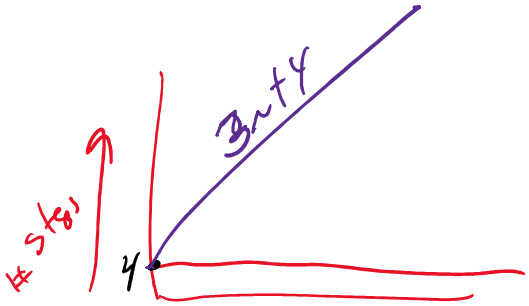
CSE12 - Lecture 9

Wednesday, April 17, 2024 10:00 AM

Exam 1 - Wednesday

- https://ucsd-cse12-sp24.github.io/lectures/exam1.html



Lecture 9

Counting Steps

N → # of elements in the structure

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;
}
```

	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	$\frac{3}{2}N+4$

worst case: index = 0, size = N = 5

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

best case → index at the end (add)

index = 5 = N

i = 4

cond	body	update
1	0	0

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 memory default init

	Best Case	Worst Case
	1	1
	1+1	1+0
	0	1+2N+2N
	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
	3	7N+6
	1+1+0	1+(N+1)+N
	0	N
	1	1
	1	1
	7	10N+10

3 + 2 = 5 7N+6 + 4 = 7N+10

add()

7 10N +10

3 + 2 = 5 7N+6+2 = 7N+8

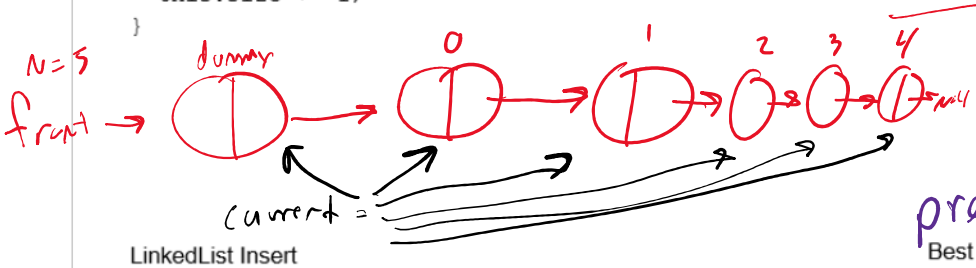
Name: _____ PID: _____ Code: 6891

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

LinkedList Add

Best Case	Worst Case	Avg Case
2	2N+4	

```
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;
}
```



LinkedList Insert

Best Case	Worst Case	Avg Case
5	3N+5	

```
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;
}
```

LinkedList Get

Best Case	Worst Case	Avg Case
4	3N+4	

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

ArrayList Get

Best Case	Worst Case	Avg Case
1	1	1

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