/\*

Jason Wilson

Thursday, February 6th 2020

CMPT435L 111 20S

\*/

Assignment 1

**Section 1: Pen-and-paper Exercises**

1. Consider the following problem:

**Input**: a 32-bit signed integer

**Output**: reverse digits of the input number.

Example 1:

Input: 123

Output: 321

Example 2:

Input: -123

Output: -321

Example 3:

Input: 120

Output: 21

Assume we are dealing with an environment which could only store integers within the 32-bit signed integer range: [−2^31, 2^31 − 1]. Assume that your function returns 0 when the reversed integer overflows.

**Design an algorithm that solves this problem.**

1. **describe the idea behind your algorithm in English (2 points);**
2. Check if the number is negative, if it is, set the negative flag variable to true
3. While the input number is bigger than 0 take the remainder of the input number divided by 10 to extract the significant number
4. Divide the input number by 10 to remove the significant figure from the input number
5. Multiply the reversed number by 10 and add the extracted significant figure to build the number in reverse
6. If the number started as a negative, multiply the reversed number by -1
7. Output: reversed input

*For this algorithm I will use the remainder function to remove the significant figure and multiplying that extracted number by the reversed number to properly and efficiently reverse the input.*

**(ii) provide pseudocode (5 points);**

**INPUT:** int x

Int sigFig, int reversed = 0, boolean, negative = false

**if** (x < 0)

negative = true

x = x \* -1

**end if**

**while** ( x > 0 )

sigFig = x % 10

x = x / 10

reversed = (reversed \* 10) + sigFig

**end while**

**if** (negative)

reversed = reversed \* -1

**end if**

**output** reversed

**Note: Most problems in assignments are classic interview questions -- it is not hard to find their answers on the internet. Please solve problems by yourself. What you submit to me should be YOUR work, not somebody else’s. You would receive 0 points and at least a reduced grade if you copy the answer of other’s or from the internet.**

1. Consider the following problem:

**Input:** a list of n-1 integers and these integers are in the range of 1 to n. There are no duplicates in list. One of the integers from 1 to n is missing in the list.

**Output:** find the missing integer

Let the input array be [2, 4, 1, 6, 3, 7, 8]. Elements in this list are in the range of 1 to 8. There are no duplicates, and 5 is missing. Your algorithm needs to return 5.

Let the input array be [6, 3, 4, 5, 1]. Elements in this list are in the range of 1 to 6. There are no duplicates, and 2 is missing. Your algorithm needs to return 2.

Design an algorithm that solves this problem.

**Note: You need to consider all the possibilities including overflow.**

**Note: There is NO requirement of “in-place” algorithms. You can use extra memory/data structure in the algorithm.**

1. **describe the idea behind your algorithm in English (2 points);**
2. Create a second array with one more element than the array with the missing number
3. Initialize the new array with numbers in increasing order from 1 to the length of the array
4. Take the all elements that exist in the input array and remove them from the test array by setting them equal to 0
5. Whichever element is not equal to 0 in the test array at this point is the missing number

**(ii) provide pseudocode (5 points);**

**input** A[]

int[] arrTest = new int[size(A) + 1] , int num = 0

**for** i in [0 , size(arrTest)

arrTest[i] = i + 1

**end for**

**for** i in [0 , size(A)]

arrTest[A[i] - 1] = 0

**end for**

**for** i in arrTest

**if** ( i != 0 )

num = I

break

**end if**

**end for**

**Output** num

1. Consider the following problem:

**Input:** an array A of n integers (positive, negative, or 0), elements sorted in ascending order.

**Output:** if there exists a majority element.

An element is a majority if it appears more than n/2 times. For example, if the input list is:

{0, 0, 0, 0, 0, 0, 1, 1, 2, 4, 7}

The output should be 0, as 0 appears 6 times (>n/2 = 11/2 times).

However, if the input list is:

{0, 0, 0, 1, 1, 2, 3, 10, 10}

The majority element does not exist.

Design an algorithm that solves this problem.

1. **describe the idea behind your algorithm in English (2 points);**
2. Get the upper and lower bounds for your search area by dividing the size of the array in half
3. While the upper bounds of the search area is less than the length of the array, check to see if the elements in the array at the location of the upper and lower bound are equal
4. If they are, you found the majority element, if not, increase the upper and lower bounds by one and continue
5. If the upper bounds of the search reaches the length of the array and the elements do not equal each other, there is no majority element

**(ii) provide pseudocode (5 points);**

**Input** = A[]

int half = size(A) / 2 , int i = 0 , int j = half

**while** ( j < size(A) )

**if** (A[i] == A[j] )

**Output** true

**end if**

i++

j++

**end while**

**Output** false

**Section 2: Java Implementation**

1. Implement problem 1 in Java.

Note:

Find a file called Problem1.java in assignment 1 folder.

Complete the method of reverse().

Test your method in the main method provided following the comments.

**Important: In all of the assignments of this course, when you are asked to implement an**

**algorithm for a problem, your code will be evaluated based on:**

**5 points - Execution**

**Each file must run without error or warning on valid input described in the main method provided.**

**5 points - Within Code Documentation**

**Is the code documented for obvious understanding of the use, preconditions, and postconditions of each function?**

**20 points - Correctness**

**Is the algorithm implemented correctly? Does your method pass the test?**

1. Implement problem 2 in Java.

Note:

Find a file called Problem2.java in assignment 1 folder.

Complete the method of missingnumber().

Test your method in the main method provided following the comments.

1. Implement problem 3 in Java.

Note:

Find a file called Problem3.java in assignment 1 folder.

Complete the method of majority().

Test your method in the main method provided following the comments.

**TURN-IN CHECKLIST:**

1. **Answers to Section 1 (.doc/.txt/.pdf), and to Section 2 (all your source Code (.java files)). Remember to include your name, the date, and the course number in comments near the beginning of your code/report.**
2. **Create a folder and name it 'FirstName\_LastName\_assignment\_1'. In the newly created folder copy and paste your files (.doc/.txt/.java files). Then compress the folder, and push it to iLearn.**