



TRIE datastructure explanation and simplified dictionary implementation in Java.

Visit us:

IN [DATASTRUCTURE](#) - ON 02:03:00 - [NO COMMENTS](#)

What is TRIE datastructure?

Trie is the data structure very similar to Binary Tree.

Trie datastructure stores the data in particular fashion, so that retrieval of

data became much faster and helps in performance.

The name "TRIE" is coined from the word *retrieve*.

What are TRIE data structure usage or applications?

1. Dictionary Suggestions OR Autocomplete dictionary

Retrieving data stored in Trie data structure is very fast, so it is most suited for application where retrieval are more frequently performed like Phone directory where contact searching operation is used frequently.

2. Searching Contact from Mobile Contact list OR Phone Directory

Auto suggestion of words while searching for anything in dictionary is very common.

If we search for word "tiny", then it auto suggest words starting with same characters like "tine", "tin", "tinny" etc.

Auto suggestion is very useful and Trie plays a nice role there, Lets see real time use.

If say, Person doesn't know the complete spelling of the some word but know few, then rest of words starting with few characters can be auto suggested using TRIE data structure.

How TRIE datastructure works?

Each word of a English language is made using 26 alphabets. Trie data structure uses this property as a base for storing words.

A word can be started (first character of a word) from either a, either b, either c or either z.

So there are total 26 possibilities, from which first character of a word will be started.

Second character can also be started from either a, either b, either c or either z.

So there are total 26 possibilities, from which second character of a word will be started.

Same for Third character and so on.

So for storing a word, we need to take an array(container) of size 26 and initially all the characters are blank as there are no words and it will look as shown below,

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z

Lets see how a word "hello" and "him" is stored,

1. "hello"

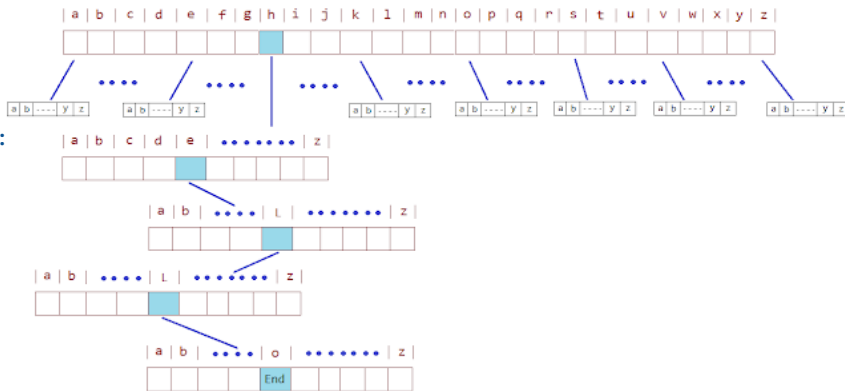
"hello" starts from "h", So we will mark the position "h" as filled, which represents the use of "h".



Second character is "e", So from "h", we will move to "e" and mark "e" in 2nd array as used.
After "e", 3rd character is "l", So mark the position "l" as used in respective array.

Repeat the above steps for all the characters of a word and Trie will look like below.

Visit us:



2. "him"

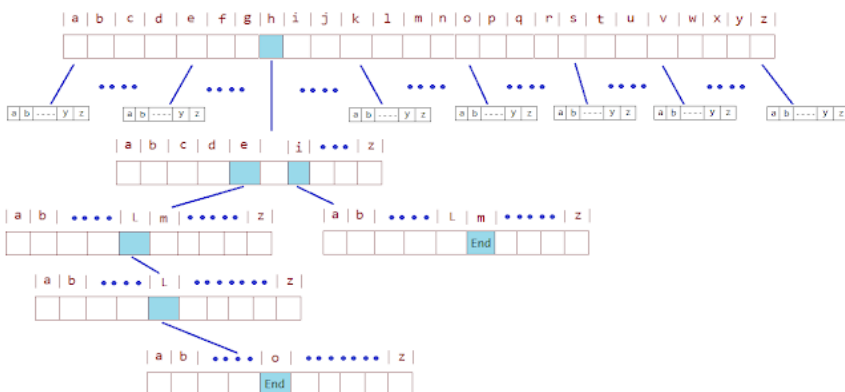
"him" starts from "h", So we will mark the position "h" as filled, but "h" is already filled by word "hello" which indicates that there exist some other word starting with "h".

Using the same position "h" as a reference, Now, for 2nd character "i",
We will move to "i" and mark position "i" in 2nd array as used.
This signifies, that there exist 2 words starting with "he" and "hi".

Now comes 3rd character "m" after "i", Now, we need to start new 26 array alphabet set again, which will hold the series starting from "hi".

So mark the position "m" as used in respective array.

After storing "hello" and "him", Trie will look like below.

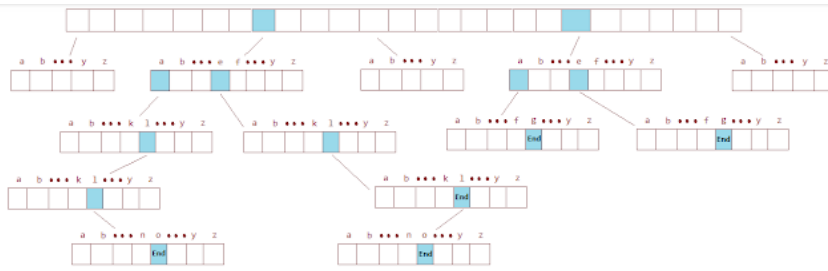


NOTE:

When we store the last character of a word, we need to mark that position as "END" to signify that this is the end of one word.

So that, while reading we will come to know that there exist some word which starts at "h" and ends at character "o" in case of "hello".

See figure below for more words.



Lets see java implementation of TRIE data structure.

TRIE Container class

Visit us:

```
1 package trie;
2
3 public class TrieContainer{
4     public boolean isEnd;
5     public TrieContainer[] series = new TrieContainer[26];
6 }
```

TRIE Operation class

Operation it supports are,

1. Insert word in TRIE datastructure,
2. Search word exist in TRIE datastructure,
3. Print all the words present in TRIE datastructure.

```
1 package trie;
2
3 public class Trie {
4
5     public static void main(String[] args) {
6         TrieContainer start = new TrieContainer();
7
8         storeWords(start, "hello");
9         storeWords(start, "hallo");
10        storeWords(start, "hell");
11        storeWords(start, "teg");
12        storeWords(start, "tag");
13
14        printWordStrings(start, "");
15
16        System.out.println("\n"+isWordPresent(start, "teg"));
17    }
18
19    private static void storeWords(TrieContainer start, String word){
20        for (int j = 0; j < word.length(); j++) {
21            char character = word.charAt(j);
22
23            //In series, check the position of character,
24            //if it is already filled then check the series of filled Trie object.
25            //if it is not filled then create new TrieContainer object and place it at con
26            //if it is end of the word, then mark isEnd = true or else false;
27            if(start.series[character-97]!=null){
28                if(word.length()-1 == j){ //if word is found till last character, then mark i
29                    start.series[character-97].isEnd = true;
30                }
31                start = start.series[character-97];
32            }else{
33                TrieContainer trie = new TrieContainer();
34                trie.isEnd = (word.length()-1 == j ? true:false);
35                start.series[character-97] = trie;
36                start = start.series[character-97];
37            }
38        }
39    }
40
41    private static boolean isWordPresent(TrieContainer start, String word){
42        boolean isFound = true;
43        for (int i = 0; i < word.length(); i++) {
44            char character = word.charAt(i);
45
46            //if at character position TrieContainer object is present then character is 1
47            //start looking for next character is word.
48            if(start.series[character-97]!=null){
49                if(word.length()-1 != i){
50                    start = start.series[character-97];
51                }else{
52                    if(!start.series[character-97].isEnd){
53                        isFound = false;
54                    }
55                }
56            }else{
57                isFound = false;
58                break;
59            }
60        }
61        return isFound;
62    }
63
64    private static void printWordStrings(TrieContainer start, String toPrint) {
65        if(start==null){
66            return;
```

[HOME](#)
[CONTACT](#)

Search here....

```

70 }
71
72 for (int i = 0; i < start.series.length; i++) {
73     TrieContainer t = start.series[i];
74     if (t != null) {
75         printWordStrings(t, toPrint + (char)(97+i));
76     }
77 }
78 }
79 }

```

You may also like to see

[Place N Queens on N×N chess board such that no two Queens attack each other.](#)
[0/1 Knapsack Problem solved using Iterative and Dynamic Programming](#)
[Visit us.](#)
[Stock Buy Sell to Maximize Profit.](#)
[Find Minimum length Unsorted Subarray, Sorting which makes the complete array sorted](#)
[Count trailing zeros in factorial of a number.](#)
[How to Detect loop in linked list.](#)
[Tower Of Hanoi Program in Java.](#)
[Print Matrix Diagonally OR Diagonal order of Matrix.](#)
[Transpose of Matrix Inplace](#)
[Implement Queue using One Stack in Java \(Recursive Implementation\)](#)
[SOAP Vs REST Web Service, when to use SOAP, When to use REST.](#)

Enjoy !!!!

If you find any issue in post or face any error while implementing, Please comment.

[Tweet](#)
[G+](#)
[Like 3](#)
[Share](#)
[Save](#)

RELATED POSTS :

How Hashmap works internally in Java...

How Hashcode and Equals works in Java...

How Hashmap works in Java

Sort Array using Insertion Sort

Array
 Input: 40 20 60 10 50 30
 Output: 10 20 30 40 50 60

Insertion Sort in Java

LABELS: [DATASTRUCTURE](#)

9 Comments [Java By Patel](#)

[Login](#)

[Recommend](#) 1 [Share](#)

[Sort by Best](#)

[Join the discussion](#)

Name

**Ajay Gu** • a year ago

Wow. Very easy to understand. I have one doubt:

I have 2 words: "hello" and "hell". While searching for the word "hello", when I reach "hell", it will be indicated as end. How will the algo knows there is a "o" after "l" ?

1 ^ | v • Reply • Share ›

**Jayesh Patel** Mod → Ajay Gu • a year ago

Hi Ajay,

Thanks.

You have a valid point, So for our Algorithm, to search "hello", It will first search for

'h', then 'e', then 'l', then 'l', at this point it will find end of string for word "hell" that is true, but that doesn't mean that our search is over at this place, there can exist other string which has same prefix as "hell",

so we need to further check is there any TrieContainer(links) coming out of last 'l' encountered, Yes there is one link that is to 'o' and also it has indication of end flag that is there exist one string "hello", So "hello" is the string which we were looking for.

If we where looking for some other string say "helloa", then our search should further continue and see is there any TrieContainer(links) coming out of last 'o' and so on.

Let me know if you still have any doubt, will be happy to help.

Regards,

Jayesh.

^ | v • Reply • Share ›

**ronald thomas** • 9 days ago

This is totally awesome.Although variety of article on this topic,this article contains some of the precious points which were never be read in other articles.

[Acronyms](#)

^ | v • Reply • Share ›

**Ankit Sharma** • a year ago

Clearly and effectively explained

^ | v • Reply • Share ›

**Jayesh Patel** Mod → Ankit Sharma • a year ago

Thanks Ankit.

^ | v • Reply • Share ›

**Kreedz Zhen** • 2 years ago

what if it is a upper case letter?

^ | v • Reply • Share ›

**Jayesh Patel** Mod → Kreedz Zhen • a year ago

Thanks Kreedz for writing me. If we need to separate lower case, upper case letter, numbers etc then we need to increase array size so that it will cover all the possible cases. Trie datastructure if not used efficiently waste lots of space.

Look at Suffix tree which is basically a compressed Trie.

^ | v • Reply • Share ›

**Sandeep Mishra** • 2 years ago

Nice explanation

^ | v • Reply • Share ›

**Jayesh Patel** Mod → Sandeep Mishra • 2 years ago

Thanks Sandeep.

^ | v • Reply • Share ›

ALSO ON JAVA BY PATEL

[wait notify notifyall example in Java Thread.](#)

3 comments • a year ago

**Saurabh Prakash** — I guess notifyAll() will wake
[Avatar](#) in all the waiting threads (i.e. ThreadB and[Why default constructor is not called during
Deserialization process in Java.](#)

2 comments • a year ago


**Jayesh Patel** — Thanks .litendra

[HOME](#)
[CONTACT](#)

Search here....


Method Overloading Interview Questions in Java

1 comment • a year ago

 **Monika Bindal** — Hi Jayesh,"You cannot widen and then box. Eg. test(int) cannot call test(Long) since to call test(Long) the compiler need to ...

Print Matrix Diagonally OR Diagonal Order of Matrix.

4 comments • a year ago

 **virat tyagi** — hiiii,Please suggest me why the above concept gives does not work.it gives me following exception. E:\java practice\Array ...

[Subscribe](#)
[Add Disqus to your site](#)
[Add Disqus](#)
[Add Disqus' Privacy Policy](#)
[Privacy Policy](#)
[Privacy](#)
[Newer Post](#)[Home](#)[Older Post](#)[Visit us:](#)

I'm Jayesh Patel, author of "JavaByPatel".

I'm not a professional blogger but when time permits, love to share in-depth solutions to common Interview questions asked.

Any questions/feedback, Please drop a mail at jayeshmaheshpatel@gmail.com



CATEGORIES

[Algorithm](#)
[Angular2](#)
[AngularJS](#)
[Array](#)
[Backtracking](#)
[Binary Search Tree](#)
[Binary Tree](#)
[Bit Manipulation](#)
[Database](#)
[Datastructure](#)
[Design Pattern](#)
[Graph](#)
[Interviews](#)
[Java](#)
[Java8](#)
[Javascript](#)
[Linked List](#)
[Matrix](#)
[Miscellaneous](#)
[Multithreading](#)
[OOPS](#)
[Puzzle](#)
[Quartz](#)
[Queue](#)
[Sorting](#)
[Spring](#)
[Stack](#)
[Stream](#)
[Strings](#)
[Typescript](#)
[Web Service](#)
[Windows](#)

POPULAR POSTS



Convert integers to roman numerals equivalent in Java

Converting Integers to Roman Numerals equivalent in Java ...



How time complexity of Hashmap get() and put() operation is O(1)? Is it O(1) in any condition?

How time complexity of Hashmap get() and put() operation...



What is Load factor and Rehashing in Hashmap?

What is Load factor and Rehashing in Hashmap? This is...



Find all subsets of a set (Power Set)

Print power set of any given set OR Print all subsets of...

HOMECONTACT

Search here....

questions.
ConcurrentHashMap Interview
Questions In Java. ...

RECENT COMMENTS

Java By Patel

661 likes

Like Page

Contact Us

Be the first of your friends to like this

Visit us:

Powered by Blogger.

BLOG ARCHIVE

› 2018 (1)

› 2017 (102)

› 2016 (63)

› 2015 (48)

› December (7)

› November (3)

› October (9)

› September (10)

› August (12)

› July (7)

› Write a program to print all permutations of a
 giv...

› What is Websocket? How Websocket works?
 Program on...

› TRIE datastructure explanation and simplified
 dict...

› How to enable or allow remote access to
 PostgreSQL...

› Convert integers to roman numerals
 equivalent in J...

› Find the element that appears once in a
 sorted arr...

› Using JQuery Datatable with AngularJS

Java By Patel

661 likes

Like Page

Contact Us

Be the first of your friends to like this

FOLLOW ME ON GOOGLE+

JavaByPatel

Follow

61

View my complete profile

© COPYRIGHT 2013 JAVABYPATEL - ALL RIGHTS RESERVED - POWERED BY BLOGGER.COM

^