Lecture 8: Linked lists

Monday, May 1, 2017 4:23 PM

LINKED LISTS

Problem Solving with Computers-I

https://ucsb-cs24-sp17.github.io/

Announcements

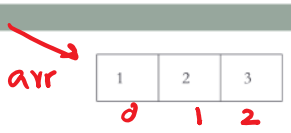
- PA3 and PA4 released
- PA3 is due on 5/8
- PA4 is due on 5/15 (a week after that)

int arr[3] = { 1, 2, 3 };

3

Linked Lists

The Drawing Of List {1, 2, 3}

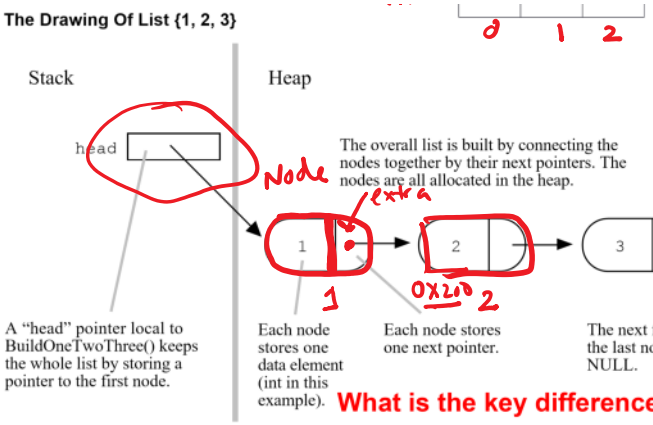


Stack | Heap

Array List

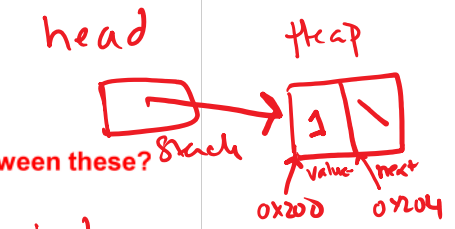
struct Node{
int value;
Node * next;

The Drawing Of List {1, 2, 3}



What is the key difference between these?

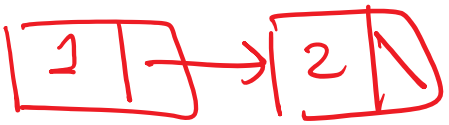
Linked List

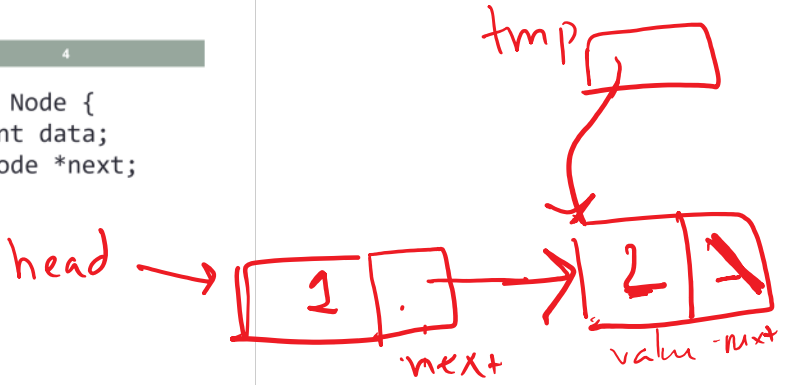


Create a two node list

- Define an empty list
- Add a node to the list with data = 10

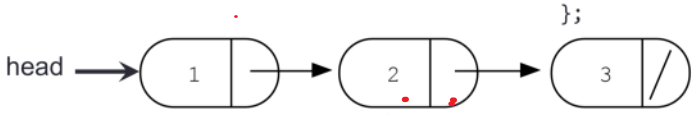
```
struct Node {
     int data;
     Node *next;
};
```





Accessing elements of a list

```
struct Node {
    int data;
    Node *next;
};
```



Assume the linked list has already been created, what do the following expressions evaluate to?

1. head->data
2. head->next->data
3. head->next->next->data
4. head->next->next->next->data

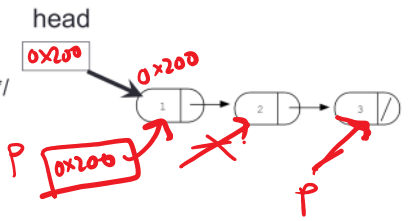
A. 1
B. 2
C. 3
D. NULL
E. Run time error

```
Node *tmp = head->next->next->next;
int arr[3]={1,2,3}; tmp->data;
for (int i=0; i<3; i++)
    cout<< arr[i];
```

arr: 0x200 | 1 | 2 | 3

Iterating through the list

```
int lengthOfList(Node * head) {
  /* Find the number of elements in the list */
    int count = 0;
    Node *p = head;
    while ( p ) {
        p = p->next; //Move p to the next node
        count ++;
    }
```

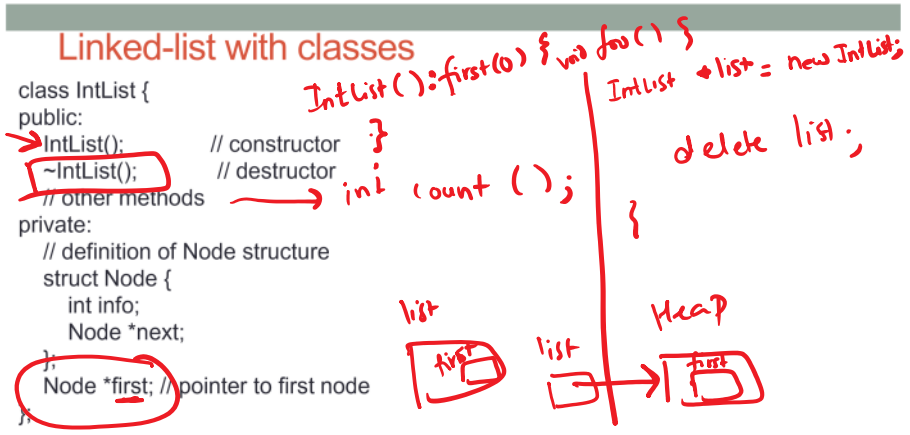



```
Node *p;
for( p = head; p != 0; p = p->next ) {
    count ++;
}
```

p: 0

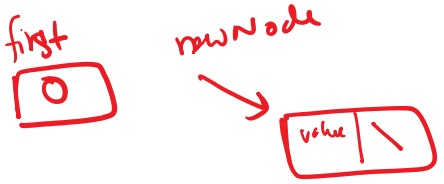
Linked-list with classes

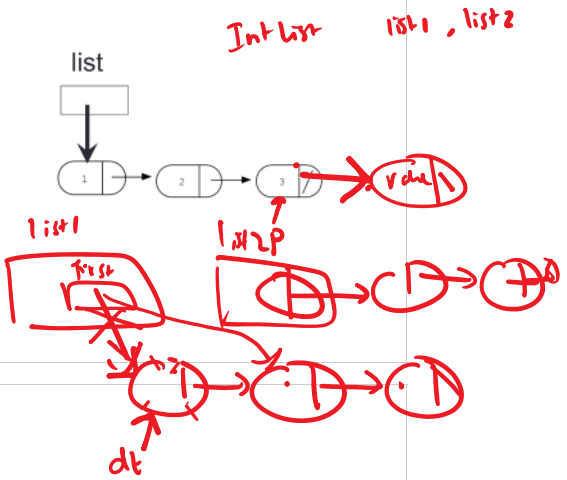
```
class IntList {
public:
   IntList();          // constructor
   ~IntList();         // destructor
   // other methods
private:
   // definition of Node structure
   struct Node {
      int info;
      Node *next;
   };
   Node *first; // pointer to first node
};
```



Deleting the list

```
Node* freeLinkedList(Node * list) {
   /* Free all the memory that was created on the heap*/
}
```





Next time

- More linked list with classes

Next time

- More linked list with classes