Day's Goals

Thursday, February 1, 2018 11:28 AM

linkedList2.cpp

Recap: - construct a singly linked list
- traverse the LL

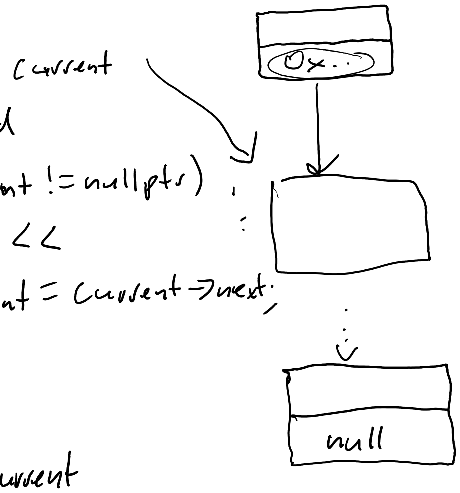
LL: - search
- insert
- delete

LL Construction

- use 3 struct pointers (Node struct)
⇒ (head, tmp, current)
- begin allocating space for each node
- set 'head' pointing to 1st node
- use 'current' point to the most recently created node (previous node)
- use 'tmp' to create now nodes
  ↳ always intialize the tmp node pointing to null
- loop through until have desired no. of nodes

Traverse:

```
current=head
 while(current != nullptr)
  {  cout <<
     current = current->next;
  }
Node * current
```

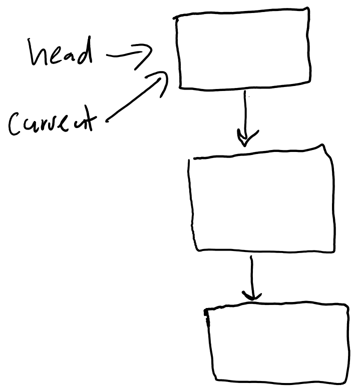


nullptr NULL

Search:

e.g. Find a node that contains "tres" and return its address.

If not found, return NULL.   target = "tres"



```
loop
while (current != NULL)
{
  if current->item = target
       return current
}

return NULL
```

Inserting a new Node

3 scenarios: A) At the beginning

B) After a given node

C) Append at the end

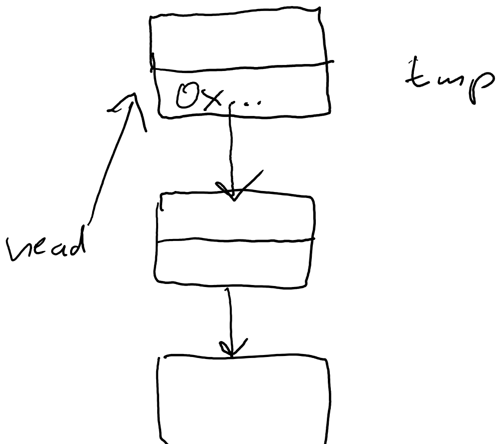
A) At the beginning

1) Create new Node

2) Link new node w/ old head

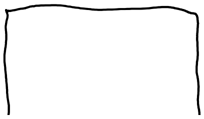
3) Make head point to new node

4) Set tmp to NULL



B) After a given node

(given previous Node addr)

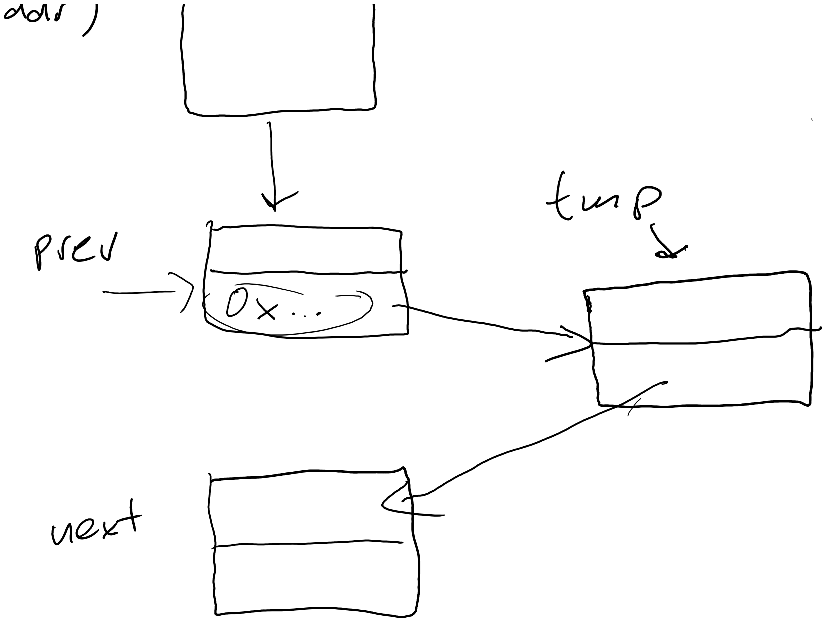


(given previous Node addr)

1) Create new node

2) Link new node w/ next node

3) Link prev w/ new Node


prev
0x...
tmp
next