Persistent red-black tree

Generated by Doxygen 1.8.18

1 Class Index **1**

1.1 Class List 1

2 Class Documentation **3**

2.1 PersistentRBTree< T > Class Template Reference 3

2.1.1 Member Function Documentation 3

2.1.1.1 getRoot() 3

2.1.1.2 search() 4

Index **5**

Chapter 1

Class Index

1.1 Class List

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

PersistentRBTree< T > 3

Chapter 2

Class Documentation

2.1 PersistentRBTree< T > Class Template Reference

Public Member Functions

- **PersistentRBTree** (const PersistentRBTree &rbt)
- void insert (T key)

  Inserts the key 'key' into the tree.
- void deleteRBTreeNode (RBTreeNode< T > ∗node)

  Delete the 'node' in the tree.
- RBTreeNode< T > ∗ search (T key)

  Returns the pointer to the node with the key 'key'.
- RBTreeNode< T > ∗ getRoot ()

  Returns the pointer to the root.
- void setRoot (RBTreeNode< T > ∗new_root)

  Set root pointer.
- void print (std::ostream &out=std::cout)

  Prints the tree according to the inorder traversal.
- void clear ()

  Deletes (free the memory) all nodes.
- RBTreeNode< T > ∗ getVersion (unsigned int index)

  Get version 'index' of tree.

2.1.1 Member Function Documentation

2.1.1.1 getRoot()

```
template<typename T >
RBTreeNode< T > ∗ PersistentRBTree< T >::getRoot  [inline]
```

Returns the pointer to the root.

If the tree is not empty, returns the not null pointer to the root node. Otherwise, returns null pointer.

2.1.1.2 search()

```
template<typename T >
RBTreeNode< T > * PersistentRBTree< T >::search (
            T key )  [inline]
```

Returns the pointer to the node with the key 'key'.

If the node with the key 'key' exists, returns the not null pointer to that node. Otherwise, returns null pointer.

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

- Persistent_RBT.h

Index

getRoot
    PersistentRBTree$<$ T $>$, 3

PersistentRBTree$<$ T $>$, 3
    getRoot, 3
    search, 3

search
    PersistentRBTree$<$ T $>$, 3