1. The \_**Linear**\_ search algorithm steps sequentially through an array, comparing each item with the search value.
2. The \_**Binary**\_ search algorithm repeatedly divides the portion of an array being searched in half.
3. The \_**Bubble**\_ search algorithm is adequate for small arrays but not large arrays.
4. The \_**Binary**\_ search algorithm requires that the array’s contents be sorted.
5. The *average* number of comparisons performed by linear search to find an item in an array of N elements is \_\_\_\_**N**\_\_.
6. The *maximum* number of comparisons performed by linear search to find an item in an array of N elements is \_\_\_\_**N**\_\_\_\_.
7. A linear search will find the value it is looking for with just one comparison if that value is stored in the \_\_\_**First**\_\_\_ array element.
8. A binary search will find the value it is looking for with just one comparison if that value is stored in the \_**Middle**\_ array element.
9. In a binary search, after three comparisons have been made, only \_\_\_\_**1/8**\_\_\_\_\_ of the array will be left to search.
10. The maximum number of comparisons that a binary search function will make when searching for a value in a 2,000-element array is \_**10.965784**\_.
11. If an array is sorted in \_**ascending**\_ order, the values are stored from lowest to highest.
12. If an array is sorted in \_**descending**\_ order, the values are stored from highest to lowest.
13. Bubble sort places \_\_**Two**\_\_ number(s) in place on each pass through the data.
14. Selection sort places \_\_**One**\_\_ number(s) in place on each pass through the data.
15. To sort N numbers, bubble sort continues making passes through the array until \_**Sorted**\_.
16. To sort N numbers, selection sort makes \_**One**\_ passes through the data.
17. Why is selection sort more efficient than bubble sort on large arrays?
    1. **Because repeated data swaps are often required to place a single item in its correct position.**
18. Which sort, bubble sort or selection sort, would require fewer passes to sort a set of data that is already in the desired order. **BUBBLE**