

Time Complexity Linked Lists Computational Complexity Theory Data Structures

Algorithms +1

What is the time complexity of inserting a node into a linked list?

Answer Follow · 3 Request

Ad by DataCamp

How can I become a data scientist?

After being dubbed “sexiest job of the 21st Century” by Harvard Business Review, data science has stirred the interest of the general public. Many people are intrigued by the job...(Continue reading in feed)

4 Answers

**Tony Flury**, Software developer since 1988

Answered July 6, 2019



Strictly speaking an insertion is simply $O(1)$.

The other answers mostly correctly state that the complexity is $O(n)$ if you need to search for the position in which to insert the new node; but in most case a linked list is never used in a situation where a search is necessary.

A good use case for a linked list is in a FIFO or LIFO queue - ie where you are only ever inserting and deleting at the start or end of the list (and assuming you are keeping track of the last item in the list) - or where you are keep an arbitrary collection of data where you don't care about the order the data is stored in, you never need to do a contains test, and when you access the data you will always only want the first one / last one or all of them.

5.7K views · View Upvoters



You upvoted this



9



Add a comment...

Add Comment

Sponsored by Whole Tomato Software

Give Visual Studio a UI boost with Visual Assist.

Download free trial!



Free Trial

**Max Zelich**

Answered December 10, 2018



Since as far as I know - the insertion and search and delete a specific unit processes within linked lists - which all contain a search process in the core of their algorithm - are all go gradually from the first node to the second node, and then from the second node to the third node and so on... - the time complexity of all of the above is $O(n)$.

In best case it's $O(1)$ - it could be even after the first checking of the first node.

The average case in these things is the same as the worst case which is $\Theta(n)$.

In the worst of the worst case - you need to go along from the first node till the nth (last) node before inserting/deleting/finding the specific node you are asking to deal with.

9.9K views · View Upvoters



2



1



1 comment from Tony Flury

Related Spaces (More Answers Below)

Related Questions

How is inserting into a linked list $O(1)$? I don't understand how this is the worst case.

What is the time complexity of appending a node in a singly linked list?

What is the time complexity of inserting a node in a doubly linked list?

What is the time complexity of appending a node in a single linked list (no extra pointers...)

Are the time complexities of doubly and singly linked lists the same?

What will be the time complexity for the deletion in sorted Linked list?

Ask Question

Quora



Search Quora



Add Question

Python & Machine Learning

Suggestions to learn Python for machine learning and data...

[Follow](#) 19.6K
Data Engineering Minds

Data Engineering Techniques, Distributed Systems & Big Data...

[Follow](#) 4.9K
Learn C#

A place to learn about the versatile C# programming language and .NET framework.

[Follow](#) 16
Data

The best c...

[Follow](#)
[View More Spaces](#) >


Inserting at beginning ----> $O(1)$

Inserting in middle---> $O(1)$ with iterator $O(n)$

Inserting at end ----> $O(1)$ or $O(n)$

The time complexity for the Inserting at the end depends if you have the location of the last node, if you do, it would be $O(1)$ other wise you will have to search through the linked list and the time complexity would jump to $O(n)$.

2.9K views · View Upvoters



[Add Comment](#)

Sponsored by ShipEngine

**Have you discovered the new standard in multi-carrier shipping?**

ShipEngine is a powerful shipping API, built for developers to power fast-growing ecommerce businesses.

[Learn More](#)
Related Questions
[More Answers Below](#)

[How is inserting into a linked list \$O\(1\)\$? I don't understand how this is the worst case.](#)

[What is the time complexity of appending a node in a singly linked list?](#)

[What is the time complexity of inserting a node in a doubly linked list?](#)



Answered December 10, 2018

Simply inserting a node is $O(1)$ for 2 operations. The pointer to next of the previous node is set to point to this node. The next of this current node is set to the next of the previous node. You can work out insertions at the head of the linked list. Insertions at the tail are $O(1)$. This of course does not include the $O(n)$ traversal of the linked list.

2.4K views · View Upvoters

3 comments from Tony Flury and more

Related Questions

[How is inserting into a linked list \$O\(1\)\$? I don't understand how this is the worst case.](#)

[What is the time complexity of appending a node in a singly linked list?](#)

[What is the time complexity of inserting a node in a doubly linked list?](#)

[What is the time complexity of appending a node in a single linked list \(no extra pointers are used\)?](#)

[Are the time complexities of doubly and singly linked lists the same?](#)

Quora

Add Question

What is the complexity of inserting a node at the beginning of doubly linked list?

When would you use a linked list?

What is linked list?

What is the time complexity to count the number of elements of a linked list?

What is the time complexity of a linked list?

Whose time complexity is better, an array or a linked list?

What is the time complexity of a linked list insertion?

Do programmers use a linked list in the real world given the time complexity relative to arrays?

What is the difference between node* and node**? When should I use node* and when should I use node** to insert node in a linked list?