Quarter 1 Final, Data Structures

Jay R Bolton

December 8, 2011

Note: This is the work I did within the class time. I will also be sos_submitting a better version later on.

1. *(a)*

```
abcde
-f-g-
--h--
-i-j-
klmno
compact = [a,b,c,d,e,f,g,h,i,j,k,l,m,n,o]
(which is each row concatenated)
```

(b)

```
get(row, col, list)
  if(row == 1)
    return index(list, col)
  else if(row == 5)
    return index(list, 10+col)
  else if(row == col)
    return index(list, row + col + 1)
  else if(row == (6 - col))
    return index(list, 4 + row)
  else
    error("Invalid index")
```

2.

```
    A
 /     \
B       D
\     / \
  C   E   G
     /   / \
    F   H   J
       /
      I
```

For some reason I wrote this algo in haskell last night.

```
Tree a = Node a (Tree a, Tree a) | Nil
rtrav [] [] = Nil
rtrav (p:ps) in = let i = find p in
                      lin = take i in
                      rin = drop (i+1) in
                      lpre = take (length lin) ps
                      rpre = drop (length lin) ps
                   in Tree p ((rtrav lpre lin),(rtrav rpre rin))
```

3. *(a)*

Index	**Key**
0	38
1	14
2	11
3	42
4	
5	
6	
7	7
8	73
9	8
10	22
11	34
12	25

(b) Not too sure about linear probe deletion. This is my best guess: collided keys migrate back towards their correct bucket.

Index	**Key**
0	38
1	14
2	
3	42
4	
5	
6	
7	7
8	8
9	34
10	
11	11
12	25

4. *(a)*

```
ElementType computeS(TreeNode t) {
  if(t == NULL) return -1 /* base case */
  else return 1 + max(computeS(t->left), computeS(t->right))
}
/* Leaves return 0. This is similar logic to the book's height balancing
```

```
       for AVL trees. */
```

(b) $\mathcal{O}(N)$

Or more closely: $\mathcal{O}(N + ceil(log(N)) * 2)$ (to account for two extra calls at each leaf).

Actually, that's wrong. I guess I'll leave it for the idea. Running out of time!

5. *(a)*

```
I'll do positive = more left
        negative = more right
       +1
      /  \
     /    \
    /      \
   +1        0
  /  \      /  \
 -1   +1   +1   0
 / \  /    /   / \
0  +1 0   0   0   0
   /
  0
```

(b)

```
Ahhhhh!! Ten minutes left!!!

1. perform insertion

       30
      /  \
     /    \
    /      \
   25       50
  /  \      /  \
 10    27  40  60
 / \  /    /   / \
5  15 26  35   55  70
   /
  12
   \
    13

2. We find as we percolate up recursively that 15 is +2.
   So we do an LR rotation
```

```
       30
      /  \
     /    \
    /      \
   25       50
  /  \      /  \
 10    27  40  60
 / \   /   /   / \
5  13  26  35  55  70
   / \
  12  15
```

```
3. We continue to percolate and find that all is balanced.
   Then we insert 14
```

```
       30
      /  \
     /    \
    /      \
   25       50
  /  \      /  \
 10    27  40  60
 / \   /   /   / \
5  13  26  35  55  70
   / \
  12  15
     /
    14
```

```
4. We find that 10 is -2
   We do a RR rotation
```

```
         30
        /  \
       /    \
      /      \
     25       50
    /  \      /  \
   13    27  40  60
   / \   /   /   / \
 10  15  26  35  55  70
 /\    /
5  12 14
```

```
5. All is balanced.
   We then insert 36.
```

```
         30
        /  \
       /    \
      /      \
     25       50
    /  \      /  \
   13     27  40  60
   / \   /   /   / \
 10  15  26  35  55  70
 /\   /       \
5  12 14      36
```

6. 40 is imbalanced
   We do an LR

And so on. Ran out of time