string color; Two kinds of color, Red and Black.*

*Node \* p; parent child.*

*Node \* left; left child.*

*Node \* right; right child.*

*};*

*Attention, if there has no child node or parent node for one node, then define it as NIL.*

*Properties:*

Below is several properties about Red Black Tree, include:

* *The color of each Node is Red or Black.*
* *The color of Root Node is Black.*
* *The color of NIL Node is Black.*
* *The color of two Children Nodes with the Red Parent Node is Black.*
* *Each route from Root to Leaf has the same number of Black Node.*

*Supplement:*

In order to make code conveniently, NIL Node needs to be set up for Red Black Tree. To make it easier, an extra T.NIL node can be set up to substitute all NIL nodes, and also the T.NIL node needs to be set up as the Parent Node of Root Node. Another thing needs to pay attention to is that its color equals to Black, and let all other properties(key, p, left and right) alone.

Another thing needs us pay attention to is the key field of each node, and the key value equals to the total number of Black Nodes from Leaf Node to Current Node – here, we call it Black Height of each node. Black Height of Current Red Black Tree is the Black Height of its Root Node.

*Example:*

Example 1 – Classic Red Black Tree

A picture containing sitting, computer, table, small

Description automatically generated

Example 2 – Red Black Tree uses NIL Node to substitute all NIL Leaf Node

A picture containing sitting, indoor, table, computer

Description automatically generated

Example 3 – Red Black Tree neglects all Leaf Nodes

A picture containing sitting, clock

Description automatically generated