AP Computer Science **Unit 6 Review**

Part I: Multiple Choice (15 questions / 4 points each)

1) What would an array look like after 3 iterations of the outer loop of the Selection Sort?
2) What would an array look like after 3 iterations of the outer loop of the Insertion Sort?
3) Why are MergeSort & QuickSort better sorting algorithms than the Selection Sort & Insertion Sort?
4) Be able to identify a rewrite of one of the 6 algorithms we studied this unit.
5) Name the necessary precondition for a Binary Search.
6) Given a sorted array and a target, determine how many comparisons are necessary to find the target using the Binary Search.
7) Know the Big O notations for each of the 6 algorithms we studied this unit.
8) Given 3 of the 6 algorithms we studied this unit, be able to list them from fastest to slowest. (Size of Array will be given)
9) Under what conditions is a Sequential Search (faster/slower) than a Binary Search?
10) Which of the 6 algorithms we studied use a "divide & conquer" approach (continually split the array in half)?
11) Given the MergeSort algorithm, determine how many calls to the method sort are required to completely sort an array of length x.
12) Given a sorted array and a target, determine how many comparisons are necessary to determine the target is not in the array using the Binary Search.
13) Be able to identify a rewrite of one of the 6 algorithms we studied this unit.
14) The Sequential Search I gave you uses a For loop. Be able to rewrite it using a While loop. Know short circuit evaluation.
15) What criteria should be considered when choosing a sorting algorithm?

Samples:

1) What would the following array look like after 3 iterations of the outer loop if you were using the Selection Sort?

63 19 25 15 75 24 5

a) 15 19 25 63 75 24 5
b) 5 19 25 15 75 24 63
c) 5 15 25 19 75 24 63
d) 5 15 19 25 75 24 63
e) 5 15 19 24 75 25 63

2) What would the following array look like after 3 iterations of the outer loop if you were using the Insertion Sort?

19 2 1 12 38 44 21

a) 2 19 1 12 38 44 21
b) 1 2 19 12 38 44 21
c) 1 2 12 19 38 44 21
d) 1 2 12 19 21 38 44
e) 1 2 12 19 21 44 38

3) Consider the following sorted array of integers:

a[0]	a[1]	...	a[55]	...	a[99]
79	124		518		927

If 518 is the target value for a Binary Search, how many iterations of the algorithm are necessary to find it?

a) 3 b) 4 c) 5 d) 6 e) 7

4) Consider the following array of sorted integers:

a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]
12	18	29	33	46	57	63	74	81	95

A Binary Search will be performed on this array. How many iterations of this algorithm will be required to determine that 36 is not in the list?

a) 1 b) 4 c) 5 d)7 e) can't be determined because of an infinite loop

5) Which of the following algorithms is matched up with it's correct Big O notation?

a) Selection Sort O(log n)
b) Insertion Sort O(n)
c) MergeSort O(n*log n)
d) Sequential Search $O(n^2)$
e) Binary Search O(n*log n)

Part II: Free Response

```
public class Sorter
{
      private int[] list;

      public Sorter(int[] x)
      {
            ...
      }
      public void selectionSort()
      {
            ...
      }
      public void insertionSort()
      {
            ...
      }
      pubic void mergeSort()
      {
            ...
      }
      public void quickSort()
      {
            ...
      }
}
```

```
public class Searcher
{
      private int[] list;

      public Searcher(int[] x)
      {
            ...
      }
      public boolean seqSearch(int t)
      {
            ...
      }
      public boolean binSearch(int t)
      {
            ...
      }
}
```

A programmer was asked to determine if a specific 4-digit integer was located in an unsorted array of 10,000 4-digit integers. This programmer was not sure which of the following two methods should be implemented.

I. Do a Sequential Search of the array
II. Do a MergeSort on the array followed by a Binary Search

Since the programmer was unaware of Big O notation, he decided to time each method in milliseconds.

Write the code that will time each method and print the results, use the Sorter and Searcher classes from above. The number you are searching for is 1234. The output should look like this...

Method #1: Found = true Time = ?????

Method #2: Found = true Time = ?????

```
import java.util.Random;

public class FreeResponse
{
      private int[] ary = new int[10000];

      public FreeResponse()
      {
            fillArray();









      }
      public static void main(String[] args)
      {
            FreeResponse app = new FreeResponse();
      }
      private void fillArray()
      {
            Random generator = new Random();

            for (int j=0; j<ary.length; j++)
            {
                  ary[j] = generator.nextInt(9000) + 1000;
            }
      }
}
```

Using your knowledge of Big O notation show how this programmer could have determined the answer to this dilemma without writing code to test each method. I want to see mathematical work!

Java Concepts Review Assignment:

Pages 659 – 660 Exercises: R14.2, R14.5, R14.6