Printout

Tuesday, April 20, 2021 1:11 PM

Name: _____ Period #: _____

AP Computer Science
Practice #38

1. Array unsortedArr contains an unsorted list of integers. Array sortedArr contains a sorted list of integers. Which of the following operations is more efficient for sortedArr than unsortedArr? Assume the most efficient algorithms are used.

   I. Inserting a new element
   II. Searching for a given element
   III. Computing the mean of the elements

   a. I only b. II only c. III only d. I and II only e. I, II, and III

Questions 2-3 refer to the insertionSort method and the private instance variable a, both in a Sorter class.

```
private Integer[] a;

/**Precondition: a[0], a[1]…a[a.length – 1] is an unsorted array of Integer objects
   Postcondition: Array a is sorted in descending order */

public void insertionSort()
{       for(int i=1; i < a.length; i++)
        {
                Integer temp = a[i];
                int j = i – 1;
                while( j >= 0 && temp.compareTo(a[j]) > 0)
                {       a[ j + 1] = a[ j ];
                        j--;
                }
                a[j+1] = temp;
        }
}
```

2. An array of Integer is to be sorted biggest to smallest using the insertionSort method. If the array originally contains: 1 7 9 5 4 12

   What will it look like after the third pass of the for loop?

   a. 9 7 1 5 4 12
   b. 9 7 5 1 4 12
   c. 12 9 7 1 5 4
   d. 12 9 7 5 4 1
   e. 9 7 12 5 4 1

1 7 9 5 4 12
1 7 1 9 5 4 12
2 9 7 1 5 4 12
3 9 7 5 1 4 12

3. When sorted biggest to smallest with insertionSort, which list will need the fewest changes of position for individual elements?

   a. 5, 1, 2, 9, 4, 9
   b. 9, 7, 1, 4, 3, 2
   c. 9, 4, 2, 5, 1, 3
   d. 9, 3, 5, 1, 4, 2
   e. 3, 2, 1, 9, 5, 4

954321

4. Which of the following is a valid reason why mergesort is a better sorting algorithm than insertion sort for sorting long, randomly ordered lists?

   I. Mergesort requires less code than insertion sort.
   II. Mergesort requires less storage space than insertion sort.
   III. Mergesorts runs faster than insertion sort.

   a. I only
   b. II only
   c. III only
   d. I and II only
   e. II and III only

5. The decision to choose a particular sorting algorithm should be made base on

   I. Run-time efficiency of the sort
   II. Size of the array
   III. Space efficiency of the algorithm

   a. I only
   b. Ii only
   c. III only
   d. I and II only
   e. I, II, and III

6. An algorithm for searching a large sorted array for a specific value x compares every third item in the array to x until it finds one that is greater than or equal to x. When a larger value is found, the algorithm compares x to the previous two items. If the array is sorted in increasing order, which of the following describes all cases when this algorithm uses fewer comparisons to find the x than would a binary search?

   a. It will never use fewer comparisons.
   b. When x is in the middle position of the array
   c. When x is very close to the beginning of the array
   d. When x is very close to the end of the array
   e. When x is not in the array

7. A large array of lowercase characters is to be searched for the pattern "pqrs". The first step in a very efficient searching algorithm is to look at characters with index

   a. 0, 1, 2, …until a "p" is encountered
   b. 0, 1, 2, ….until any letter in "p"…"s" is encountered
   c. 3, 7, 11, …until an "s" is encountered.
   d. 3, 7, 11, … until any letter in "p"…"s" is encountered
   e. 3, 7, 11,… until any letter other than "p"…"s" is encountered.