

# CSE 12 – Basic Data Structures and Object-Oriented Design

## Lecture 11

Greg Miranda, Fall 2020

# Announcements

- Quiz 11 due Wednesday @ 9am
- Survey 4 due Friday @ 11:59pm
- PA3 due Wednesday @ 11:59pm
- Exam 1 - Friday
  - Released @ 8am on Friday
  - Closes @ 12pm on Saturday
  - See Piazza post for details

# Topics

- Questions on Lecture 11?
- Big Theta
- Sorting

Questions on Lecture 11?

Count how many times each line executes, then say which  $\Theta( )$  statement(s) is(are) true.

	Line #
<code>int maxDifference(int[] arr){</code>	1
<code>    max = 0;</code>	2
<code>    for (int i=0; i&lt;arr.length; i++) {</code>	3
<code>        for (int j=0; j&lt;arr.length; j++) {</code>	4
<code>            if (arr[i] - arr[j] &gt; max)</code>	5
<code>                max = arr[i] - arr[j];</code>	6
<code>        }</code>	7
<code>    }</code>	8
<code>    return max;</code>	9
<code>}</code>	

A.  $f(n) = \theta(2^n)$

B.  $f(n) = \theta(n^2)$

C.  $f(n) = \theta(n)$

D.  $f(n) = \theta(n^3)$

E. Other/none/more

*(assume  $n = arr.length$ )*

Count how many times each line executes, then say which  $\Theta(\ )$  statement(s) is(are) true.

```
int sumTheMiddle(int[] arr){
    int range = 100;
    int start = arr.length/2 - range/2;
    int sum = 0;
    for (int i=start; i<start+range; i++)
    {
        sum += arr[i];
    }
    return max;
}
```

- A.  $f(n) = \theta(2^n)$
- B.  $f(n) = \theta(n^2)$
- C.  $f(n) = \theta(n)$

- D.  $f(n) = \theta(1)$
- E. None of these  
*(assume  $n = arr.length$ )*

# Selection Sort – what does it print out?

```
import java.util.Arrays;
public class Sort {
public static void sortA(int[] arr) {
    for(int i = 0; i < arr.length; i += 1) {
        System.out.print(Arrays.toString(arr) + " -> ");
        int minIndex = i;
        for(int j = i; j < arr.length; j += 1) {
            if(arr[minIndex] > arr[j]) { minIndex = j; }
        }
        int temp = arr[i];
        arr[i] = arr[minIndex];
        arr[minIndex] = temp;
        System.out.println(Arrays.toString(arr));
    }
}
}
```

```
Sort.sortA(new int[]{ 53, 83, 15, 45, 49 });
[53, 83, 15, 45, 49] ->
```

# Insertion Sort – what does it print out?

```
import java.util.Arrays;
public class Sort {
public static void sortB(int[] arr) {
    for(int i = 0; i < arr.length; i += 1) {
        System.out.print(Arrays.toString(arr) + " -> ");
        for(int j = i; j > 0; j -= 1) {
            if(arr[j] < arr[j-1]) {
                int temp = arr[j-1];
                arr[j-1] = arr[j];
                arr[j] = temp;
            }
        }
        System.out.println(Arrays.toString(arr));
    }
}
```

```
Sort.sortB(new int[]{ 53, 83, 15, 45, 49 });
[53, 83, 15, 45, 49] ->
```