

h5 RB Delete, Git, B-Trees

Started: Mar 4 at 2:04pm

Quiz Instructions

Question 1

2 pts

In a B-Tree with a branching factor of 5, **seven** numbers are added, in no particular order.

After these insertions, how many keys are in the root?

Question 2

1 pts

A B-Tree with a branching factor of 5 is created.

The following numbers are inserted, in this order: 3, 9, 8, 2, 1, 6, 4, 7, 5

In the node that contains the number 8, how many keys are there?


Hint: draw the tree

Question 3

3 pts

Match each git command with its use case.

git status

lists files in the current working 

git log

displays a list of commits for the

git add

copy files to the staging area

git commit

copies staged files to the local repository

git push

copies any changes in the local repository to the remote repository

git pull

retrieves updated files from the remote repository

Question 4

3 pts

A B-Tree has a branching factor of 7, which means that any node can have as many as 7 children.

Assume the tree is not empty.

Any node that is not the root must have a minimum of

keys.

Any node can have a maximum of keys.

The fewest number of keys that the root can have is .

Question 5

4 pts

Assume that a B-Tree uses the following node type:

```
class BTreeNode<K, V> {  
    private List<K> keyList;
```

```

private List<V> valueList;
private List<BTreeNode <K,V> childList;
private BTreeNode() {
    keyList = new ArrayList<K>();
    valueList = new ArrayList<V>();
    childList = new ArrayList<<BTreeNode<K,V>>>();
private boolean isLeaf(){
    return childList.size() == 0;
}
}

```

You are writing a recursive inorder traversal method that you would call with the command

```
print_inorder(root);
```

Fill in the missing parts of the code using the choices given.

```

// prints out the keys, in order
private void print_inorder(BSTNode<K,V> current) {
    if (current == null) { return; }
    int i=0;
    while (i < )
    {
        if ( ! current.isLeaf())
            print_inorder()
        );
        System.out.println()
        + " " );
        i++;
    }
    print_inorder()
    );
}

```



Question 6

1 pts

Which case of a Red-Black Tree Delete is the most complicated (has the most cases to consider) ?

- ☒ deleting a black leaf
- ☐ deleting a node with exactly one child
- ☐ deleting a red leaf
- ☐ deleting a node with two children

Question 7

2 pts

Match each Version Control term with its definition as given in lecture.

merge

add your changes to a repository ↕

conflict

when merge fails because your ↕

resolution

choose which content will prevail ↕

checking in

putting new and changed files in ↕

Question 8

4 pts

You are trying to delete a node in a Red-Black tree.

Let d be a reference to the node you are trying to delete.

Let g be a reference to d 's parent and assume g 's right child is d .

Let k be the only child of d .

Complete the algorithm for deletion when you are certain that node d has exactly one child.

1. set to

2. set the color of to

Quiz saved at 2:24pm

[Submit Quiz](#)