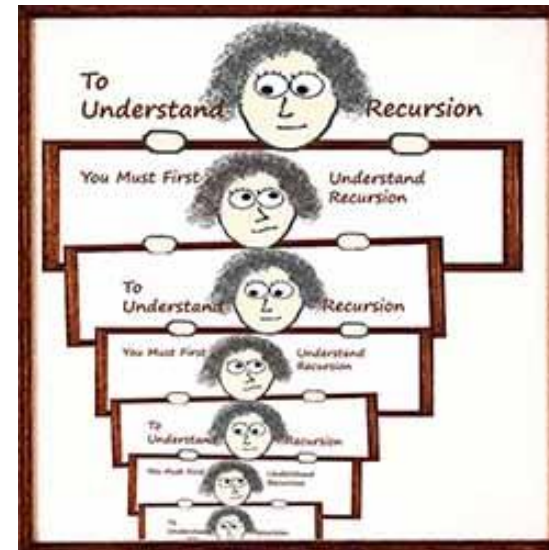


RECURSION



Problem Solving with Computers-I

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



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

- 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

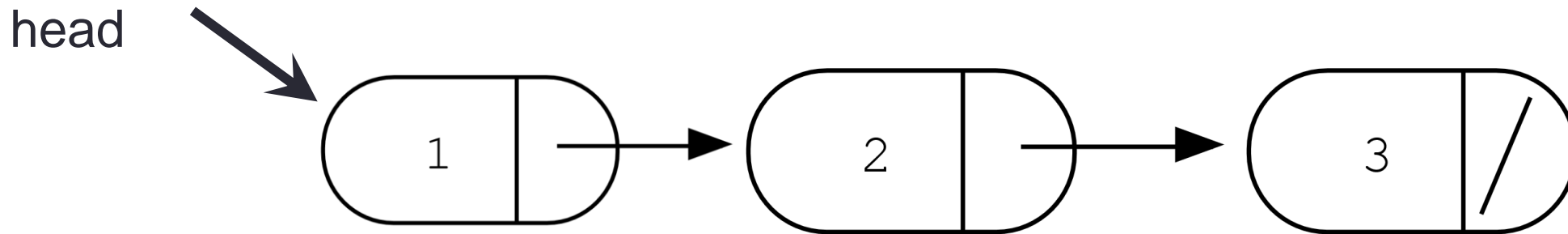
Thinking recursively!

- 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 small^{est} 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!



What is the smallest version of this problem?

Step 1: Base case!

//Write code for the smallest version of the problem

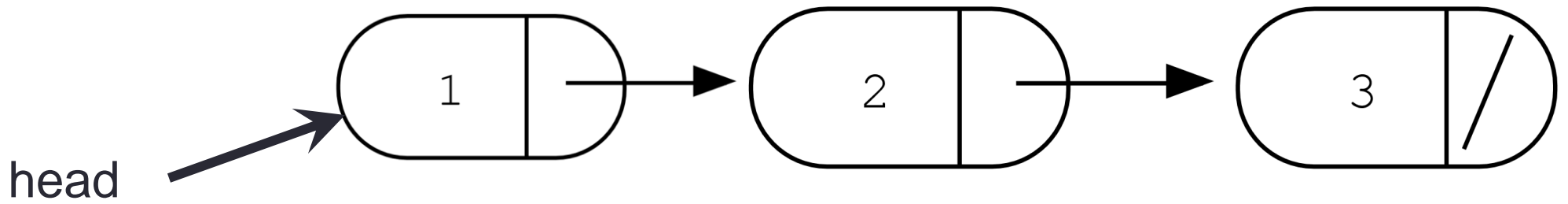
```
void printBackwards(Node * head){
```

```
}
```

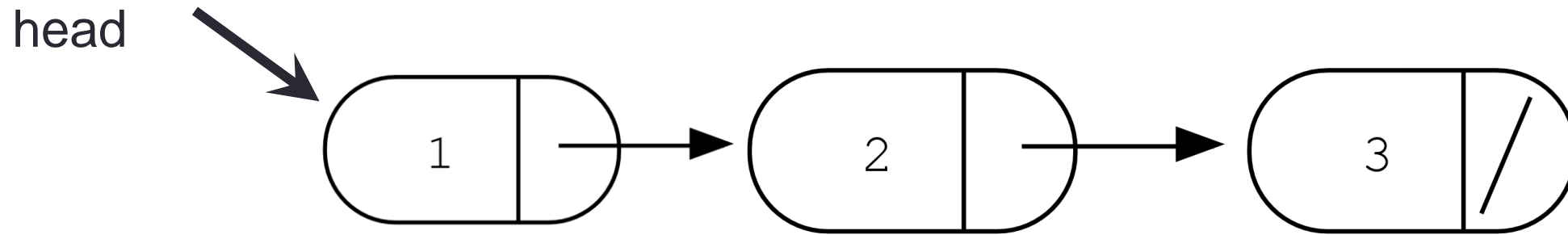
Step 2: Write the recursive case !

- 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



Step 1: Base case!

- Write code for the smallest version of the problem

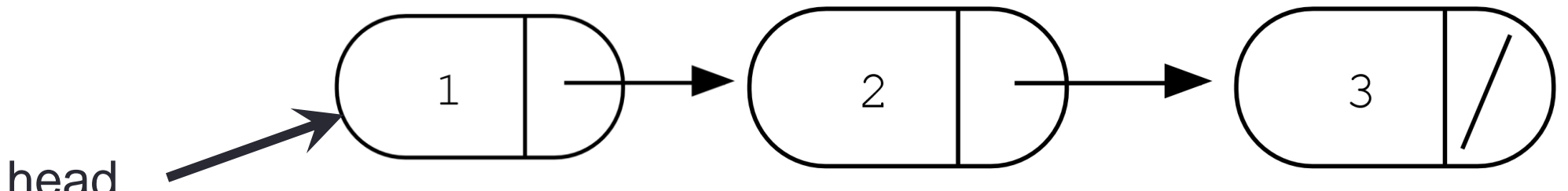
```
int sum(Node * head){
```

```
}
```


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
```



Example 3: Backwards with arrays

name	'B'	'o'	'n'	'd'	'0'	'0'	'7'
------	-----	-----	-----	-----	-----	-----	-----

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

```
    if(len<=0){ //Base case
```

```
        return;
```

```
    }
```

```
    //Write your code here
```

```
}
```

Next time

- Binary Search Trees