Lab 18

Insertion sort would be the best algorithm. With insertion sort less comparisons would have to be made than with selection sort and less swaps too. With insertion sort you would compare the number at the current position with the number before it and if it is greater than that number then you can keep it there and move on to the next position. With selection sort you have to compare the number at the position you are at with every number following that position and in some cases there will be no swap after all those comparisons. For example, because 1 is at the second to last position, with selection sort you would have compared every number with 4 just to swap 4 and 1 whereas with insertion sort you could have swapped those numbers after the first comparison. On the second run through of the array with selection sort, you would have compared the 1 in the second position with the rest of the numbers just to have it stay where it is. With insertion sort, you would have just swapped the 4 and 2 on the second round.