(1)

a. Insertion sort

p1	p2	p3	p4	p5	p6	p6	p7
64	32	32	32	32	32	32	32
*32	64	64	64	64	64	46	46
79	*79	79	79	67	67	64	64
83	83	*83	83	79	79	67	67
67	67	67	*67	83		79	79
46	46	46	46	*46		83	83
96	96	96	96	96		*96	96
55	55	55	55	55		55	*55
68	68	68	68	68		68	68
12	12	12	12	12		12	12

p8	p9	
32	32	12
46	46	32
55	55	46
64	64	55
67	67	64
79	68	67
83	79	68
96	83	79
*68	96	83
12	*12	96

No. of elements = 10, passes = n-1 = 10-1 = 9

b. Shell sort 5,3,1

5 sort

64	46	46	46	46	46
32	32	32	32	32	
79	79	79	55	55	55
83	83	83	83	68	68
67	67	67	67	67	12
46	64	64	64	64	64
96	96	96	96	96	96
55	55	55	79	79	79
68	68	68	68	83	83
12	12	12	12	12	67

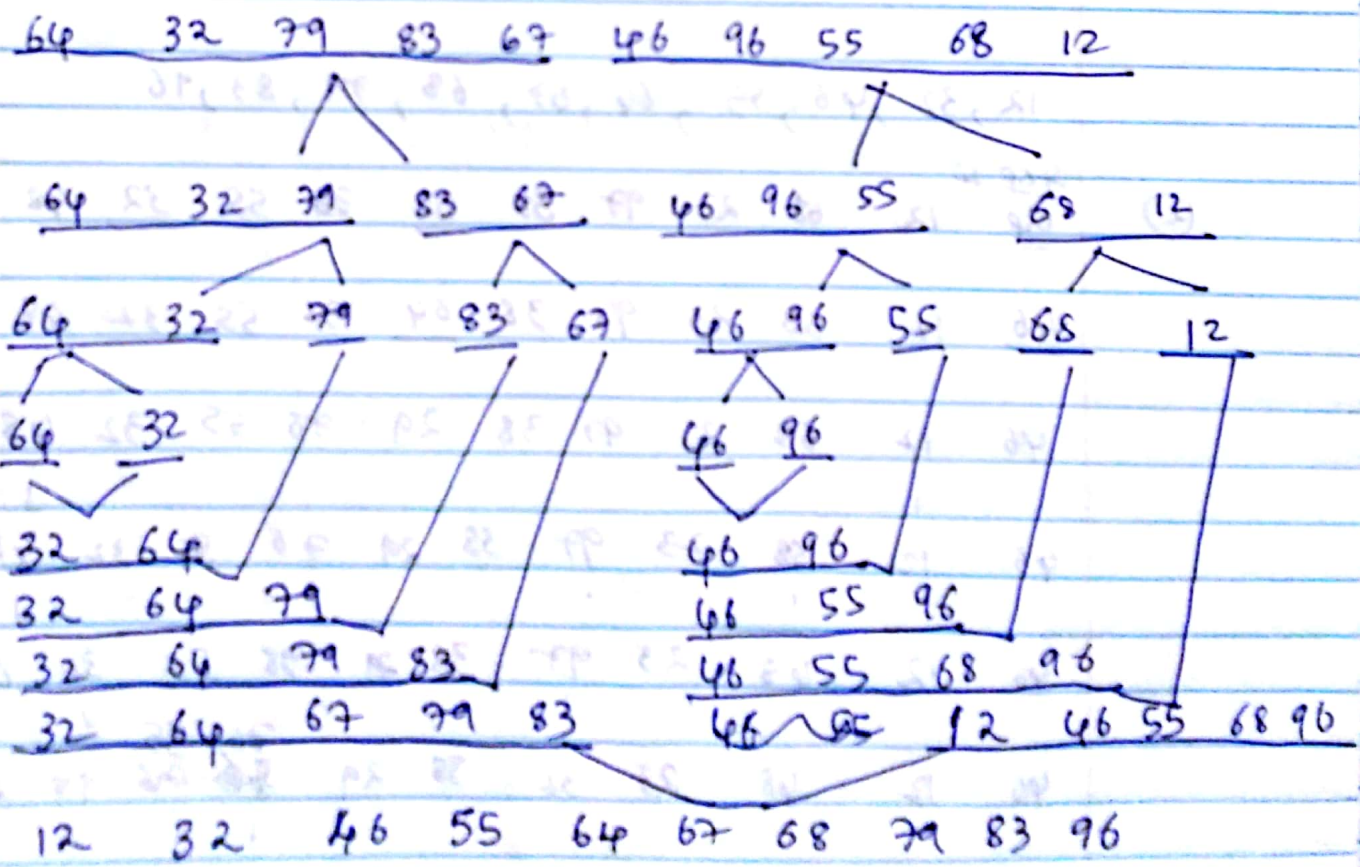
3 sort

46	46	46	46	46	46	46	46
32	12	12	12	12	12	12	12
55	55	55	55	55	55	55	55
68	68	68	68	68	68	68	67
12	32	32	32	32	32	32	32
64	64	64	64	64	64	64	64
96	96	96	96	96	96	67	68
79	79	79	79	79	79	79	79
83	83	83	83	83	83	83	83
67	67	67	67	67	67	96	96

1-sort

46	12	12	12	12	12	12	12	12
12	46	46	46	32	32	32	32	32
55	55	55	55	46	46	46	46	46
67	67	67	67	55	55	55	55	55
32	32	32	32	67	64	64	64	64
64	64	64	64	64	67	67	67	67
68	68	68	68	68	68	68	68	68
79	79	79	79	79	79	79	79	79
83	83	83	83	83	83	83	83	83
96	96	96	96	96	96	96	96	96

c)



d) Radix Sort

	1st digit Sort		2nd digit Sort	
64	0		0	
32	1		1	12
79	2	12, 32	2	
83	3	~~12~~, ~~32~~ 83	3	32
67	4	64	4	46
46	5	55	5	55
96	6	~~64~~ 46, 96	6	64, 67, 68
55	7	~~7~~ 67	7	~~79~~
68	8	68	8	83
12	9	~~79~~	9	96

12, 32, 46, 55, 64, 67, 68, 79, 83, 96

② ~~Step i:~~

64 12 68 23 97 38 81 76 55 32 48 29 46

46 12 68 23 97 38 64 76 55 32 68 29 81

46 12 68 23 97 38 29 76 55 32 48 64 81
i j

46 12 68 23 97 38 29 76 55 32 48 64 81
i j

46 12 48 23 97 38 29 76 55 32 68 46 81
i j

46 12 48 23 32 38 29 ~~76~~ 55 ~~46~~ 97 68 64 81
i j

46 12 48 23 32 38 29 55 76 97 68 64 81
j i

46 12 48 23 32 38 29 55 64 97 68 76 81
j i

Step 2:

46 12 48 23 32 38 29 55

23 12 48 46 32 38 29 55

23 12 48 29 32 38 46 55

23 12 38 29 32 48 46 55

23 12 38 29 32 48 46 55

23 12 38 29 32 46 48 55

Step 3:

23 12 38 29 32

23 12 32 29 38

23 12 32 29 38

23 12 32 29 38

23 12 29 32 38 ⇒ list size <=3 ⇒ 12 23 29 32 38

12 23 29 32 38 46 48 55

97 68 76 81

68 81 76 97

68 76 81 97

list size <=3

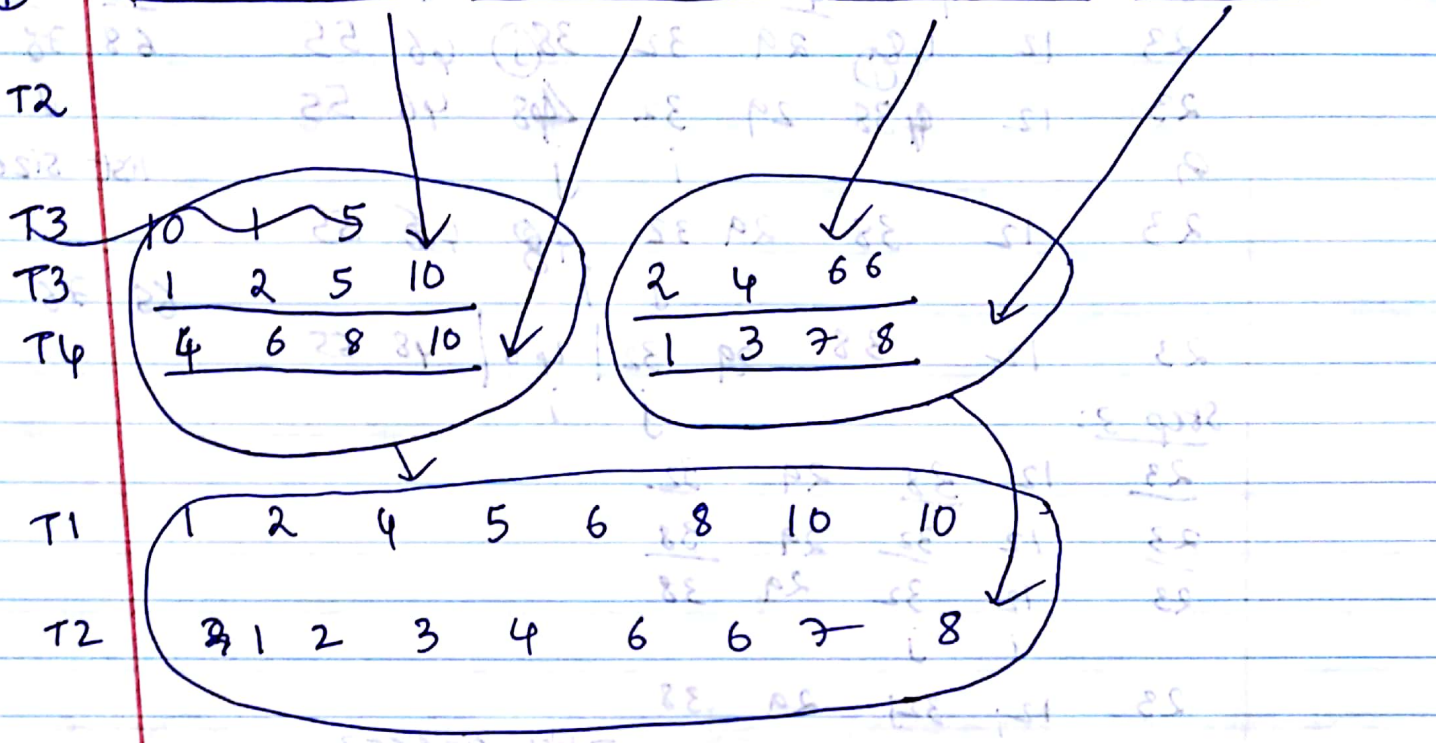
68 76 81 97

12 23 29 32 38 46 64 68 76 81 97

(3)

8	0		2
7	1		2
4	2	11	4
2	3		4
5	4	11	5
5	5	111	5
2	6		5
4	7	11	7
5	8	11	7
7			8
8			8

④ T1 10 1 5 2 6 8 4 10 6 6 2 4 1 8 7 3



T3 1 1 2 2 3 4 4 5 6 6 7 8 8 10 10

⑤ M = 3

10 1 5 2 6 8 4 10 6 6 2 4 1 8 7

10	10	
1	10 1	
5	10 1 5	Run 1
2	10 2 5	1
6	10 6 5	2
8	10 6 8	5
4	10 4* 8	6
10	10 4* 10	8
6	6* 4* 10	10
6	6* 4* 6*	10

6* 4* 6*

2 6 4 6

4 6 2* 6

1 4* 2* 6

4* 2* 1*

Run 2

Run → 4

4 6

6

4 2 1

8 4 2 8

7 4 7 8

Run 3

1

2

4

7

8

(5) ~~4 items can be compared in 4! ways i.e 24 ways~~

(6) 4 items have 4! possible arrangments.

This would lead to a tree with 4! = 24 leaves & log(4!) depth ⇒ hence log(4!) comparisons.

$$\log(4!) = 4.584 \simeq 5$$

It would take 5 comparisons at worst to sort them.