Implementation of Red Black Trees in C

Anant Maheshwari 13CO111 Shrukul Habib 13CO143

Generated by Doxygen 1.8.7

Wed Feb 24 2016 00:40:53

Contents

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 2

File Index

2.1 File List

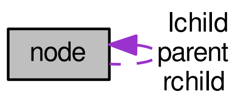
Here is a list of all documented files with brief descriptions:

Chapter 3

Class Documentation

3.1 node Struct Reference

Collaboration diagram for node:



Public Types

- enum { **black**, **red** }

Public Attributes

- enum node:: { ... } **colour**
- int info
- struct node ∗ lchild
- struct node ∗ rchild
- struct node ∗ parent

3.1.1 Member Data Documentation

3.1.1.1 int node::info

Stores whether the node is red or black

3.1.1.2 struct **node**∗ node::lchild

Stores the value of the node

3.1.1.3 struct node∗ node::parent

Pointer of the right node

3.1.1.4 struct node∗ node::rchild

Pointer of the left node

The documentation for this struct was generated from the following file:

- red_black_tree.c

Chapter 4

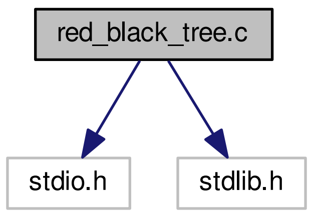
File Documentation

4.1 red_black_tree.c File Reference

Implementation of Red Black Tree.

```
#include <stdio.h>
#include <stdlib.h>
```

Include dependency graph for red_black_tree.c:



Classes

- struct node

Functions

- int find (int item, struct node ∗∗loc)

  Check whether a node with the given value exists in the tree or not.
- void insert (int item)

  Inserts a new node. Incase of duplicate node, error is displayed.
- void insert_balance (struct node ∗nptr)

  The function performs series of rotations needed after node insertion.
- void del (int item)

  Delete a node from the tree.
- void del_balance (struct node ∗ptr)

  The function performs series of rotations needed after node deletion.

- void rotate_left (struct node ∗ptr)

  Rotates the tree about the given node in left direction.
- void rotate_right (struct node ∗ptr)

  Rotates the tree about the given node in right direction.
- struct node ∗ succ (struct node ∗ptr)

  Finds the inorder successor of the given node.
- void inorder (struct node ∗ptr)

  Displays Inorder traversal of the tree.
- void display (struct node ∗ptr, int level)

  Level order traversal of the tree.
- int main ()

  The Main Function of the Program.

Variables

- struct node ∗ **root**
- struct node ∗ sentinel

4.1.1 Detailed Description

Implementation of Red Black Tree.

Author

Shrukul Habib 13CO143
Anant Maheshwari 13CO111

4.1.2 Function Documentation

4.1.2.1 void del (int *item*)

Delete a node from the tree.

Parameters

item	Value of the node to be deleted

Returns

Doesn't return anything, void.

4.1.2.2 void del_balance (struct **node** ∗ *ptr*)

The function performs series of rotations needed after node deletion.

Parameters

ptr	value of the node to be deleted

Returns

Doesn't return anything, void.

4.1.2.3 void display (struct **node** ∗ *ptr,* int *level*)

Level order traversal of the tree.

Parameters

ptr	The pointer of the root node
level	The Level of the the current node (1, if starting from root)

Returns

Doesn't return anything, void.

4.1.2.4 int find (int *item*, struct node $**$ *loc*)

Check whether a node with the given value exists in the tree or not.

Parameters

item	The value of the node to be checked
loc	The location of the found node

Returns

Doesn't return anything, void.

4.1.2.5 void inorder (struct node $*$ *ptr*)

Displays Inorder traversal of the tree.

Parameters

ptr	The pointer of the root node.

Returns

Doesn't return anything, void.

4.1.2.6 void insert (int *item*)

Inserts a new node. Incase of duplicate node, error is displayed.

Parameters

item	The value of the node to be inserted.

Returns

Doesn't return anything, void.

4.1.2.7 void insert_balance (struct node $*$ *nptr*)

The function performs series of rotations needed after node insertion.

Parameters

nptr	The pointer of the new node.

Returns

Doesn't return anything, void.

4.1.2.8 int main ( )

The Main Function of the Program.

for parent of root node and NULL nodes

Returns

Doesn't return anything, void.

4.1.2.9 void rotate_left (struct node ∗ *ptr*)

Rotates the tree about the given node in left direction.

Parameters

ptr	The pointer of the node about which the tree is to be rotated.

Returns

Doesn't return anything, void.

4.1.2.10 void rotate_right (struct node ∗ *ptr*)

Rotates the tree about the given node in right direction.

Parameters

ptr	The pointer of the node about which the tree is to be rotated.

Returns

Doesn't return anything, void.

4.1.2.11 struct node ∗ succ (struct node ∗ *ptr*)

Finds the inorder successor of the given node.

Parameters

ptr	The pointer of the given node.

Returns

The pointer of the successor node.

4.1.3 Variable Documentation

4.1.3.1 struct node∗ sentinel

This is the pointer of the root node

Index