CO322 Lab07- Discussion

E/14/317

1) Memory Space Usage

	Trie (bytes)	Radix Trie (bytes)
Size of the trieNode	108	112
Number of nodes	968	467
Total Memory Space	104544	52344

If the Trie is dense and all inner nodes contain 1 character both implementation does not make a big difference. Because in that scenario both tries will look same. But if the trie is less dense and there are more letters on a given node Radix trie implementation will save a lot more memory space.

Radix trie saves nearly twice the amount of Memory required by the regular trie. So if the trie is small and not dense Radix trie is memory efficient

2. Time Consumption

For this test a different text file is used with larger size.(wordlist2.txt)

	Trie	Radix Trie
Time to traverse	6.037 s	5.657 s
Time to insertion	0.269 s	0.229 s

Considering the inserting time we can see that Radix trie is faster than the regular trie. Insertion time is also a bit faster but it is not much different.