Timothy Allec 1/12/2024

Name, SID, Date ....

In Class Assignment 3: Binary Counter

Benjamin Sanders, MS November 25, 2020

1 Introduction

You will need to work individually to complete this assignment. Write your name at the top of all pages for this assignment. Turn in all work to Blackboard on or before the deadline to receive credit.

You may use additional libraries and online resources, if you get them approved in writing, over email, from the instructor first. If you have received approval from the instructor, write the approved libraries and any references in the space below.

import java.lang.InterruptedException;
import java.util.concurrent.TimeUnit;

....
....
....
....

2 Assignment Description

2.1 Big Picture

This method exhibits an automatic way to enumerate the input values in a truth table.

2.2 Algorithm Implementation

Implement the following algorithm in Java, using the Vector data structure for any 1-D array, 2-D array, or linear algebra purposes.

```
INCREMENT(A)
$i = 0$
while $i < A.length$ and $A[i] == 1$
    $A[i] = 0$
    $i = i + 1$
if $i < A.length$
    $A[i] = 1$
```

Note that A is an array of symbols, 0 and 1.

```
public class BinaryCounter{
   // this function increments a 4 bit by 1
public static int []function (int[] array){
   int i = 0;
   while (i < array.length && array[i] == 1){
      array[i] = 0;
      i = i +1;
   }
   if ( i < array.length -1){
         array[i] = 1;
      }
      return array;
   }
   // this main initiliazes an array and calls the function method to output the incremented 4 bit
public static void main(String[] args) {
   int [] arrayOne = {0,1,1,0};
   int [] arrayTwo = function(arrayOne);

   for (int i = 0; i < arrayTwo.length; i++){
   System.err.print(arrayTwo[i] + " ");
   }
}
}
```

2.3 Time Complexity Analysis

Where n is the number of data points in A, analyze the time complexity of the given algorithm with respect to n. Write the result of your analysis in big-O notation, i.e. $O(n^2)$ in the space below.

O(n)
n = A.length

....

2.4 Space Complexity Analysis

Where n is the number of data points in A, analyze the space complexity of the given algorithm with respect to n. Write the result of your analysis in big-O notation, i.e. $O(n \cdot log(n))$ in the space below.

....

O(n)

2.5 New Algorithm Design

In the space below, create an algorithm to print out all the values of the binary counter, for a length of 8 bits, with a 2-second pause in-between each print. Use pseudocode written in a style similar to the given algorithm, and implement it in Java. You may use as many additional pages as necessary for this purpose.

```
import java.lang.InterruptedException;
import java.util.concurrent.TimeUnit;

public class BinaryCounterOptimization{

    public class TimerTest{
        public TimerTest(){
        }
    public void waittest(){
        try{
            TimeUnit.SECONDS.sleep(2);
        }
        catch ( java.lang.InterruptedException e){
            System.out.println(e);
        }
    }
}

// this method increments an 8 bit
public static int []function (int[] array){
    int i = 0;
    while (i < array.length && array[i] == 1){
        array[i] = 0;
        i = i +1;
    }
    if ( i < array.length -1){
            array[i] = 1;
        }
        return array;
    }
    // this main initiliazes an array and calls the function method to output the incremented 4 bit
public static void main(String[] args) {

    int [] arrayOne = {0,1,1,0,1,1,1,0};
    int [] arrayTwo = function(arrayOne);

    System.out.println("before timer \nresult:");
    for (int i = 0; i < arrayTwo.length; i++){
    System.err.print(arrayTwo[i] + " ");
    }

    // this creates a 2 second break in between  of the 8 bit using the time unit library
    TimerTest timer = new BinaryCounterOptimization().new TimerTest();
timer.waittest();
System.out.println();
    System.out.println("after timer \nresult:");
    for (int j = 0; j < arrayTwo.length; j++){
        System.err.print(arrayTwo[j] + " ");
        timer.waittest();
    }
}
}
```

3 What to Turn In

Turn in one PDF or Word document on Blackboard, containing the following items.

1. All pages scanned or photographed of the In Class Assignment completed document.
2. Any additional pages you used to complete the assignment.
3. All code created for the assignment, along with test cases.
4. One statement indicating which parts of your implementation(s) are working, and which parts are not.
5. Screenshots demonstrating the code working, if it is working.