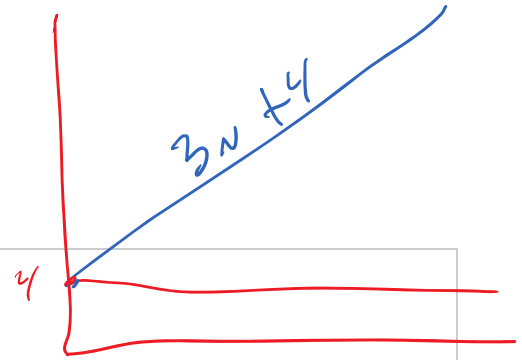


# CSE12 - Lecture 9 - B00

Wednesday, October 12, 2022 9:00 AM

PA3 released today  
PA2 hard-deadline is tonight

$N \rightarrow$  # 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;
}
```

Best Case Worst Case Avg Case

$$\begin{array}{ccc} 1 + 1 + 0 & 1 + (N+1) + N & 1 + (\frac{N}{2} + 1) + \frac{N}{2} \\ 0 & N & \frac{N}{2} \\ \hline 4 & 3N + 4 & \frac{3}{2}N + 4 \end{array}$$

Worst Case  
index = 0

cond  $\frac{N}{2}$  update  $\frac{N}{2}$  body  $N$   
best case  
index = 3  
 $N = 5$   
 $i = 4$   
 $i = 3$   
 $i = 2$   
 $i = 1$   
 $i = 0$   
 $-1$   
cond update body  
1 0 0  
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;
}
```

Best Case Worst Case ~~Avg Case~~

$$\begin{array}{ccc} 1 & 1 & \\ 1 + 1 & 1 + 0 & \\ 0 & 1 + 2N + 2N & \\ 0 & 1 + (N+1) + N & \\ 0 & N & \\ 0 & 1 & \\ \hline 3 & 7N + 6 & \end{array}$$

allocate room  
init default

### 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~~

$$\begin{array}{ccc} 3 & 7N + 6 & \\ (1 + 0) & 1 + (N+1) + N & \\ 0 & N & \\ 1 & 1 & \\ 1 & 1 & \\ \hline 7 & 10N + 10 & \end{array}$$

$\rightarrow$  worst 6

add()

$$3 + (2) \quad 7N + 6 + (2)$$

add()

$$3 + \textcircled{2}$$

$$5$$

$$7n + 6 + \textcircled{2}$$

$$7n + 8$$

### Counting Steps - where size of the contents is $n$

#### LinkedList Add

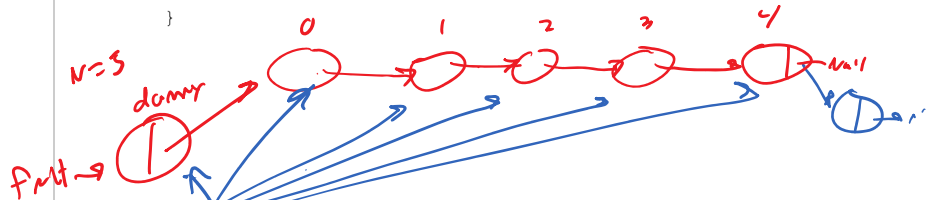
```
public void add(String s) {
    Node current = this.front;
    while(current.next != null) {
        current = current.next;
    }
    current.next = new Node(s, null);
    this.size += 1;
}
```

Best Case

Worst Case

Avg Case

$$\begin{array}{c} 1 \\ n+1 \\ n \\ 1 \\ 1 \\ \hline 2n+4 \end{array}$$



#### 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~~

$$\begin{array}{c} 1 \\ 1+1+0 \\ 0 \\ 1 \\ \hline 5 \end{array} \quad \begin{array}{c} 1 \\ 1+(n+1)+n \\ n \\ 1 \\ \hline 3n+5 \end{array}$$

#### 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  
Best Case

index=n-1  
Worst Case

index =  $\frac{n}{2}$   
Avg Case

$$\begin{array}{c} 1 \\ 1+1+0 \\ 1 \\ \hline 4 \end{array} \quad \begin{array}{c} 1 \\ 1+n+(n-1) \\ n-1 \\ 1 \\ \hline 3n+1 \end{array} \quad \begin{array}{c} 1 \\ 1+(\frac{n}{2}+1)+\frac{n}{2} \\ \frac{n}{2} \\ 1 \\ \hline \frac{3}{2}n+4 \end{array}$$

#### ArrayList Get

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

Best Case

Worst Case

Avg Case

$$\begin{array}{c} 1 \\ 1 \\ 1 \end{array}$$