

## **Report**

A Trie is a special data structure used to store strings that can be visualized like a graph. It consists of nodes and edges. Each node consists of at max 26 children and edges connect each parent node to its children. These 26 pointers are nothing but pointers for each of the 26 letters of the English alphabet. A separate edge is maintained for every edge.

Strings are stored in a top to bottom manner on the basis of their prefix in a trie. All prefixes of length 1 are stored at until level 1, all prefixes of length 2 are stored at until level 2 and so on.

A radix tree is a compressed version of a trie. In a trie, on each edge store a single letter, while in a PATRICIA tree (or radix tree) store whole words.

In the codes provided memory space usage and time taken to store and search are compared below. For searching every scenario used “**he**”

### **wordlist1000**

For trie -:

Memory space usage: 326808 Bytes

Time taken to store: 0.000000 s

Time taken to search: 5.311000 s

For radix -:

Memory space usage: 130256 Bytes

Time taken to store: 0.001000 s

Time taken to search: 0.040000 s

### **wordlist10000**

For trie -:

Memory space usage: 2611332 Bytes

Time taken to store: 0.015000 s

Time taken to search: 5.456000 s

For radix -:

Memory space usage: 1277472 Bytes

Time taken to store: 0.008000 s

Time taken to search: 0.218000 s

### **wordlist70000**

For trie -:

Memory space usage: 23355000 Bytes

Time taken to store: 0.058000 s

Time taken to search: 2.061000 s

For radix -:

Memory space usage: 9403072 Bytes

Time taken to store: 0.046000 s

Time taken to search: 2.145000 s

It is clear the Radix tree structure uses less memory space and less time to search than trie tree structure, but for insertion trie uses less time compared to radix.