Lecture 14: Recursion

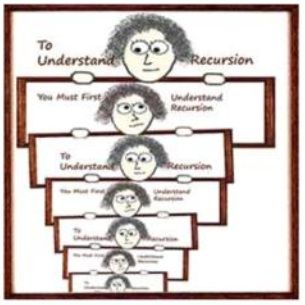
Monday, May 22, 2017 10:14 AM

RECURSION

it's like magic!

Problem Solving with Computers-I

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



Monday, May 22, 2017 10:14 AM

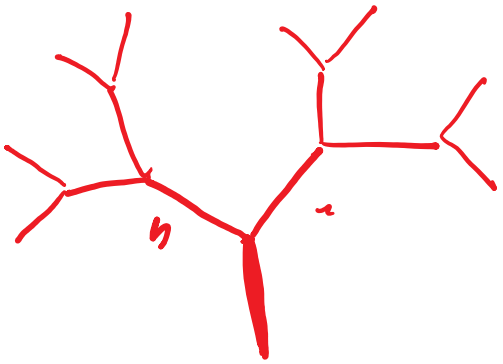
How much more time do you need to get 80% or more on PA4?

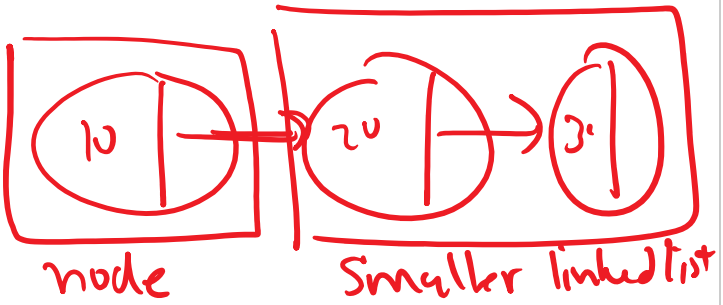
Phelps 35zi

A. I already have that score
B. I am on track to complete the PA tonight
C. One more day
D. One more week
E. I plan to let this PA slide

Thinking recursively!

- Many structures in nature and CS that are recursive
- A recursive solution to a problem is all about describing the problem in terms of a smaller version of itself!

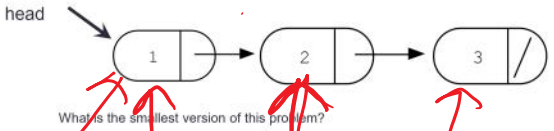




Thinking recursively!

1. Base case: solve the smallest version(s) of the problem
2. Recursive case: describe the problem in terms of itself!
   - Assume you have a solution with smaller input size!
   - Describe the problem in terms of a smaller version of itself.

Example problem: Print all the elements of a linked-list backwards!




```
void printItemBack(Node *head){
  if (head == 0)
    return;
  if (head->next == 0){
    cout << head->info;
    return;
  }
  printItemBack(head->next);
  cout << head->info;
  return
}
```

Base case

Step 1: Base case!

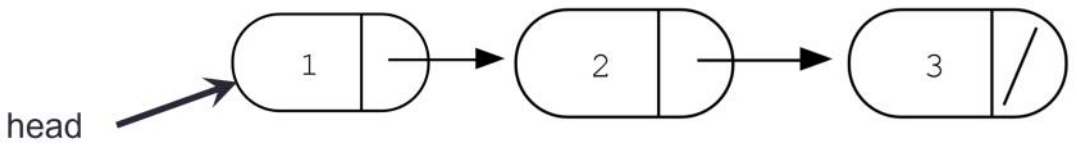
```
//Write code for the smallest version of the problem
void printBackwards(Node * head){




}
```

Monday, May 22, 2017 10:14 AM

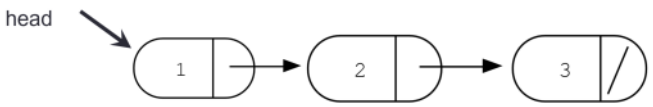
Step 2: Write the recursive case !

- Assume you have a solution for a smaller version of the problem!!!!
- Describe the problem in terms of a smaller version of itself

```
void printBackwards(Node * head){
      if (head == NULL)  //Base case
          return;
```



Example 2: Find the sum of the elements of a linked-list




```
int sum( Node *head ) {
  int result = 0;
  for( Node *p = head; p != 0; p = p->next ) {
    result = result + p->info;
  }
  return result;
}
```

Monday, May 22, 2017 10:14 AM

Step 1: Base case!

- Write code for the smallest version of the problem

```
int sum(Node * head){
    if (head == 0)
        return 0;
    if (head->next == 0)
        return head->info;
    int result = sum(head->next);
    return result + head->info;
}
```

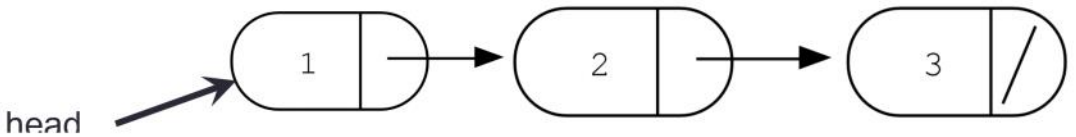
Base case

Monday, May 22, 2017 10:14 AM

Step 2: Write the recursive case !

- Assume you have a solution for a smaller version of the problem!!!!
- Describe the problem in terms of a smaller version of itself

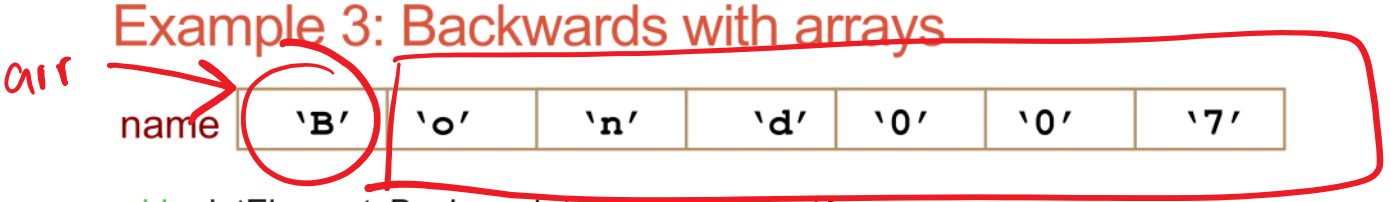
```
void sum(Node * head){
    if (head == NULL)  //Base case
```



Monday, May 22, 2017 10:14 AM

Example 3: Backwards with arrays

arr




```
void printElementsBackwards(char *arr, int len){

   if(len<=0){ //Base case
       return;
   }
   //Write your code here


}
```

printElementsBackwards(arr+1, len-1)
cout << arr[0];

Monday, May 22, 2017 10:14 AM

Next time

- Binary Search Trees