REPORT

2. Compare performance of the original trie and the compressed trie. Measure time to generate suggested word list for word "the" in both scenarios and report it.

Time Measurements:

	Time to Insert	Time to traverse 'the'
Original Trie	0.117197	0.005478
Compressed Trie	0.270488	0.002216

According to my implementation, the insertion time is lesser for the original trie compared to the insertion time of the compressed trie. But in contrast, the traversal time is lesser for the compressed trie with respect to the traversal time of original trie for the given word 'the'.

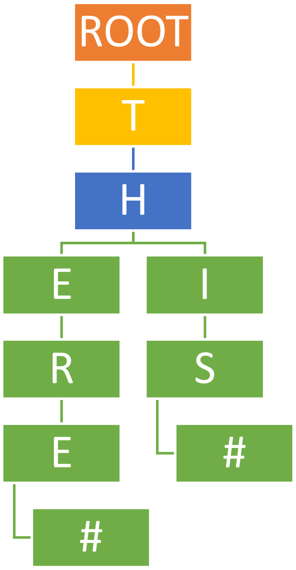
3. Discuss memory consumptions of a regular trie and a Radix Tree and compare them.

In a regular trie, all nodes will contain only one character. In contrast, a radix tree can store multiple characters per one node. Threrefore, for a given number of words the radix tree implementation will hav a lesser number of nodes compared to a regular trie. Hence, the trie structure will be more denser than the radix tree for a given number of words to be stored. As a result, radix tree implementation will consume lesser amount of memeory space with respect to a regular trie.

Consider an example where the following two words needs to be stored.

- THERE
- THIS

Regular Trie Implementation:



Memory consumption:

Amount for addresses of 26 children = 26 * 8 bytes

Amount to store 1 character = 1 byte

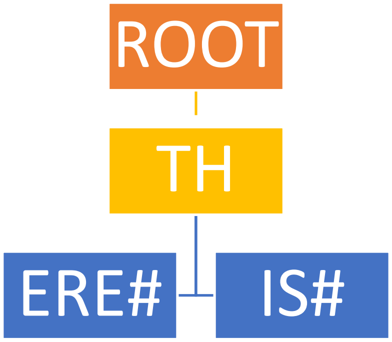
Amount for a single node = 26 * 8 + 1 bytes

= 209 bytes

Total Amount for all nodes = 10 * 209 bytes

= 2090 bytes

Radix Tree Implementation:



Memory consumption:

Amount for addresses of 26 children = 26 * 8 bytes (Assuming all 26 addreses are stored always)

Amount to store a character address = 8 byte

Amount for a single node = 26 * 8 + 8 bytes

= 216 bytes

Total Amount for all nodes = 4 * 216 bytes

= 864 bytes

Therefore, the radix tree implementation consumes only lesser than half the amount of memeory space required by a regular trie. Hence, the radix implementation saves more memory for the same amount of data to be stored. Hence, the radix tree implementation is more memory efficient with respect to the regular trie.