Q1. Selection Sort

43	7	10	23	18	4	19	5	66	14
[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]

1.

4	7	10	23	18	43	19	5	66	14
[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]

2.

4	5	10	23	18	43	19	7	66	14
[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]

3.

4	5	7	23	18	43	19	10	66	14
[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]

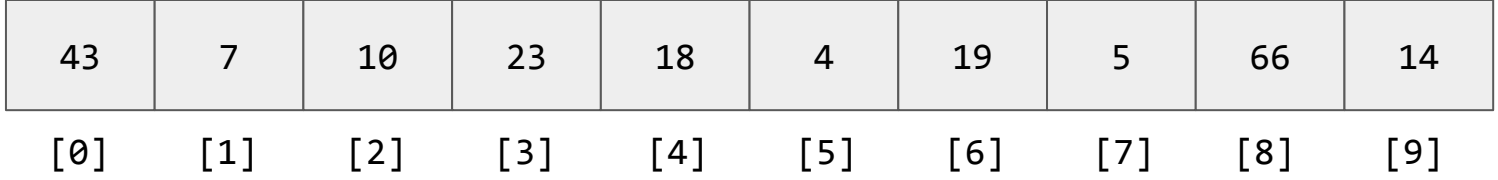
4.

4	5	7	10	18	43	19	23	66	14
[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]

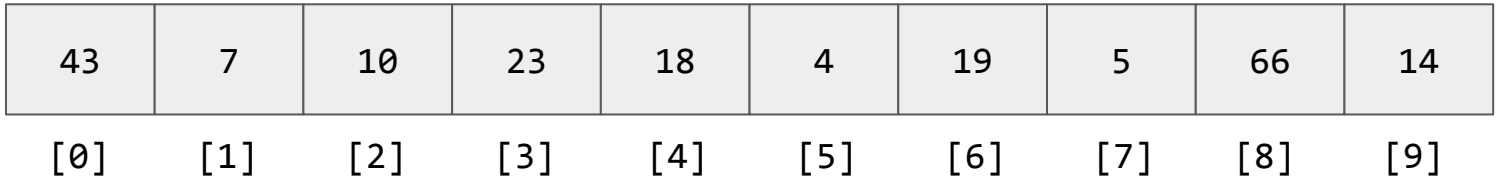
Q2. Bubble Sort

	43	7	10	23	18	4	19	5	66	14
	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
1.	4	43	7	10	23	18	5	19	14	66
	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
2.	4	5	43	7	10	23	18	14	19	66
	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
3.	4	5	7	43	10	14	23	18	19	66
	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
4.	4	5	7	10	43	14	18	23	19	66
	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]

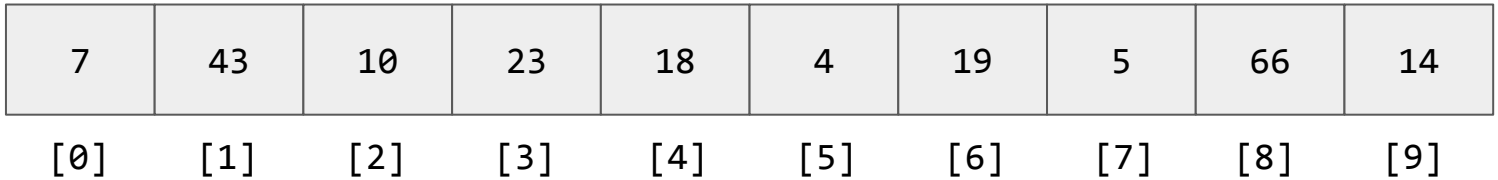
Q3. Insertion Sort



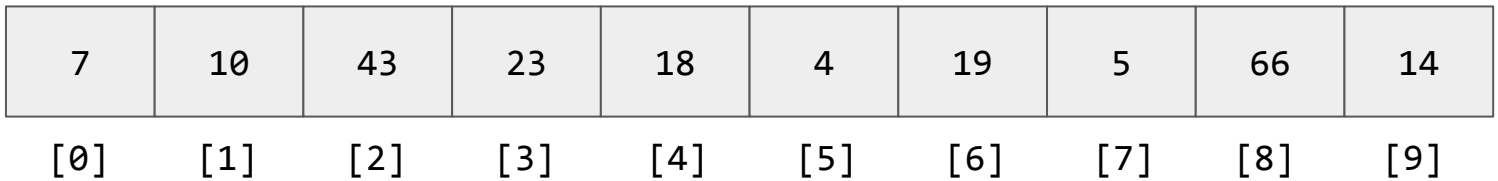
1.



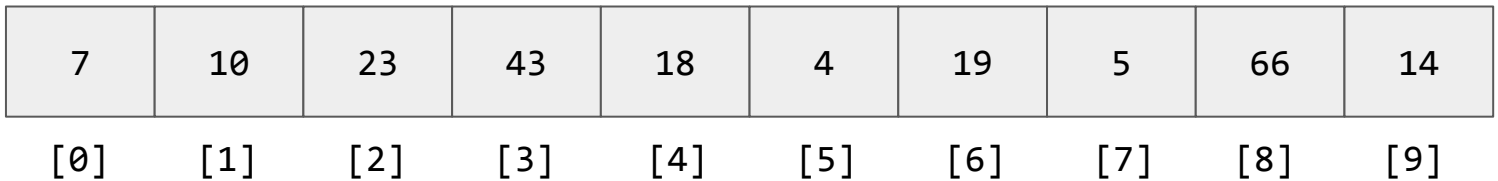
2.



3.



4.



Given the following array:

26	24	3	17	25	24	13	60	47	1
[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]

tell which sorting algorithm would produce the following results after four iterations:

a.

1	3	13	17	26	24	24	25	47	60
[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]

b.

1	3	13	17	25	24	24	60	47	26
[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]

c.

3	17	24	26	25	24	13	60	47	1
[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]

a. Bubble Sort

	26	24	3	17	25	24	13	60	47	1
	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]

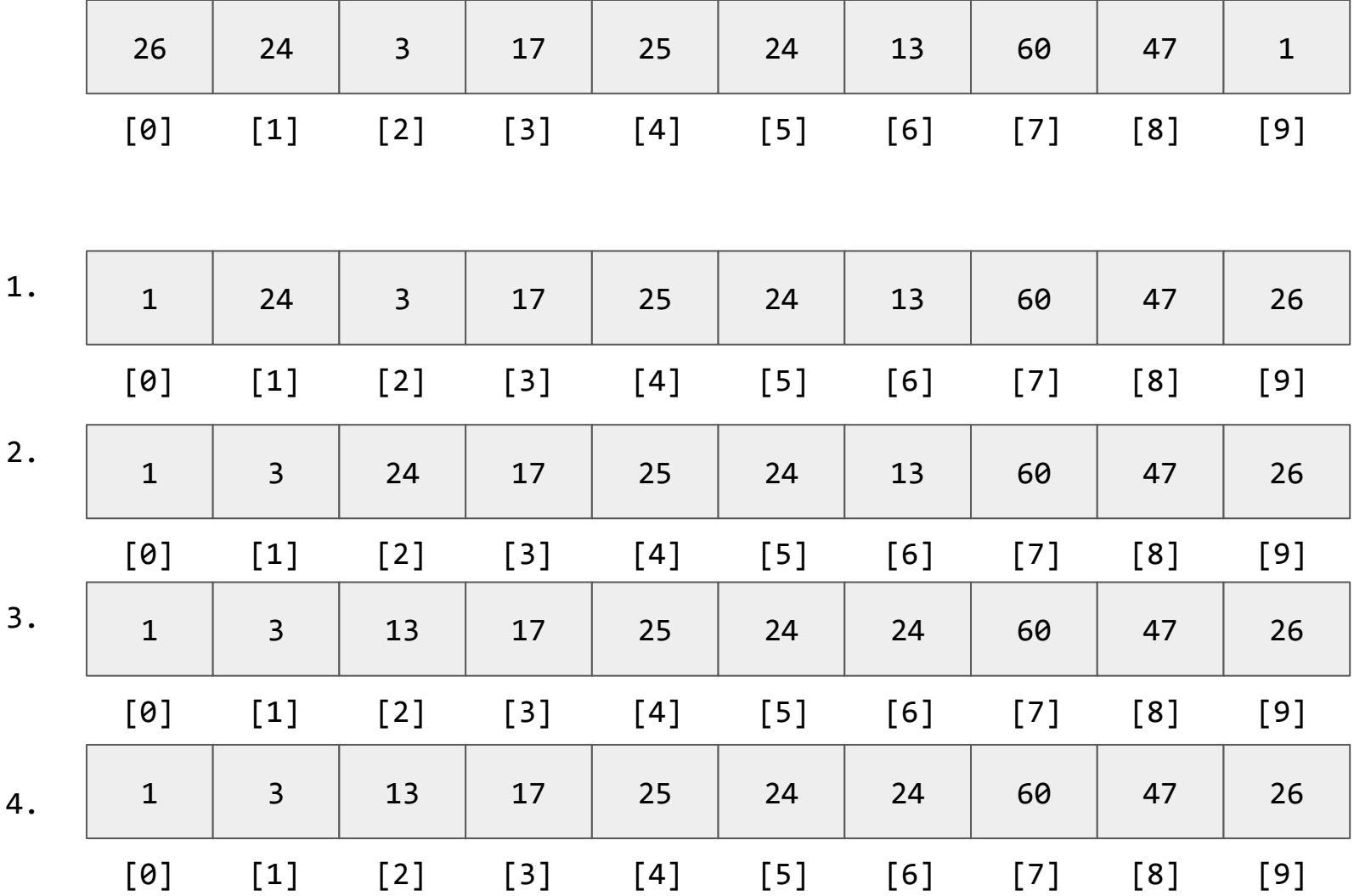
1.	1	26	24	3	17	25	24	13	60	47
	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]

2.	1	3	26	24	15	17	25	24	47	60
	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]

3.	1	3	13	26	24	17	24	25	47	60
	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]

4.	1	3	13	17	26	24	24	25	47	60
	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]

b. Selection Sort



c. Insertion Sort

	26	24	3	17	25	24	13	60	47	1
	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]

1.

26	24	3	17	25	24	13	60	47	1
[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]

2.

24	26	3	17	25	24	13	60	47	1
[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]

3.

3	24	26	17	25	24	13	60	47	1
[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]

4.

3	17	24	26	25	24	13	60	47	1
[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]

A very large array of elements is to be sorted. The program will be run on a personal computer with limited memory. Which sort would be a better choice: a heap sort or a merge sort? Why?

HeapSort would be a better choice than MergeSort, because MergeSort needs a copy of the array for its processing whereas HeapSort sorts within the original array.