Computer Science 501
Data Structures & Algorithms
The College of Saint Rose
Fall 2014

Lab 5: More Analysis
Due: 6:00 PM, Tuesday, September 30, 2014

This "mini" lab, which is mostly a problem set moreso than a lab, will give you a chance to practice with more of the analysis tools we have considered in class. You will also do a small amount of programming in preparation for upcoming lab experiments.

You may work individually or in a group of 2 or 3 on this lab, but it is essential that each individual understands how to do the lecture assignment and problem set questions, as exam questions will be very similar. As you know, exams are not group efforts.

Getting Set Up

To get your BlueJ environment set up for this week's lab assignment, start BlueJ and choose "New Project" from the "Project" menu. Navigate to your folder for this course and choose the name "Lab5" (no spaces) for the project.

Create a document where you will record your answers to the lecture assignment and lab questions. If you use plain text, call it "lab5.txt". If it's a Word document, you can call it whatever you'd like, but when you submit, be sure you convert it to a PDF document "lab5.pdf" before you submit it.

Lecture Assignment Questions

We will usually discuss these questions at the start of class on the lab due date, so no credit can be earned for late submissions of lecture assignment questions.

> **? LA Question 1:**
> Bailey Exercise 5.34, p. 114, parts a-c. Set up and solve a recurrence to answer part b. (5 points)

> **? LA Question 2:**
> Solve the following recurrence relations. (5 points, 1 each)
> a. $x(n) = x(n-1) + 6$ for $n > 1$, $x(1) = 0$
> b. $x(n) = 2x(n-1)$ for $n > 1$, $x(1) = 4$
> c. $x(n) = x(n-1) + n$ for $n > 0$, $x(0) = 0$
> d. $x(n) = x(n/2) + n$ for $n > 1$, $x(1) = 1$ (solve for $n = 2^k$)
> e. $x(n) = x(n/3) + 2$ for $n > 1$, $x(1) = 1$ (solve for $n = 3^k$)

Problem Set Problems

Consider the following recursive algorithm:

```
int mystery(int A[0..n-1])
  if (n==1) return A[0]
  else
    temp = mystery(A[0..n-2])
    if (temp <= A[n-1]) return temp
    else return A[n-1]
```

> **? Question 1:**
> What does this algorithm compute? (3 points)

> **? Question 2:**
> Set up and solve a recurrence relation for the algorithm's basic operation count. (10 points)

Consider the following recursive algorithm:

```
int S(n)
  if (n==1) return 1
  else return S(n-1) + n*n*n
```

> **? Question 3:**
> Set up and solve a recurrence relation for the algorithm's basic operation count. (10 points)

Generating Example Arrays

Soon, you will be asked to perform empirical analysis on the sorting algorithms we will be studying. As part of that task, you will be required to generate input arrays for the sorting algorithms. In order to test the best, worst, and average cases of some of these algorithms, you will need to generate input arrays with various characteristics. For simplicity, we will sort arrays of int.

- an array filled with n random values within a given range

- an array filled with n values sorted in ascending order

- an array filled with n values sorted in descending order

- an array filled with n values "nearly" sorted in ascending order

> **Practice Program:**
> Write a Java class IntArrayGenerator that includes static methods to generate and return arrays of int with each of the above characteristics, and a main method that tests these methods for various values of n and ranges. (12 points)

Submitting

Before 6:00 PM, Tuesday, September 30, 2014, submit your lab for grading. There are two things you need to do to complete the submission: (*i*) Copy your file with the answers to the lecture assignment and lab questions into your project directory. Be sure to use the correct file name. If you prepared your answers in Word, export to a PDF file and submit that. (*ii*) Upload a copy of your lab (a `.7z` or `.zip` file containing your project directory) using Submission Box at `http://sb.teresco.org` under assignment "Lab5".

Grading

This assignment is worth 45 points, which are distributed as follows:

Feature	Value	Score
LA Question 1 (5.34)	5	
LA Question 2 (recurrences)	5	
Question 1: `mystery` computes	3	
Question 2: `mystery` recurrence	10	
Question 3: `S` recurrence	10	
`IntArrayGenerator` required methods	10	
`IntArrayGenerator` main method	2	
Total	45	