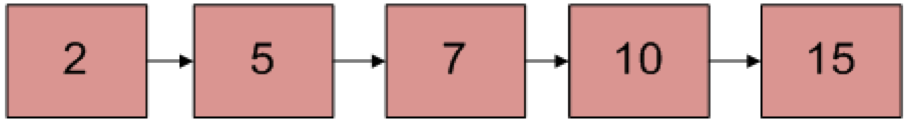
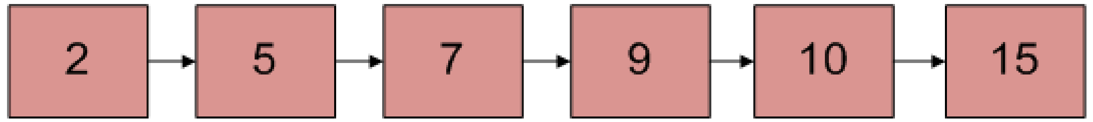
Given a linked list which is sorted, how will you insert in sorted way

Algorithm:
Let input linked list is sorted in increasing order.

```
1) If Linked list is empty then make the node as head and return it.
2) If value of the node to be inserted is smaller than value of head node
    then insert the node at start and make it head.
3) In a loop, find the appropriate node after which the input node (let 9) is
    to be inserted. To find the appropriate node start from head, keep moving
    until you reach a node GN (10 in the below diagram) who's value is
    greater than the input node. The node just before GN is the appropriate
    node (7).
4) Insert the node (9) after the appropriate node (7) found in step 3.
```

Initial Linked List


Linked List after insertion of 9


Implementation:

Google™ Custom Search

GeeksforGeeks
53,528 people like GeeksforGeeks.

Interview Experiences
Advanced Data Structures
Dynamic Programming
Greedy Algorithms
Backtracking
Pattern Searching
Divide & Conquer
Mathematical Algorithms
Recursion
Geometric Algorithms

```c
/* Program to insert in a sorted list */
#include<stdio.h>
#include<stdlib.h>

/* Link list node */
struct node
{
    int data;
    struct node* next;
};

/* function to insert a new_node in a list. Note that this
  function expects a pointer to head_ref as this can modify the
  head of the input linked list (similar to push())*/
void sortedInsert(struct node** head_ref, struct node* new_node)
{
    struct node* current;
    /* Special case for the head end */
    if (*head_ref == NULL || (*head_ref)->data >= new_node->data)
    {
        new_node->next = *head_ref;
        *head_ref = new_node;
    }
    else
    {
        /* Locate the node before the point of insertion */
        current = *head_ref;
        while (current->next!=NULL &&
               current->next->data < new_node->data)
        {
            current = current->next;
        }
        new_node->next = current->next;
        current->next = new_node;
    }
}

/* BELOW FUNCTIONS ARE JUST UTILITY TO TEST sortedInsert */

/* A utility function to create a new node */
struct node *newNode(int new_data)
{
    /* allocate node */
    struct node* new_node =
        (struct node*) malloc(sizeof(struct node));

    /* put in the data  */
```

Popular Posts

- All permutations of a given string
- Memory Layout of C Programs
- Understanding “extern” keyword in C
- Median of two sorted arrays
- 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

```
    new_node->data  = new_data;
    new_node->next =  NULL;

    return new_node;
}

/* Function to print linked list */
void printList(struct node *head)
{
    struct node *temp = head;
    while(temp != NULL)
    {
        printf("%d  ", temp->data);
        temp = temp->next;
    }
}

/* Drier program to test count function*/
int main()
{
    /* Start with the empty list */
    struct node* head = NULL;
    struct node *new_node = newNode(5);
    sortedInsert(&head, new_node);
    new_node = newNode(10);
    sortedInsert(&head, new_node);
    new_node = newNode(7);
    sortedInsert(&head, new_node);
    new_node = newNode(3);
    sortedInsert(&head, new_node);
    new_node = newNode(1);
    sortedInsert(&head, new_node);
    new_node = newNode(9);
    sortedInsert(&head, new_node);
    printf("\n Created Linked List\n");
    printList(head);

    getchar();
    return 0;
}
```

Shorter Implementation using double pointers

Thanks to Murat M Ozturk for providing this solution. Please see Murat M Ozturk's comment below for complete function. The code uses double pointer to keep track of the next pointer of the previous node (after which new node is being inserted).

705

Subscribe

Note that below line in code changes *current* to have address of next pointer in a node.

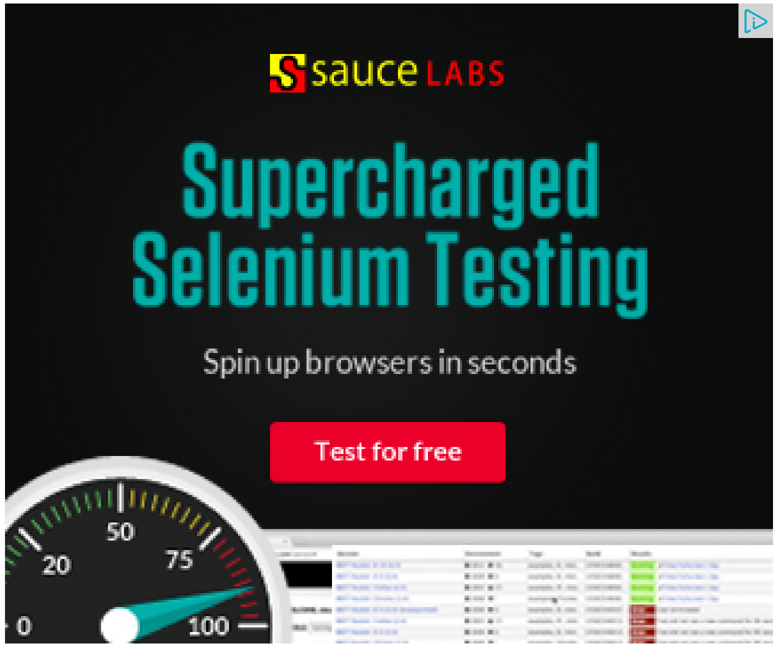
```
current = &((*current)->next);
```

Also, note below comments.

```
new_node->next = *current; /* Copies the value-at-address current to n
*current = new_node;  /* Fix next pointer of the node (using it's addr
```

Time Complexity: O(n)

References:
http://cslibrary.stanford.edu/105/LinkedListProblems.pdf



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
- Self Organizing List | Set 1 (Introduction)
- Merge a linked list into another linked list at alternate positions

Recent Comments

Abhi You live US or India?

Google (Mountain View) interview · 49 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

- 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

11 Tweet 0 0

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

30 Comments GeeksforGeeks

Sort by Newest

Join the discussion…

ravi m • 17 days ago

```
#include<stdio.h>

#define s sizeof(int)

void main()

{

int i = -1;

// printf("%d", sizeof(short int));

if( i < s)

printf("t");

else

printf("f");
```

}

// answer is f how it is possible to get out put f, condition is if(-1 < sizeof(int)) it
me solution...

Reply • Share ›

mareen ravi m • 16 days ago

"f" gets printed because "sizeof()" returns unsigned value when it is co
of "i" is high and of "s" is low ,so "i" is greater

Reply • Share ›

mareen • 17 days ago

*current = new_node;
i do not get this line . how can it make the node's (before current) next point to

Reply • Share ›

AMIT JAMBOTKAR • 24 days ago

IMPLEMENTED IN JAVA GENERIC WAY:

public class LinkedList<e extends="" number=""> implements Cloneable{

Node<e> head = null;

//Adding at the End

class Node<t extends="" number=""> {

T value;

Node<t> nextReference;

public Node(T value) {

this.value = value;

this.nextReference = null;

}

see more

^ | v • Reply • Share ›

Dark Protocol • 2 months ago

For Larger List size (n>10000000), Skip list is more appropriate

^ | v • Reply • Share ›

Himanshu Dagar • 3 months ago

can refer to below code

http://ideone.com/R5rl9g

1 ^ | v • Reply • Share ›

Daniel YIn • 7 months ago

```
node * sortedInsert(node * n, int d){
if (n == NULL || n->data >d) return new node(d,n);
else if (n->data == d) return n;
else {
n->next = sortedInsert(n->next,d);
return n;
}
}
```

^ | v • Reply • Share ›

mahi2 • 8 months ago

This problem can be solved if we maintain 2 pointers...and move one pointer (
loop..and compare the value of the node to be inserted with the data value of t
(node)> data(tmp1) and data(node) < data(tmp2)..

insert the node at that point!

Reply • Share ›

Xristos Mpalis • 9 months ago

I want this code in java, please.

Reply • Share ›

AMIT JAMBOTKAR → Xristos Mpalis • 24 days ago

```
public class LinkedList<e extends="" number=""> implements Cloneal

Node<e> head = null;

//Adding at the End

class Node<t extends="" number=""> {

T value;

Node<t> nextReference;

public Node(T value) {

this.value = value;

this.nextReference = null;

}

public Node(T value, Node<t> ref) {
```

see more

Reply • Share ›

nitin • a year ago

#include

```
#include
struct node
{
int data;
struct node *link;
};
void insert1(struct node **p,int data)
{
struct node *temp,*t,*s;
temp=(struct node *)malloc(sizeof(struct node));
temp->data=data;
temp->link=NULL;
if((*p)==NULL)
{
*p=temp;
}
else
```

see more

Reply • Share ›

Chuantao Zang • a year ago

This does not work if the node is the largest, you should add sereral lines mor
/* Locate the node before the point of insertion */.

current = *head_ref;.

while (current->next!=NULL && current->next->data < new_node->data).

{.

current = current->next;.

}.

if(current->next!=NULL).

current0>next=new_node; //add to tail.
else.

{.

new_node->next = current->next;.

current->next = new_node;.

}.

• Reply • Share ›

Amit Kumar • a year ago

thats is what we all do...
if you are asking for insertion before a node then For that you can keep track o
linked_list_pointer->next->value to compare with..

• Reply • Share ›

Pallavee Gogoi • a year ago

insert into linked list after a given node.

• Reply • Share ›

Hina Jain • a year ago

@Murat M- I think your solutionn wont work when the node to be inserted turns
few checks for this condition....comments would be welcomed...

void sortedInsert(struct node** head_ref, struct node* new_node).
{
if (head_ref == NULL).
{.

return;.
}.

/* Locate the node before the point of insertion or if last node is reached we st

struct node** current = head_ref;.

while ((*current)->next!=NULL && (*current)->data < data).

{.

current = &((*current)->next);.

see more

Reply · Share ›

ff · 4 years ago
hi ... please i want sorted with with only int

this funnction: sortedinsert(int)

Reply · Share ›

Shekhu ff · 4 years ago
can you please explain your requirement with an example?

Reply · Share ›

GeeksforGeeks · 4 years ago
@Murat M Ozturk: Thanks for the short and nice solution. We have added the

3 · Reply · Share ›

rikitic GeeksforGeeks · 11 months ago
it can be done in less time by using binary search on linked list....corre

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

^ | v • Reply • Share ›

GeeksforGeeks → rikitic • 11 months ago
Binary Search can not be applied on Linked Lists. That is why v
(http://www.geeksforgeeks.org/s...

^ | v • Reply • Share ›

rikitic → GeeksforGeeks • 11 months ago
its almost binary search

```
/* Paste your code here (You may delete these li
```

^ | v • Reply • Share ›

Murat M Ozturk • 4 years ago
Here is a simplified version of the sortedInsert() method:

```
 void sortedInsert(struct node** head_ref, struct node* new_node)
{
  if (head_ref == NULL)
  {
          return;
  }


        /* Locate the node before the point of insertion */
        struct node** current = head_ref;
        while (*current !=NULL && (*current)->data < data)
        {
          current = &((*current)->next);
        }
        new_node->next = *current;
        *current = new_node;
}
```

1 ^ | ⌄ • Reply • Share ›

hina ➦ Murat M Ozturk • 10 months ago

I think this wont work when the node to be inserted turns out to be the l
this condition....Correct me if I am wrong...

^ | ⌄ • Reply • Share ›

hina ➦ hina • 10 months ago

Code with all the checks:
Correct me if i am wrong

void sortedInsert(struct node** head_ref, struct node* new_noc
{
/*if LL is empty */
if (head_ref == NULL)
{
*head_ref = new_node;
}

/* Locate the node before the point of insertion or if last node is
struct node** current = head_ref;

//if new node is to be inserted at first position
if((*current)->data > new_node ->data)
{
new_node -> next = *current;
*head_ref = new_node;

see more

Reply • Share ›

olra Murat M Ozturk • 2 years ago

```
 /*
checking: if (head_ref == NULL) is included in while loop
so the code is :
*/
void sortedInsert(struct node** head_ref, struct node* new_node
{
    /* Locate the node before the point of insertion */
    struct node** current = head_ref;
    while (*current !=NULL && (*current)->data < data)
    {
      current = &((*current)->next);
    }

    new_node->next = *current;
    *current = new_node;
}
```

Reply • Share ›

Viky olra • a year ago

The second method of double pointer doesn't work for all cases

```
/* Paste your code here (You may delete these lines if r
```

Reply • Share ›

GeeksforGeeks Viky • a year ago

^ | ⌄ • Reply • Share ›

Viky ➦ GeeksforGeeks • a year ago

If the list is empty, we should make head as the new no
NULL.

Also, Adding element to the end of the list doesn't work

> ```
> /* Paste your code here (You may delete these lir
> ```

^ | ⌄ • Reply • Share ›

Bunty ➦ Viky • 2 months ago

#include<stdio.h>

#include<conio.h>

struct node

{

int data;

struct node *next;

};

void printList(struct node *n)

{

while(n!=NULL)

see more

1 ⌃ | ⌄ • Reply • Share ›

bunty ➦ Bunty • 2 months ago

neglect the </conio.h></stdio.h> at the end

⌃ | ⌄ • Reply • Share ›

Subscribe   Add Disqus to your site