2.3. Problem 2: SelectSortApp.java

Are swap always needed?

I think that swaps are not always needed. If the element at the current position of the outer loop is already the minimum (or maximum) element in the unsorted portion of the array, then there's no need to swap it with itself. In other words, if the current element is already in its correct sorted position, there is no need to perform a swap operation.

2.5. Problem 4:

COPIES / COMPARISONS / SWAPS						
	Bubble Sort		Selection Sort		Insertion Sort	
	Comparisons	Swaps	Comparisons	Swaps	Comparisons	Swaps
10000	49994999	24898040	49995000	9999	24991732	24991732
15000	112492499	55671085	112492500	14999	56046176	56046176
20000	199989999	100424266	199990000	19999	99681057	99681057
25000	312487499	156301931	312487500	24999	156054211	156054211
30000	449984999	225363046	449985000	29999	225707534	225707534
35000	612482499	306053598	612482500	34999	306381568	306381568
40000	799979999	395696673	799980000	39999	398883105	398883105
45000	1012477499	506491922	1012477500	44999	504186538	504186538
50000	1249974999	625825839	1249975000	49999	627644514	627644514

The trend of the three algorithms:

- Bubble Sort:

  + Comparisons: The number of comparisons increases quadratically with the increase in input size. This is evident from the fact that the number of comparisons is approximately proportional to n^2, where n is the input size.

  + Swaps: The number of swaps also increases quadratically with the increase in input size. However, the number of swaps is slightly less than the number of comparisons, as not every comparison results in a swap.

- Selection Sort:

  + Comparisons: Similar to Bubble Sort, the number of comparisons increases quadratically with the increase in input size. It is also approximately proportional to n^2.

  + Swaps: The number of swaps remains relatively constant for Selection Sort, irrespective of the input size. This is because Selection Sort performs swaps only after finding the minimum element in each iteration, resulting in a fixed number of swaps.

- Insertion Sort:

  + Comparisons: The number of comparisons increases quadratically with the increase in input size, similar to Bubble Sort and Selection Sort.

+ Swaps: The number of swaps for Insertion Sort also increases quadratically with the increase in input size. However, it tends to be slightly lower compared to Bubble Sort and Selection Sort, as it involves fewer swaps in average cases.