GeeksforGeeks

A computer science portal for geeks

GeeksQuiz

Login

| Home | Algorithms | DS | GATE | Interview Corner | Q&A | C | C++ | Java | Books | Contribute | Ask a Q | About |

| Array | Bit Magic | C/C++ | Articles | GFacts | Linked List | MCQ | Misc | Output | String | Tree | Graph |

Skip List | Set 1 (Introduction)

Can we search in a sorted linked list in better than O(n) time?
The worst case search time for a sorted linked list is O(n) as we can only linearly traverse the list and cannot skip nodes while searching. For a Balanced Binary Search Tree, we skip almost half of the nodes after one comparison with root. For a sorted array, we have random access and we can apply Binary Search on arrays.

Can we augment sorted linked lists to make the search faster? The answer is Skip List. The idea is simple, we create multiple layers so that we can skip some nodes. See the following example list with 16 nodes and two layers. The upper layer works as an "express lane" which connects only main outer stations, and the lower layer works as a "normal lane" which connects every station. Suppose we want to search for 50, we start from first node of "express lane" and keep moving on "express lane" till we find a node whose next is greater than 50. Once we find such a node (30 is the node in following example) on "express lane", we move to "normal lane" using pointer from this node, and linearly search for 50 on "normal lane". In following example, we start from 30 on "normal lane" and with linear search, we find 50.
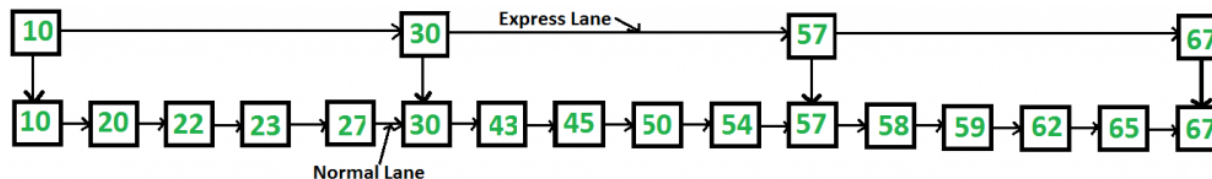
What is the time complexity with two layers? The worst case time complexity is number of nodes on "express lane" plus number of nodes in a segment (A segment is number of "normal lane" nodes between two "express lane" nodes) of "normal lane". So if we have n nodes on "normal lane", $\sqrt{n}$ nodes on "express lane" and we equally divide the "normal lane", then there
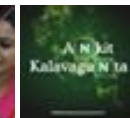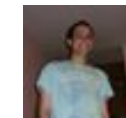
GeeksforGeeks

53,527 people like GeeksforGeeks.

will be $\sqrt{n}$ nodes in every segment of "normal lane" . $\sqrt{n}$ is actually optimal division with two layers. With this arrangement, the number of nodes traversed for a search will be $O(\sqrt{n})$. Therefore, with $O(\sqrt{n})$ extra space, we are able to reduce the time complexity to $O(\sqrt{n})$.

Can we do better?
The time complexity of skip lists can be reduced further by adding more layers. In fact, the time complexity of search, insert and delete can become O(Logn) in average case. We will soon be publishing more posts on Skip Lists.

References
MIT Video Lecture on Skip Lists
http://en.wikipedia.org/wiki/Skip_list

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

Related Tpoics:

- Given a linked list, reverse alternate nodes and append at the end
- Pairwise swap elements of a given linked list by changing links

Popular Posts

- Self Organizing List | Set 1 (Introduction)
- Merge a linked list into another linked list at alternate positions
- QuickSort on Singly Linked List
- Delete N nodes after M nodes of a linked list
- Design a stack with operations on middle element
- Swap Kth node from beginning with Kth node from end in a Linked List

[f] ⟨41 🐦 **Tweet** ⟨3 ⟨1

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

20 Comments　　　**GeeksforGeeks**

Sort by Newest ▾

Join the discussion…

Vinay Singh · 3 months ago

nice stuff...good 2 learn a new datastructure.
∧ | ∨ · Reply · Share ›

nigam · 3 months ago

next post??
when?
where?
1 ∧ | ∨ · Reply · Share ›

Vsprabhakaran Prabhu · 4 months ago

This is superb stuff. Though it resembles a bit of indexing technique, who wou
layers of linked list. :)
∧ | ∨ · Reply · Share ›

tausif akram · 5 months ago

tausi akram · 5 months ago

this is simply awesome !!

∧ | ∨ · Reply · Share ›

tripathi · 7 months ago

http://www.dreamincode.net/for...

∧ | ∨ · Reply · Share ›

nishant08 · 10 months ago

awesome

∧ | ∨ · Reply · Share ›

ultimate_coder · 11 months ago

> @geeksforgeeks Could u post c/c++ code **or** algo **for** insertion/deletion
> It will be highly appreciated ! !

∧ | ∨ · Reply · Share ›

Manisha Barnwal · 11 months ago

You will need middle at each level of the search,where the final level of middle

∧ | ∨ · Reply · Share ›

Kushagra Singhal · a year ago

Why cant, we augment the list with middle pointer then the worst case wud be

∧ | ∨ · Reply · Share ›

abhishek08aug · a year ago

Intelligent :D

∧ | ∨ · Reply · Share ›

Recent Comments

Abhi You live US or India?

Google (Mountain View) interview · 44 minutes ago

Aman Hi, Why arent we checking for conditions...

Write a C program to Delete a Tree. · 1 hour ago

kzs please provide solution for the problem...

Backtracking | Set 2 (Rat in a Maze) · 1 hour ago

Sanjay Agarwal bool tree::Root_to_leaf_path_given_sum(tree...

Root to leaf path sum equal to a given number · 1 hour ago

GOPI GOPINATH @admin Highlight this sentence "We can easily...

Count trailing zeroes in factorial of a number · 1 hour ago

newCoder3006 If the array contains negative numbers also. We...

Find subarray with given sum · 2 hours ago

► Linked List

► C++ Source Code

► Tutorial C++

CODER_1 · a year ago

gr8 and nice tutorial.

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

∧ | ∨ · Reply · Share ›

gh05t · a year ago

Extremely similar to B+ tree..if you relax a few rules followed by a B+ tree it be
please point out a fundamental difference?

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

∧ | ∨ · Reply · Share ›

cooldude · a year ago

```java
[sourcecode language="java"]
class Node
{
int data;
Node next;
Node(int data,Node next)
{
this.data=data;
this.next=next;
}
}
class SpecialNode extends Node
{
SpecialNode right;
SpecialNode(int data,Node next,SpecialNode right)
{
```

```
super(data,next);
this.right=right;
```

∧ | ∨ · Reply · Share ›

bitlasrinivas · a year ago

fine.

∧ | ∨ · Reply · Share ›

ibnipun10 · a year ago

Just wondering how is it different from a BST?
Also there is a requirement for linked list to be sorted

∧ | ∨ · Reply · Share ›

aayushkumar · a year ago

want awesomee tutorial for B,B+ tree like this ....:)

∧ | ∨ · Reply · Share ›

Barry Fruitman · a year ago

I like this a lot. I&#039ve been looking for a list data structure with the insert/de
lookup speed of a binary search.

1 ∧ | ∨ · Reply · Share ›

Uddhav Arote · a year ago

How would this be different from the B+ trees which is more or less similar to
trees, the siblings from different parents are connected via a link ..

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

∧ | ∨ · Reply · Share ›

Uddhav ➤ Uddhav Arote · a year ago

ok understood .. nice stuff ! :)

⌃ | ⌄ · Reply · Share ›

Niks · a year ago

Awesome ....Please provide tutorials on B-trees,AB trees too

⌃ | ⌄ · Reply · Share ›

✉ Subscribe   ⓓ Add Disqus to your site