Gebze Technical University
Computer Engineering

CSE 222 - 2018 Spring

HOMEWORK 6 REPORT

AHMET ERGANİ
161044011

Course Assistant: Fatma Nur Esirci

1 Worst RedBlack Tree

I created a Red Black tree with a full black left subtree.

1.1 Problem Solution Approach

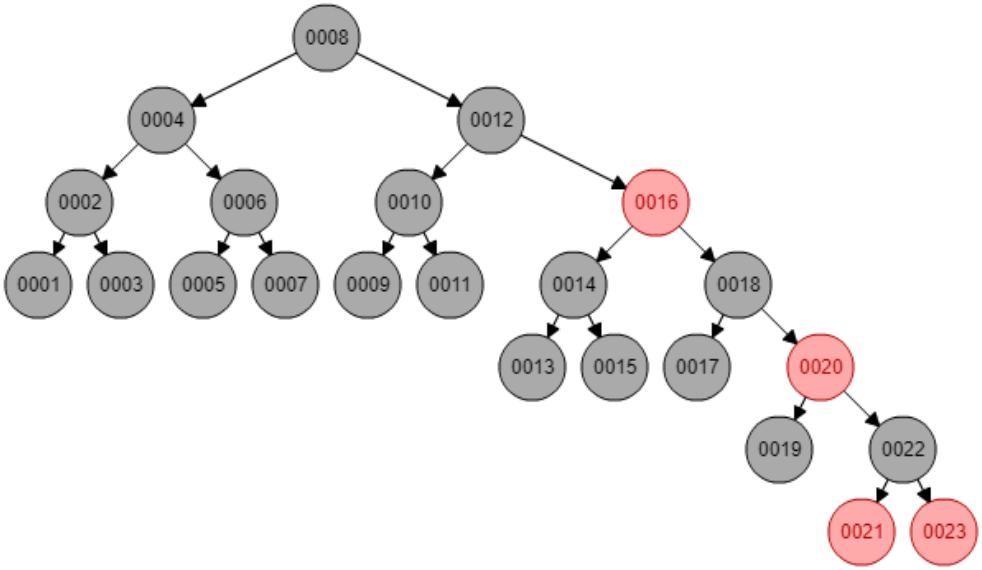
Since the homework pdf did not ask me to implement in this part, I used our book's instructor codes. The classes I used are : SearchTree, BinaryTree, BinarySearchTree, BinarySearchTreeWithRotate and RedBlackTree. I did not simply copy paste them. Instead, I analyzed them and tried to understand how each method works. I realized that the main difference between BinarySearchTree and RedBlackTree is RedBlack tree being much more "bushy" tree. By "bushy" I mean left and right subtree of each node can't be too different and assymetrical like a BinarySearchTree. This difference occurs because the amount of black nodes are equal in each node's left and right subtree in a RedBlackTree. So in order to get the most assymetrical and different subtrees and therefore achieve the Worst Case Scenario, one of the subtrees should be normal while the other one is fully black. While dealing with 6 height, I achieved it by adding integers 1n to 23n.

1.2 Test Cases

Test Case 1: Add {1,2,3,4.....23} to tree1; print tree1;
Test Case 2: Add {3,6,9,12....69} to tree2; print tree2;

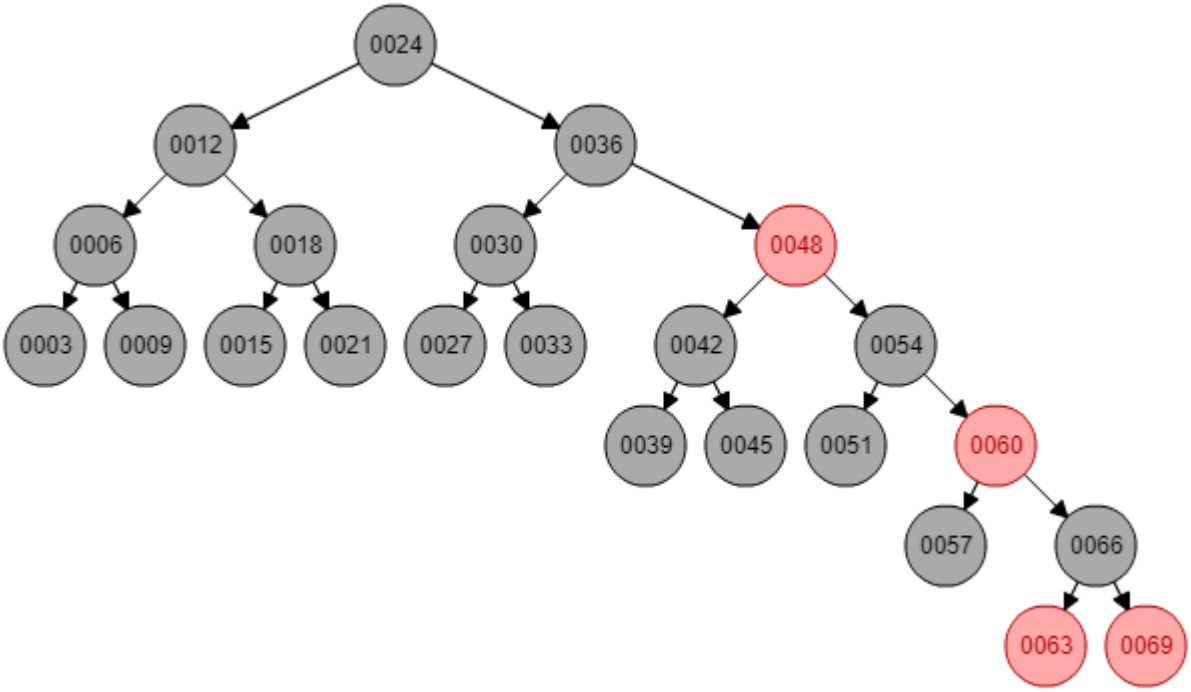
1.3 Running Commands and Results

Test Case 1:



```
**********TEST CASE 1**********
Black: 8
  Black: 4
    Black: 2
      Black: 1
        null
        null
      Black: 3
        null
        null
    Black: 6
      Black: 5
        null
        null
      Black: 7
        null
        null
  Black: 12
    Black: 10
      Black: 9
        null
        null
      Black: 11
        null
        null
    Red  : 16
      Black: 14
        Black: 13
          null
          null
        Black: 15
          null
          null
      Black: 18
        Black: 17
          null
          null
        Red  : 20
          Black: 19
            null
            null
          Black: 22
            Red  : 21
              null
              null
            Red  : 23
              null
              null
```

Test Case 2:


0024
0012
0036
0006
0018
0030
0048
0003
0009
0015
0021
0027
0033
0042
0054
0039
0045
0051
0060
0057
0066
0063
0069

```
**********TEST CASE 2**********
Black: 24
  Black: 12
    Black: 6
      Black: 3
        null
        null
      Black: 9
        null
        null
    Black: 18
      Black: 15
        null
        null
      Black: 21
        null
        null
  Black: 36
    Black: 30
      Black: 27
        null
        null
      Black: 33
        null
        null
    Red  : 48
      Black: 42
        Black: 39
          null
          null
        Black: 45
          null
          null
      Black: 54
        Black: 51
          null
          null
        Red  : 60
          Black: 57
            null
            null
          Black: 66
            Red  : 63
              null
              null
            Red  : 69
              null
              null
```

2 binarySearch method

I implemented a binarySearch() method for insertion into B-Tree

2.1 Problem Solution Approach

```
binarySearch(item,data,from,to)
{
   if(from < to)
         middle = from + (to - from) / 2
         if(item < data[middle])
             return binarySearch(item,data,from,mid);
         if(item > data[middle])
             return binarySearch(item,data,mid+1,to);
         if(item == data[middle)
             return middle;
    return from;
}
```

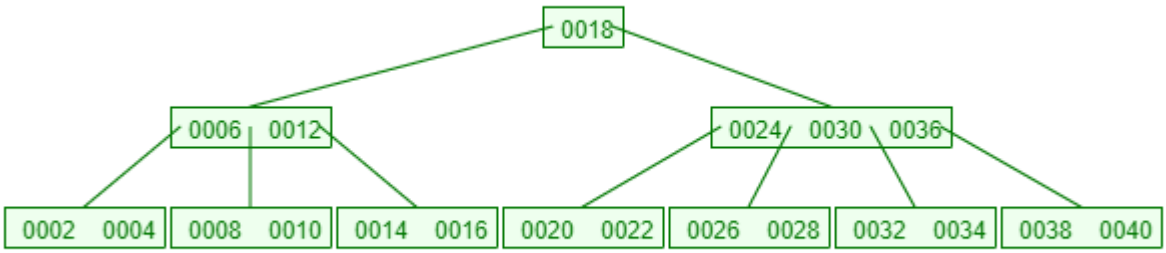
I used Btree.java from student code. It is really similar to basic binarySearch for array but the last line creates the difference.

2.2 Test Cases

Test Case 1: Add {2,4,6,8,10,12,14,16,18,20,22,24,26,28,30,32,34,36,38,40} to tree1 (order: 5)
Test Case 2: Add {20,19,18,17,16,15,14,13,12,11,1,3,5,25,45,6,46,7,9} to tree2 (order : 4)

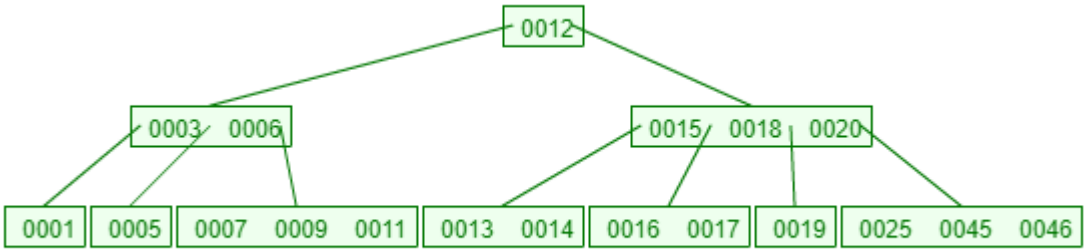
2.3 Running Commands and Results

Test Case 1:



```
"C:\Program Files\Java\jdk1.8.0_151\bin\java.exe" ...
**********TEST CASE 1**********
18
  6, 12
    2, 4
      null
      null
      null


    8, 10
      null
      null
      null


    14, 16
      null
      null
      null




  24, 30, 36
    20, 22
      null
      null
      null


    26, 28
      null
      null
      null


    32, 34
      null
      null
      null


    38, 40
      null
      null
      null
```

Test Case 2:




```
**********TEST CASE 2**********
12
  3, 6
    1
      null
      null


    5
      null
      null


    7, 9, 11
      null
      null
      null
      null



  15, 18, 20
    13, 14
      null
      null
      null


    16, 17
      null
      null
      null


    19
      null
      null


    25, 45, 46
      null
      null
      null
      null
```

3 Project 9.5 in book

I have failed to implement this part

3.1 Problem Solution Approach

null

3.2 Test Cases

null

3.3 Running Commands and Results

null