

GeeksforGeeks

A computer science portal for geeks

GeeksQuiz

- [Home](#)
- [Algorithms](#)
- [DS](#)
- [GATE](#)
- [Interview Corner](#)
- [Q&A](#)
- [C](#)
- [C++](#)
- [Java](#)
- [Books](#)
- [Contribute](#)
- [Contests](#)
- [Jobs](#)

[Array](#)

[Bit Magic](#)

[C/C++](#)

[Articles](#)

[GFactS](#)

[Linked List](#)

[MCQ](#)

[Misc](#)

[Output](#)

[String](#)

[Tree](#)

[Graph](#)

Trie | (Insert and Search)

[Trie](#) is an efficient information *retrieval* data structure. Using trie, search complexities can be brought to optimal limit (key length). If we store keys in binary search tree, a well balanced BST will need time proportional to $M * \log N$, where M is maximum string length and N is number of keys in tree. Using trie, we can search the key in $O(M)$ time. However the penalty is on trie storage requirements.

Every node of trie consists of multiple branches. Each branch represents a possible character of keys. We need to mark the last node of every key as leaf node. A trie node field *value* will be used to distinguish the node as leaf node (there are other uses of the *value* field). A simple structure to represent nodes of English alphabet can be as following,

```
struct trie_node
{
```

```

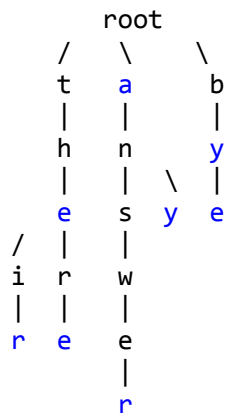
int value; /* Used to mark leaf nodes */
trie_node_t *children[ALPHABET_SIZE];
};

```

Inserting a key into trie is simple approach. Every character of input key is inserted as an individual trie node. Note that the *children* is an array of pointers to next level trie nodes. The key character acts as an index into the array *children*. If the input key is new or an extension of existing key, we need to construct non-existing nodes of the key, and mark leaf node. If the input key is prefix of existing key in trie, we simply mark the last node of key as leaf. The key length determines trie depth.

Searching for a key is similar to insert operation, however we only compare the characters and move down. The search can terminate due to end of string or lack of key in trie. In the former case, if the *value* field of last node is non-zero then the key exists in trie. In the second case, the search terminates without examining all the characters of key, since the key is not present in trie.

The following picture explains construction of trie using keys given in the example below,



In the picture, every character is of type *trie_node_t*. For example, the *root* is of type *trie_node_t*, and it's children *a*, *b* and *t* are filled, all other nodes of root will be NULL. Similarly, "a" at the next level is having only one child ("n"), all other children are NULL. The leaf nodes are in blue.

Insert and search costs **O(key_length)**, however the memory requirements of trie is **O(ALPHABET_SIZE * key_length * N)** where N is number of keys in trie. There are efficient representation of trie nodes (e.g. compressed trie, [ternary search tree](#), etc.) to minimize memory requirements of trie.

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define ARRAY_SIZE(a) sizeof(a)/sizeof(a[0])

// Alphabet size (# of symbols)
#define ALPHABET_SIZE (26)
// Converts key current character into index
// use only 'a' through 'z' and lower case
#define CHAR_TO_INDEX(c) ((int)c - (int)'a')

// trie node
typedef struct trie_node trie_node_t;

```

```
struct trie_node
{
    int value;
    trie_node_t *children[ALPHABET_SIZE];
};

// trie ADT
typedef struct trie trie_t;
struct trie
{
    trie_node_t *root;
    int count;
};

// Returns new trie node (initialized to NULLs)
trie_node_t *getNode(void)
{
    trie_node_t *pNode = NULL;

    pNode = (trie_node_t *)malloc(sizeof(trie_node_t));

    if( pNode )
    {
        int i;

        pNode->value = 0;

        for(i = 0; i < ALPHABET_SIZE; i++)
        {
            pNode->children[i] = NULL;
        }

        return pNode;
    }
}

// Initializes trie (root is dummy node)
void initialize(trie_t *pTrie)
{
    pTrie->root = getNode();
    pTrie->count = 0;
}

// If not present, inserts key into trie
// If the key is prefix of trie node, just marks leaf node
void insert(trie_t *pTrie, char key[])
{
    int level;
    int length = strlen(key);
    int index;
    trie_node_t *pCrawl;

    pTrie->count++;
```

```

pCrawl = pTrie->root;

for( level = 0; level < length; level++ )
{
    index = CHAR_TO_INDEX(key[level]);
    if( !pCrawl->children[index] )
    {
        pCrawl->children[index] = getNode();
    }

    pCrawl = pCrawl->children[index];
}

// mark last node as leaf
pCrawl->value = pTrie->count;
}

// Returns non zero, if key presents in trie
int search(trie_t *pTrie, char key[])
{
    int level;
    int length = strlen(key);
    int index;
    trie_node_t *pCrawl;

    pCrawl = pTrie->root;

    for( level = 0; level < length; level++ )
    {
        index = CHAR_TO_INDEX(key[level]);

        if( !pCrawl->children[index] )
        {
            return 0;
        }

        pCrawl = pCrawl->children[index];
    }

    return (0 != pCrawl && pCrawl->value);
}

// Driver
int main()
{
    // Input keys (use only 'a' through 'z' and lower case)
    char keys[][8] = {"the", "a", "there", "answer", "any", "by", "bye", "the"};
    trie_t trie;

    char output[][32] = {"Not present in trie", "Present in trie"};

    initialize(&trie);

```

```
// Construct trie
for(int i = 0; i < ARRAY_SIZE(keys); i++)
{
    insert(&trie, keys[i]);
}

// Search for different keys
printf("%s --- %s\n", "the", output[search(&trie, "the")] );
printf("%s --- %s\n", "these", output[search(&trie, "these")] );
printf("%s --- %s\n", "their", output[search(&trie, "their")] );
printf("%s --- %s\n", "thaw", output[search(&trie, "thaw")] );

return 0;
}
```

— [Venki](#). Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

Related Topics:

- [How to handle duplicates in Binary Search Tree?](#)
- [Handshaking Lemma and Interesting Tree Properties](#)
- [Advantages of BST over Hash Table](#)
- [Given a binary tree, how do you remove all the half nodes?](#)
- [K'th Largest Element in BST when modification to BST is not allowed](#)
- [Vertex Cover Problem | Set 2 \(Dynamic Programming Solution for Tree\)](#)
- [Check whether a binary tree is a complete tree or not | Set 2 \(Recursive Solution\)](#)
- [Check whether a binary tree is a full binary tree or not](#)

Tags: [Advance Data Structures](#)



Writing code in comment? Please use ideone.com and share the link here.

 Recommend **2**  Share

Sort by Newest ▾



Join the discussion...

**name** · 11 days ago

can you give me java code for the same

  · Reply · Share ›**Koustav Chatterjee** → name · 11 days ago<http://ideone.com/LnkdcT>  · Reply · Share ›**Cracker** · 15 days ago

C++ implementation of Trie:

<http://algods-cracker.blogspot...>  · Reply · Share ›**Siddhant** · 22 days ago

why 2 structures are used?

  · Reply · Share ›**Ajitesh Mandal** · 2 months ago

another way of code

<http://ideone.com/u3zc4Z>  · Reply · Share ›**John** → Ajitesh Mandal · 2 months ago

Hello! is there a way you can help me?

Im reusing your code and learning from it, can you make a function that can print the trie?

I would be so gratefull!

  · Reply · Share ›**Sheetal** → John · a month ago<http://ideone.com/HcvHBw>  · Reply · Share ›**Sheetal** → John · a month ago

public class Test123 {

```

public static void main(String[] args) throws ParseException {
    Test123 test = new Test123();
    Trie trie = test.new Trie();

    dateFormat = new SimpleDateFormat("yyyyMMdd");
    String[] str = { "Prakash", "Venkat", "Ullman", "AnandRajaraman",
        "JureLeskovec", "Prasanna", "Pandya" };

    for (String str : str) {
        trie.insert(str);
    }
    System.out.println(trie.search("P R A K A S H"));
    System.out.println("-----");
    trie.traverseUtil();
    System.out.println("-----");
    trie.prefixFetch("p");
}

```

[see more](#)[^](#) | [v](#) • [Reply](#) • [Share](#) ›**Anonymous** • 4 months ago

Is Trie the best data structure for predicting word (T9) type applications?

[^](#) | [v](#) • [Reply](#) • [Share](#) ›**Abhishek** • 5 months ago

Can i use the same this trie code for dictionary purpose?

[^](#) | [v](#) • [Reply](#) • [Share](#) ›**Sun** → **Abhishek** • 5 months ago

Yes, You Can !

[^](#) | [v](#) • [Reply](#) • [Share](#) ›**manuel velasquez** → **Sun** • 4 months ago

how can i assign a simple linked list to that nodes (leaf)?

[^](#) | [v](#) • [Reply](#) • [Share](#) ›**Hello_world** • 5 months ago

what does value represent in trie_node DS ?

[^](#) | [v](#) • [Reply](#) • [Share](#) ›**Anurag Singh** → **Hello_world** • 5 months ago

Few statements from above articlie

"A trie node field value will be used to distinguish the node as leaf node"

"int value; /* I used to mark leaf nodes */"

int value, / USED TO MARK LEAF NODES /

A node with non-zero value represents a leaf node.

^ | v • Reply • Share ›



Tokala sai teja • 6 months ago

Java Implementation For Java Folks

<http://ideone.com/cgDpzN>

2 ^ | v • Reply • Share ›



Guest • 6 months ago

In the search function, I understood the condition `pCrawl->value`, so you test whether the string is a leaf node or not..

For eg.

string present in trie- `geeksforgeeks`
and i try to search `geeks`

so its not in the tree correct? as 's' node of `geeks` wont have the leaf node value..

But what does the `0!=pCrawl` condition check?

Isn't it the same as doing

```
if( !pCrawl->children[index] )
{
    return 0;
}
```

inside the loop?

Thankyou :)

^ | v • Reply • Share ›



Jasdeep Singh ➔ Guest • 5 months ago

Yes it is. This is a preference of style. What this statement will return is a bool value which some people prefer. The code will not break even if you do `return pCrawl->value..`

1 ^ | v • Reply • Share ›



Aditya Goel • 7 months ago

<http://ideone.com/eF7Nih>

my implemetation of Trie(Insertion, deletion and Searching)

It is working fine in C-Free but giving Runtime error on ideone. Not sure why. I don't have time to debug the issue. Please if anybody found the issue, comment back here..

^ | v • Reply • Share ›



Vikas Malviya → Aditya Goel • 7 months ago

1. Return the node pointer from your getnewnode function.

2. Never use global data

It is running now.

<http://ideone.com/U9ryd1>

2 ^ | v • Reply • Share ›



vishal → Vikas Malviya • 6 months ago

traversing ?? in trie

^ | v • Reply • Share ›



Aditya Goel → Vikas Malviya • 7 months ago

Thanks bud.. not sure how I missed that :)

^ | v • Reply • Share ›



Ankit Aggarwal • 7 months ago

how to insert number like 911 in this trie? This program is giving run-time error for this type of insertion.

^ | v • Reply • Share ›



Vikas Malviya → Ankit Aggarwal • 7 months ago

Hello Ankit .

You can insert numbers for sure.

Just modify the macro to

```
#define CHAR_TO_INDEX(c) ((int)c - (int)'0')
```

Also , if you want to store all the ASCII characters in the Trie , then ,increase the size of the children array.

^ | v • Reply • Share ›



Guest → Vikas Malviya • 7 months ago

Also reduce ALPHABET_SIZE to 10 to save memory

^ | v • Reply • Share ›



Aditya Goel → Ankit Aggarwal • 7 months ago

Number cannot be inserted in this Trie. Only string and that too in lower-case. Trie is basically used for implementing dictionaries, etc, so it is implemented in a way to accept only stream of characters. If u want to insert numbers, u can by modifying some code but then u can't insert an String

but then I can't insert an string.

^ | v • Reply • Share ›



Phoenix • 7 months ago

Yes it does handle the case...or else it wouldn't have been a trie at first place -_-

^ | v • Reply • Share ›



Srikar ED • 8 months ago

Does the insert method handle the case when we insert a key which is already present in the tree?? I dont think so....

^ | v • Reply • Share ›



Abhishek Kashyap Jha • 8 months ago

What will be the logic of implementing a phonebook using a trie? How will I search a number given a trie of names? How to store the phone numbers and how to retrieve them?

^ | v • Reply • Share ›



neelabhsingh • 9 months ago

Java Solution

```
import java.util.Scanner;
class trie_node
{
int value;
trie_node[] children;
}
public class TrieTree {
public static void main(String [] args){
Scanner scan=new Scanner(System.in);
TrieTree tt=new TrieTree();
String word;
trie_node root = new trie_node();
root.value=0;
root.children=new trie_node[26];
for(int i=0;i<26;i++){
word=scan.next();
```

[see more](#)

^ | v • Reply • Share ›



vrg • 9 months ago

Where are we storing the char in the node. ?

for example I was expecting something like in the trie node

```

struct trie_node
{
char data // contains the current node character

int value;
trie_node_t *children[ALPHABET_SIZE];
};

```

^ | v • Reply • Share ›



Divino César → vrg • 9 months ago

Chars are implicitly stored in the "children" vector. The idea is to keep a one-to-one mapping between the alphabet chars and the positions of the "children" vector.

^ | v • Reply • Share ›



Saurabh • 9 months ago

```

void base::prefixSearch(const char prefix[],ptrie crawl, int done)
{
ptrie temp = crawl;
ptrie temp2 = NULL;
int i = 0;
if(crawl == NULL)
return;

string s(prefix);

if(crawl->value)
{
cout<<" "<<s.c_str()<<endl; }="" for(i="0;i<26;i++)" {="" if(temp="" &&="" temp="">children[i])
{
s.push_back('a' + i);
prefixSearch(s.c_str(),temp->children[i],0);
s.erase(s.size()-1);
}
}
}

```

^ | v • Reply • Share ›



kaushik Lele • 10 months ago

A java code implementation can be found here

<https://community.oracle.com/t...>

^ | v • Reply • Share ›



Umashankar • 10 months ago

Hi, Thanks for a nice piece of code. Any suggestions on using words which have special

characters like "can't" -: " ' " or '-'. They don't seem to be supported right now.

The macro 'CHAR_TO_INDEX' you have used seems to give a negative value in such cases.

^ | v • Reply • Share ›



Anuj → Umashankar • 9 months ago

That's because ASCII value of 'a' is greater than the characters that you have mentioned .

^ | v • Reply • Share ›



kick • 10 months ago

if(!pCrawl->children[index]) what this line supposed to do??/

^ | v • Reply • Share ›



Ekta Goel → kick • 10 months ago

The line checks if the child with index INDEX for the trie node pcrawl exists or not. Eg, while searching for the word 'the', if there is no word starting with 't' the statement will return 0. I.e. there must be the word present as a whole (true when pcrawl->value is true) or some prefix of the word must be present for searching the next character of the word till the end. Here, if 't' is present, the next index is for 'h', if there is no word or a prefix with an 'h' after t, it will return false. Similarly it continues..

^ | v • Reply • Share ›



MK • 10 months ago

What is the necessity of checking (pcrawl != 0) in search function at the end?

^ | v • Reply • Share ›



Ekta Goel → MK • 10 months ago

Even i don't understand the use of that, the possible reason can be because we need a return value in terms of bool, so they are just taking AND of pcrawl value which will be true if the control reaches that line and the value field will also be set. However, i changed the code to this <http://ideone.com/nKQeuy> (line 108-109) and it works fine. Any problematic case, let me know..!!

^ | v • Reply • Share ›



Ekta Goel → Ekta Goel • 10 months ago

Yeah, above is correct. See the below comment by Luas and Venki .

^ | v • Reply • Share ›



Guest • a year ago

how is insertion m logn suppose a words abcd bcde gfhd hgjf..time will be length of string * no. of strings!

^ | v • Reply • Share ›



Guest · a year ago

if you search "thei", the program returns "Present in the Trie"
there's a bug, please somebody confirm.

^ | v · Reply · Share ›



Ekta Goel → Guest · 10 months ago

It isn't <http://ideone.com/nKQeuy>

^ | v · Reply · Share ›



Preethi · a year ago

Nice Information

^ | v · Reply · Share ›



c89 user · a year ago

please add declaration of variable i to the top of the main(), because in C89, you must declare all of your variables at the beginning of a scope block.

^ | v · Reply · Share ›



Prasanth Manoharan · a year ago

Could anyone give me the same code in php?

^ | v · Reply · Share ›



Milind · a year ago

nice work !!

^ | v · Reply · Share ›



Leo Jefferson Cardoso Marinho · a year ago

In this part of the code: return (0 != pCrawl && pCrawl->value);

0 != pCrawl is UNNECESSARY

This condition is ALWAYS TRUE!

1 ^ | v · Reply · Share ›



krsnabb → Leo Jefferson Cardoso Marinho · 9 months ago

"0 != pCrawl" is equivalent to checking "pCrawl != NULL" it is just a verification to check whether getnode() function was able to assign required memory has been allocated to store the leaf value. IF for some reason memory was unable to allocate the the leaf value will not be stored. there by even the key exist since leaf value (count) is not stored and the output will be "not present in trie"

Please correct me if i am wrong

^ | v · Reply · Share ›



AlienOnEarth → Leo Jefferson Cardoso Marinho · a year ago

Its not unnecessary. Usually we dont use root for traversal. We use another pointer like

its not unnecessary. Usually we dont use root for traversal. we use another pointer like pCrawl or current to traverse any data structure. For arrays we use i, for linked list we use current, for trees we use crawl or current.

^ | v • Reply • Share ›

Load more comments

 [Subscribe](#)

 [Add Disqus to your site](#)

 [Privacy](#)

DISQUS

Google™ Custom Search



-
- - [Interview Experiences](#)
 - [Advanced Data Structures](#)
 - [Dynamic Programming](#)
 - [Greedy Algorithms](#)
 - [Backtracking](#)
 - [Pattern Searching](#)
 - [Divide & Conquer](#)
 - [Mathematical Algorithms](#)
 - [Recursion](#)
 - [Geometric Algorithms](#)

-

• Popular Posts

- [All permutations of a given string](#)
- [Memory Layout of C Programs](#)
- [Understanding “extern” keyword in C](#)
- [Median of two sorted arrays](#)
- [Tree traversal without recursion and without stack!](#)
- [Structure Member Alignment, Padding and Data Packing](#)
- [Intersection point of two Linked Lists](#)
- [Lowest Common Ancestor in a BST](#)
- [Check if a binary tree is BST or not](#)
- [Sorted Linked List to Balanced BST](#)
- [Follow @GeeksforGeeks](#)

• Recent Comments

- [Mr. Lazy](#)

Does not work for most of the cases given in...

[Amazon Interview Experience | Set 184 \(Off Campus for SDE1\)](#) · [1 hour ago](#)

- [Mr. Lazy](#)

Round 3, Q1 Solution $O(n)$... Covers all most...

[Amazon Interview Experience | Set 184 \(Off Campus for SDE1\)](#) · [1 hour ago](#)

- [Youssef Yossry](#)

First, You need to know some terminologies in...

[Algorithms](#) · [1 hour ago](#)

- [Abhishek](#)

how about this: initialize first and second to...

[Find the largest pair sum in an unsorted array](#) · [1 hour ago](#)

- [jaspreet singh](#)

You are missing an "=" sign in the inner while...

[Next Greater Element](#) · [1 hour ago](#)

- [GeeksforGeeks](#)

Please take a closer look. It produces 7 only....

[Calculate \$7n/8\$ without using division and multiplication operators](#) · [1 hour ago](#)

•

@geeksforgeeks, [Some rights reserved](#) [Contact Us!](#) [About Us!](#)

Powered by [WordPress](#) & [MooTools](#), customized by geeksforgeeks