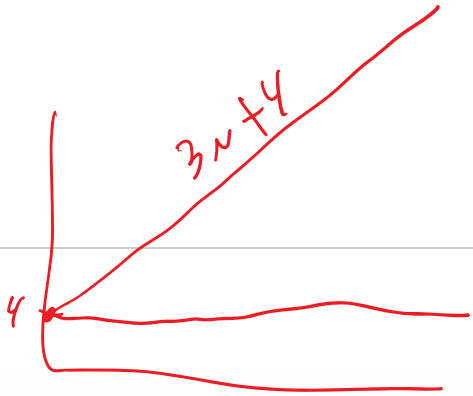
CSE12 - Lecture 9 - A00

Wednesday, October 12, 2022 8:00 AM

1 PA3 released today
PA2 hard-deadline → tonight



N → # of elements in the structure

Counting Steps

ArrayList Insert - ignore ExpandCapacity

```
public void insert(int index, String s) {
  //expandCapacity();   //ignore
  for (int i = size - 1; i >= index ; i--) {
    this.elements[i+1] = this.elements[i];
  }
  this.elements[index] = s;
  this.size += 1;
}
```

worst index = 0

N	i	cond	update	body
5	4	1		
	3			
	2			
	1			
	0			
	-1	N+1	N	N

best index = 5

N	i	cond	update	body
5	4	1		
		1	0	0

Best Case	Worst Case	Avg Case
0	0	0
1+1+0	1+(N+1)+N	1+($\frac{N}{2}$+1)+$\frac{N}{2}$
0	N	$\frac{N}{2}$
1	1	1
1	1	1
4	3N+4	$\frac{3}{2}$N+4

ArrayList ExpandCapacity

```
private void expandCapacity() {
  int currentCapacity = this.elements.length;
  if(this.size < currentCapacity) { return; }
  String[] expanded = new String[currentCapacity * 2];
  for(int i = 0; i < this.size; i += 1) {
    expanded[i] = this.elements[i];
  }
  this.elements = expanded;
}
```

allocate storage, init default

Best Case	Worst Case	~~Avg Case~~
1	1	
1+1	1+0	
0	2N+2N+1	
0	1+(N+1)+N	
0	N	
0	1	
3	7N+6	

ArrayList Insert - with ExpandCapacity

```
public void insert(int index, String s) {
  expandCapacity();
  for (int i = size - 1; i >= index ; i--) {
    this.elements[i+1] = this.elements[i];
  }
  this.elements[index] = s;
  this.size += 1;
}
```

Best Case	Worst Case	~~Avg Case~~ → week 6
3	7N+6	
1+1+0	1+(N+1)+N	
0	N	
1	1	
1	1	
7	10N+10	

	1	10N + 10
add()	3 + 2 5	7N + 6 + 2 7N + 8

Counting Steps - where size of the contents is *n*

LinkedList Add

```
public void add(String s) {
  Node current = this.front;
  while(current.next != null) {
    current = current.next;
  }
  current.next = new Node(s, null);
  this.size += 1;
}
```

Best Case	Worst Case	Avg Case
1 N+1 N 1 1 2N+4		?

LinkedList Insert

```
public void insert(int index, String s) {
  Node current = this.front;
  for(int i = 0; i < index; i += 1) {
    current = current.next;
  }
  current.next = new Node(s, current.next);
  this.size += 1;
}
```

index = 0 index = N

Best Case	Worst Case	Avg Case
1 1+1+0 0 1 1 5	1 1(N+1)+N N 1 1 3N+5	

LinkedList Get

```
public String get(int index) {
  Node current = this.front.next;
  for(int i = 0; i < index; i += 1) {
    current = current.next;
  }
  return current.value;
}
```

index = 0 index = N-1 index = $\frac{N}{2}$

Best Case	Worst Case	Avg Case
1 1+1+0 0 1 4	1 1+(N+1)+N N 1 3N+4	1 1+($\frac{N}{2}$+1)+$\frac{N}{2}$ $\frac{N}{2}$ 1 $\frac{3N}{2}$+4

ArrayList Get

```
public String get(int index) {
  return this.elements[index];
}
```

Best Case	Worst Case	Avg Case
1	1	1