

Lecture 11 (review session for intro to computing)

1)

(Big-Oh notation continued)

Assume that each of the following operations takes 1 time unit: an arithmetic op, a comparison, a logical op, an assignment, accessing a particular row in an array, accessing a particular column in an array, a variable declaration (allocation of space in memory for the declared variable).

```
public void fillArray(int[][] arr, int n)
{
    int row, col;

    for (row=0; row < n; row++)
        for (col=0; col < n; col++)
            arr[row][col] = 10*row + col;
}
```

- a) Find the worst-case time, $T(n)$ for the method fillArray above.
- b) $T(n)$ is $O(?)$

Solution:

a)
We have two int declaration operations (allocation of space for variables).
For each loop, there is one initialization (row = 0 or col = 0), $n+1$ for loop comparisons (row < n or col < n) and n increments (row++ or col++). Each increment is equivalent to an assignment and an arithmetic op. Each loop repeats the statements/operations contained within it n times.

within the inner for loop there are two operations to access a particular row and column of the 2 dimensional array (arr[row][col]), one assignment of a value to that particular array location, one multiplication, and one addition.

$$\begin{aligned} T(n) &= 2 + 1 + n+1 + 2n + n * (1 + n+1 + 2n + 5)) \\ &= 2 + 2 + 3n + n * (7 + 3n) \\ &= 2 + 2 + 3n + 7n + 3n^2 \\ &= 3n^2 + 10n + 4 \end{aligned}$$

$$3n^2 + 10n + 4 \leq 3n^2 + n^2 + n^2$$

$$\text{for } c = 5, n_0 = 6, \text{ and } T(n) \leq c*f(n) \\ (3n^2 + 10n + 4 \leq 5n^2)$$

b)
therefore $T(n)$ is $O(n^2)$

2)

(2-dimensional arrays)

what are the contents of the array arr above (from the method fillArray) when $n = 5$?

Solution:

lecture11solutions

```
0 1 2 3 4
10 11 12 13 14
20 21 22 23 24
30 31 32 33 34
40 41 42 43 44
```

3)

(Big-Oh notation)

We have an algorithm which has time $T(n) = 4n + 4$.

- To find the worst case time, if we pick the value 5 as the constant c , for the inequality $T(n) \leq c \cdot f(n)$ ($4n + 4 \leq 5n$ in this case) what is the value of n_0 ?
- If we pick the value 6 as the constant c , for the inequality $T(n) \leq c \cdot f(n)$ ($4n + 4 \leq 6n$ in this case) what is the value of n_0 ?

Solution:

- $n_0 = 4$
- $n_0 = 2$

4)

(pre-increment/post-increment pre-decrement/post-decrement operators)

What are the values of result1 through result4?

```
public static void main(String[] args)
{
    int i=25, j=30, k=4, l = 9;
    int result1, result2, result3, result4;

    result1 = 5 + ++i;
    result2 = 87 / --j;
    result3 = 50 % k++;
    result4 = 6 * l--;
}
```

Solution:

```
result1 is 31 (5 + 26)
result2 is 3 (87 / 29)
result3 is 2 (50 % 4)
result4 is 54 (6 * 9)
```

5)

(Primitive and class types and call/return-by-value)

Remember:

In Java you make a class cloneable by having it implement the Cloneable interface; e.g., to make class Person cloneable you would write:

```
public class Person implements Cloneable {
    // the usual class variables, constructors, and methods
    // go here
}
```

- What are the 3 lines that the main method in class TestNumber prints

lecture11solutions

- out?
b) What are the 2 lines that the main method in class TestVector prints out?
c) What changes would you make in class TestVector to ensure that the vector v in that class cannot be inappropriately modified?

```
public class TestNumber {
    private int number;

    public void setNumber(int n)
    {
        number = n;
    }

    public int getNumber()
    {
        return number;
    }

    public static void main(String[] args)
    {
        TestNumber newNum = new TestNumber();
        newNum.setNumber(25);
        System.out.println("number is: " + newNum.getNumber());
        int n1 = newNum.getNumber();
        n1 = 90;
        System.out.println("number is: " + newNum.getNumber());
        System.out.println("n1 is: " + n1);
    }
}
```

```
public class TestVector {
    private Vector v;

    public void setVector(int[] arr)
    {
        v = new Vector(arr.length);
        for (int i = 0; i < arr.length; i++)
            v.addElement(new Integer(arr[i]));
    }

    public Vector getVector()
    {
        return v;
    }

    public void printVector()
    {
        Integer n;
        System.out.print("Vector v has contents: ");
        for (int i=0; i < v.size(); i++) {
            n = (Integer)v.elementAt(i);
            System.out.print(" " + n.intValue());
        }
        System.out.println(".");
    }

    public static void main(String[] args)
    {
        int[] arr1 = {1, 2, 3, 4, 5};
        TestVector newVec = new TestVector();
        newVec.setVector(arr1);
        newVec.printVector();
        Vector n1 = newVec.getVector();
        n1.addElement(new Integer(90));
        n1.addElement(new Integer(-25));
        newVec.printVector();
    }
}
```

lecture11solutions

```
}
```

Solution:

a)

```
number is 25
number is 25
n1 is 90
```

b)

```
Vector v has contents: 1 2 3 4 5.
Vector v has contents: 1 2 3 4 5 90 -25.
```

c)

```
public Vector getVector()
{
    return v.clone();
}
```

=====
6)

(static and non-static variables)

a) what is the line printed out by the main method of class TestChar1?

b) what is the line printed out by the main method of class TestChar2?

```
public class TestChar1 {
    private char c;

    public void setChar(char ch)
    {
        c = ch;
    }

    public char getChar()
    {
        return c;
    }

    public static void main(String[] args)
    {
        TestChar1 newChar1 = new TestChar1();
        newChar1.setChar('a');
        TestChar1 newChar2 = new TestChar1();
        newChar2.setChar('z');
        char c1 = newChar1.getChar();
        char c2 = newChar2.getChar();
        System.out.println("c1 is " + c1 + " c2 is " + c2);
    }
}
```

```
public class TestChar2 {
    public static char c;

    public void setChar(char ch)
    {
        c = ch;
    }

    public char getChar()
    {
        return c;
    }

    public static void main(String[] args)
```

lecture11solutions

```
{
    TestChar2 newChar1 = new TestChar2();
    newChar1.setChar('a');
    TestChar2 newChar2 = new TestChar2();
    newChar2.setChar('z');
    char c1 = newChar1.getChar();
    char c2 = newChar2.getChar();
    System.out.println("c1 is " + c1 + " c2 is " + c2);
}
}
```

Solution:

a) c1 is 'a' c2 is 'z'
b) c1 is 'z' c2 is 'z'

For a class that has a static variable, only one version of that variable exists for all instantiations of objects of that class. This means that if there are 20 objects created of type TestChar2 as given directly above, all these instantiations will have a variable called number with the same value. Static variables can be used by different objects of the same class for synchronization.

7)

(spot the errors -- 7 compile-time, 2 run-time)

```
import java.util.*;
public class Test{

    String s1 = "hey";
    Vector strVec = new Vector(20);
    char ch = "a";

    public void setVector(String[] strs)
    {
        int i = 0;
        do {
            strVec.addElement(strs[i]);
        } while (i < strs.length)
        strVec.trimToSize();
    }

    public static void main(String[] args)
    {
        String a = {"one", "two", "three", "four"};
        setVector(a);
        String b[] = {};
        setVector(b);
    }
}
```

Solution (errors):

```
import java.util.*;
public class Test{

    String s1 = "hey";
    Vector strVec = new Vector(20);
    char ch = "a"; // should be char ch = 'a'; (compile)

    public void setVector(String[] strs)
    {
        int i = 0;
```

lecture11solutions

```
do {
// attempt to access array element before finding out
// whether its length is greater than 0 (run-time)
    strVec.addElement(strs[i]);
    // forgot to increment i (run-time -- infinite loop)
} while (i < strs.length) // forgot semi-colon after while (compile)

strVec.trimToSize();
}

public static void main(String[] args)
{
String a = {"one", "two", "three", "four"}; // forgot square brackets
(compile)
setVector(a);    // cannot call non-static method from within static one
(compile)
    string b[] = {}; // method is expecting an array, not a string (compile)
    setVector(b); // incorrect placement of square brackets (compile)
(compile)
}
}
```

Valid code:

```
import java.util.*;
public class Test{

    String s1 = "hey";
    Vector strVec = new Vector(20);
    char ch = 'a';

    public void setVector(String[] strs)
    {
        int i = 0;
        while (i < strs.length) {
            strVec.addElement(strs[i]);
            i++;
        }
        strVec.trimToSize();
    }

    public static void main(String[] args)
    {
        String[] a = {"one", "two", "three", "four"};
        Test newT = new Test();
        newT.setVector(a);
        String[] b = {};
        newT.setVector(b);
    }
}
```

=====
8)

(iteration and recursion)

Write a method `int square(int n)` that given a number `n` returns the value of `n*n` (`n` squared) without using the multiplication operator. You may use your knowledge of multiplication via addition from homework 1.

- Your method `int square(int n)` should call a recursive method that actually computes the square of `n`.
- Your method `int square(int n)` should call an iterative method

that actually computes the square of n.

Solution:

a)

```
public int square(int n)
{
    return product(n,n);
}

public int product(int n, int m)
{
    // n and m are whole numbers greater than or equal to 0
    // base case
    if (m==1) {
        return n;
    }
    else {
        return(n + product(n, m-1));
    }
}
```

b)

```
public int square(int n)
{
    return product(n,n);
}

public int product(int n, int m)
{
    int p = 0;
    for(int i=0; i < m; i++)
        p+=n;
    return p;
}
```

=====
 (Tracing through a recursive method)

9)

write out a trace of the return values when fib(5) is called.

```
public int fib(int n)
{
    // calculates fib(n) where n >=0
    if (n == 0)
        return 0;
    else if (n == 1)
        return 1;
    else
        return fib(n-1) + fib(n-2)
}
```

Solution:

$$\begin{aligned}
 \text{fib}(5) &= \text{fib}(4) + \text{fib}(3) \\
 &= [\text{fib}(3) + \text{fib}(2)] + [\text{fib}(2) + \text{fib}(1)] \\
 &= [[\text{fib}(2) + \text{fib}(1)] + [\text{fib}(1) + \text{fib}(0)]] + [[\text{fib}(1) + \text{fib}(0)] + [1]] \\
 &= [[[1] + [0]] + [1]] + [1 + 0] + [[1 + 0] + [1]] \\
 &= [[[1] + [1]] + [1]] + [[1] + [1]] \\
 &= [[[2]] + [1]] + [[2]] \\
 &= [[3]] + [[2]] \\
 &= 5
 \end{aligned}$$

10)

(inheritance)

Imagine that you have defined a class called BaseClass which has the following attributes:

```
private int num;
private String arg;
```

and the following methods:

```
public void doSomething(String app);
private void appendArg(String app);
public int getNum();
public void setNum(int n);
public String getArg();
public void setArg(String a);
```

You create a new class DerivedClass that is a subclass of BaseClass and define a variable newClass in DerivedClass's main method as follows:

```
DerivedClass newClass = new DerivedClass();
```

which of the following calls in main would be invalid:

```
int a = newClass.getNum();
newClass.setNum(56);
newClass.setArg("happy happy");
newClass.appendArg("joy joy");
String newStr=newClass.getArg();
```

Solution:

```
newClass.appendArg("joy joy");
// you cannot access private methods or variables of a base class
// within a derived class
```

11)

(method overloading)

what's wrong here?

```
import java.util.*;
public class Area{

    double area(int s1, double s2)
    {
        return(s1*s2);
    }

    double area(int s1, int s2)
    {
        return(s1*s2*1.0);
    }

    double area(double r)
    {
        return(Math.PI*r*r);
    }

    int area(int s1, int s2)
    {
        return(s1*s2);
    }
}
```


lecture11solutions

```
}
```

Solution:

```
int area(int s1, int s2)
{ // can't overload on return type only
  // this method and double area(int s1, int s2)
  // differ only in terms of their return type
  // so one of them has to go
  return(s1*s2);
}
```

=====
12)

(exceptions)

what's the difference in output between a call to
double quot = divide1(5, 0);
and
double quot = divide2(5, 0);

```
public double divide1(int numerator, int denominator)
{
  double quotient = SENTINEL;
  try
  {
    if (denominator == 0)
      throw new DivideByZeroException();
    quotient = numerator/(double)denominator;
    System.out.println(numerator + "/"
                      + denominator
                      + " = " + quotient);
  }
  catch(DivideByZeroException e)
  {
    System.out.println(e.getMessage());
    System.exit(0);
  }
  return(quotient);
}

public double divide2(int numerator, int denominator)
{
  double quotient = SENTINEL;
  try
  {
    if (denominator == 0)
      throw new DivideByZeroException();
    quotient = numerator/(double)denominator;
    System.out.println(numerator + "/"
                      + denominator
                      + " = " + quotient);
  }
  catch(DivideByZeroException e)
  {
    System.out.println(e.getMessage());
  }
  finally {
    return(quotient);
  }
}
```

Solution:

divide1 exits after printing out the error message,
Page 9

lecture11solutions

divide2 prints the error message and then returns the value SENTINEL when the finally block is executed.

13)
(command line input to a program)

a) If I've defined a java class TestNew and compiled it, what is the value of args.length in main after I type in the following command?:

```
java TestNew input.txt output.txt
```

b) what would the values stored in args be?

Solution:

a)

```
args.length = 2
```

b)

```
args[0] = "input.txt"  
args[1] = "output.txt"
```

14)

We want to copy the contents of one file to another in the class defined below. what expression should go in the parentheses after the while statement. what statements should occur in the body of the while loop? what happens when a FileNotFoundException occurs in main given the class definition below (if for example, input.txt cannot be found)?:

```
public class CopyFile{  
    public static void main(String[] args)  
    {  
        try {  
            FileReader fr = new FileReader("input.txt");  
            FileOutputStream fw = new FileOutputStream("output.txt");  
            BufferedReader in = new BufferedReader(fr);  
            PrintWriter out = new PrintWriter(fw);  
            String getLine = in.readLine();  
            while (          ) {  
  
                }  
                in.close();  
                out.close();  
            } catch (Exception e2) {  
                System.err.println("An exception occurred");  
                return;  
            } catch (IOException e1) {  
                System.err.println("An IO exception occurred");  
                return;  
            } catch (FileNotFoundException e) {  
                System.err.println("File not found!");  
                return;  
            }  
        }  
    }  
}
```

Solution:

In the body of the loop we want to copy the line we've

lecture11solutions

read from the input file to the output file, and read another line (we check whether we have reached the end of the input file by testing if `getLine == null` in the while loop condition).

The error message "An exception occurred" is printed. To point out that it is a `FileNotFoundException`, we would need to re-order the catch-blocks so that the more specific exceptions like `FileNotFoundException` and `IOException` come before the more general one, `Exception`.

```
public class CopyFile{
    public static void main(String[] args)
    {
        try {
            FileReader fr = new FileReader("input.txt");
            FileOutputStream fw = new FileOutputStream("output.txt");
            BufferedReader in = new BufferedReader(fr);
            PrintWriter out = new PrintWriter(fw);
            String getLine = in.readLine();
            while ( getLine != null ) {
                out.println(getLine);
                getLine = in.readLine();
            }
            in.close();
            out.close();
        } catch (FileNotFoundException e) {
            System.err.println("File not found!");
            return;
        } catch (IOException e1) {
            System.err.println("An IO exception occurred");
            return;
        } catch (Exception e2) {
            System.err.println("An exception occurred");
            return;
        }
    }
}
```