Sorting

Array: 1 , 7 , 8 , 5 , 3 , 6 , 4 , 2

● swap
● iteration

a) Selection Sort

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

b) Insertion Sort

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

● sorted partition

● swap

↓ where is it placed

1   3 5   6   7   8   4   2

1   3   4   5   6   7   8   2

1   2   3   4   5   6   7   8

c) Merge sort

1   7   8   5 | 3   6   4   2

1   7 | 8   5 | 3   6 | 4   2

| 1 | 7 | 8 | 5 | 3 | 6 | 4 | 2 |

1   7 | 5   8 | 3   6 | 2   4

1   5   7   8 | 2   3   4   6

1   2   3   4   5   6   7   8

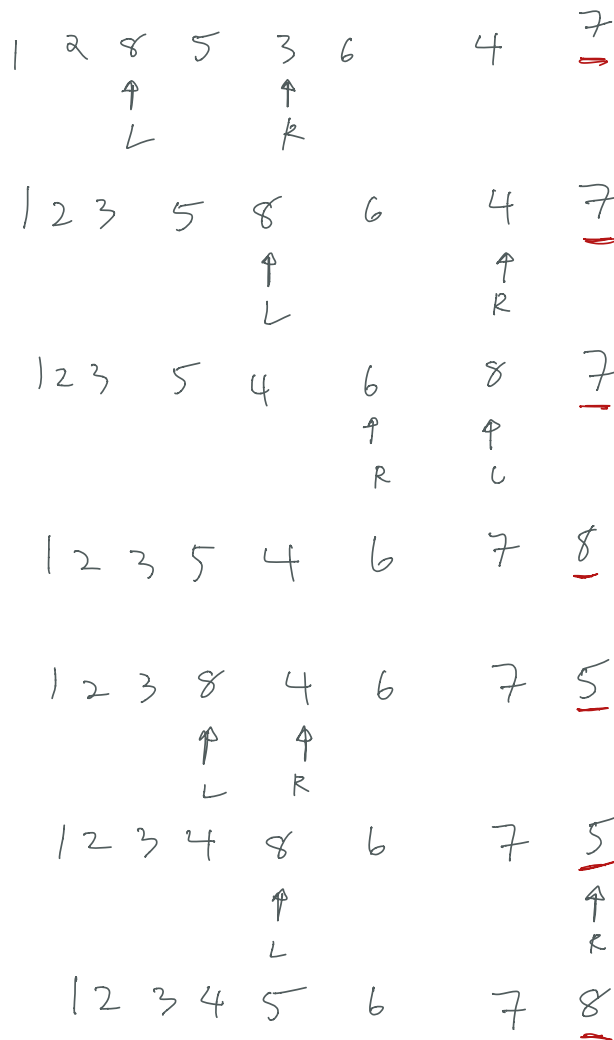d) Quick sort

1   7   8   5   3   6   4   2

2   7   8   5   3   6   4   1
↑   ↗
L   ↑
                            R

☐1 7   8   5   3   6   4   2

1 2 8 5 3 6   4   7
      ↑   ↑
      L   R

1 2 3 5 8   6   4   7
         ↑   ↑
         L   R

1 2 3 5 4   6   8   7
               ↑   ↑
               R   L

1 2 3 5 4   6   7   8

1 2 3 8 4   6   7   5
      ↑   ↑
      L   R

1 2 3 4 8   6   7   5
         ↑   ↑
         L   R

1 2 3 4 5   6   7   8

e)   Heap Sort

   1 7 8 5 3 6 4 2   → build max heap
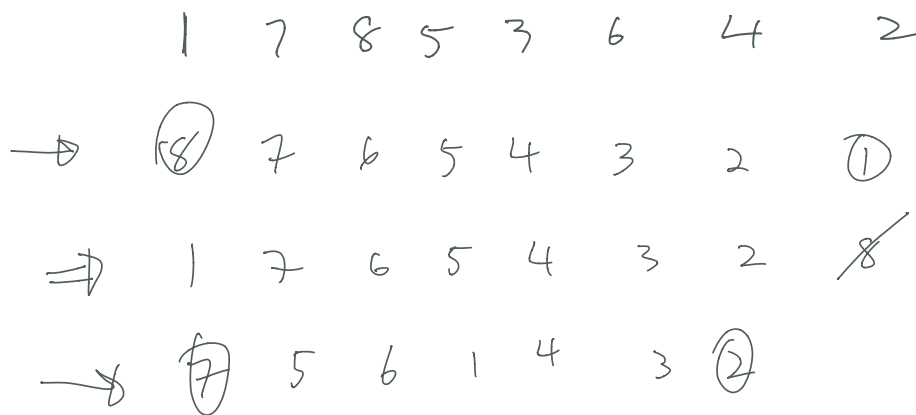
→ ⑧ 7 6 5 4 3 2 ①   ✓ remove from array consider sorted

⇒ 1 7 6 5 4 3 2 ⊗   ○ swap

→ ⑦ 5 6 1 4 3 ②   ⇒ heapify

⇒ ② 5 ⑥ 1 4 3 ̶7̶

→ 5 2 6   1 4 3

⑥ 5 4   3 2 ①

⇒ 1 5 4   3 2 ̶6̶

↛ 5 3 4   1 2

⇒ ⑤ 4 3 1 ②

2 4 3 1 ̶5̶

→ 4 3 2 1

1 3 2 ̶4̶

3 1 2

3 2 1

1 2 ̶3̶

2 1

2 1   ⟶ take all /

1 2   1 2 3 4 5 6 7 8