ECU178 Computer Science: 210CT - Programming, Algorithms and Data Structures Portfolio

Due on Monday, December 15th, 2014

Dr James Shuttleworth

Robert Rigler: 4939377

Contents

Item 1: Week 3 - Linear Search and Duplicate Finder	3
Pre-Homework 1: Write a Program that displays your name 10 times	3
Pre-Homework 2: Write a function that draws a square of stars given as a parameter	5
Pre-Homework 3: Write a program to open a file and display it's contents in capitals	7
1. Pseudocode for linear search	8
2. Pseudocode for finding duplicates in a list	8
Item 2: Week 4 - Time complexities and Big-O notation	9
1. Describe the runtime bounds of the linear search algorithm	Ć
2. Describe the runtime bounds of the duplicate finder algorithm	Ć
Additional work: Critical values of relative runtimes	10
Item 3: Week 6 - Harmonic Series or Pivot Selection	12
Pre-Homework 1.: 3-Bit binary number	12
Pre-Homework 2.: Boolean sequence binary number	12
1. Harmonic Series (Pseudocode)	13
2. Harmonic Series (JAVA Implementation)	14
Item 4: Week 7 - Heapworksheet or RPN Calculator	16
Item 5. Week 8 - Linked List Delete function or Linked List Sortings	17

Item 1: Week 3 - Linear Search and Duplicate Finder

Pre-Homework 1: Write a Program that displays your name 10 times

Listing 1: NameReapeat class JAVA code

```
/**
1
    * Created by Rob on 23/10/2014.
2
  public class NameRepeat {
5
       public static void main(String[] args) {
           NameRepeat myObject = new NameRepeat(); /*Create Object*/
        /*Use object to call PrintName() method*/
10
           myObject.PrintName("Rob");
11
12
13
       public void PrintName(String _name) {
14
15
           for (int i = 0; i<10;i++) { /* Loop 10 times*/</pre>
16
        /*, print the number && _name parameter each time.*/
                System.out.println((i+1) + " " + _name);
19
21
22
23
24
25
26
```



Pre-Homework 2: Write a function that draws a square of stars given as a parameter

Listing 2: StarSquare class JAVA code

```
* Created by Rob on 23/10/2014.
2
  public class StarSquare {
5
        //Create variable to hold asterisk character.
       public char ast = '*';
       public static void main(String[] args) {
           StarSquare sSquare = new StarSquare(); /*Create Object*/
10
11
        /*Use object to call writeSquare() method*/
12
           sSquare.writeSquare(10);
13
14
15
       }
16
17
       public void writeSquare(int size){
18
19
            for (int i =0; i < size; i++) { /*OuterLoop 'size' times*/</pre>
20
                for (int j = 0; j < size; j++) { /*InnerLoop 'size' times*/</pre>
21
22
                    System.out.print(ast); /*Print line of asterisks*/
24
                }
25
              /* Start new line when inner loop finishes*/
27
                System.out.println();
28
           }
30
31
```


Pre-Homework 3: Write a program to open a file and display it's contents in capitals

Listing 3: RtoCaps class JAVA code

```
* Created by Rob on 23/10/2014.
2
    */
  import java.io.File;
  import java.io.FileNotFoundException;
5
  import java.util.Scanner;
  public class RtoCaps {
8
   public static void main(String[] args)throws FileNotFoundException
10
11
           File inFile = new File("input.txt");
12
                    /*Create a file object */
13
           RtoCaps obj = new RtoCaps(); /*Create class object*/
15
        /*Use Class object to call rInput() method*/
16
           obj.rInput(inFile);
17
18
19
       public void rInput(File inFile) throws FileNotFoundException{
20
21
        /*Create a new scanner to read from the file*/
22
           Scanner in = new Scanner(inFile);
24
        /*Loop WHile there is still lines left in the document*/
25
           while (in.hasNextLine())
26
           {
27
             /* Place the next line in a strin varibale*/
28
               String line = in.nextLine();
30
              /* Print the line in uppercase*/
31
               System.out.println(line.toUpperCase());
32
33
34
35
36
37
38
```

Evidence bash - "riglerr-univer × + riglerr@university-work:~/workspace/210CT_Programming/Portfolio/Item_1/com.Pre-Homework/src (master) \$ java RtoCaps HELLO WORLD! FROM INPUT.TXT riglerr@university-work:~/workspace/210CT_Programming/Portfolio/Item_1/com.Pre-Homework/src (master) \$ |

1. Pseudocode for linear search

```
Algorithm 1 LinearSearch

1: procedure BOOL LINEARSEARCH(item, list[])

2: for each element i in list do

3: if list[i] = item then

4: return true

5: end if

6: end for
```

2. Pseudocode for finding duplicates in a list

Algorithm 2 Examining for duplicates

7: return false8: end procedure

```
    procedure BOOL EXFORDUPES(list[])
    for each element i in list[] do
    for each element j in list[] do
    if list[i] = list[j] then
    return true
    end if
    end for
    end procedure
```

Item 2: Week 4 - Time complexities and Big-O notation

1. Describe the runtime bounds of the linear search algorithm

```
Algorithm 3 LinearSearch
1: procedure BOOL LINEARSEARCH(item, list[])
      for each element i in list do
                                       (n)
3:
          if list[i] = list then t
                                       (n)
4:
            return true
                                       (n)
          end if
6:
      end for
8: return false
                                       (1)
9: end procedure
```

The time complexity of the algorithm is O(n)

2. Describe the runtime bounds of the duplicate finder algorithm

```
Algorithm 4 Examining for duplicates
 1: procedure BOOL EXFORDUPES(list[])
       for each element i in list[] do
 2:
                                                 (n)
          for each element j in list[] do
                                                 (n*n)
 3:
             if list[i] = list[j] then
                                                 (n*n)
 4:
            return true
                                                 (n*n)
 5:
             end if
 6:
 7:
          end for
       end for
 8:
                                                (1)
 9: return false
10: end procedure
```

The time complexity of the algorithm is $O(n^2)$

Additional work: Critical values of relative runtimes

Robert Rigler: 4939377

Write a function that determines the critical value at which the relative runtime of two linear algorithms swap.

Page 10 of 17

Algorithm 5 Relative runtime comparison algorithm

```
1: procedure CRITVAL(m1, k1, m2, k2)
       switch \leftarrow false
 2:
       n \leftarrow 0
 3:
 4:
       // Which Expression has a greater value for n=0
 5:
       if (((m1*n) + k1) > ((m2*n) + k2)) then
 6:
 7:
          //While Expression 1 (m1, k1) is greater than Expression 2(m2, k2), do:
 8:
          while !switch do
 9:
10:
              // If Expression 1 become less than Expression 2 for that value of n
11:
12:
              if (((m1*n) + k1) < ((m2*n) + k2)) then
13:
                  //switch becomes true, which exits the loop and both if statements
14:
                  switch \leftarrow true
15:
              else
16:
                  n + +
17:
              end if
18:
          end while
19:
20:
       else
21:
22:
          //While Expression 2 (m1, k1) is greater than Expression 1(m2, k2), do:
23:
          while !switch do
24:
25:
              // If Expression 2 become less than Expression 1 for that value of n
26:
              if (((m1*n) + k1) > ((m2*n) + k2)) then
27:
28:
                  //switch becomes true, which exits the loop and both if statements
29:
                  switch \leftarrow true
30:
              else
31:
                  n + +
32:
33:
              end if
          end while
34:
35:
       end if
36:
37:
       // Return the value of n at which either while loop was fulfilled.
38: return n
39:
40: end procedure
```

Item 3: Week 6 - Harmonic Series or Pivot Selection

Pre-Homework 1.: 3-Bit binary number

Write a function that takes 3 boolean parameters, a, b and c and returns an integer value they represent if they are the three bits of a three-bit number, with a being the most significant and c being the least.

Algorithm 6 3-Bit Binary Number

Robert Rigler: 4939377

```
1: procedure PRE1(bool a, bool b, bool c)
        total \leftarrow 0
 2:
        if a then
 3:
 4:
            total \leftarrow total + 1
        end if
 5:
        if b then
 6:
            total \leftarrow total + 2
 7:
 8:
        end if
        if c then
 9:
            total \leftarrow total + 4
10:
        end if
11:
12: return total
13: end procedure
```

Pre-Homework 2.: Boolean sequence binary number

Write a function that takes a sequence of values of any given length and returns the integer value they represent.

Algorithm 7 Return integer value from list a of boolean value

```
    procedure PRE2(bool list[])
    total ← 0
    len ← LengthOf.list[]
    for i ← len to 0 do
    if list[i] = true then
    total ← total + 2i
    end if
    end for
    end procedure
```

1. Harmonic Series (Pseudocode)

Robert Rigler: 4939377

Use pseudocode to specify a recursive algorithm to compute the nth value of the harmonic series, for some integer n.

Algorithm 8 Computing nth value of harmonic series

```
procedure HARM(float t, float n)

while n > 0 do

t \leftarrow t + (1/n)

n - -

HARM(t, n)

end while

return t

end procedure
```

2. Harmonic Series (JAVA Implementation)

The Harmonic Series computation algorithm implemented in Java

Listing 4: harms java class file

```
public class harms{
2
       public static void main(String[] args) {
3
           System.out.println(f(0,3));
5
       public static float f(float t, float n) {
           t always has a value of 0 on the initial method call.
10
           n is the nth term, which decreases by 1 each recursive call
11
12
           When n = 0, stop recursive calling and return the value t.
13
           */
14
           while (n>0)
15
                t+= (1/n);
17
                f(t,--n);
18
           }
19
           return t;
20
22
23
24
```

Evidence of the Harmonic Series computation java implementation. The nth value passed to the method was 3.

Robert Rigler: 4939377

```
Terminal × +

riglerr@university-work:~/workspace/210CT_Programming/Portfolio/Item_3/6_Code (master) $ java harms

1.8333334

riglerr@university-work:~/workspace/210CT_Programming/Portfolio/Item_3/6_Code (master) $
```

Item 5: Week 8 - Linked List Delete function or Linked List Sortings

Robert Rigler: 4939377

Page 17 of 17