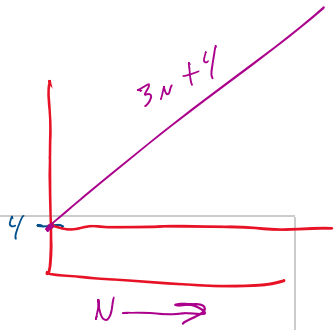
CSE12 - Lecture 9

Friday, April 21, 2023 8:00 AM

Exam 1 → next Friday 4/28



N → # of elements in the structure

Counting Steps

ArrayList Insert - ignore ExpandCapacity

	Best Case	Worst Case	Avg Case
	0	0	0
	1+1+0	1+(N+1)+N	1+(N/2+1)+N/2
	0	N	N/2
	1	1	1
	1	1	1
	4	3N+4	3/2 N + 4

```
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;
}
```

cond ||||1 update |||| body |||| N+1 N N

Worst case index = 0: N = 5, i = 4 3 2 1 0 -1

Best case index = 5, N = 5: i = 5 → cond 1, update 0, body 0

ArrayList ExpandCapacity

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

N+2

```
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;
}
```

1) allocate room 2) init default

ArrayList Insert - with ExpandCapacity

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

→ week 6

```
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;
}
```

add()

3 + (2) → 5 7N+6+(2) → 7N+8

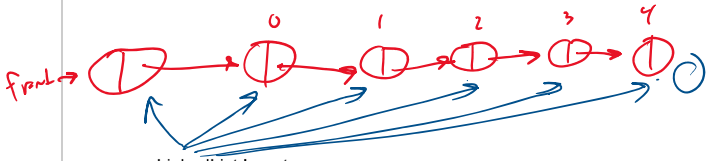
Name: _____ PID: _____ Code: 7066

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	
1	1	
	2N + 3	



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;
}
```

Best Case	Worst Case	Avg Case
index = 0	index = N-1	index = N/2
1	1	
1 + 1 + 0	1 + (N+1) + N	
0	N	
1	1	
1	1	
5	3N + 5	

replace N with N-1

Actually, do not replace with n-1 - because you can insert at the end f the list, it's still okay to loop to n.

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;
}
```

Best Case	Worst Case	Avg Case
index = 0	index = N-1	index = N/2
1	1	
1 + 1 + 0	1 + (N-1+1) + N-1	
0	N-1	
1	1	
1	1	
4	3N + 1	

ArrayList Get

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

Best Case	Worst Case	Avg Case
1	1	1