

COMP 6651 DD 2172 Programming Assignment 1

Student Name: Shyam Kantesariya

Student ID# 40042715

Date: 8th Oct 2017

Attachment:

Please find code in “code” folder.

Test results and input files used for each individual test case are in test_files folder.

Modifications required to run code:

Please modify values in “Constant class” as per host machine directory set up.

Runtime complexity

1. Linear:

While searching for correct word, if we go linearly then complexity would be:
 $O(n*m*L)$ where n = dictionary size, m = average size of correct word, L = length of misspelled word

2. BKTree:

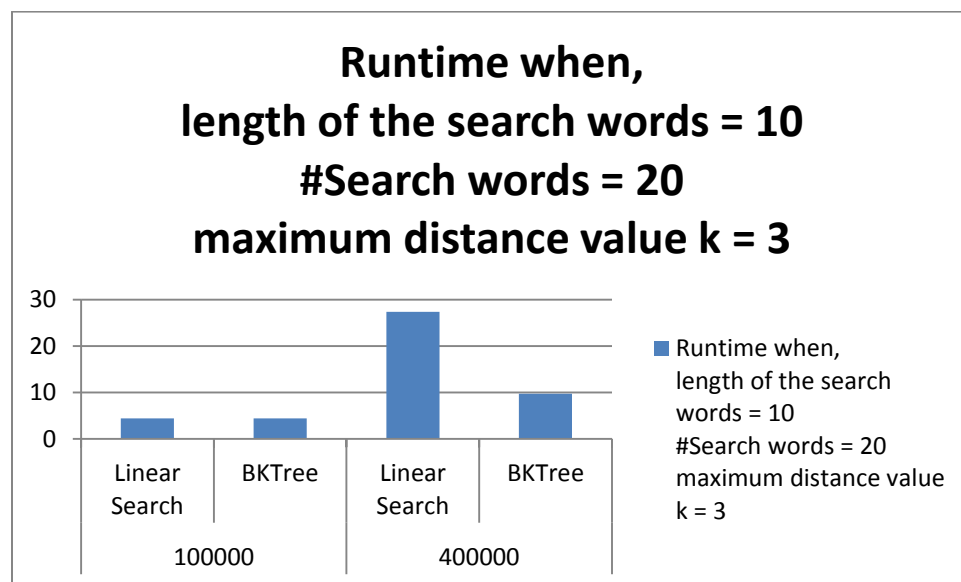
BKTree has $O(L_1*L_2*\log n)$ complexity, where L_1 = average length of word in our dictionary,
 L_2 = the length of misspelled word and n = dictionary size

- Analyze the impact of dictionary size on runtime while keeping maxDistance and input sentence words same

Following are the results on running both linear and BKTree search programs on same set of words.

Test Case#	Dictionary Size	Algorithm Type	Runtime when, length of the search words = 10 #Search words = 20 maximum distance value k = 3
TC#1	100000	Linear Search	4.425
		BKTree	4.429
TC#2	400000	Linear Search	27.392
		BKTree	9.749

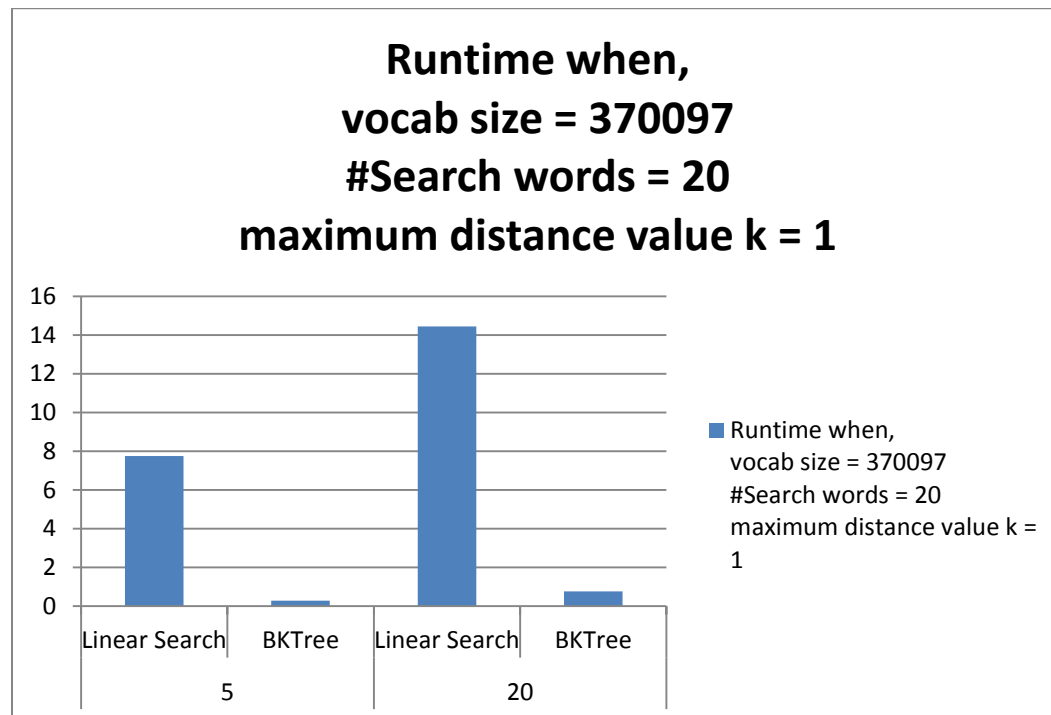
Here is the chart to better visualize the runtime difference.



Observation: We can see that for small set of dictionary words, runtime is almost similar for both the algorithms but while increasing the dictionary size, BKTree out performs linear search.

- Analyze the impact of search word length while keeping dictionary size and maxDistance constant for different search word size

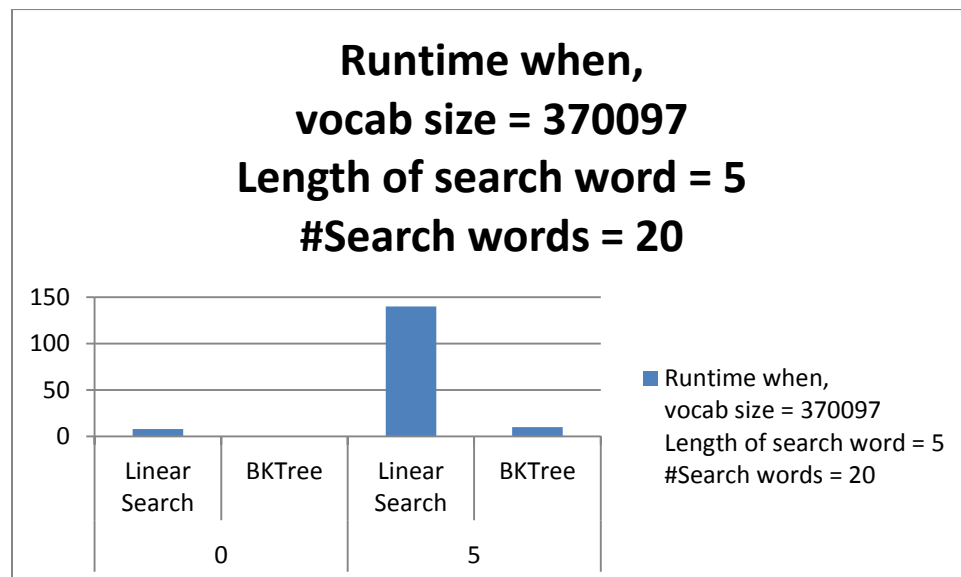
Test Case#	Length of search word	Algorithm Type	Runtime when, vocab size = 370097 #Search words = 20 maximum distance value k = 1
TC#3	5	Linear Search	7.755
		BKTree	0.284
TC#4	20	Linear Search	14.45
		BKTree	0.766



Observation: As we can see that in both the cases BKTree works well.

- Analyze the impact of maxDistance k while keeping dictionary size and search word length constant for different maxDistance values

maximum distance value k	Algorithm Type	Runtime when, vocab size = 370097 Length of search word = 5 #Search words = 20
0	Linear Search	7.853
	BKTree	0.024
5	Linear Search	140.02
	BKTree	9.985



Observation: We can see a drastic difference in runtime for both the algorithms while increasing k value. The change is exponential in fact.