COMPSCI 105: Principles of Computer Science
Summer 2007

Tutorial Five: Big-O & Sorting

Helen Gu and Slobodan Vukanovic
{ygu029, svuk002}@ec.auckland.ac.nz

This tutorial is not being assessed. It provides you with an opportunity to become familiar with concepts introduced in lectures.

Question One: Big-O

What is the Big-O complexity of each of the following methods?

```
public static void q1A(int n) {
   int count = 0;
   for(int i=0; i<20000; i++)
      count++;
}
```
O( )

```
public static void q1B(int n) {
   int count = 0;
   n = n/10;
   for(int i=0; i<n; i++)
      for (int j=0; j<100; j++)
         count++;
}
```
O( )

```
public static void q1C(int[] a) {
   int result = 0;
   int n = a.length;
   for(int i=0; i<n; i++)
      result = result + a[i];
}
```
O( )

```
public static void q1D(int[][] a) {
   int result = 0;
   int n = a.length;
   for (int i=0; i < n; i++) {
      for (int j=0; j < i; j++)
         result = result + a[i][j];
   }
}
```
O( )

```
public static void q1E(int n) {
   int count = 0;
   for(int i=0; i<n; i++)
      count++;
   for (int j=0; j<n; j++)
      count++;
}
```
O( )

```
public static void q1F(int[] a) {
   int result = 0;
   int n = a.length;
   for (int i=0; i < n; i++)
      result = doG( a[i]);
}
private static int doG(int a) {
   int result = 0;
   for (int i=0; i < a; i++)
      result += i;
   return result;
}
```
O( )

Question Two: Sorting

a) Trace the **selection sort** as it sorts the following array into ascending order:
2 3 1 6 4 5

b) Trace the **insertion sort** as it sorts the following array into ascending order:
2 3 1 6 4 5

c) Trace the **bubble sort** as it sorts the following array into ascending order:
2 3 1 6 4 5

Question Three: Bubble Sort

The following is a list of birthdays in DD/MM format. Trace the **bubble sort** of the list into ascending order by month; for the cases of more than one birthday within a month, please further sort them by day.

20/09
12/01
12/03
25/12
01/01

Question Four: Merge Sort

Given the source code for the method merge() – see also file: `T05Q3.java`. Write the recursive method of Merge sort (in `T05Q3.java`).

```
private void merge(int[] a, int start, int mid, int end) {
      int temp[] = new int[end-start];
      int i = start;
      int j = mid;
      for(int k = 0; k < temp.length; k++) {
            if (j >= end || i < mid && a[i] <= a[j])
                  temp[k] = a[i++];
            else
                  temp[k] = a[j++];
      }
      for(int k = start; k < end; k++)
            a[k] = temp[k-start];
}
public static void mergeSort (int[] a, int start, int end) {
      /* please complete this method */
```