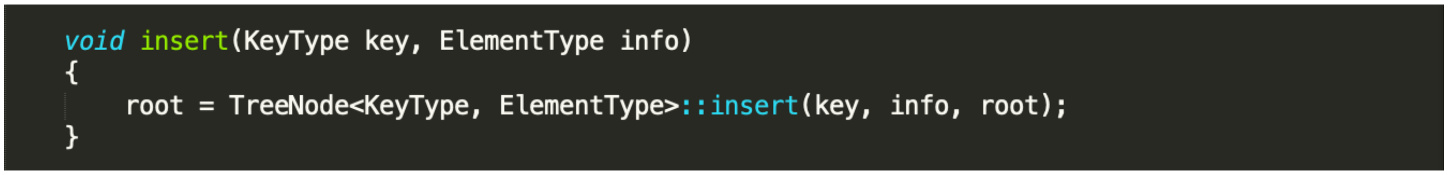
HOMEWORK 6

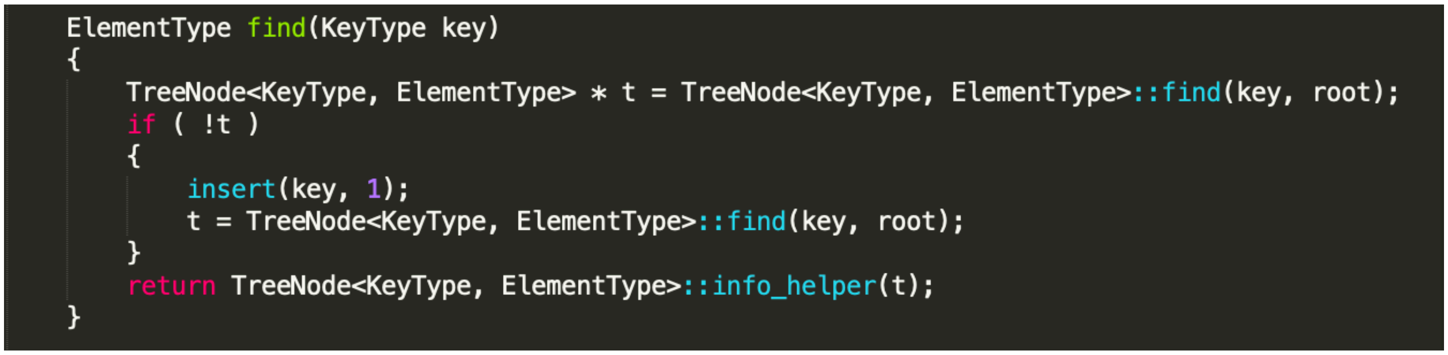
YANG TANG ID: 53979886

Part 1: Code Implementation & Big-O Time complexity Analysis



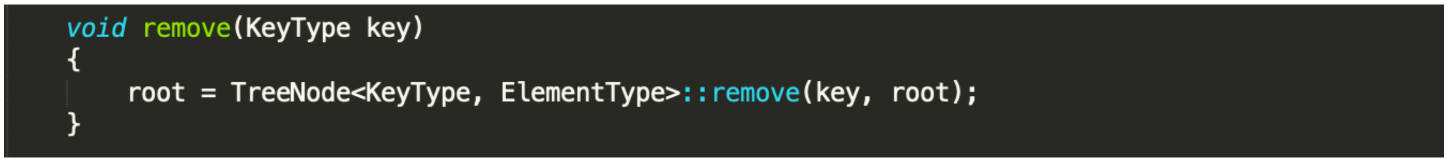
```
void insert(KeyType key, ElementType info)
{
    root = TreeNode<KeyType, ElementType>::insert(key, info, root);
}
```

O(N) = O(logN)   T(N) = 1010logN+5



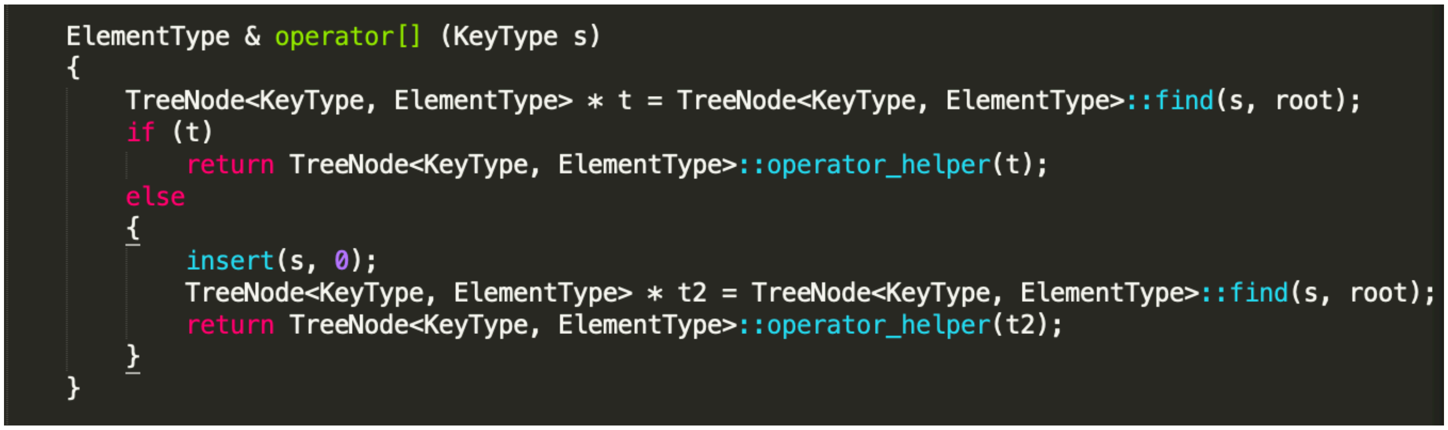
```
ElementType find(KeyType key)
{
    TreeNode<KeyType, ElementType> * t = TreeNode<KeyType, ElementType>::find(key, root);
    if ( !t )
    {
        insert(key, 1);
        t = TreeNode<KeyType, ElementType>::find(key, root);
    }
    return TreeNode<KeyType, ElementType>::info_helper(t);
}
```

O(N) = O(logN)   T(N) = 1020logN+13



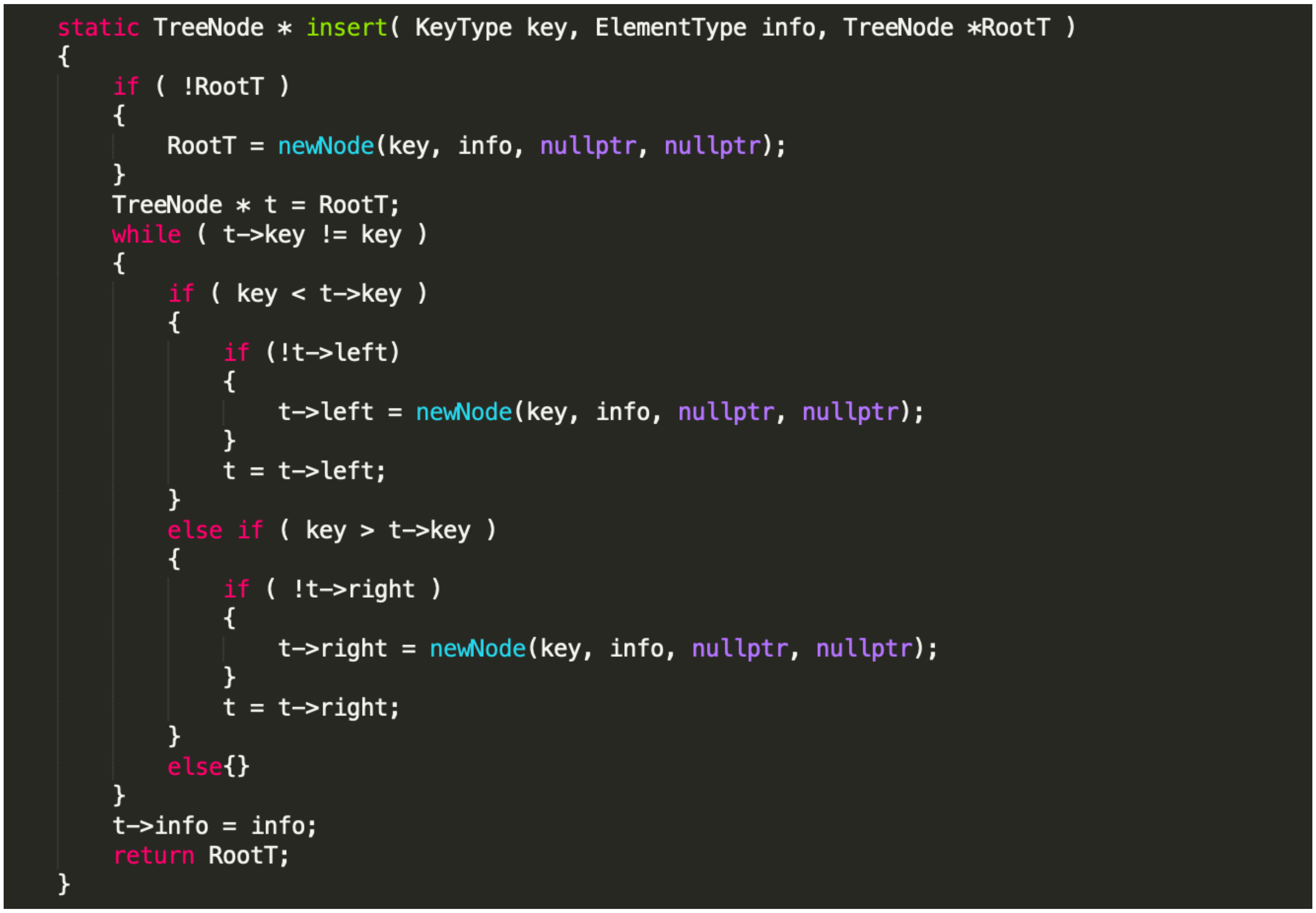
```
void remove(KeyType key)
{
    root = TreeNode<KeyType, ElementType>::remove(key, root);
}
```

O(N) = O(logN)   T(N) = / (recursive)



```
ElementType & operator[] (KeyType s)
{
    TreeNode<KeyType, ElementType> * t = TreeNode<KeyType, ElementType>::find(s, root);
    if (t)
        return TreeNode<KeyType, ElementType>::operator_helper(t);
    else
    {
        insert(s, 0);
        TreeNode<KeyType, ElementType> * t2 = TreeNode<KeyType, ElementType>::find(s, root);
        return TreeNode<KeyType, ElementType>::operator_helper(t2);
    }
}
```

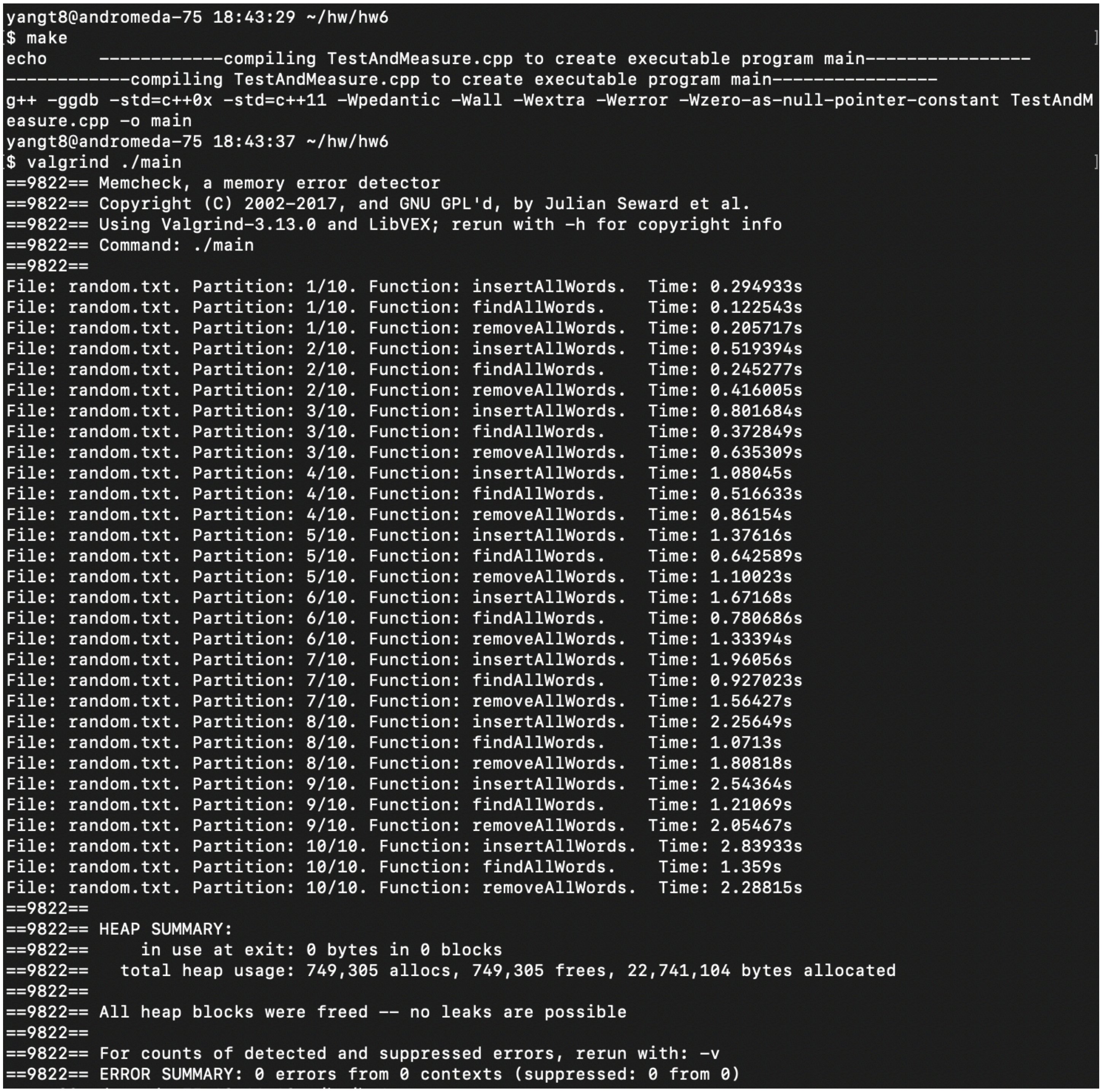
O(N) = O(logN)   T(N) = 1020logN+14



```
static TreeNode * insert( KeyType key, ElementType info, TreeNode *RootT )
{
    if ( !RootT )
    {
        RootT = newNode(key, info, nullptr, nullptr);
    }
    TreeNode * t = RootT;
    while ( t->key != key )
    {
        if ( key < t->key )
        {
            if (!t->left)
            {
                t->left = newNode(key, info, nullptr, nullptr);
            }
            t = t->left;
        }
        else if ( key > t->key )
        {
            if ( !t->right )
            {
                t->right = newNode(key, info, nullptr, nullptr);
            }
            t = t->right;
        }
        else{}
    }
    t->info = info;
    return RootT;
}
```

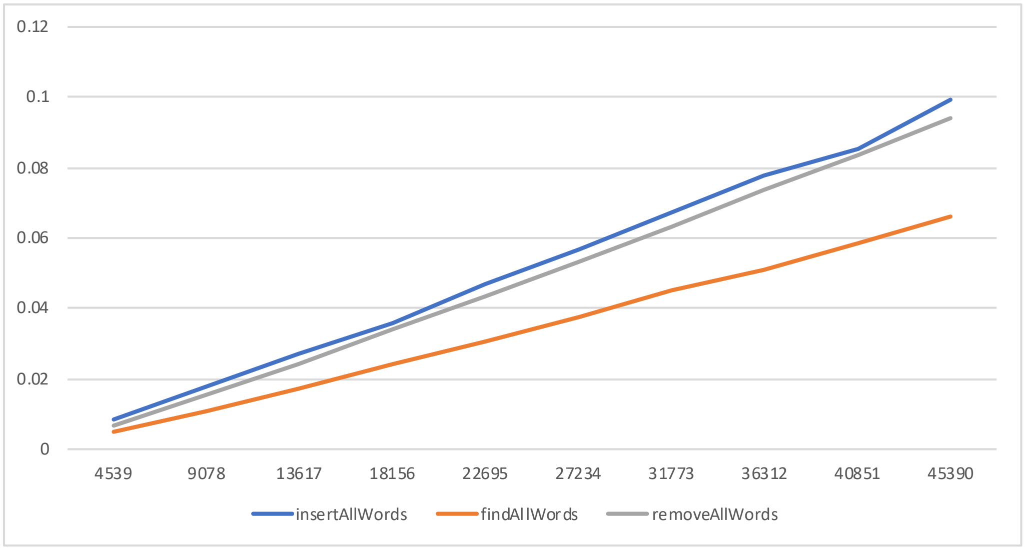
O(N) = O(logN)   T(N) =1010logN+4

Part 2: Testing BST under valgrind with no memory leaks

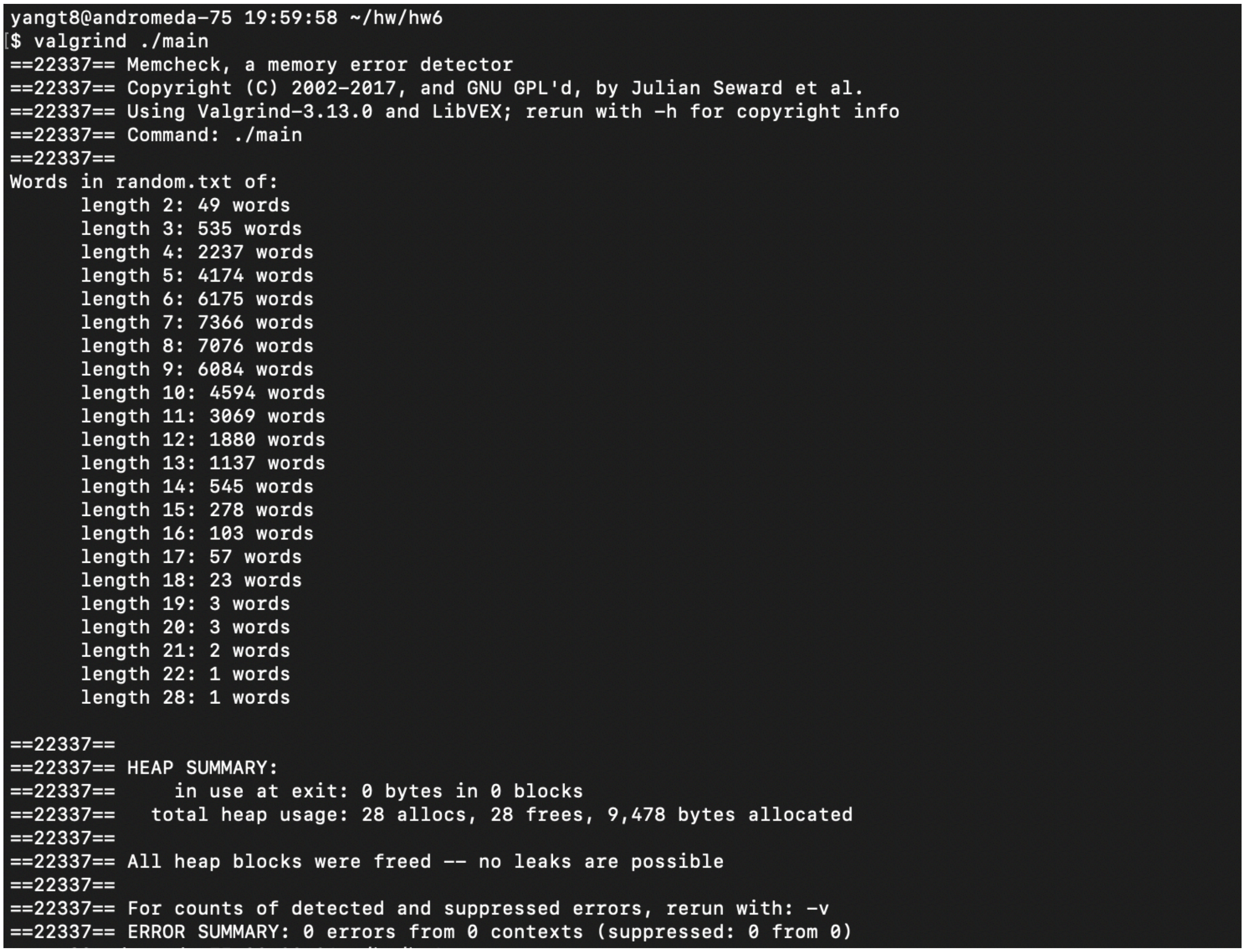


```
yangt8@andromeda-75 18:43:29 ~/hw/hw6
$ make
echo     ------------compiling TestAndMeasure.cpp to create executable program main----------------
------------compiling TestAndMeasure.cpp to create executable program main----------------
g++ -ggdb -std=c++0x -std=c++11 -Wpedantic -Wall -Wextra -Werror -Wzero-as-null-pointer-constant TestAndM
easure.cpp -o main
yangt8@andromeda-75 18:43:37 ~/hw/hw6
$ valgrind ./main
==9822== Memcheck, a memory error detector
==9822== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==9822== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==9822== Command: ./main
==9822==
File: random.txt. Partition: 1/10. Function: insertAllWords.  Time: 0.294933s
File: random.txt. Partition: 1/10. Function: findAllWords.    Time: 0.122543s
File: random.txt. Partition: 1/10. Function: removeAllWords.  Time: 0.205717s
File: random.txt. Partition: 2/10. Function: insertAllWords.  Time: 0.519394s
File: random.txt. Partition: 2/10. Function: findAllWords.    Time: 0.245277s
File: random.txt. Partition: 2/10. Function: removeAllWords.  Time: 0.416005s
File: random.txt. Partition: 3/10. Function: insertAllWords.  Time: 0.801684s
File: random.txt. Partition: 3/10. Function: findAllWords.    Time: 0.372849s
File: random.txt. Partition: 3/10. Function: removeAllWords.  Time: 0.635309s
File: random.txt. Partition: 4/10. Function: insertAllWords.  Time: 1.08045s
File: random.txt. Partition: 4/10. Function: findAllWords.    Time: 0.516633s
File: random.txt. Partition: 4/10. Function: removeAllWords.  Time: 0.86154s
File: random.txt. Partition: 5/10. Function: insertAllWords.  Time: 1.37616s
File: random.txt. Partition: 5/10. Function: findAllWords.    Time: 0.642589s
File: random.txt. Partition: 5/10. Function: removeAllWords.  Time: 1.10023s
File: random.txt. Partition: 6/10. Function: insertAllWords.  Time: 1.67168s
File: random.txt. Partition: 6/10. Function: findAllWords.    Time: 0.780686s
File: random.txt. Partition: 6/10. Function: removeAllWords.  Time: 1.33394s
File: random.txt. Partition: 7/10. Function: insertAllWords.  Time: 1.96056s
File: random.txt. Partition: 7/10. Function: findAllWords.    Time: 0.927023s
File: random.txt. Partition: 7/10. Function: removeAllWords.  Time: 1.56427s
File: random.txt. Partition: 8/10. Function: insertAllWords.  Time: 2.25649s
File: random.txt. Partition: 8/10. Function: findAllWords.    Time: 1.0713s
File: random.txt. Partition: 8/10. Function: removeAllWords.  Time: 1.80818s
File: random.txt. Partition: 9/10. Function: insertAllWords.  Time: 2.54364s
File: random.txt. Partition: 9/10. Function: findAllWords.    Time: 1.21069s
File: random.txt. Partition: 9/10. Function: removeAllWords.  Time: 2.05467s
File: random.txt. Partition: 10/10. Function: insertAllWords.  Time: 2.83933s
File: random.txt. Partition: 10/10. Function: findAllWords.    Time: 1.359s
File: random.txt. Partition: 10/10. Function: removeAllWords.  Time: 2.28815s
==9822==
==9822== HEAP SUMMARY:
==9822==     in use at exit: 0 bytes in 0 blocks
==9822==   total heap usage: 749,305 allocs, 749,305 frees, 22,741,104 bytes allocated
==9822==
==9822== All heap blocks were freed -- no leaks are possible
==9822==
==9822== For counts of detected and suppressed errors, rerun with: -v
==9822== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```

Part 3: Graphing the data



Part 4: Testing TreeNode and BinarySearchTree as a template



```
yangt8@andromeda-75 19:59:58 ~/hw/hw6
[$ valgrind ./main
==22337== Memcheck, a memory error detector
==22337== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==22337== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==22337== Command: ./main
==22337==
Words in random.txt of:
      length 2: 49 words
      length 3: 535 words
      length 4: 2237 words
      length 5: 4174 words
      length 6: 6175 words
      length 7: 7366 words
      length 8: 7076 words
      length 9: 6084 words
      length 10: 4594 words
      length 11: 3069 words
      length 12: 1880 words
      length 13: 1137 words
      length 14: 545 words
      length 15: 278 words
      length 16: 103 words
      length 17: 57 words
      length 18: 23 words
      length 19: 3 words
      length 20: 3 words
      length 21: 2 words
      length 22: 1 words
      length 28: 1 words

==22337==
==22337== HEAP SUMMARY:
==22337==     in use at exit: 0 bytes in 0 blocks
==22337==   total heap usage: 28 allocs, 28 frees, 9,478 bytes allocated
==22337==
==22337== All heap blocks were freed -- no leaks are possible
==22337==
==22337== For counts of detected and suppressed errors, rerun with: -v
==22337== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```