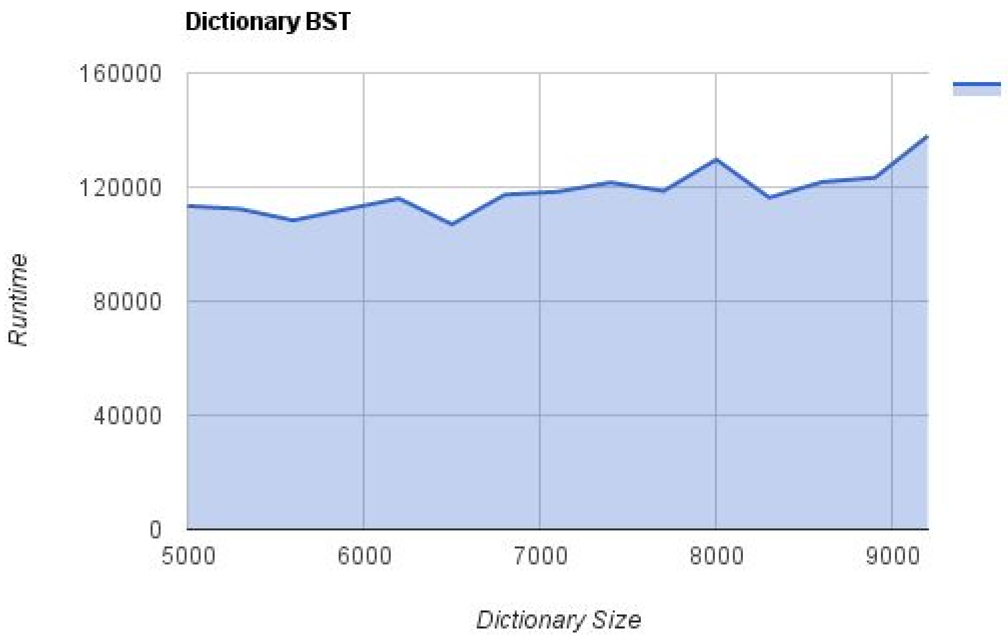
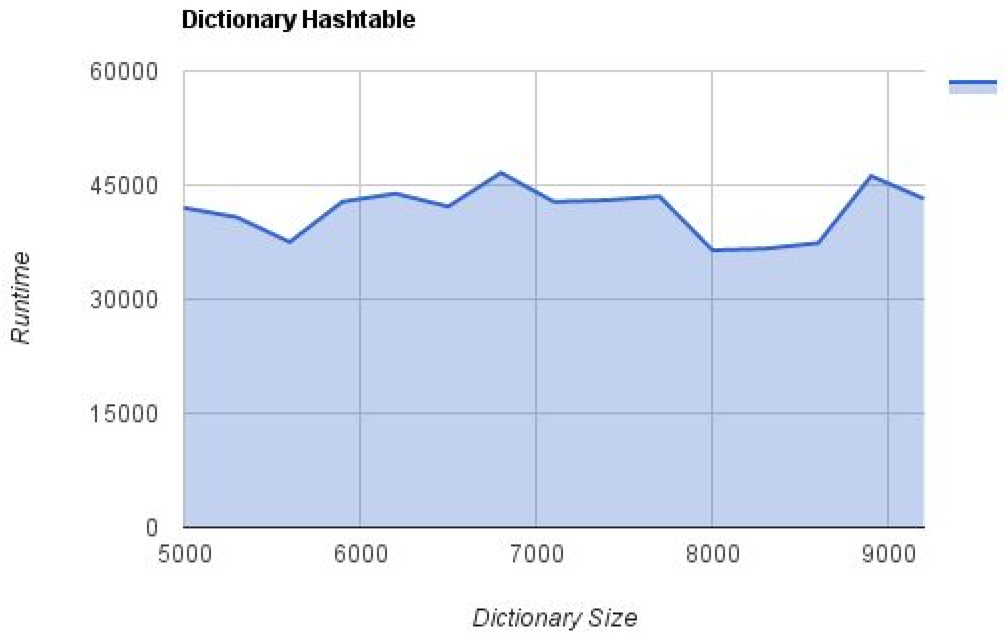
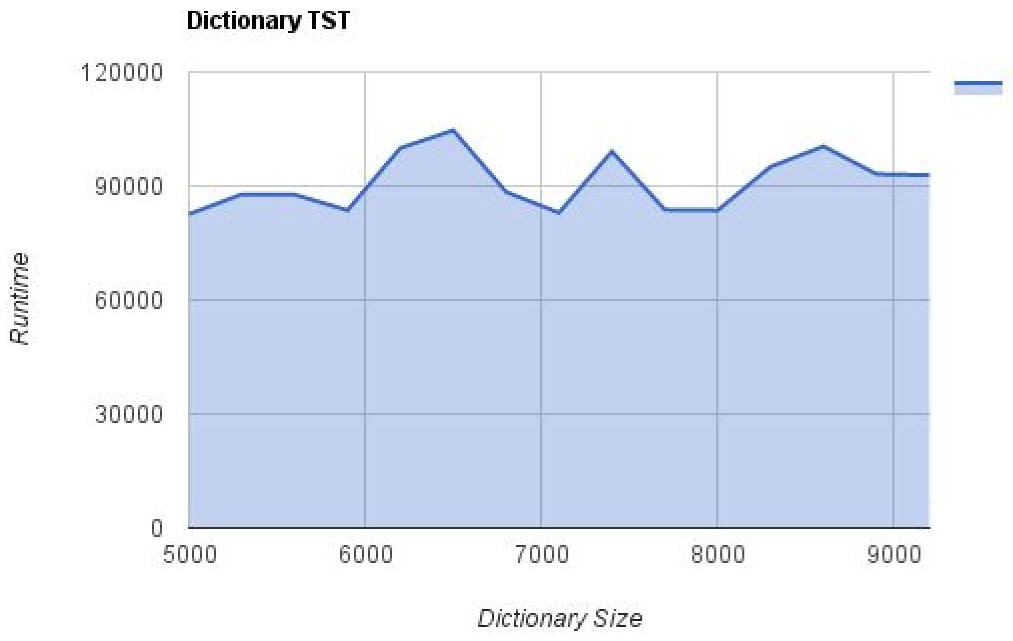
1.



Dictionary BST
160000
120000
80000
40000
0
Runtime
5000
6000
7000
8000
9000
Dictionary Size



Dictionary Hashtable
60000
45000
30000
15000
0
Runtime
5000
6000
7000
8000
9000
Dictionary Size



So according to the graph results, the hash table is actually pretty close to O(1). The Binary Search Tree one is a little bit up and down but in general, it grows up gradually so that it is also close to O(log(n)). But the TST graph is kind of messy maybe because of the instability of my CPU but generally it also grows up, which can be interpreted as O(log(n)).

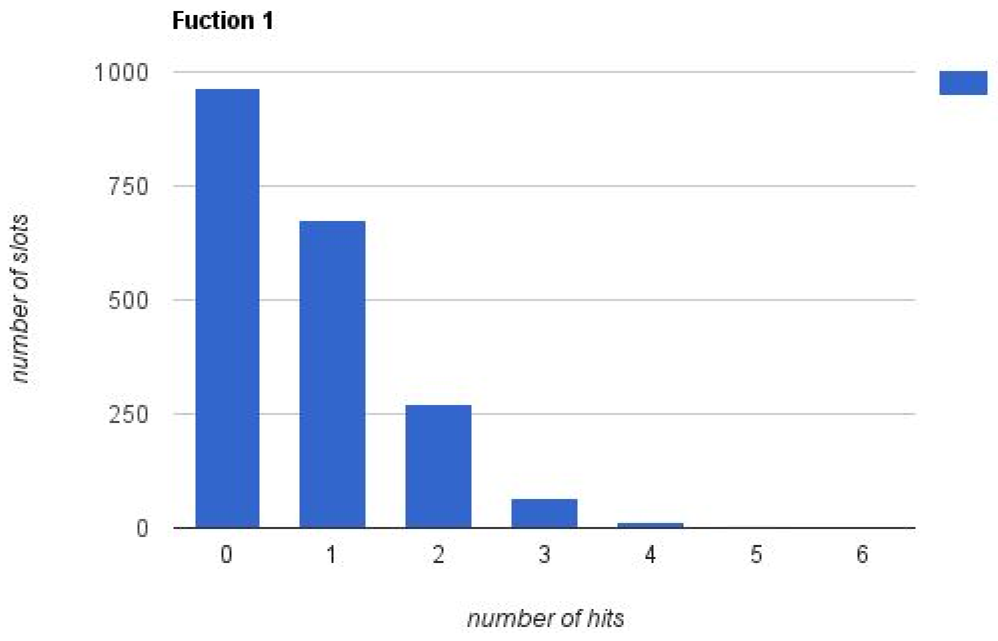
a. For hash function one, it basically multiplies the hash value by 37 and then adds up the ASCII values of each character in each iteration. Then mod the hash value by table size. Check if the hash value is negative. If it is, add it by table size and make it positive. Finally, return the value.
   For hasStringBase256, in every iteration, it moves hash Value 8 bits left, which means it is multiplied by 2^8 then plus the char value. Then mod the hash value by table size. After all of the iteration, return the value as final hashValue.
b. Suppose we have 3 strings: “rido”, “zhuru”, “jiz”, the table size is 10. The hash value for each of them in function 1 is 8, 2, and 1. In function 2 is 5, 9, 8
   As the test output shows out:

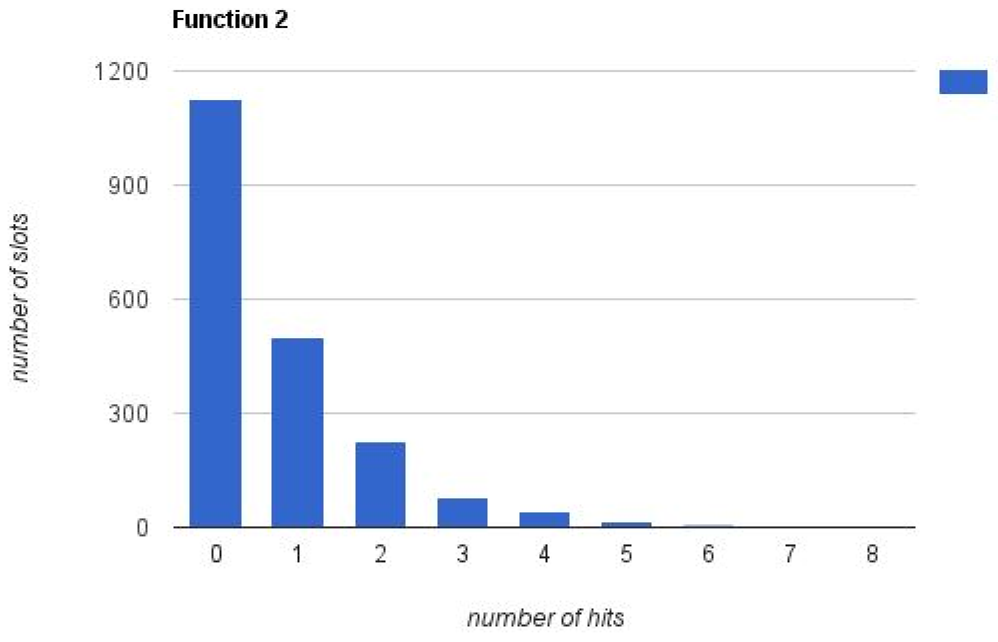
```
./benchhash
8
2
1
5
9
8
```

c.

The output:

```
Printing the statistics for hash Function1 with hash table size 2000
#hits    #slots receiving the #hits
0       966
1       676
2       273
3       67
4       14
5       3
6       1
The average number of steps for a successful search for hash function 1 would be 1.402
The worst case steps that would be needed to find a word is 6
Printing the statistics for hash Function2 with hash table size 2000
#hits    #slots receiving the #hits
0       1125
1       500
2       228
3       81
4       42
5       15
6       6
7       2
8       1
The average number of steps for a successful search for hash function 2 would be 1.68867
The worst case steps that would be needed to find a word is 8
```





d.
The hash function 1 is actually better since it comes out with less collisions. The keys are more separate in table than the second one.