C Program - Red Black Tree Insertion.

Date: 18/11/20

ADS - LAB - 8 Adith Pavi Pai 1BM18CS005

```
struct node { // To represent each node in Red Black Tree
    int d; //data
    int c; // 1-red, 0-black
    struct node* p; //parent
    struct node* r; //right-child
    struct node *l; // left child
};
struct node *root = NULL;
struct node* bst (struct node* trav, struct node *temp)
{ if(trav //function to perform BST insertion of node
    if (trav == NULL)
        return temp; // if tree is empty, return new node

    // or else recurse down the tree
    if (temp -> d < trav -> d)
    {
        trav -> l = bst(trav -> l, temp);
        trav -> l -> p = trav;
    }
    else if (temp -> d > trav -> d)
    {
        trav -> r = bst(trav -> r, temp);
        trav -> r -> p = temp; trav;
    }
    return trav; // unchanged node pointer
}
```

Date: / /

```
void rightrotate (struct node* temp)
{
    struct node* left = temp -> l;
    temp -> l = left -> r;
    if (temp -> l)
        temp -> l -> p = temp;
    left -> p = temp -> p;
    if (!temp -> p)
        root = left;
    else if (temp == temp -> p -> l)
        temp -> p -> l = left;
    else
        temp -> p -> r = left;
    left -> r = temp;
    temp -> p = left;
}
void leftrotate (struct node* temp)
{ struct node* right = temp -> r;
  temp -> r = right -> l;
  if (temp -> r)
      temp -> r -> p = temp;
  if (!temp -> p)
      root = right;
  else if (temp == temp -> p -> l)
      temp -> p -> l = right;
  else temp -> p -> r = right;
  right -> l = temp;
  temp -> p = right;
}
```