2016 Fall CS 401 FINAL WRITTEN EXAM 2016-12-14 (instructor: Tim Hoffman)

YOUR NAME _____

YOUR PITT EMAIL ADDRESS _____

EXAM DATE: Wednesday December 14th 2016 12pm-1:50pm POSVAR Hall

6pts

Q-1: In your Lab#3 assignment you wrote a resize method that allocated new array 2 X the length of the original array and then copied all the values from the smaller array into the larger one.

The big O cost of that operation is:

A: O(1)
B: O(log2 N)
C: O(N)
D: O(N squared)

Q-2: 6pts In your Lab#4 you searched for the first occurrence of a duplicate in an array of Strings. You sorted the array using Arrays.sort(). After the array was sorted, In order to find the index of the first duplicate, you did which of the following operations:

A: a binarySearch on each element of the array
B: an insertInOrder operation on each element of the array
C: a scan from left to right looking at each pair of elements at [i] and [i+1]
D: an indexOf() operation on each element of the array

Q-3: 6pts In your Lab#5 you performed the same task, but this time you stored the Strings into a HashSet. Why were you not asked to report the index position of the first duplicate in the set?

A: HashSet does not require elements to be stored anywhere

B: Retrieving the index of an element is an expensive operation i.e. O(N)

C: Retrieving the index of an element is an expensive operation i.e. O(log2 N)

D: The duplicate element will never get stored in the set and thus has no index in the set

Q-4: 6pts In your project#3 generated an array of counters to record how many words of length 1

,2 3 etc were read in from a text file. The .length of the array of counters started out at zero and was upsized only as needed so that a word of length() i would cause histogram[I] to be incremented. Assuming in input file containing 1,000 words, what would be maximum possible number of resize operations needed to correctly store all the length counts? **Assume the longest word in the file is 28 chars long**.

A: 1,000 B:
1

6pts

C: 28

D: log 2 of 1,000

Q-5: In your project#4 you wrote an insertInOrder operation that adds an element into the array by placing it in its proper sorted position. The two operations that had to be done on each element in the array were a comparison operation that compared the new incoming element against elements already in the array – and the second operation is a copy operation that copies an element from [i] to [i+1] in order to make room for the new element. You also wrote a binary search method to make your insertInOrder a bit more efficient. Which of the following statements best describes HOW/WHY that binary search makes your program do the job faster?

A: the binary search reduces the number of copy operations

B: the binary search reduces the number comparison operations

C: the binary search reduces the complexity from N squared down to O(N)

D: the binary search does nothing in any way to improve performance

Q-6: 6pts In your project#6 you wrote a Fraction class that represents a rational number using an int for the numerator and an int for denominator. Which statement below best illustrates what you are trying to prevent when you declared numer and denom as private?

A: Fraction f = new Fraction(22, 0);

B: f.setDenom(0) ;

C: f.denom = 0;

D: this(22, 0);

Q-7: 6pts In your project#8 the Swamp you used backtracking and recursion to find all possible paths from a given point of origin to the edge of the grid. Assuming a 4x 4 grid what would the maximum depth of the call stack if you are only counting calls to your recursive method?

A: 16

B: 4

C: 2 to the 16^{th} power

6pts

D: none of the above

Q-8: 6pts In your project#9 - Boggle , you again used backtracking and recursion to find all possible strings inside the grid. Assuming a 4x 4 grid what would the maximum depth of the call stack if you are only counting calls to your recursive method?

A: 16

B: 4

C: 2 to the 16th power

D: None of the above

Q9: Here is a recursive method that you must trace. Below the code, select the answer that best describes the output or behavior of the method.

```
1  import java.io.*;
2  public class Recursion
3  {
4      public static void main( String[] args )
5      {
6          System.out.println(  mystery( 7532 ) );
7
8      } // END MAIN
9      static int mystery( int n )
10     {
11         if (n==0) return 0;
12         return n - mystery( n/10 );
13     }
14 } // END RECURSION
```

A: 6699

B: 7

C: crashes because it underflows negative D:

none of the above

Q-10: 6pts Here is a recursive method that you must trace. Below the code, select the answer that best describes the output or behavior of the method.

6pts

```
1  import java.io.*;
2  public class Recursion
3  {
4      public static void main( String[] args )
5      {
6          System.out.println(  Q10( "fooBar", 0 ) );
7      } // END MAIN
8      static boolean Q10( String s, int i )
9      {   if (i == s.length()) return true;
10         return Character.isLowerCase(s.charAt(i)) &&
11                Character.isLowerCase(s.charAt(i+1));
12     }
13 } // END RECURSION
```

A: true
B: false
C: crashes

Q-11: Trace the code of the method below and then select the choice that most accurately describes its output or behavior.

```
1  import java.io.*;
2  public class Final
3  {
4      public static void main( String[] args )
5      {
6          int[] arr  =  { 1,2,3,4,5 };
7          for (int i=0 ; i < arr.length ; ++i)
8              for ( int j=0 ; j < i ; ++j )
9                  System.out.print( arr[i] + " ");
10         System.out.println();
11     } // END MAIN
12 }
```

A: 1 2 3 4 5 4 3 2 1

B: 5 4 3 2 1 2 3 4 5

C: 5 5 4 4 3 3 2 2 1

D: 2 3 3 4 4 4 5 5 5 5

6pts

Q-12: 6pts Looking at the code above, what is the runtime complexity with respect to the input length N.

A: O(1)
B: O(N)
C: O(N log2N)
D: O(N squared)
E: O(2 to the N)

Q-13: 6pts Based on the following scenario, which of the four choices is the most efficient approach?

Scenario: You are the software architect for the one world government's social security department in the next decade. The Earth's population about 8 billion. You have a single point website that receives messages about births and deaths in real time as they occur. In coming updates of new SSNs (a birth) or requests to remove an SSN (a death) come at a rate of thousands per second. On your server is the collection of all social security numbers. With each SSN there is also a large collection of data associated with that person. It is your job to make the decision about what type of data structure to store this information in and maintain and process the updates in strict real time. The most common operations are adding/removing a SSN. Sorting and ordering the data is not important.

A: Use a pair of arrays. The first array stores the SSN at index [i]. The 2nd array stores a reference to the associated database at it's index [i].
B: Use a HashMap where the key is the SSN and the value is a reference to that person's database. C: Use a TreeSet with key and value the same as the HashSet D: Use a pair of ArrayLists rather than plain arrays.

Q-14: 6pts You are given a text file that represents a mapping of auto makers to some of their models.

FORD F150 TAURUS MERCURY FIESTA
GM SIVERADO BLAZER SUBURBAN
SUBARU FORESTER OUTBACK IMPREZZA

Your job is to build a map that is the inverse of the above map. You must build a map that has each specific model of car/truck as the key and the associated value being the manufacturer.

Can this be accomplished by processing (read then store) each token only ONCE?

A: YES. You only need to read and store each token once.
B: NO. You must make more than one pass or process some tokens more than once.

Q-15 6pts What is the output of this program?

```
1  import java.io.*;
2  public class Final
3  {
4      public static void main( String[] args )
5      {
6          int[][] grid  =  new int[3][3];
7          for (int r=0 ; r < grid.length ; ++r)
8              for (int c=0 ; c < grid.length ; ++c)
9                  grid[r][c] = r*c;
10
11         for (int r=0 ; r < grid.length ; ++r)
12         {   for (int c=0 ; c < grid.length ; ++c)
13                 System.out.print( grid[r][c] + " " );
14             System.out.println();
15         }
16     } // END MAIN
17 }
```

A: 0 0 0
0 1 2
0 2 4

B: 0 1 2

0 2 4

0 3 6

C: 0 2 4

0 3 6

0 2 4

Q-16: 5pts Why is it inefficient to use recursion to solve a problem that you could have solved with a loop?

A: you will write more code with recursion and therefore it is harder to debug and maintain

B: recursion uses more memory on the call stack making copies of the method C:

recursion generates less efficient machine instructions

Q-17: 1pt Hashing figures out where to put a new value into an array without having to compare the incoming value to the multiple other elements already in the array that stores the values.

A: TRUE

B: FALSE

Q-18: 1pt Hashing requires O(1) time to place a value into the array like structure.

A: TRUE

B: FALSE

Q-19: 1pt Assuming that hashing is being done on strings, the hashing algorithm is O(1) with respect to the length of the string.

A: TRUE

B: FALSE

Q-20: 1pt Hashing requires that all the elements in the hash table or set have a test for equality defined on them so that any element in the hash map/set can be tested for equality against any new incoming value.

A: TRUE

B: FALSE

CONTINUED ON NEXT PAGE

Q-21: 1pt HashMap requires that the there are no duplicates in the value (the 2nd column) fields.

A: TRUE
B: FALSE

Q-22: 1pt Hashing figures out where to put a new value into an array without having to compare the incoming value to the multiple other elements already in the array.

A: TRUE
B: FALSE