CSDS 233 Spring Session 1

SI Leader: Jakob Danninger

1/26/2022

Disclosure: This is a supplement to class, not a replacement. This should not be your only study activity for exams, it should aid you in studying. I do not have the actual exam so questions here will differ from those on the exam.

Session Objectives:

1) Be able interpret and create recursive functions

2) Understand what big O is and determine the big of basic pieces of code

Practice Problems:

1) Compare the runtime of the following $2^n, \log(n), n^2, n, nlog(n), 1$

$$\square < \square < \square < \square < \square < \square$$

2) Determine the big O of the following

```
public void example1 (int N) {
        for (int i = 0; i < N; i++) {
                System.out.println(”do something ”);
        }
        for (int i = 0; i < N; i++) {
                System.out.println(”do something ”);
        }
}
```

3) Determine the big O of the following

```
public void example2 (int N) {
```

```
        for (int i = 0; i < N; i++) {
                for (int j = 0; i < N; j++) {
                        System.out.println(”do something ”);
                }
        }
}
```

4) The following is a recursive algorithm circle the base case

```
public boolean example3 (int N) {
        if (N < 1) {
                return True;
        }
        N = N / 2
        // Recursion
        example3(N)
}
```

5) What is the value of N at each run for the function example3(4)

Simplify the following big O expression

6) $9999^2 + n + n\log(n)$

7) $\log n + n + 4n$

8) $300n + 30n^2 + 3n^3$

9) $1000 + n\log n + 2^n + 9999^2 n$

10) Determine the big O expression of the following code:

```
public void example4 (int N, int M) {
    for (int i = 0; i < N; i++) {
        for (int j = 0; M >= 0; j++) {
                M = M/2
        }
    }
}
```

11)

Coding activity found here:

https://github.com/jdanninger/CSDS233-Supplemental-Instruction/blob/main/Session%201%20-%20Intro%20to%20Recursion%20and%20Runtime%20Analysis/RecursivePractice.java

Once you reach the site copy and paste the code into your preferred development environment. . . if you do not have one set up go to https://replit.com/ and use their web ide

What you need to code

- void countDown(int n) → This method should recursively take N and print in the console a count down from n to 0. For example is you get countdown(3) it should print 3, 2, 1
- int sumOfNumbers(int n) → this method should return the sum of all numbers between 0 and n. For example sumOfNumbers(3) should return 6 since 6 = 1+2+3
- boolean isAscending(int[] arr, int index) → should return whether or not all the ints in the array are ascending. The index variable in the test case will initially be 1