

WhatsApp

2-3 Trees - Data

Trie Data Structu

Trie Data Structu

FREE AI Code Ge

ChatGPT

Online C Compil

programiz.com/c-programming/online-compiler/

Programiz
C Online Compiler

Programiz PRO

Premium Coding
Courses by Programiz

Learn More

Programiz PRO >

main.c

Share

Run

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #define RED 0
4 #define BLACK 1
5 typedef struct RBNode {
6     int data;
7     int color; // RED or BLACK
8     struct RBNode *left, *right, *parent;
9 } RBNode;
10 RBNode* createNode(int data) {
11     RBNode *newNode = (RBNode *)malloc(sizeof(RBNode));
12     newNode->data = data;
13     newNode->color = RED;
14     newNode->left = newNode->right = newNode->parent = NULL;
15     return newNode;
16 }
17 void leftRotate(RBNode **root, RBNode *x) {
18     RBNode *y = x->right;
19     x->right = y->left;
20     if (y->left != NULL) {
21         y->left->parent = x;
22     }
23     y->parent = x->parent;
24     if (x->parent == NULL) {
25         *root = y;
26     } else if (x == x->parent->left) {
27         x->parent->left = y;
28     } else {
29         x->parent->right = y;
30     }
31     y->left = x;
32     x->parent = y;
33 }
34 void rightRotate(RBNode **root, RBNode *y) {
35     RBNode *x = y->left;
36     y->left = x->right;
```

Output

Clear

/tmp/xBghnDugUC.o
In-order Traversal: 10 (BLACK) 15 (RED) 20 (BLACK) 25 (RED) 30 (BLACK)

=== Code Execution Successful ===

Feels hotter
Now

Search

14:11
01-08-2024

```
main.c
34- void rightRotate(RBNode **root, RBNode *y) {
35-     RBNode *x = y->left;
36-     y->left = x->right;
37-     if (x->right != NULL) {
38-         x->right->parent = y;
39-     }
40-     x->parent = y->parent;
41-     if (y->parent == NULL) {
42-         *root = x;
43-     } else if (y == y->parent->right) {
44-         y->parent->right = x;
45-     } else {
46-         y->parent->left = x;
47-     }
48-     x->right = y;
49-     y->parent = x;
50- }
51- void fixInsert(RBNode **root, RBNode *z) {
52-     while (z->parent != NULL && z->parent->color == RED) {
53-         if (z->parent == z->parent->parent->left) {
54-             RBNode *y = z->parent->parent->right;
55-             if (y != NULL && y->color == RED) {
56-                 z->parent->color = BLACK;
57-                 y->color = BLACK;
58-                 z->parent->parent->color = RED;
59-                 z = z->parent->parent;
60-             } else {
61-                 if (z == z->parent->right) {
62-                     z = z->parent;
63-                     leftRotate(root, z);
64-                 }
65-                 z->parent->color = BLACK;
66-                 z->parent->parent->color = RED;
67-                 rightRotate(root, z->parent->parent);
68-             }
69-         } else {
```

Output Clear

```
/tmp/xBghnDugUC.o
In-order Traversal: 10 (BLACK) 15 (RED) 20 (BLACK) 25 (RED) 30 (BLACK)

=== Code Execution Successful ===
```

main.c

```
70 RBNode *y = z->parent->parent->left;
71 if (y != NULL && y->color == RED) {
72     z->parent->color = BLACK;
73     y->color = BLACK;
74     z->parent->parent->color = RED;
75     z = z->parent->parent;
76 } else {
77     if (z == z->parent->left) {
78         z = z->parent;
79         rightRotate(root, z);
80     }
81     z->parent->color = BLACK;
82     z->parent->parent->color = RED;
83     leftRotate(root, z->parent->parent);
84 }
85 }
86 }
87 (*root)->color = BLACK;
88 }
89 void insert(RBNode **root, int data) {
90     RBNode *z = createNode(data);
91     RBNode *y = NULL;
92     RBNode *x = *root;
93
94     while (x != NULL) {
95         y = x;
96         if (z->data < x->data) {
97             x = x->left;
98         } else {
99             x = x->right;
100         }
101     }
102     z->parent = y;
103     if (y == NULL) {
104         *root = z;
105     } else if (z->data < y->data) {
106         y->left = z;
```

Share

Run

Output

Clear

```
/tmp/xBghnDugUC.o
In-order Traversal: 10 (BLACK) 15 (RED) 20 (BLACK) 25 (RED) 30 (BLACK)

=== Code Execution Successful ===
```


exness Think Next Level Trading Think Exness Upgrade now

```
main.c
96- if (z->data < x->data) {
97-     x = x->left;
98- } else {
99-     x = x->right;
100- }
101- }
102- z->parent = y;
103- if (y == NULL) {
104-     *root = z;
105- } else if (z->data < y->data) {
106-     y->left = z;
107- } else {
108-     y->right = z;
109- }
110- fixInsert(root, z);
111- }
112- void inorderTraversal(RBNode *root) {
113-     if (root != NULL) {
114-         inorderTraversal(root->left);
115-         printf("%d (%s) ", root->data, root->color == RED ? "RED" : "BLACK");
116-         inorderTraversal(root->right);
117-     }
118- }
119- int main() {
120-     RBNode *root = NULL;
121-     insert(&root, 10);
122-     insert(&root, 20);
123-     insert(&root, 30);
124-     insert(&root, 15);
125-     insert(&root, 25);
126-     printf("In-order Traversal: ");
127-     inorderTraversal(root);
128-     printf("\n");
129-     return 0;
130- }
131-
```

Output

/tmp/xBghnDugUC.o
In-order Traversal: 10 (BLACK) 15 (RED) 20 (BLACK) 25 (RED) 30 (BLACK)

=== Code Execution Successful ===