Data Structures and Algorithms
INFO 6205
Homework1
Due: September 12, 2019

Name: Qian Cai
NUID:001389278

Put all your java, compiled class files and documentation files into a zip file Homework1.zip and submit it via the drop box on the blackboard before the END of due date. Put your name on all .java files. There will be a short quiz on this homework.

1. Read references on Java, Stacks, Algorithm Analysis and Slides provided

2. What is an Algorithm? Give three examples

**Algorithm is a solution or a list of commands that could scientifically solve problems and perform a computation.**

**Three success algorithm examples are Kruskal's algorithm, Discrete Fourier transform and N-body simulation.**

**I also have three algorithm examples in programming.**

**Firstly, write an algorithm to find the largest number between a and b.**

```
Step 1: Start.
Step 2: Declare variables a and b.
Step 3: Read variables a and b.
Step 4: if a>b
            Display a is the largest number.
        If b>a
            Display b is the largest number.
Step 5: Stop.
```

**Secondly, write an algorithm to judge whether number a is an even number.**

```
Step 1: Start.
Step 2: Declare variables a.
Step 3: Read variables a.
Step 4: if a%2=0
            Display a is an even number.
        else
            Display a is an odd number.
Step 5: Stop.
```

**Thirdly, write an algorithm to judge whether 2003 is a leap year.**

```
Step 1: Start.
Step 2: Declare variables a.
Step 3: Initialize variables a=2003.
Step 4: if a%2003=0
              Display 2003 is a leap year.
```

```
        else
            Display 2003 is not a leap year
Step 5: Stop.
```

3. What is time and space complexity of an algorithm?

**Time complexity of an algorithm quantifies the amount of time taken by an algorithm to run as a function of the length of the input.**
**Space complexity of an algorithm quantifies the amount of space or memory taken by an algorithm to run as a function of the length of the input.**

4. What is the time complexity of the following code, and why?

```
public makeSentence ( String[] words) {
        String sentence="";
        for (String w:words) {
                sentence+=w;
        }
        return sentence;
}
```

**The time complexity of the following code is O(n*n).**
**Because the loop count of "for (String w: words)" is n and the loop count of "sentence+=w;" is n，too. So the time complexity of the following code is O(n*n).**

5. What are all Stack operations, explain.

**The all Stack operations including pop and push.**
**Pop means remove objects from first in the stack.**
**Push means add object to the stack before first, which is reverse of pop.**
**The stack operation follows principle that last in first out.**

6. Consider String "It was the best of time". Start with the first word, design a Stack such that when you read back the words, the order of string does not change. Write Java code for all necessary operations of Stack. Compile and run the code.

7. Consider the following Node data structure, build a Stack linkedList with the following data: {31, "Name1"}, {24, "Name2"}, {10, "Name3"}, {44, "Name4"}, {81, "Name5"}.

a) Write java implementation for all necessary Stack operations including stack pointers.
b) Compile and run your program.
c) What is Stack linkedList time and space complexity?

```
class Node {
        int Age;
        String Name;
        Node next;
 }
```

**The time complexity of Stack linkedList is O(1).**

**The space complexity of Stack linkedList is O(n).**

8. Consider data: {31, “Name1”}, {24, “Name2”}, {10, “Name3”}, {44, “Name4”}, {81, “Name5”}.

a) Provide Array implementation of Stack.
b) Compile and run the code.
c) What is time and space complexity of Stack Array implementation?

**The time complexity of Stack Array is O(n*n).**
**The space complexity of Stack Array is O(n).**

9. Suppose in problem-8 above, the array size was: a) too large, or b) too small. How would you manage resizing the array for (a) and (b). Write the code, compile and test the program. Discuss the running time/space complexity of your approach.

I**f resizing the array for(a), the time complexity of Stack Array is O(n), the space complexity of Stack Array is O(n).**
I**f resizing the array for(b), the time complexity of Stack Array is O(n), the space complexity of Stack Array is O(n).**