1. Sort the sequence 3, 1, 4, 7, 5, 9, 2, 6, 8 using Insertion Sort (Please present the sorting procedure as shown on page 15 in slides of Course_09) and calculate the number of swaps.

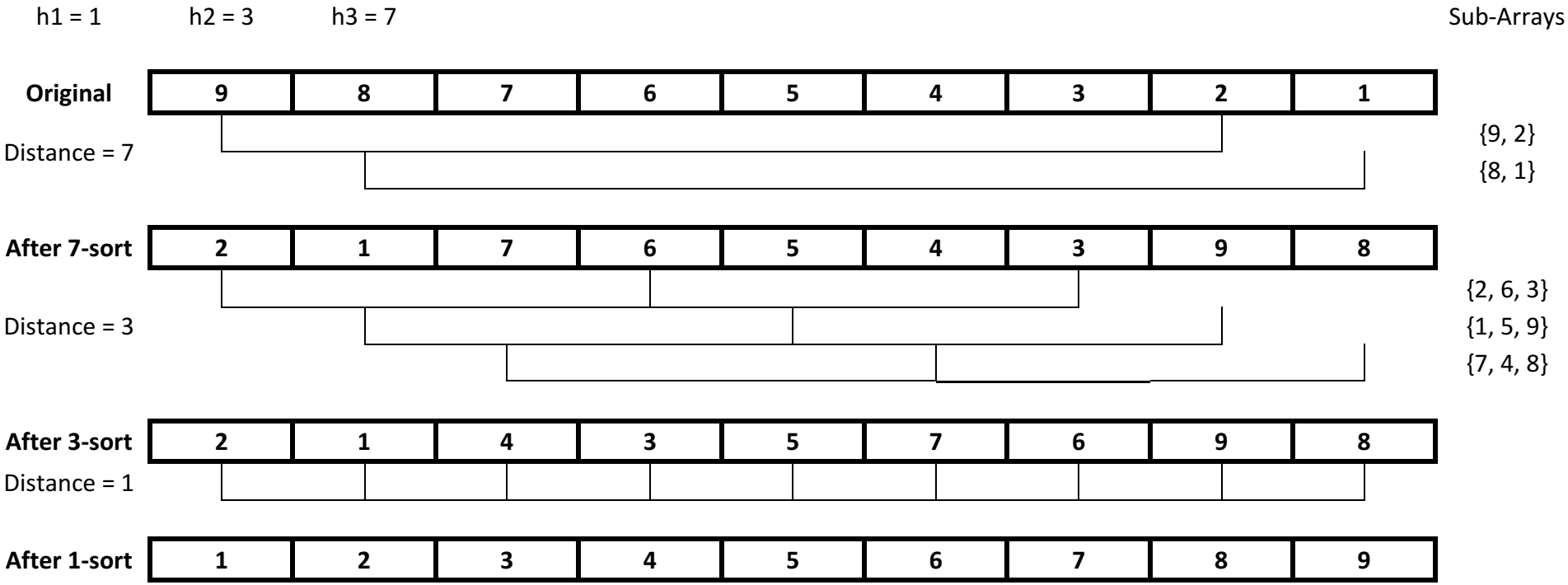
The procedure:

- Insertion sort consists of 8 passes.
- For pass P = 1 through 8, insertion sort ensures that the elements in position 0 to P are in sorted order.

Original	3	1	4	7	5	9	2	6	8	Swap	Comparison
After P = 1	1	3	4	7	5	9	2	6	8	1	2
After P = 2	1	3	4	7	5	9	2	6	8	0	1
After P = 3	1	3	4	7	5	9	2	6	8	0	1
After P = 4	1	3	4	5	7	9	2	6	8	1	2
After P = 5	1	3	4	5	7	9	2	6	8	0	1
After P = 6	1	2	3	4	5	7	9	6	8	5	6
After P = 7	1	2	3	4	5	6	7	9	8	2	3
After P = 8	1	2	3	4	5	6	7	8	9	1	2
Sum	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**10**	**18**

2. Sort the sequence 9, 8, 7, 6, 5, 4, 3, 2, 1 using Shell Sort with the increments {7, 3, 1} (Please present the sorting procedure as shown on page 34 in slides of Course_09) and calculate the number of swaps.

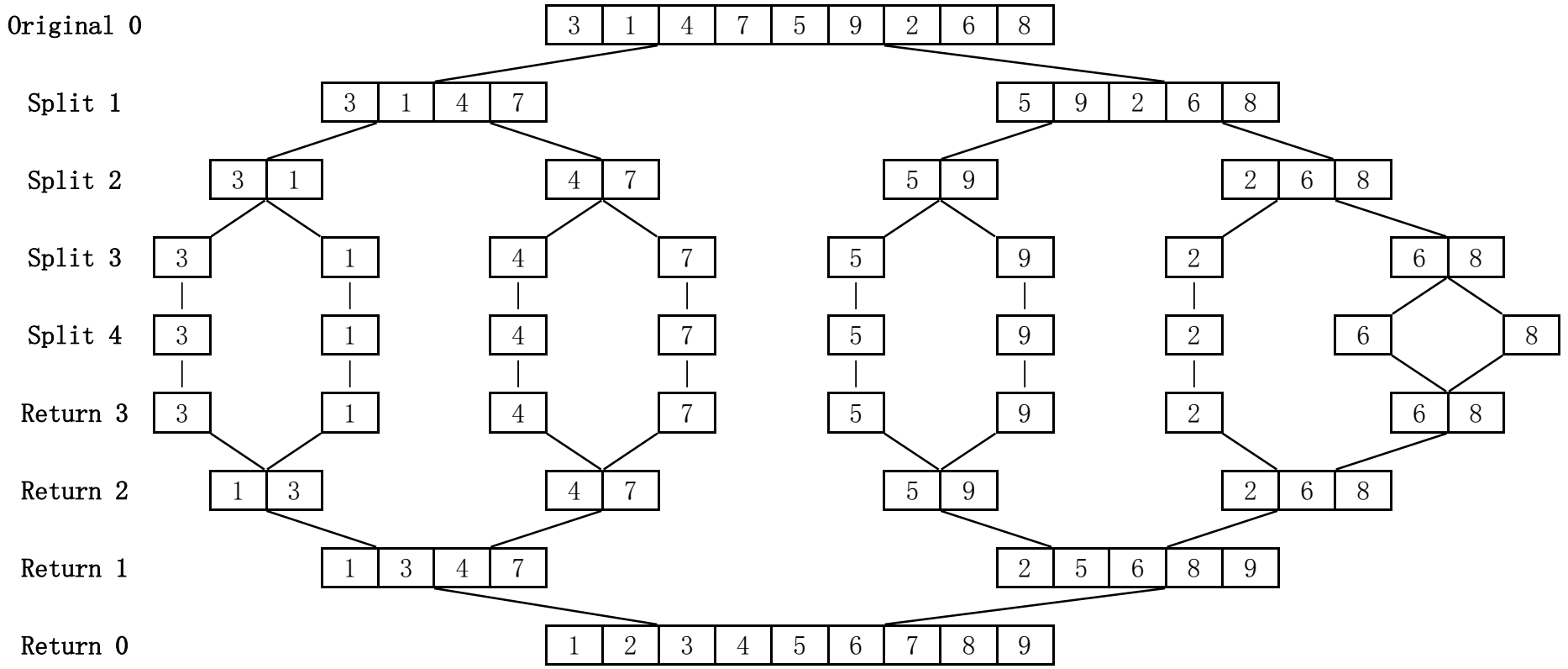
The procedure:



Number of swaps = (1 + 1) + (1 + 0 + 1) + (1 + 1 + 1 + 1) = 8

3. Sort 3, 1, 4, 7, 5, 9, 2, 6, 8 using Merge Sort (Please present the sorting procedure as shown on page 45 in slides of Course_09).

The procedure:

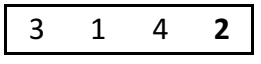


4. Sort 3, 1, 4, 7, 5, 9, 2, 6, 8 using Quick Sort with median-of-three pivot selection and the partitioning strategy (Please present the sorting procedure as shown in pages 67 and 68 in slides of Course_09).

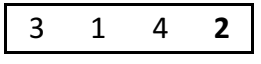
The procedure: （ppt 67 页的 while 边界有问题，导致 n=2 时出现错误，因次以下答案将 while 边界改为了“i <= j”）

Original A[9]	3	1	4	7	5	9	2	6	8
1st Partition :									
median(A[0], A[4],A[8]) = median(3,5,8) = 5 = A[4]									
pivot = A[4]; left = 0; right = 9; swap(A[4], A[9]);									
i = left; j = right -1;	3	1	4	7	8	9	2	6	**5**
	i							j	
Before 1st swap	3	1	4	7	8	9	2	6	**5**
				i			j		
After 1st swap	3	1	4	2	8	9	7	6	**5**
				i			j		
Before 2nd swap	3	1	4	2	8	9	7	6	**5**
				j	i				
swap(A[i], A[right])	3	1	4	2	**5**	9	7	6	8
				j	i				
After 1st Partition	3	1	4	2	**5**	9	7	6	8
2-1st Partition :	3	1	4	2					
median(A[0], A[1],A[3]) = median(3,1,2) = 2 = A[3]									
pivot = A[3]; left = 0; right = 3									

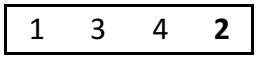
i = left; j = right -1;



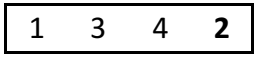
Before 1st swap



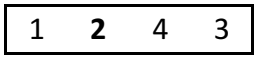
After 1st swap



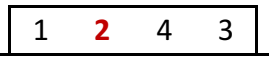
Before 2nd swap



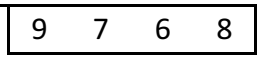
swap(A[i], A[right])



After 2-1st Partition



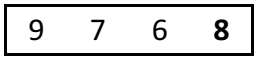
2-2nd Partition :



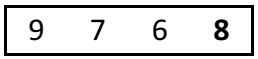
median(A[5], A[6],A[8]) = median(9,7,8) = 8 = A[8]

pivot = A[8]; left = 5; right = 8

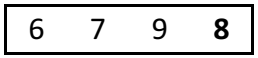
i = left; j = right -1;



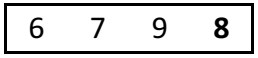
Before 1st swap



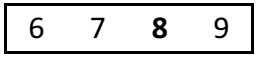
After 1st swap



Before 2nd swap



swap(A[i], A[right])



							j	i	
After 2-2nd Partition							6	7	**8**	9
3-1st Partition :			4	3					
median(A[2], A[2],A[3]) = median(4,4,3) = 4 = A[2]									
pivot = A[2]; left = 2; right = 3; swap(A[2], A[3])									
i = left; j = right -1;			3	**4**					
			ij						
Before 1st swap			3	**4**					
			j	i					
swap(A[i], A[right]) // swap(A[3], A[3])			3	**4**					
			j	i					
After 3-1st Partition			3	**4**					
3-2nd Partition :							6	7	
median(A[5], A[5],A[6]) = median(6,6,7) = 6 = A[5]									
pivot = A[5]; left = 5; right = 6; swap(A[5], A[6])									
i = left; j = right -1;							7	**6**	
							ij		
Before 1st swap							7	**6**	
						j	i		
swap(A[i], A[right])							**6**	7	
						j	i		
After 3-2ndPartition							**6**	7	
Final Sorted Output	1	2	3	4	5	6	7	8	9