Testing for linearSearch of the IntSortSearch class

Num Comps = 1
In searching for 44 in the Array containing one element only, and the value being searched for is the element in the array, aka [44], 44 was found at position 0.
Num Comps = 1
In searching for -28 in the Array containing one element only, and the value being searched for is NOT the element in the array, aka [80], -28 was not found.
Num Comps = 1
In searching for -38 in the Array containing two unequal elements, and the value being searched for is one of the elements of the array, aka [-38, -60], -38 was found at position 0.
Num Comps = 1
In searching for -52 in the Array containing two unequal elements, and the value being searched for is NOT ANY of the elements of the array, aka [82, 73], -52 was not found.
Num Comps = 2
In searching for -47 in the Array containing 3 elements, and the value being searched for is the very first element of the array, aka [-47, 80, -33], -47 was found at position 0.
Num Comps = 2
In searching for 59 in the Array containing 3 elements, and the value being searched for is the very last element of the array, aka [-6, 26, 59], 59 was found at position 2.
Num Comps = 1
In searching for -26 in the Array containing 3 elements, and the value being searched for is the middle element, aka [18, -26, -56], -26 was found at position 1.
Num Comps = 2
In searching for 7 in the Array containing 3 elements, and the value being searched for is NOT ANY of the elements in the array, aka [-15, -59, -5], 7 was not found.

****

Num Comps = 1
In searching for -19 in the Array containing one element only, and the value being searched for is the element in the array, aka [-19], -19 was found at position 0.
Num Comps = 1
In searching for -25 in the Array containing one element only, and the value being searched for is NOT the element in the array, aka [-20], -25 was not found.
Num Comps = 1
In searching for 64 in the Array containing two unequal elements, and the value being searched for is one of the elements of the array, aka [64, -8], 64 was found at position 0.
Num Comps = 2
In searching for 5 in the Array containing two unequal elements, and the value being searched for is NOT ANY of the elements of the array, aka [-3, 49], 5 was not found.
Num Comps = 2
In searching for -27 in the Array containing 3 elements, and the value being searched for is the very first element of the array, aka [-27, 44, 32], -27 was found at position 0.
Num Comps = 2
In searching for 26 in the Array containing 3 elements, and the value being searched for is the very last element of the array, aka [-70, -51, 26], 26 was found at position 2.
Num Comps = 1
In searching for -75 in the Array containing 3 elements, and the value being searched for is the middle element, aka [41, -75, -17], -75 was found at position 1.
Num Comps = 2
In searching for 73 in the Array containing 3 elements, and the value being searched for is NOT ANY of the elements in the array, aka [70, -26, -63], 73 was not found.

****

Num Comps = 1

In searching for 33 in the Array containing one element only, and the value being searched for is the element in the array, aka [33], 33 was found at position 0.
Num Comps = 1
In searching for 60 in the Array containing one element only, and the value being searched for is NOT the element in the array, aka [4], 60 was not found.
Num Comps = 1
In searching for -24 in the Array containing two unequal elements, and the value being searched for is one of the elements of the array, aka [-24, -74], -24 was found at position 0.
Num Comps = 1
In searching for -47 in the Array containing two unequal elements, and the value being searched for is NOT ANY of the elements of the array, aka [94, -54], -47 was not found.
Num Comps = 2
In searching for 84 in the Array containing 3 elements, and the value being searched for is the very first element of the array, aka [84, -55, 51], 84 was not found.
Num Comps = 2
In searching for 72 in the Array containing 3 elements, and the value being searched for is the very last element of the array, aka [15, -81, 72], 72 was found at position 2.
Num Comps = 1
In searching for -2 in the Array containing 3 elements, and the value being searched for is the middle element, aka [86, -2, -8], -2 was found at position 1.
Num Comps = 2
In searching for 44 in the Array containing 3 elements, and the value being searched for is NOT ANY of the elements in the array, aka [-82, -11, 26], 44 was not found.

****

Testing for binarySearch of the IntSortSearch class

Num Comps = 0

Num Comps = 0
Num Comps = 2
Num Comps = 2
Num Comps = 5
Num Comps = 5
Num Comps = 5
Num Comps = 5
Num Comps = 1
In searching for -52 in the Array containing one element only, and the value being searched for is the element in the array, aka [-52], -52 was found at position 0.
Num Comps = 1
In searching for 10 in the Array containing one element only, and the value being searched for is NOT the element in the array, aka [60], 10 was not found.
Num Comps = 2
In searching for 68 in the Array containing two unequal elements, and the value being searched for is one of the elements of the array, aka [-82, 68], 68 was found at position 1.
Num Comps = 2
In searching for -33 in the Array containing two unequal elements, and the value being searched for is NOT ANY of the elements of the array, aka [-93, -8], -33 was not found.
Num Comps = 2
In searching for 65 in the Array containing 3 elements, and the value being searched for is the very first element of the array, aka [-44, 20, 65], 65 was found at position 2.
Num Comps = 1
In searching for 30 in the Array containing 3 elements, and the value being searched for is the very last element of the array, aka [-21, 30, 49], 30 was found at position 1.
Num Comps = 2
In searching for -69 in the Array containing 3 elements, and the value being searched for is the middle element, aka [-69, -50, -40], -69 was found at position 0.
Num Comps = 2

In searching for 48 in the Array containing 3 elements, and the value being searched for is NOT ANY of the elements in the array, aka [-63, -47, 95], 48 was not found.

****

Num Comps = 0
Num Comps = 0
Num Comps = 2
Num Comps = 2
Num Comps = 5
Num Comps = 5
Num Comps = 5
Num Comps = 5
Num Comps = 1
In searching for 79 in the Array containing one element only, and the value being searched for is the element in the array, aka [79], 79 was found at position 0.
Num Comps = 1
In searching for -100 in the Array containing one element only, and the value being searched for is NOT the element in the array, aka [18], -100 was not found.
Num Comps = 1
In searching for -76 in the Array containing two unequal elements, and the value being searched for is one of the elements of the array, aka [-76, 67], -76 was found at position 0.
Num Comps = 1
In searching for -11 in the Array containing two unequal elements, and the value being searched for is NOT ANY of the elements of the array, aka [45, 80], -11 was not found.
Num Comps = 2
In searching for 88 in the Array containing 3 elements, and the value being searched for is the very first element of the array, aka [-88, 58, 88], 88 was found at position 2.
Num Comps = 2

In searching for -66 in the Array containing 3 elements, and the value being searched for is the very last element of the array, aka [-66, -48, 21], -66 was found at position 0.
Num Comps = 2
In searching for 54 in the Array containing 3 elements, and the value being searched for is the middle element, aka [-12, 28, 54], 54 was found at position 2.
Num Comps = 2
In searching for -36 in the Array containing 3 elements, and the value being searched for is NOT ANY of the elements in the array, aka [16, 75, 90], -36 was not found.

****

Num Comps = 0
Num Comps = 0
Num Comps = 2
Num Comps = 2
Num Comps = 5
Num Comps = 5
Num Comps = 5
Num Comps = 5
Num Comps = 1
In searching for -80 in the Array containing one element only, and the value being searched for is the element in the array, aka [-80], -80 was found at position 0.
Num Comps = 1
In searching for 21 in the Array containing one element only, and the value being searched for is NOT the element in the array, aka [37], 21 was not found.
Num Comps = 1
In searching for -63 in the Array containing two unequal elements, and the value being searched for is one of the elements of the array, aka [-63, 62], -63 was found at position 0.
Num Comps = 2

In searching for 51 in the Array containing two unequal elements, and the value being searched for is NOT ANY of the elements of the array, aka [-31, 14], 51 was not found.
Num Comps = 2
In searching for 18 in the Array containing 3 elements, and the value being searched for is the very first element of the array, aka [18, 48, 86], 18 was found at position 0.
Num Comps = 2
In searching for -97 in the Array containing 3 elements, and the value being searched for is the very last element of the array, aka [-97, -80, -50], -97 was found at position 0.
Num Comps = 2
In searching for -71 in the Array containing 3 elements, and the value being searched for is the middle element, aka [-71, 3, 20], -71 was found at position 0.
Num Comps = 2
In searching for -87 in the Array containing 3 elements, and the value being searched for is NOT ANY of the elements in the array, aka [-95, 0, 32], -87 was not found.

****

Testing for selectionSort of the IntSortSearch class

Num Comps = 0
In sorting the Array containing one element only, aka [-67], the expected array is [-67]. The actual sorted array using the selectionSort method is [-67].
Num Comps = 2
In sorting the Array containing two unequal elements, aka [26, 3], the expected array is [3, 26]. The actual sorted array using the selectionSort method is [3, 26].
Num Comps = 5
In sorting the Array containing 3 elements, already sorted, aka [-73, -50, -11], the expected array is [-73, -50, -11]. The actual sorted array using the selectionSort method is [-73, -50, -11].
Num Comps = 5

In sorting the Array containing 3 elements, already sorted in the reverse order, aka [-7, -81, -87], the expected array is [-87, -81, -7]. The actual sorted array using the selectionSort method is [-87, -81, -7].
Num Comps = 5
In sorting the Array containing 3 elements, all of them negative, aka [-40, -52, -12], the expected array is [-52, -40, -12]. The actual sorted array using the selectionSort method is [-52, -40, -12].
Num Comps = 5
In sorting the Array containing 3 elements, one negative, one zero, one positive, aka [-67, 0, 10], the expected array is [-67, 0, 10]. The actual sorted array using the selectionSort method is [-67, 0, 10].
Num Comps = 5
In sorting the Array containing 3 elements, all of them the same value (like: 5,5,5), aka [99, 99, 99], the expected array is [99, 99, 99]. The actual sorted array using the selectionSort method is [99, 99, 99].

****

Num Comps = 0
In sorting the Array containing one element only, aka [65], the expected array is [65]. The actual sorted array using the selectionSort method is [65].
Num Comps = 2
In sorting the Array containing two unequal elements, aka [88, -98], the expected array is [-98, 88]. The actual sorted array using the selectionSort method is [-98, 88].
Num Comps = 5
In sorting the Array containing 3 elements, already sorted, aka [-62, -31, 53], the expected array is [-62, -31, 53]. The actual sorted array using the selectionSort method is [-62, -31, 53].
Num Comps = 5
In sorting the Array containing 3 elements, already sorted in the reverse order, aka [98, 81, -100], the expected array is [-100, 81, 98]. The actual sorted array using the selectionSort method is [-100, 81, 98].
Num Comps = 5
In sorting the Array containing 3 elements, all of them negative, aka [-35, -8, -44], the expected array is [-44, -35, -8]. The actual sorted array using the selectionSort method is [-44, -35, -8].

Num Comps = 5
In sorting the Array containing 3 elements, one negative, one zero, one positive, aka [-54, 0, 7], the expected array is [-54, 0, 7]. The actual sorted array using the selectionSort method is [-54, 0, 7].
Num Comps = 5
In sorting the Array containing 3 elements, all of them the same value (like: 5,5,5), aka [6, 6, 6], the expected array is [6, 6, 6]. The actual sorted array using the selectionSort method is [6, 6, 6].

****

Num Comps = 0
In sorting the Array containing one element only, aka [25], the expected array is [25]. The actual sorted array using the selectionSort method is [25].
Num Comps = 2
In sorting the Array containing two unequal elements, aka [49, 17], the expected array is [17, 49]. The actual sorted array using the selectionSort method is [17, 49].
Num Comps = 5
In sorting the Array containing 3 elements, already sorted, aka [-75, -17, 61], the expected array is [-75, -17, 61]. The actual sorted array using the selectionSort method is [-75, -17, 61].
Num Comps = 5
In sorting the Array containing 3 elements, already sorted in the reverse order, aka [7, -71, -91], the expected array is [-91, -71, 7]. The actual sorted array using the selectionSort method is [-91, -71, 7].
Num Comps = 5
In sorting the Array containing 3 elements, all of them negative, aka [-40, -34, -52], the expected array is [-52, -40, -34]. The actual sorted array using the selectionSort method is [-52, -40, -34].
Num Comps = 5
In sorting the Array containing 3 elements, one negative, one zero, one positive, aka [-79, 0, 75], the expected array is [-79, 0, 75]. The actual sorted array using the selectionSort method is [-79, 0, 75].
Num Comps = 5

In sorting the Array containing 3 elements, all of them the same value (like: 5,5,5), aka [21, 21, 21], the expected array is [21, 21, 21]. The actual sorted array using the selectionSort method is [21, 21, 21].

****