GeeksforGeeks
A computer science portal for geeks

IDE   Q&A   GeeksQuiz

Trie | (Insert and Search)

Trie is an efficient information re*trie*val 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.
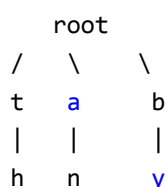
Every node of trie consists of multiple branches. Each branch represents a possible character of keys. We need to mark the last node of every key as leaf node. A trie node field *value* will be used to distinguish the node as leaf node (there are other uses of the *value* field). A simple structure to represent nodes of English alphabet can be as following,
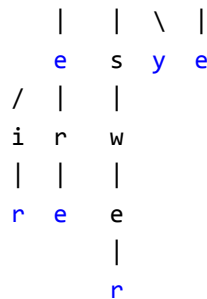
```
struct trie_node
{
    int value; /* Used to mark leaf nodes */
    trie_node_t *children[ALPHABET_SIZE];
};
```

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,

```
       root
      /   \    \
     t     a    b
     |     |    |
     h     n    y
```

```
              |   |  \  |
              e   s  y  e
            /  |     |
            i  r     w
            |  |     |
            r  e     e
                     |
                     r
```

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.

```c
#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;
        }
    }
```

```c
        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", "their"};
    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;
}
```

Run on IDE

— **Venki**. Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

107 Comments Category: Trees Tags: Advance Data Structures

About Venki

Software Engineer

View all posts by Venki →

Related Questions:

- Count BST subtrees that lie in given range
- Count BST nodes that lie in a given range
- Data Structure for a single resource reservations
- 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

Like ‹ 15   **Tweet** ‹ 3 g+1 ‹ 2

107 Comments   **GeeksforGeeks**   Priyanka Khire ⌄

♥ Recommend 4   ↗ Share   Sort by Newest ▾

Join the discussion…

Tarasha · 5 days ago
How can we free all the malloc-ed memory?
∧ | ∨ · Reply · Share ›

Krishna · 16 days ago
This is an awesome implementation of trie tree in C++ stl.... With the delete operation also explained...

C++ stl - http://theoryofprogramming.com...
C++ class - http://theoryofprogramming.com...
1 ∧ | ∨ · Reply · Share ›

Ayushi Srivastava · 18 days ago
what does this mean?? (0 != pCrawl && pCrawl->value)
∧ | ∨ · Reply · Share ›

  atul ➔ Ayushi Srivastava · 17 days ago
  It is checking whether last character of the string is present in the tree and it is marked as leaf node in the tree.
  1 ∧ | ∨ · Reply · Share ›

    Ayushi Srivastava ➔ atul · 17 days ago
    heyy what does 0 != pCrawl mean? thnx btw :)
    ∧ | ∨ · Reply · Share ›

      arpit ➔ Ayushi Srivastava · 15 days ago
      it is not necessary because we have already checked whether the leaf node is present or node in the loop...
      and it means if(pCrawl!=NULL)...
      1 ∧ | ∨ · Reply · Share ›

name · 2 months ago
can you give me java code for the same
∧ | ∨ · Reply · Share ›

  Pavan Yadiki ➔ name · 24 days ago
  https://gist.github.com/Panthe...
  ∧ | ∨ · Reply · Share ›

  Koustav Chatterjee ➔ name · 2 months ago
  http://ideone.com/LnkdaT

http://ideone.com/LnkdcT
∧ | ∨ · Reply · Share ›

Cracker · 2 months ago

C++ implementation of Trie:
http://algods-cracker.blogspot...
∧ | ∨ · Reply · Share ›

Siddhant · 2 months ago

why 2 structures are used?
∧ | ∨ · Reply · Share ›

Ajitesh Mandal · 4 months ago

another way of code

http://ideone.com/u3zc4Z
∧ | ∨ · Reply · Share ›

John → Ajitesh Mandal · 4 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 · 3 months ago
http://ideone.com/HcvHBw
∧ | ∨ · Reply · Share ›

Sheetal → John · 3 months 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[] strs = { "Prakash", "Venkat", "Ullman", "AnandRajaraman",
"JureLeskovec", "Prasanna", "Pandya" };

for (String str : strs) {
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");

```
trie.prefixFetch( p ),
}
```

see more

∧ | ∨ • Reply • Share ›

Anonymous · 5 months ago

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

∧ | ∨ • Reply • Share ›

huih ➜ Anonymous · 21 hours ago

Yes, it is very good, except lots of memory usage.

∧ | ∨ • Reply • Share ›

Abhishek · 6 months ago

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

∧ | ∨ • Reply • Share ›

Sun ➜ Abhishek · 6 months ago

Yes, You Can !

∧ | ∨ • Reply • Share ›

manuel velasquez ➜ Sun · 6 months ago

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

∧ | ∨ • Reply • Share ›

Hello_world · 7 months ago

what does value represent in trie_node DS ?

∧ | ∨ • Reply • Share ›

Anurag Singh ➜ Hello_world · 7 months ago

Few statements from above articlie
"A trie node field value will be used to distinguish the node as leaf node"
"int value; /* Used to mark leaf nodes */"

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

∧ | ∨ • Reply • Share ›

Tokala sai teja · 7 months ago

Java Implementation For Java Folks

http://ideone.com/cgDpzN

2 ∧ | ∨ • Reply • Share ›

Guest · 7 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 :)

ᐱ | ᐯ · Reply · Share ›

Jasdeep Singh ➔ Guest · 7 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 ᐱ | ᐯ · Reply · Share ›

Aditya Goel · 8 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 degbug the issue. Please if anybody found the issue, comment back
here..

ᐱ | ᐯ · Reply · Share ›

Klaus ➔ Aditya Goel · 8 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 ᐱ | ᐯ · Reply · Share ›

vishal ➔ Klaus · 7 months ago
traversing ?? in trie ....

ᐱ | ᐯ · Reply · Share ›

Aditya Goel → Klaus · 8 months ago

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

∧ | ∨ · Reply · Share ›

Ankit Aggarwal · 9 months ago

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

∧ | ∨ · Reply · Share ›

Klaus → Ankit Aggarwal · 8 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.

∧ | ∨ · Reply · Share ›

Guest → Klaus · 8 months ago

Also reduce ALPHABET_SIZE to 10 to save memory

∧ | ∨ · Reply · Share ›

Aditya Goel → Ankit Aggarwal · 8 months ago

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

∧ | ∨ · Reply · Share ›

Phoenix · 9 months ago

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

∧ | ∨ · Reply · Share ›

Srikar ED · 9 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....

∧ | ∨ · Reply · Share ›

Abhishek Kashyap Jha · 9 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?

∧ | ∨ · Reply · Share ›

neelabhsingh · 10 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<8;i++){
word=scan.next();
```

see more

∧ | ∨ · Reply · Share ›

vrg · 10 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];
};
```

∧ | ∨ · Reply · Share ›

Divino César → vrg · 10 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.

∧ | ∨ · Reply · Share ›

Saurabh · 10 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&lt;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);
}
}
}
```

∧ │ ∨ · Reply · Share ›

kaushik Lele · a year ago

A java code implementation can be found here
https://community.oracle.com/t...

∧ │ ∨ · Reply · Share ›

Umashankar · a year ago

Hi, Thanks for a nice piece of code. Any suggestions on using words which have
special characters like "can't" -: " ' " or '-'. They dont seem to be supported right
now.

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

∧ │ ∨ · Reply · Share ›

Anuj → Umashankar · a year ago

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

∧ │ ∨ · Reply · Share ›

kick · a year ago

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

∧ │ ∨ · Reply · Share ›

Ekta Goel → kick · a year 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..

⌃ | ⌄ • Reply • Share ›

MK · a year ago

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

⌃ | ⌄ • Reply • Share ›

Ekta Goel → MK · a year 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 contro 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..!!

⌃ | ⌄ • Reply • Share ›

Ekta Goel → Ekta Goel · a year ago

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

⌃ | ⌄ • Reply • Share ›

Guest · a year ago

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

⌃ | ⌄ • Reply • Share ›

Guest · a year ago

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

⌃ | ⌄ • Reply • Share ›

Load more comments