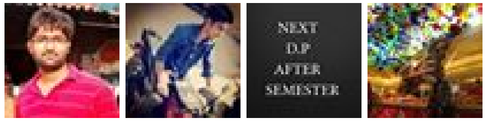
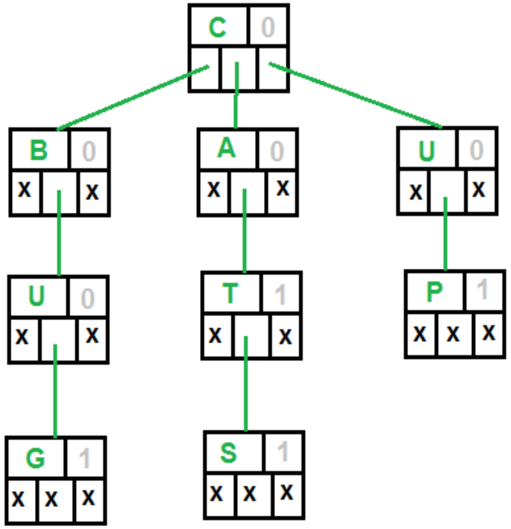
Ternary Search Tree

A ternary search tree is a special trie data structure where the child nodes of a standard trie are ordered as a binary search tree.

Representation of ternary search trees:
Unlike trie(standard) data structure where each node contains 26 pointers for its children, each node in a ternary search tree contains only 3 pointers:
1. The left pointer points to the node whose value is less than the value in the current node.
2. The equal pointer points to the node whose value is equal to the value in the current node.
3. The right pointer points to the node whose value is greater than the value in the current node.

Apart from above three pointers, each node has a field to indicate data(character in case of dictionary) and another field to mark end of a string.
So, more or less it is similar to BST which stores data based on some order. However, data in a ternary search tree is distributed over the nodes. e.g. It needs 4 nodes to store the word “Geek”.
Below figure shows how exactly the words in a ternary search tree are stored?





Following are the 5 fields in a node

1) The data (a character)

2) isEndOfString bit (0 or 1). It may be 1 for nonleaf nodes (the node with character T)

3) Left Pointer

4) Equal Pointer

5) Right Pointer

Ternary Search Tree for CAT, BUG, CATS, UP

One of the advantage of using ternary search trees over tries is that ternary search trees are a more space efficient (involve only three pointers per node as compared to 26 in standard tries). Further, ternary search trees can be used any time a hashtable would be used to store strings.

Tries are suitable when there is a proper distribution of words over the alphabets so that spaces are utilized most efficiently. Otherwise ternary search trees are better. Ternary search trees are efficient to use(in terms of space) when the strings to be stored share a common prefix.

Applications of ternary search trees:
1. Ternary search trees are efficient for queries like “Given a word, find the next word in dictionary(near-neighbor lookups)” or “Find all telephone numbers starting with 9342 or “typing few starting characters in a web browser displays all website names with this prefix”(Auto complete feature)”.

2. Used in spell checks: Ternary search trees can be used as a dictionary to store all the words. Once the word is typed in an editor, the word can be parallely searched in the ternary search tree to check for correct spelling.

Implementation:
Following is C implementation of ternary search tree. The operations implemented are, 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

insert and traversal.

```c
// C program to demonstrate Ternary Search Tree (TST) insert, travese
// and search operations
#include <stdio.h>
#include <stdlib.h>
#define MAX 50

// A node of ternary search tree
struct Node
{
    char data;

    // True if this character is last character of one of the words
    unsigned isEndOfString: 1;

    struct Node *left, *eq, *right;
};

// A utility function to create a new ternary search tree node
struct Node* newNode(char data)
{
    struct Node* temp = (struct Node*) malloc(sizeof( struct Node ));
    temp->data = data;
    temp->isEndOfString = 0;
    temp->left = temp->eq = temp->right = NULL;
    return temp;
}

// Function to insert a new word in a Ternary Search Tree
void insert(struct Node** root, char *word)
{
    // Base Case: Tree is empty
    if (!(*root))
        *root = newNode(*word);

    // If current character of word is smaller than root's character,
    // then insert this word in left subtree of root
    if ((*word) < (*root)->data)
        insert(&( (*root)->left ), word);

    // If current character of word is greate than root's character,
    // then insert this word in right subtree of root
    else if ((*word) > (*root)->data)
        insert(&( (*root)->right ), word);

    // If current character of word is same as root's character,
```

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

```
    else
    {
        if (*(word+1))
            insert(&( (*root)->eq ), word+1);

        // the last character of the word
        else
            (*root)->isEndOfString = 1;
    }
}

// A recursive function to traverse Ternary Search Tree
void traverseTSTUtil(struct Node* root, char* buffer, int depth)
{
    if (root)
    {
        // First traverse the left subtree
        traverseTSTUtil(root->left, buffer, depth);

        // Store the character of this node
        buffer[depth] = root->data;
        if (root->isEndOfString)
        {
            buffer[depth+1] = '\0';
            printf( "%s\n", buffer);
        }

        // Traverse the subtree using equal pointer (middle subtree)
        traverseTSTUtil(root->eq, buffer, depth + 1);

        // Finally Traverse the right subtree
        traverseTSTUtil(root->right, buffer, depth);
    }
}

// The main function to traverse a Ternary Search Tree.
// It mainly uses traverseTSTUtil()
void traverseTST(struct Node* root)
{
    char buffer[MAX];
    traverseTSTUtil(root, buffer, 0);
}

// Function to search a given word in TST
int searchTST(struct Node *root, char *word)
{
    if (!root)
```

695 Subscribe

Recent Comments

affiszerv Your example has two 4s on row 3, that's why it...

Backtracking | Set 7 (Sudoku) · 35 minutes ago

RVM Can someone please elaborate this Qs from above...

Flipkart Interview | Set 6 · 55 minutes ago

Vishal Gupta I talked about as an Interviewer in general,...

Software Engineering Lab, Samsung Interview | Set 2 · 55 minutes ago

@meya Working solution for question 2 of 4f2f round....

Amazon Interview | Set 53 (For SDE-1) · 1 hour ago

sandeep void rearrange(struct node *head) {...

Given a linked list, reverse alternate nodes and append at the end · 2 hours ago

```
        return 0;

    if (*word < (root)->data)
        return searchTST(root->left, word);

    else if (*word > (root)->data)
        return searchTST(root->right, word);

    else
    {
        if (*(word+1) == '\0')
            return root->isEndOfString;

        return searchTST(root->eq, word+1);
    }
}

// Driver program to test above functions
int main()
{
    struct Node *root = NULL;

    insert(&root, "cat");
    insert(&root, "cats");
    insert(&root, "up");
    insert(&root, "bug");

    printf("Following is traversal of ternary search tree\n");
    traverseTST(root);

    printf("\nFollowing are search results for cats, bu and cat respec
    searchTST(root, "cats")? printf("Found\n"): printf("Not Found\n");
    searchTST(root, "bu")?   printf("Found\n"): printf("Not Found\n");
    searchTST(root, "cat")?  printf("Found\n"): printf("Not Found\n");

    return 0;
}
```

Output:

```
Following is traversal of ternary search tree
bug
cat
cats
up
```

Neha I think that is what it should return as, in...

Find depth of the deepest odd level leaf node · 3 hours ago

```
Following are search results for cats, bu and cat respectively
Found
Not Found
Found
```

Time Complexity: The time complexity of the ternary search tree operations is similar to that of binary search tree. i.e. the insertion, deletion and search operations take time proportional to the height of the ternary search tree. The space is proportional to the length of the string to be stored.

Reference:
http://en.wikipedia.org/wiki/Ternary_search_tree

This article is compiled by **Aashish Barnwal** and reviewed by GeeksforGeeks team. Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

Related Tpoics:

- Print a Binary Tree in Vertical Order | Set 2 (Hashmap based Method)
- Print Right View of a Binary Tree
- Red-Black Tree | Set 3 (Delete)
- Construct a tree from Inorder and Level order traversals
- Print all nodes at distance k from a given node
- Print a Binary Tree in Vertical Order | Set 1
- Interval Tree
- Check if a given Binary Tree is height balanced like a Red-Black Tree

31 Tweet 3 1

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

33 Comments GeeksforGeeks

Sort by Newest ▾

Join the discussion…

1234 • a month ago

please explain me the traverse function.suppose string stored in cat,the a will would become act,please explaim m confused

^ | ˅ • Reply • Share ›

yzzzd • 2 months ago

could you share how to delete node in ternary search tree?

^ | ˅ • Reply • Share ›

cooldude • 4 months ago

Java implementation
http://ideone.com/NxFOT4

^ | ˅ • Reply • Share ›

Amit Bgl • 9 months ago

wow code :D

1 ^ | ˅ • Reply • Share ›

Aashish Barnwal • 9 months ago

TST searches for existence of only prefix strings, be its dictionary word or not. in the algorithm. Do not look for endOfString. However, it violates the principle TST is to store dictionary(meaningful) words. So, if you store "apple" in the TS work. Suffix trees work well to find existence of substring in main string. I woul easier and cleaner to implement.

1 ^ | ˅ • Reply • Share ›

Prity Bhudolia • 10 months ago

Hi, Can someone help. Can we use TST for partial search. If yes , then out of better? I mean to check whether a substring exists in main string or not.

^ | ˅ • Reply • Share ›

atiq • 10 months ago

I tried on the input
BOAT BOATS BOA BOSS BOOM BOOK MAN

and search BOA... output was "NOT FOUND".

I tried it differently you can see on my blog.
http://atiqwhiz.blogspot.in/se...

```
/* Paste your code here (You may delete these lines if not writing co
```

^ | ˅ • Reply • Share ›

atiq ➦ atiq • 10 months ago

sorry it's working .....great work

bateesh · 10 months ago

Can anybody explain me the complexity for inserting and searching in TST?

⌃ | ⌄ • Reply • Share ›

bateesh · 10 months ago

@GeeksforGeeks
In the Search function
When we have reached the end of string and the current node is not leaf node
current word as NULL.I think you can modify it to return
root->isEndOfString instead of 1.Here is the modified version

//when word finishes,then return whether current node is leaf or not
if (*(word+1) == '\0'
return root->isEndOfString;

return searchTST(root->eq, word+1);

⌃ | ⌄ • Reply • Share ›

GeeksforGeeks ➜ bateesh · 10 months ago

@bateesh: Thanks for your inputs. We have updated the code. Keep it

⌃ | ⌄ • Reply • Share ›

lizard · 10 months ago

In the insert function you are inserting in the equal subtree when the node valu
word....
but in the figure shown above you have shown
C
|
A
Please clarify....I have a bit confusion in this part.

| • Reply • Share ›

Aashish ➦ lizard • 10 months ago

Lets say we want to insert word CAT in the TST. We match the charac
deepen into the the equal pointer to insert the remaining suffix AT.

1 ⌃ | ⌄ • Reply • Share ›

Aashish Barnwal • 11 months ago

Please tell us which operation is not clear?

⌃ | ⌄ • Reply • Share ›

Maryam Syed • 11 months ago

can any one help me in finding operations of ternary search tree..

⌃ | ⌄ • Reply • Share ›

abhishek08aug • 11 months ago

Intelligent :D

⌃ | ⌄ • Reply • Share ›

code_ignitor • a year ago

When the word "CUP" is searched the search returns 1 though the word was
...

```
/* Paste your code here (You may delete these lines if not writing co
```

⌃ | ⌄ • Reply • Share ›

Aashish ➦ code_ignitor • a year ago

Please take a closer look at the search part. When the word "CUP" is s
Its because, when the character 'C' is referenced, it will deepen into th
"UP" will be searched. Observe the subTree with root node 'A'. Search

⌃ | ⌄ • Reply • Share ›

code_ignitor → Aashish • a year ago

yeah Got it :) Great Thank you :D

^ | v • Reply • Share ›

apsc • a year ago

Keep up the good work. Thanks for this. Please keep them coming especially

^ | v • Reply • Share ›

anandhakumar.P • a year ago

great explanation man . Thanks for it .

can u answer this question

"which data structure is the best for implementing telephone directory " --> wit

^ | v • Reply • Share ›

Aashish → anandhakumar.P • a year ago

If time is at its premium, TRIE is more efficient. The time complexity w telephone number to be searched.

1 ^ | v • Reply • Share ›

anonymous • a year ago

What happens when you add BBC after you add the above words ?

^ | v • Reply • Share ›

anonymous → anonymous • a year ago

Works perfectly... Carefully see the traversal

^ | v • Reply • Share ›

Aashish Barnwal • a year ago

Distortion is possible in ternary search tree. But, pattern matching is still possi algorithm. If a char is found(&#039c&#039 here), go into the node through equ

searched). Then if the next char(&#039o&#039 here) is greater(in ascii), go in
greater pointer(Do not increment the pattern to be searched). See here: https:

˄ | ˅ • Reply • Share ›

Rohit Jain • a year ago

Suppose I want to add another string say "Comma" to tree in example, this wil
&#0390&#039 to go to left of &#039u&#039, it will disturb structure of tree and

˄ | ˅ • Reply • Share ›

monika • a year ago

Nice article.
Can u please specify the applications where ternary search tree is better than

˄ | ˅ • Reply • Share ›

Aashish ➦ monika • a year ago

Ternary search tree is applicable to all those applications where TRIE
depends on the type of operation and the density of data(determines s
is distributed over the alphabets(can be digits or other characters). Th
If time is at its premium, go for TRIE.
If space is at its premium, go for Ternary search tree.

1 ˄ | ˅ • Reply • Share ›

Riya Chowdhury • a year ago

wow! really nice way to make things clear. good job.

˄ | ˅ • Reply • Share ›

Ravi • a year ago

Nice Approach. Are you posting Everything about Data Structures?

˄ | ˅ • Reply • Share ›

Aashish ➦ Ravi • a year ago

We are posting important aspects of data structures.

^ | ⌄ • Reply • Share ›

Rahul • a year ago

can you give the diagram, how the tree is created after every string added ?

^ | ⌄ • Reply • Share ›

Aashish ➦ Rahul • a year ago

I encourage you to draw the tree yourself on paper. Please note that th
which words are inserted in changed. e.g. Try to generate the tree with

^ | ⌄ • Reply • Share ›

✉ Subscribe   Ⓓ Add Disqus to your site