Name _____ Period_____

---

1. Consider the following code segment from an insertion sort program.

```
for (int j = 1; j < arr.length; j++)
{
    int insertItem = arr[j];
    int k = j - 1;
        while (k >= 0 && insertItem < arr[k])
        {
            arr[k + 1] = arr[k];
            k--;
        }
    arr[k + 1] = insertItem;
    /* end of for loop */
}
```

Assume that array arr has been defined and initialized with the values {5, 4, 3, 2, 1}.

What are the values in array arr after each pass of the for loop (i.e., when j = 1, when j = 2, when j = 3, when j = arr.length-1)?

[4, 5, 3, 2, 1]
[3, 4, 5, 2, 1]
[2, 3, 4, 5, 1]
[1, 2, 3, 4, 5]

---

2. Consider the following sort method. This method correctly sorts the elements of array data into increasing order.

```
public static void sort(int[] data)
{
    for (int j = 0; j < data.length - 1; j++)
    {
        int m = j;
        for (int k = j + 1; k < data.length; k++)
        {
            if (data[k] < data[m])
            /* Compare values */
            {
                m = k;
            }
        }
    int temp = data[m];
    /* Assign to temp */
    data[m] = data[j];
    data[j] = temp;
    /* End of outer loop */
    }
```

(a) Assume that sort is called with the array {6, 3, 2, 5, 4, 1}. What will the value of data be
after each pass of the outer loop (i.e., when j = 1, when j = 2, when j = 3, when j = arr.length-1)?

[1, 3, 2, 5, 4, 6]
[1, 2, 3, 5, 4, 6]
[1, 2, 3, 5, 4, 6]
[1, 2, 3, 4, 5, 6]
[1, 2, 3, 4, 5, 6]

(a) Assume that sort is called with the array {1, 2, 3, 4, 5, 6}. How many times will the expression indicated
by /* Compare values */ and the statement indicated by /* Assign to temp */ execute?

15
5

3. Given the following array as input, illustrate how the Mergesort algorithm performs. To illustrate the
Mergesort's behavior, start with the dividing of the array until the end condition of the recursive function is
met and then show how the merge is performed.

3 8 4 10 1 5 6 9
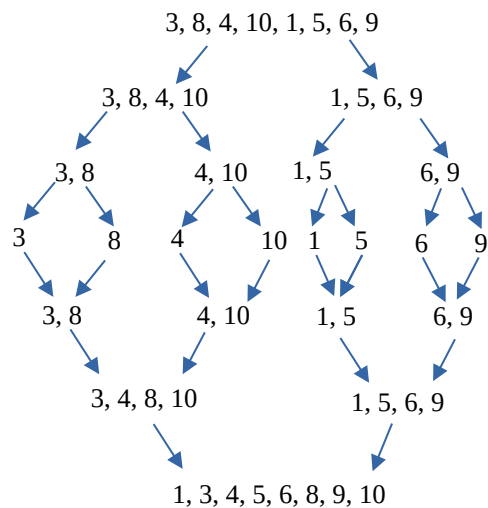
4. Given the following array [10, 5, 3, 9, 22, 24, 28, 27, ?] and assuming that Quicksort will be used to sort this array in ascending order,

(a) Select a value for the last element of the array (indicated by "?") such that the partitioning performed by Quicksort is most balanced. Explain why this makes Quicksort perform efficiently.

Any value between 11 and 22. This will ensure that approximately half the values will land on either side of the pivot.

(b) Show the results of the first two rounds of the Quicksort algorithms based on the number you have chosen.

For ? = 15

10, 5, 3, 9, 22, 24, 28, 27, 15 (pivot = 15)
10, 5, 3, 9, 15, 24, 28, 27, 22 (pivot moved. 1st partition complete.)
10, 5, 3, 9, 15, 24, 28, 27, 22 (pivot = 9)
3, 5, 10, 9, 15, 24, 28, 27, 22 (pivot = 9)
3, 5, 9, 10, 15, 24, 28, 27, 15 (pivot moved. 2nd partition complete.)