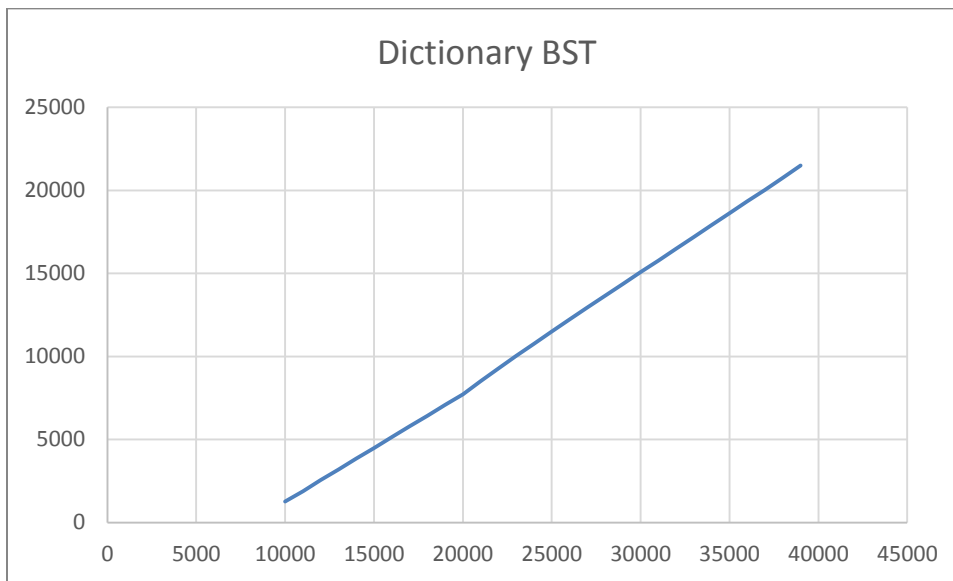


Final Report PA2

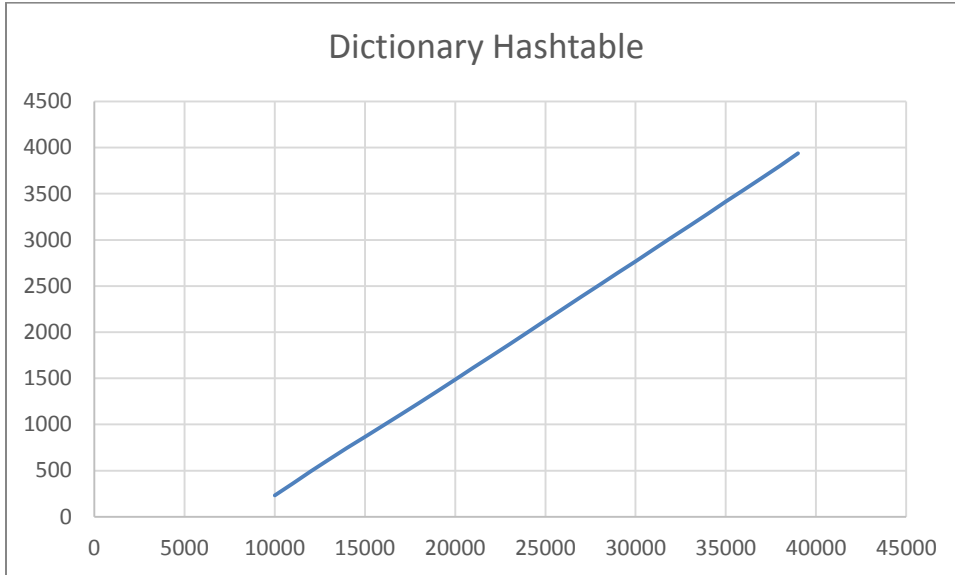
3.4 (1)

DictionaryBST



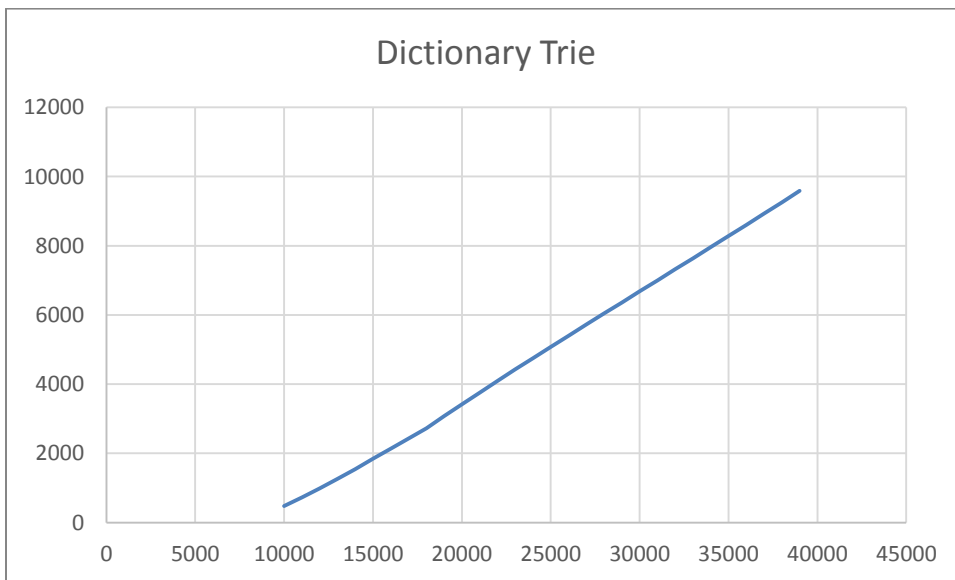
The DictionaryBST graph is not consistent with the analytical running time expectation, $O(\log n)$, because our results in the graph show that maybe in the shuffled_freq_dict.txt, the first letters of each string or word might be different, but the strings in general might have had a linear pattern. This linear pattern leads to a linear time graph.

DictionaryHashtable



The Dictionary Hashtable graph is not consistent with the analytical running time expectation, $O(1)$, because the distribution in the hash table was not even throughout the hash table. This unevenness was a result of possible lack of random shuffling in the shuffled_freq_dict.txt file for the same reason as the Dictionary BST. Since we are assuming separate chaining for the hash table, some of the slots in the hash table have linked lists which results in this time complexity of $O(n)$.

DictionaryTrie



The Dictionary Trie graph is consistent with the analytical running time expectation, $O(K)$, because the graph shows a linear time complexity. It is linear time because since K

is the length of the longest string, it most likely traversed through the longest string of length K.

(1)

(a) <http://research.cs.vt.edu/AVresearch/hashing/strings.php>

This hash function sums up the ASCII values of the characters in a string. Then, the modulus operator will be applied to the sum with the table size, M. ($\text{sum} \% M$).

(b)

<https://courses.cs.washington.edu/courses/cse326/00wi/handouts/lecture15/sld013.htm>

This hash function starts from the end of the string and works its way to the beginning of the string. First, h will be multiplied by 128 and added to the ASCII value of the indicated character of the string at the specific index. Then, the modulus operator will be applied to that result with the table size, tableSize.

(2)

We verified the correctness of each hash function's implementation by testing the strings: bob, an, hat.

Assume that the tableSize is 100. For the string "bob" we expected the result 7 with hash function (a). Using hash function (b), the expected result is 38.

For the string "an" we expected the result 7 with hash function (a). Using hash function (b), the expected result is 77.

For the string "hat" we expected the result 17 with hash function (a). Using hash function (b), the expected result is 64.

After we tested these cases by calculating the results by hand, we tested these strings with our code and the same results held.

(3)

Using freq1.txt

Printing the statistics for hashFunction1 with hash table size: 60000

#hits #slots receiving the #hits

0	58062
1	181
2	124
3	93
4	109
5	85
6	70
7	57
8	66
9	62
10	59

11	50
12	48
13	60
14	54
15	55
16	47
17	53
18	47
19	46
20	45
21	39
22	35
23	34
24	29
25	23
26	25
27	46
28	26
29	25
30	24
31	17
32	21
33	24
34	19
35	19
36	18
37	16
38	12
39	14
40	10
41	10
42	7
43	5
44	3
45	5
46	8
47	3
48	4
49	1
50	1
51	1
54	1
57	2

The average number of steps for a successful search for hash function one is 11.4254
The worst case steps that would be needed to find a word is 57

Printing the statistics for hashFunction2 with hash table size: 60000

#hits #slots receiving the #hits

0 58125

4	2
5	5
6	7
7	18
8	51
9	77
10	91
11	116
12	160
13	153
14	176
15	169
16	192
17	182
18	143
19	107
20	72
21	44
22	37
23	33
24	12
25	14
26	5
27	3
28	5
30	1

The average number of steps for a successful search for hash function two is 7.62067
The worst case steps that would be needed to find a word is 30

Using freq2.txt

Printing the statistics for hashFunction1 with hash table size: 60000

#hits #slots receiving the #hits

0	58353
1	127
2	126
3	99
4	80
5	72
6	72
7	57
8	61
9	51
10	46
11	31
12	44
13	34
14	28
15	30

16	29
17	32
18	37
19	30
20	22
21	25
22	29
23	23
24	26
25	19
26	14
27	20
28	24
29	18
30	16
31	17
32	9
33	11
34	13
35	15
36	18
37	11
38	21
39	17
40	11
41	10
42	23
43	12
44	13
45	8
46	7
47	7
48	7
49	5
50	10
51	6
52	10
53	4
54	4
55	8
56	4
57	6
58	9
59	5
60	4
61	4
62	2
63	2
64	3
65	1
67	1

68	2
69	2
72	1
73	1
75	1

The average number of steps for a successful search for hash function one is 15.3205

The worst case steps that would be needed to find a word is 75

Printing the statistics for hashFunction2 with hash table size: 60000

#hits #slots receiving the #hits

0	58125
4	1
5	4
6	8
7	14
8	36
9	53
10	98
11	130
12	158
13	183
14	194
15	184
16	194
17	145
18	149
19	96
20	78
21	53
22	33
23	29
24	15
25	6
26	9
27	1
28	1
29	2
31	1

The average number of steps for a successful search for hash function two is 7.5817

The worst case steps that would be needed to find a word is 31

Using freq3.txt

Printing the statistics for hashFunction1 with hash table size: 60000

#hits #slots receiving the #hits

0	58332
---	-------

1	201
2	145
3	113
4	86
5	100
6	73
7	59
8	59
9	40
10	43
11	41
12	30
13	20
14	39
15	22
16	29
17	18
18	21
19	21
20	17
21	12
22	16
23	13
24	20
25	21
26	13
27	13
28	11
29	12
30	8
31	12
32	17
33	14
34	16
35	15
36	13
37	10
38	13
39	10
40	12
41	9
42	9
43	8
44	8
45	6
46	13
47	8
48	10
49	12
50	7
51	6

52	7
53	11
54	3
55	6
56	5
57	6
58	7
59	2
60	4
61	5
62	7
63	7
64	1
65	3
66	5
67	5
68	2
69	9
70	3
71	4
72	4
73	2
74	2
75	2
76	3
77	1
78	3
81	1
83	1
84	1
86	1
88	1

The average number of steps for a successful search for hash function one is 17.8794
The worst case steps that would be needed to find a word is 88

Printing the statistics for hashFunction2 with hash table size: 60000

#hits #slots receiving the #hits

0	58125
4	3
5	4
6	10
7	23
8	42
9	85
10	88
11	112
12	152
13	177

14	206
15	161
16	153
17	166
18	124
19	116
20	79
21	60
22	37
23	37
24	15
25	10
26	3
27	5
28	3
29	3
30	1

The average number of steps for a successful search for hash function two is 7.64433
The worst case steps that would be needed to find a word is 30

Using freq_dict.txt

Printing the statistics for hashFunction1 with hash table size: 60000

#hits #slots receiving the #hits

0	57729
1	215
2	165
3	119
4	103
5	95
6	95
7	75
8	80
9	84
10	57
11	71
12	92
13	75
14	92
15	84
16	79
17	81
18	69
19	59
20	53
21	63
22	47
23	46
24	46

25	41
26	24
27	30
28	36
29	21
30	16
31	10
32	15
33	6
34	5
35	7
36	2
37	4
38	1
39	3
40	3
41	1
42	1

The average number of steps for a successful search for hash function one is 8.85907

The worst case steps that would be needed to find a word is 42

Printing the statistics for hashFunction2 with hash table size: 60000

#hits #slots receiving the #hits

0	58125
2	1
5	4
6	18
7	23
8	50
9	56
10	89
11	123
12	156
13	179
14	173
15	179
16	179
17	161
18	129
19	104
20	79
21	57
22	44
23	25
24	18
25	13
26	5
27	7
28	1

29 2

The average number of steps for a successful search for hash function two is 7.5138

The worst case steps that would be needed to find a word is 29

Using freq_dict.txt

Printing the statistics for hashFunction1 with hash table size: 60000

#hits #slots receiving the #hits

0	58681
1	135
2	118
3	82
4	72
5	43
6	51
7	49
8	37
9	47
10	29
11	25
12	19
13	17
14	23
15	18
16	19
17	17
18	17
19	20
20	13
21	21
22	13
23	22
24	14
25	9
26	9
27	9
28	8
29	13
30	12
31	11
32	14
33	7
34	9
35	7
36	9
37	7
38	3
39	8
40	3

41	7
42	10
43	13
44	3
45	7
46	3
47	8
48	7
49	8
50	7
51	5
52	4
53	7
54	5
55	4
56	3
57	3
58	6
59	6
60	9
61	5
62	3
63	5
64	2
65	1
66	7
67	6
68	5
69	3
70	6
71	4
72	6
73	3
74	4
75	5
76	3
77	3
78	2
79	5
80	2
81	3
83	3
84	5
85	4
86	2
87	4
88	3
89	4
90	1
91	1
93	5

94	1
95	4
96	1
97	2
98	2
99	1
100	3
102	1
103	1
106	1
107	1
110	1
112	1

The average number of steps for a successful search for hash function one is 23.6903

The worst case steps that would be needed to find a word is 112

Printing the statistics for hashFunction2 with hash table size: 60000

<u>#hits</u>	<u>#slots receiving the #hits</u>
--------------	-----------------------------------

0	52985
1	4878
2	1579
3	441
4	92
5	17
6	7
7	1

The average number of steps for a successful search for hash function two is 0.4537

The worst case steps that would be needed to find a word is 7

(4) Hash function (b) is better, based on our output, because the average number of steps for a successful search is more efficient in hash function (b). Also, the worst case steps to find a word using hash function (b) is less than the worst case steps to find a word using hash function (a).