CS 401 Algorithms
Assignment1
Due Date: April 30, 2021

Part I: Algorithm Analysis

1. Please provide running time functions, order of growth rates in Big-O, Omega and Theta Notations for the following code fragments. Please show your work.

   a.
   ```
   int i = n;
   int sum = 0;
   while (i >0) {
      sum++;
      i = i/2;
   }
   ```

   b.
   ```
   int j = n;
   int sum = 0;
   while (j > 0) {
      for (int i = 0; i < n; i++)
            sum++;
      j = j / 2;
    }
   ```

   c.
   ```
   while (n > 0) {
      for (int k = 0; k < n; k++)
            printReport(); // runs in O(n/2) time
       n = n / 2;
   }
   ```

   d.
   ```
   void aMethod(int n){
      if (n <= 1)
            return;
       anotherMethod(); //runs in O(1) time
      aMethod(n/2);
      aMethod(n/2);
   }
   ```

2. You are given two linked lists with O(n) elements. You are asked to find the intersection of two lists. Assume that the sets don’t have duplicates and not sorted, and also assume that comparison operation is a simple statement and it takes O(1) time.
   a. Please provide a pseudo code that has the growth rate of O(n^2)
   b. Improve your solution, and give a pseudo code that has a better growth rate which is O(nlogn).
   c. Is it possible to make it O(n)? Please provide a pseudo code for your solution, if any.

   Note: You can call other functions use other data structures, you don’t need to give pseudocodes for those functions. Just know their run time complexity.

3. You are given the following pseudocode for an algorithm:

```
method1(n)
    if n==0
        return 0
    else
        return method2(n-1, n-1) – (n-1)

method2(m, k)
    if m==0
        return method1(k)
    else
        return 1 + method2(m-1, k)
```

a. What would be the output for the following method call?

```
System.out.println(method1(4));
```

b. Please provide the recurrence form for the running time function of this algorithm.

c. Please find the growth rate of the algorithm in Big-O notation.

CS 401 Algorithms
Assignment1

Part II: The Ability of Soil to Hold Water

Soil scientists characterize and classify soils into different groups. Each group of soil has differing ability to hold water depending on the particles inside. For the sake of simplicity, you are given the internal structure of soil as *n*-by-*n* grid of *cells*. Each cell is either *1* or *0*.

- 1 means, the cell allows water to drain.
- 0 means that cell will hold the water.

You are asked to write a Java program. The program shall

- Read n-by-n grid of cells from a text file
- Determine if the soil allow the water to drain or hold the water
  - Prints “Allow water to drain” or “Don’t allow water to drain” as a console output.

You are asked to implement and use the Weighted Quick Union with Path Compression union-find algorithm to solve whether a soil example allows water to drain. You are supposed to build your solution on the given partial implementation provided with src.zip file. You are not allowed to use union-find implementations provided in alg4.jar.

Sample Input File-1:

```
  1    0    1    0    1
  1    1    0    1    0
  0    1    1    0    1
  1    0    1    0    1
  1    0    1    1    1
```

Sample Output-1:

```
        Allows water to drain
```

Sample Input File-2:

```
 1    0    0    1    1
 0    1    1    1    0
 1    0    0    0    1
 1    0    0    0    1
 1    1    0    1    1
```

Sample Output-2:

```
        Don’t allow water to drain
```

Sample Input File-3:

```
 1     0     0     1     1     1     1     0     0     0
 0     1     1     1     0     0     0     1     1     0
 1     0     0     1     1     0     0     1     0     0
 1     0     0     0     1     1     1     0     0     0
 1     1     0     1     1     1     1     0     0     0
 1     0     1     1     0     1     1     1     1     0
 0     0     0     0     1     1     0     0     0     0
 1     0     1     1     1     1     1     0     0     0
 0     1     0     1     1     0     1     0     1     0
 1     1     0     1     0     1     1     0     0     0
```

Sample Output-3:

```
        Allows water to drain
```

CS 401 Algorithms
Assignment1

HOW TO SUBMIT

You are supposed to submit your work as a single zip file via CANVAS. Zip file will include the following two archive files for each part:

- Part1.zip (word/pdf file)
- Part2.zip (src.zip)

Please use the following file format while naming the zip file:
LastNameFirstnameX_Y.zip where LastNameFirstname is your last name with the first letter in capital, followed by your first name with the first letter in capital; the X is the course code; the Y is the assignment #. (ex: SerceFatmaCS401_2.zip)