

Sorting Quiz

Question 1

Consider the following *algorithm*, which represents a particular sort routine. The algorithm will sort an integer array `a[]` into ascending order.

```
public void sorter (int [] a, int n) // n represents the
{                                     // number of data items
                                     // in the array

    int temp;
    do
    {
        for (int j = 1; j < n; j++)
        {
            if (a [j-1] > a[j])
            {
                temp = a[j-1];
                a[j-1]= a[j];
                a[j] = temp;
            }
        }
        n = n-1;
    }
    while (n>1);
}
```

Assume that `n` represents the number of elements in the array, and that the array `a[]` already contains the following 4 items:

`a[0] = 8, a[1] = 2, a[2] = 5, a[3] = 1`

- a. Complete the trace table showing the result of the first pass through this sort routine.

j	j < n ?	a[0]	a[1]	a[2]	a[3]
1	true	2	8	5	1
2					
3					
4					

- b. The choice of an algorithm for a particular process will often depend upon its efficiency in dealing with that process.
- Identify the characteristics of this algorithm that determine its efficiency.
 - Discuss the choice of this sort routine for a company that regularly sorts large amounts of data.

At the moment, this algorithm will not terminate early, even if the array becomes sorted.

- c. By making use of a Boolean variable, rewrite the algorithm including the code that will allow the algorithm to terminate if the array becomes sorted.

Consider an application which holds integer data, sorted in ascending order, in the array `b[]`. At times, a new value, `item`, is passed to the method `addItem()`, which places this value in its correct (sorted) position in this array.

You can assume that there are enough spaces in the array to hold the new data.

- d. Construct the method `addItem()`, that has been started below.

```
public void addItem (int[] b, int item, int n)
                        // n represents the number
                        // of elements in the array
```

Question 2

Consider the class SelectionSort shown below.

```
public class SelectionSort
{
    public static void main(String[] args)
    {
        int[] unsortedArray = {3,6,2,8,5};
        int[] sortedArray = selection(unsortedArray);
    }
    public static int[] selection(int[] list) // sorts array in ascending order
    {
        for (int x = 0; x < 4; x++)
        {
            int p = x;
            int smallest = list[p];
            for (int y = x + 1; y < 5; y++)
            {
                if (list[y] < smallest)
                {
                    p = y;
                    smallest = list[y];
                }
            }
            // end of 'y' loop
            if (p != x)
            {
                list[p] = list[x];
                list[x] = smallest;
            }
            // end of 'x' loop
        }
        return list;
    }
}
```

- Describe the information given in the following line.
public int[] selection(int[] list)
- State the values of the array list[] when the end of the 'x' loop is reached for the first time.
- Explain what would happen if the array unsortedArray[] was already sorted.
- Explain, without writing code, how the method selection() could be modified to sort an integer array of any size.