Exercise: Quadratic Sorting Algorithms

Exercise A. Show the contents of the array after the **fourth iteration of the outer loop** for (i) Bubble Sort, (ii) Selection Sort, and (iii) Insertion Sort.

i. Bubble sort
ii. Selection sort
iii. Insertion Sort

50	15	17	30	25	11	26	12	73	21
(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)

Answer

i. Bubble sort

50	15	17	30	25	11	26	12	73	21	
15	50									Compared and swapped
	17	50								Compared and swapped
		30	50							Compared and swapped
			25	50						Compared and swapped
				11	50					Compared and swapped
					26	50				Compared and swapped
						12	50			Compared and swapped
							50	73		Compared, NOT swapped
								21	73	Compared and swapped
15	17	30	25	11	26	12	50	21	**73**	End 1st iteration
15	17									Compared, NOT swapped
	17	30								Compared, NOT swapped
		25	30							Compared and swapped
			11	30						Compared and swapped
				26	30					Compared and swapped
					12	30				Compared and swapped
						30	50			Compared, NOT swapped
							21	50		Compared and swapped
15	17	25	11	26	12	30	21	**50**	**73**	End 2nd iteration
15	17									Compared, NOT swapped
	17	25								Compared, NOT swapped
		11	25							Compared and swapped
			25	26						Compared, NOT swapped
				12	26					Compared and swapped
					26	30				Compared, NOT swapped
						21	30			Compared, NOT swapped
15	17	11	25	12	26	21	**30**	**50**	**73**	End 3rd iteration
15	17									Compared, NOT swapped
	11	17								Compared and swapped
		17	25							Compared, NOT swapped
			12	25						Compared and swapped
				25	26					Compared, NOT swapped
					21	26				Compared and swapped
15	11	17	12	25	21	**26**	**30**	**50**	**73**	End 4th iteration

ii. Selection sort

50	15	17	30	25	11	26	12	73	21	
50	15	17	30	25	11	26	12	73	21	Find smallest
11					50					Swap
11	15	17	30	25	50	26	12	73	21	End 1st iteration
	15	17	30	25	50	26	12	73	21	Find smallest
	12						15			Swap
11	12	17	30	25	50	26	15	73	21	End 2nd iteration
		17	30	25	50	26	15	73	21	Find smallest
		15					17			Swap
11	12	15	30	25	50	26	17	73	21	End 3rd iteration
			30	25	50	26	17	73	21	Find smallest
			17				30			Swap
11	12	15	17	25	50	26	30	73	21	End 4th iteration

iii. Insertion sort

50	15	17	30	25	11	26	12	73	21	
50	15	17	30	25	11	26	12	73	21	Start with left side sorted
	15									Hold 15,
50		17	30	25	11	26	12	73	21	Compare with 50
	50	17	30	25	11	26	12	73	21	Shift 50
15	50	17	30	25	11	26	12	73	21	Put 15 in place
15	50	17	30	25	11	26	12	73	21	End 1st iteration
		17								Hold 17
15	50									Compare with 50
15		50								Shift 50
15		50								Compare with 15 – no shift
15	17	50	30	25	11	26	12	73	21	Put 17 in place
15	17	50	30	25	11	26	12	73	21	End 2nd iteration
			30							Hold 30
15	17	50		25	11	26	12	73	21	Compare with 50
15	17		50	25	11	26	12	73	21	Shift 50
15	17		50	25	11	26	12	73	21	Compare with 17 – no shift
15	17	30	50	25	11	26	12	73	21	Put 30 in place
15	17	30	50	25	11	26	12	73	21	End 3rd iteration
				25						Hold 25
15	17	30	50		11	26	12	73	21	Compare with 50
15	17	30		50						Shift 50
15	17	30		50						Compare with 30
15	17		30	50						Shift 30
15	17		30	50						Compare with 17 – no shift
15	17	25	30	50						Put 25 in place
15	17	25	30	50	11	26	12	73	21	End 4th iteration

Exercise B. A sorting function is called to sort a list of 170 integers that have been read from a file. If all 170 values are zero, what would the running time (in terms of O-notation) for each of the algorithms below?

i. Bubble sort (flagged version) **O(n)**
ii. Selection sort **O(n^2)**
iii. Insertion sort **O(n)**

Exercise C. How many comparisons would be needed to sort an array containing 100 elements using **Selection sort**, if the original array values were already sorted? **4950**

Explanation:

- Go down the list to find the smallest number and swap it with the first element (99 comparisons)
- Do this again but starting from the second index (98 comparisons)
- Do this again but starting from the third index (97 comparisons)
- Do this again until you are left with the last 2 indices (1 comparison)
- So, you have made 99 + 98 + 97 + … + 1 comparisons

$$\frac{n(n+1)}{2} = \frac{99(99+1)}{2} = \frac{99*100}{2} = \frac{9900}{2} = \mathbf{4950}$$

Exercise D. The elements in a long list of integers are roughly sorted in decreasing order. No more than 5 percent of the elements are out of order. Which of the following is a valid reason for using an insertion sort rather than a selection sort to sort this list into decreasing order?

I. There will be fewer comparisons of elements for insertion sort.
II. There will be fewer changes of position of elements for insertion sort.
III. There will be less space required for insertion sort.