Dhanushka S.M.V.
E/16/083

Lab 04 - Tree ADT - Report

Assumptions:

- assume each word given does not exceed 50 characters (MAX_WORD_SIZE 50)
- remove all non alphabetic characters and convert the words to lowercase before the words are stored to the dictionary

(a) Memory space usage

File Name	Part 1 implementation(bytes)	Part 2 implementation(bytes)
wordlist1000.txt	216000	124320
wordlist10000.txt	2160216	1809024
wordlist70000.txt	15099264	7644224

As the amount of different words are increased, the memory space usage gets increased due to the increasing number of nodes. Here the part 2 implementation is a compressed version of a trie (radix) and it is obvious that the memory usage is relatively low in the part 2 implementation compared to the part 1 implementation.

Ex:
Memory usage dropped for wordlist70000.txt in part 2 compared to part 1 = 49.3735 %

(b) Time taken to store the dictionary

File Name	Part 1 implementation (s)	Part 2 implementation (s)
wordlist1000.txt	0.00365900	0.00339600
wordlist10000.txt	0.02502300	0.02137100
wordlist70000.txt	0.10146400	0.06337100

When the number of inserting words is low, the storing time does not show considerable difference. But when the number of inserting words are increased, part 2 implementation shows a considerable amount less time compared to part 1 implementation.

Ex:
Time drop of storing the dictionary for wordlist70000.txt in part 2 compared to part 1
= 37.5433 %

(c) Time taken to print a list of suggestions for chosen word prefixes

1. For word prefix - ‘att’

File Name	Part 1 implementation	Part 2 implementation
wordlist1000.txt	0.000047	0.000035
wordlist10000.txt	0.000374	0.000319
wordlist70000.txt	0.000707	0.000574

2. For word prefix - ‘to’

File Name	Part 1 implementation	Part 2 implementation
wordlist1000.txt	0.000144	0.000102
wordlist10000.txt	0.000530	0.000554
wordlist70000.txt	0.005316	0.003332

3. For word prefix - ‘ec’

File Name	Part 1 implementation	Part 2 implementation
wordlist1000.txt	0.000023	0.000027
wordlist10000.txt	0.000176	0.000094
wordlist70000.txt	0.000890	0.000716

If we observe the above tables, the time taken to print a list of suggestions for chosen word prefixes, sometimes part 2 implementation records better time compared to part 1 implementation and sometimes part 1 implementation records better time compared to part 2 implementation.

But the difference of the time is negligible. Hence we can assume that the both implementations take the same amount of time to print the suggestions.