Shell Sort:

A:

1)

Gap: 5

nextPos: 5 (6)

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

2)

Gap: 5

nextPos: 6 (7)

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

3)

Gap: 5

nextPos: 7 (8)

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

4)

Gap: 5

nextPos: 8 (9)

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

5)

Gap: 5

nextPos: 9 (10)

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

6)

Gap: 2

nextPos: 2 (3)

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

7)

Gap: 2

nextPos: 3 (4)

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

8)

Gap: 2

nextPos: 4 (5)

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

9)

Gap: 2

nextPos: 5 (6)

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

10)

Gap: 2

nextPos: 6 (7)

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

11)

Gap: 2

nextPos: 7 (8)

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

12)

Gap: 2

nextPos: 8 (9)

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

13)

Gap: 2

nextPos: 9 (10)

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

14)

Gap: 1

nextPos: 1 (2)

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

15)

Gap: 1

nextPos: 2 (3)

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

16)

Gap: 1

nextPos: 3 (4)

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

17)

Gap: 1

nextPos: 4 (5)

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

18)

Gap: 1

nextPos: 5 (6)

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

19)

Gap: 1

nextPos: 6 (7)

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

20)

Gap: 1

nextPos: 7 (8)

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

21)

Gap: 1

nextPos: 8 (9)

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

22)

Gap: 1

nextPos: 9 (10)

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

23)

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

Gap: 0

B:

1)

[10, 9, 8, 7, 6,

 5, 4, 3, 2, 1]

Gap: 5

nextPos: 5 (10) <---> 0 (5)

[5, 9, 8, 7, 6, 10, 4, 3, 2, 1]

2)

[5, 9, 8, 7, 6,

 10, 4, 3, 2, 1]

Gap: 5

nextPos: 6 (9) <---> 1 (4)

[5, 4, 8, 7, 6, 10, 9, 3, 2, 1]

3)

[5, 4, 8, 7, 6,

 10, 9, 3, 2, 1]

Gap: 5

nextPos: 7 (8) <---> 2 (3)

[5, 4, 3, 7, 6, 10, 9, 8, 2, 1]

4)

[5, 4, 3, 7, 6,

 10, 9, 8, 2, 1]

Gap: 5

nextPos: 8 (7) <---> 3 (2)

[5, 4, 3, 2, 6, 10, 9, 8, 7, 1]

5)

[5, 4, 3, 2, 6,

```
 10, 9, 8, 7, 1]

Gap: 5

nextPos: 9 (6) <---> 4 (1)

[5, 4, 3, 2, 1, 10, 9, 8, 7, 6]


6)

[5, 4,

 3, 2, 1, 10, 9, 8, 7, 6]

Gap: 2

nextPos: 2 (5) <---> 0 (3)

[3, 4, 5, 2, 1, 10, 9, 8, 7, 6]


7)

[3, 4,

 5, 2, 1, 10, 9, 8, 7, 6]

Gap: 2

nextPos: 3 (4) <---> 1 (2)

[3, 2, 5, 4, 1, 10, 9, 8, 7, 6]


8)

[3, 2,

 5, 4, 1, 10, 9, 8, 7, 6]

Gap: 2

nextPos: 4 (5) <---> 0 (1)

[1, 2, 3, 4, 5, 10, 9, 8, 7, 6]


9)

Gap: 2

nextPos: 5 (10)

[1, 2, 3, 4, 5, 10, 9, 8, 7, 6]
```

10)

Gap: 2

nextPos: 6 (9)

[1, 2, 3, 4, 5, 10, 9, 8, 7, 6]

11)

[1, 2,

3, 4, 5, 10, 9, 8, 7, 6]

Gap: 2

nextPos: 7 (10) <---> 5 (8)

[1, 2, 3, 4, 5, 8, 9, 10, 7, 6]

12)

[1, 2,

3, 4, 5, 8, 9, 10, 7, 6]

Gap: 2

nextPos: 8 (9) <---> 6 (7)

[1, 2, 3, 4, 5, 8, 7, 10, 9, 6]

13)

[1, 2,

3, 4, 5, 8, 7, 10, 9, 6]

Gap: 2

nextPos: 9 (10) <---> 5 (6)

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

14)

Gap: 1

nextPos: 1 (2)

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

```
15)
Gap: 1
nextPos: 2 (3)
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

16)
Gap: 1
nextPos: 3 (4)
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

17)
Gap: 1
nextPos: 4 (5)
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

18)
Gap: 1
nextPos: 5 (6)
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

19)
Gap: 1
nextPos: 6 (7)
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

20)
Gap: 1
nextPos: 7 (8)
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

21)
```

Gap: 1

nextPos: 8 (9)

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

22)

Gap: 1

nextPos: 9 (10)

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

23)

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

Gap: 0

C:

1)

Gap: 6

nextPos: 6 (6)

[5, 2, 13, 9, 1, 7, 6, 8, 1, 15, 4, 11]

2)

Gap: 6

nextPos: 7 (8)

[5, 2, 13, 9, 1, 7, 6, 8, 1, 15, 4, 11]

3)

[5, 2, 13, 9, 1, 7,

6, 8, 1, 15, 4, 11]

Gap: 6

nextPos: 8 (13) <---> 2 (1)

[5, 2, 1, 9, 1, 7, 6, 8, 13, 15, 4, 11]

4)

Gap: 6

nextPos: 9 (15)

[5, 2, 1, 9, 1, 7, 6, 8, 13, 15, 4, 11]

5)

Gap: 6

nextPos: 10 (4)

[5, 2, 1, 9, 1, 7, 6, 8, 13, 15, 4, 11]

```
6)
Gap: 6
nextPos: 11 (11)
[5, 2, 1, 9, 1, 7, 6, 8, 13, 15, 4, 11]

7)
[5, 2,
 1, 9, 1, 7, 6, 8, 13, 15, 4, 11]
Gap: 2
nextPos: 2 (5) <---> 0 (1)
[1, 2, 5, 9, 1, 7, 6, 8, 13, 15, 4, 11]

8)
Gap: 2
nextPos: 3 (9)
[1, 2, 5, 9, 1, 7, 6, 8, 13, 15, 4, 11]

9)
[1, 2,
 5, 9, 1, 7, 6, 8, 13, 15, 4, 11]
Gap: 2
nextPos: 4 (5) <---> 2 (1)
[1, 2, 1, 9, 5, 7, 6, 8, 13, 15, 4, 11]

10)
[1, 2,
 1, 9, 5, 7, 6, 8, 13, 15, 4, 11]
Gap: 2
nextPos: 5 (9) <---> 3 (7)
[1, 2, 1, 7, 5, 9, 6, 8, 13, 15, 4, 11]
```

```
11)
Gap: 2
nextPos: 6 (6)
[1, 2, 1, 7, 5, 9, 6, 8, 13, 15, 4, 11]

12)
[1, 2,
 1, 7, 5, 9, 6, 8, 13, 15, 4, 11]
Gap: 2
nextPos: 7 (9) <---> 5 (8)
[1, 2, 1, 7, 5, 8, 6, 9, 13, 15, 4, 11]

13)
Gap: 2
nextPos: 8 (13)
[1, 2, 1, 7, 5, 8, 6, 9, 13, 15, 4, 11]

14)
Gap: 2
nextPos: 9 (15)
[1, 2, 1, 7, 5, 8, 6, 9, 13, 15, 4, 11]

15)
[1, 2,
 1, 7, 5, 8, 6, 9, 13, 15, 4, 11]
Gap: 2
nextPos: 10 (13) <---> 4 (4)
[1, 2, 1, 7, 4, 8, 5, 9, 6, 15, 13, 11]
```

```
16)

[1, 2,

 1, 7, 4, 8, 5, 9, 6, 15, 13, 11]

Gap: 2

nextPos: 11 (15) <---> 9 (11)

[1, 2, 1, 7, 4, 8, 5, 9, 6, 11, 13, 15]


17)

Gap: 1

nextPos: 1 (2)

[1, 2, 1, 7, 4, 8, 5, 9, 6, 11, 13, 15]


18)

[1,

 2, 1, 7, 4, 8, 5, 9, 6, 11, 13, 15]

Gap: 1

nextPos: 2 (2) <---> 1 (1)

[1, 1, 2, 7, 4, 8, 5, 9, 6, 11, 13, 15]


19)

Gap: 1

nextPos: 3 (7)

[1, 1, 2, 7, 4, 8, 5, 9, 6, 11, 13, 15]


20)

[1,

 1, 2, 7, 4, 8, 5, 9, 6, 11, 13, 15]

Gap: 1

nextPos: 4 (7) <---> 3 (4)

[1, 1, 2, 4, 7, 8, 5, 9, 6, 11, 13, 15]
```

21)

Gap: 1

nextPos: 5 (8)

[1, 1, 2, 4, 7, 8, 5, 9, 6, 11, 13, 15]

22)

[1,

1, 2, 4, 7, 8, 5, 9, 6, 11, 13, 15]

Gap: 1

nextPos: 6 (8) <---> 4 (5)

[1, 1, 2, 4, 5, 7, 8, 9, 6, 11, 13, 15]

23)

Gap: 1

nextPos: 7 (9)

[1, 1, 2, 4, 5, 7, 8, 9, 6, 11, 13, 15]

24)

[1,

1, 2, 4, 5, 7, 8, 9, 6, 11, 13, 15]

Gap: 1

nextPos: 8 (9) <---> 5 (6)

[1, 1, 2, 4, 5, 6, 7, 8, 9, 11, 13, 15]

25)

Gap: 1

nextPos: 9 (11)

[1, 1, 2, 4, 5, 6, 7, 8, 9, 11, 13, 15]

26)

Gap: 1

nextPos: 10 (13)

[1, 1, 2, 4, 5, 6, 7, 8, 9, 11, 13, 15]

27)

Gap: 1

nextPos: 11 (15)

[1, 1, 2, 4, 5, 6, 7, 8, 9, 11, 13, 15]

28)

[1, 1, 2, 4, 5, 6, 7, 8, 9, 11, 13, 15]

Gap: 0

D:

1)

[S, B, I, M, H, Q,

C, L, R, E, P, K]

Gap: 6

nextPos: 6 (S) <---> 0 (C)

[C, B, I, M, H, Q, S, L, R, E, P, K]

2)

Gap: 6

nextPos: 7 (L)

[C, B, I, M, H, Q, S, L, R, E, P, K]

3)

Gap: 6

nextPos: 8 (R)

[C, B, I, M, H, Q, S, L, R, E, P, K]

4)

[C, B, I, M, H, Q,

S, L, R, E, P, K]

Gap: 6

nextPos: 9 (M) <---> 3 (E)

[C, B, I, E, H, Q, S, L, R, M, P, K]

5)

Gap: 6

nextPos: 10 (P)

[C, B, I, E, H, Q, S, L, R, M, P, K]

6)

[C, B, I, E, H, Q,

S, L, R, M, P, K]

Gap: 6

nextPos: 11 (Q) <---> 5 (K)

[C, B, I, E, H, K, S, L, R, M, P, Q]

7)

Gap: 2

nextPos: 2 (I)

[C, B, I, E, H, K, S, L, R, M, P, Q]

8)

Gap: 2

nextPos: 3 (E)

[C, B, I, E, H, K, S, L, R, M, P, Q]

9)

[C, B,

I, E, H, K, S, L, R, M, P, Q]

Gap: 2

nextPos: 4 (I) <---> 2 (H)

[C, B, H, E, I, K, S, L, R, M, P, Q]

10)

Gap: 2

nextPos: 5 (K)

[C, B, H, E, I, K, S, L, R, M, P, Q]

11)

Gap: 2

nextPos: 6 (S)

[C, B, H, E, I, K, S, L, R, M, P, Q]

12)

Gap: 2

nextPos: 7 (L)

[C, B, H, E, I, K, S, L, R, M, P, Q]

13)

[C, B,

H, E, I, K, S, L, R, M, P, Q]

Gap: 2

nextPos: 8 (S) <---> 6 (R)

[C, B, H, E, I, K, R, L, S, M, P, Q]

14)

Gap: 2

nextPos: 9 (M)

[C, B, H, E, I, K, R, L, S, M, P, Q]

15)

[C, B,

H, E, I, K, R, L, S, M, P, Q]

Gap: 2

nextPos: 10 (S) <---> 6 (P)

[C, B, H, E, I, K, P, L, R, M, S, Q]

16)

Gap: 2

nextPos: 11 (Q)

[C, B, H, E, I, K, P, L, R, M, S, Q]

17)

[C,

B, H, E, I, K, P, L, R, M, S, Q]

Gap: 1

nextPos: 1 (C) <---> 0 (B)

[B, C, H, E, I, K, P, L, R, M, S, Q]

18)

Gap: 1

nextPos: 2 (H)

[B, C, H, E, I, K, P, L, R, M, S, Q]

19)

[B,

C, H, E, I, K, P, L, R, M, S, Q]

Gap: 1

nextPos: 3 (H) <---> 2 (E)

[B, C, E, H, I, K, P, L, R, M, S, Q]

20)

Gap: 1

nextPos: 4 (I)

[B, C, E, H, I, K, P, L, R, M, S, Q]

21)

Gap: 1

nextPos: 5 (K)

[B, C, E, H, I, K, P, L, R, M, S, Q]

22)

Gap: 1

nextPos: 6 (P)

[B, C, E, H, I, K, P, L, R, M, S, Q]

23)

[B,

C, E, H, I, K, P, L, R, M, S, Q]

Gap: 1

nextPos: 7 (P) <---> 6 (L)

[B, C, E, H, I, K, L, P, R, M, S, Q]

24)

Gap: 1

nextPos: 8 (R)

[B, C, E, H, I, K, L, P, R, M, S, Q]

25)

[B,

C, E, H, I, K, L, P, R, M, S, Q]

Gap: 1

nextPos: 9 (R) <---> 7 (M)

[B, C, E, H, I, K, L, M, P, R, S, Q]

26)

Gap: 1

nextPos: 10 (S)

[B, C, E, H, I, K, L, M, P, R, S, Q]

27)

[B,

C, E, H, I, K, L, M, P, R, S, Q]

Gap: 1

nextPos: 11 (S) <---> 9 (Q)

[B, C, E, H, I, K, L, M, P, Q, R, S]

28)

[B, C, E, H, I, K, L, M, P, Q, R, S]

Gap: 0

Merge Sort:

A: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

[1, 2, 3, 4, 5] [6, 7, 8, 9, 10]

[1, 2] [3, 4, 5] [6, 7] [8, 9, 10]

[1] [2] [3] [4, 5] [6] [7] [8] [9, 10]

[4] [5] [9] [10]

[1] [2] [3] [4, 5] [6] [7] [8] [9, 10]

[1, 2] [3, 4, 5] [6, 7] [8, 9, 10]

[1, 2, 3, 4, 5] [6, 7, 8, 9, 10]

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

B: [10, 9, 8, 7, 6, 5, 4, 3, 2, 1]

[10, 9, 8, 7, 6, 5, 4, 3, 2, 1]

[10, 9, 8, 7, 6] [5, 4, 3, 2, 1]

[10, 9] [8, 7, 6] [5, 4] [3, 2, 1]

[10] [9] [8] [7, 6] [5] [4] [3] [2, 1]

[7] [6] [2] [1]

[10] [9] [8] [6, 7] [5] [4] [3] [1, 2]

[9, 10] [6, 7, 8] [4, 5] [1, 2, 3]

[6, 7, 8, 9, 10] [1, 2, 3, 4, 5]

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

C: [5, 2, 13, 9, 1, 7, 6, 8, 1, 15, 4, 11]

[5, 2, 13, 9, 1, 7, 6, 8, 1, 15, 4, 11]

[5, 2, 13, 9, 1, 7] [6, 8, 1, 15, 4, 11]

[5, 2, 13] [9, 1, 7] [6, 8, 1] [15, 4, 11]

[5] [2, 13] [9] [1, 7] [6] [8, 1] [15] [4, 11]

[2] [13] [1] [7] [8] [1] [4] [11]

[5] [2, 13] [9] [1, 7] [6] [1, 8] [15] [4, 11]

[2, 5, 13] [1, 7, 9] [1, 6, 8] [4, 11, 15]

[1, 2, 5, 7, 9, 13] [1, 4, 6, 8, 11, 15]

[1, 1, 2, 4, 5, 6, 7, 8, 9, 11, 13, 15]

D: [S, B, I, M, H, Q, C, L, R, E, P, K]

[S, B, I, M, H, Q, C, L, R, E, P, K]

[S, B, I, M, H, Q] [C, L, R, E, P, K]

[S, B, I] [M, H, Q] [C, L, R] [E, P, K]

[S] [B, I] [M] [H, Q] [C] [L, R] [E] [P, K]

[B] [I] [H] [Q] [L] [R] [P] [K]

[S] [B, I] [M] [H, Q] [C] [L, R] [E] [K, P]

[B, I, S] [H, M, Q] [C, L, R] [E, K, P]

[B, H, I, M, Q, S] [C, E, K, L, P, R]

[B, C, E, H, I, K, L, M, P, Q, R, S]

Heap Sort:

A:

Building a Max Heap:

```
      1
   2     3
 4   5   6   7
8    9     10
```

Child: 2 [1]

Parent: 1 [0]

Swap: 2<--->1

```
      2
   1     3
 4   5   6   7
8    9     10
```

Child: 3 [2]

Parent: 2 [0]

Swap: 3<--->2

```
      3
   1     2
 4   5   6   7
8    9     10
```

Child: 4 [3]

Parent: 1 [1]

Swap: 4<--->1

```
Child: 4 [1]

Parent: 3 [0]

Swap: 4<--->3



        4

     3       2

 1   5   6   7

8     9     10

Child: 5 [4]

Parent: 3 [1]

Swap: 5<--->3


Child: 5 [1]

Parent: 4 [0]

Swap: 5<--->4



        5

     4       2

 1   3   6   7

8     9     10

Child: 6 [5]

Parent: 2 [2]

Swap: 6<--->2


Child: 6 [2]

Parent: 5 [0]

Swap: 6<--->5
```

```
      6
   4     5
 1  3  2  7
8   9    10
Child: 7 [6]
Parent: 5 [2]
Swap: 7<--->5

Child: 7 [2]
Parent: 6 [0]
Swap: 7<--->6


      7
   4     6
 1  3  2  5
8   9    10
Child: 8 [7]
Parent: 1 [3]
Swap: 8<--->1

Child: 8 [3]
Parent: 4 [1]
Swap: 8<--->4

Child: 8 [1]
Parent: 7 [0]
Swap: 8<--->7


      8
```

```
    7      6
 4   3   2   5
1     9     10
Child: 9 [8]
Parent: 4 [3]
Swap: 9<--->4

Child: 9 [3]
Parent: 7 [1]
Swap: 9<--->7

Child: 9 [1]
Parent: 8 [0]
Swap: 9<--->8


        9
    8      6
 7   3   2   5
1     4     10
Child: 10 [9]
Parent: 3 [4]
Swap: 10<--->3

Child: 10 [4]
Parent: 8 [1]
Swap: 10<--->8

Child: 10 [1]
Parent: 9 [0]
Swap: 10<--->9
```

```
         10
     9       6
 7   8   2   5
1     4     3

Shrinking the Max Heap:
n = 10
         10
     9       6
 7   8   2   5
1     4     3

Swap: 10<--->3
         3
     9       6
 7   8   2   5
1     4     10

RightChild: 9 [1]
LeftChild: 6 [2]
MaxChild: 9 [1]
Parent: 3 [0]
Swap: 3<--->9
         9
     3       6
 7   8   2   5
1     4     10

RightChild: 7 [3]
```

```
LeftChild: 8 [4]

MaxChild: 8 [4]

Parent: 3 [1]

Swap: 3<--->8

        9

    8      6

 7   3   2   5

1     4     10


n = 9

        9

    8      6

 7   3   2   5

1     4     10


Swap: 9<--->4

        4

    8      6

 7   3   2   5

1     9     10


RightChild: 8 [1]

LeftChild: 6 [2]

MaxChild: 8 [1]

Parent: 4 [0]

Swap: 4<--->8

        8

    4      6

 7   3   2   5

1     9     10
```

```
RightChild: 7 [3]

LeftChild: 3 [4]

MaxChild: 7 [3]

Parent: 4 [1]

Swap: 4<--->7

       8

    7      6

 4   3   2   5

1     9     10


RightChild: 1 [7]

LeftChild: 9 [8]

MaxChild: 1 [7]

Parent: 4 [3]

n = 8

       8

    7      6

 4   3   2   5

1     9     10


Swap: 8<--->1

       1

    7      6

 4   3   2   5

8     9     10


RightChild: 7 [1]

LeftChild: 6 [2]

MaxChild: 7 [1]

Parent: 1 [0]

Swap: 1<--->7
```

```
        7
    1       6
 4   3   2   5
8    9     10

RightChild: 4 [3]
LeftChild: 3 [4]
MaxChild: 4 [3]
Parent: 1 [1]
Swap: 1<--->4
        7
    4       6
 1   3   2   5
8    9     10

n = 7
        7
    4       6
 1   3   2   5
8    9     10

Swap: 7<--->5
        5
    4       6
 1   3   2   7
8    9     10

RightChild: 4 [1]
LeftChild: 6 [2]
MaxChild: 6 [2]
Parent: 5 [0]
```

```
Swap: 5<--->6
       6
    4      5
 1   3   2   7
8    9     10

RightChild: 2 [5]
LeftChild: 7 [6]
MaxChild: 2 [5]
Parent: 5 [2]
n = 6
       6
    4      5
 1   3   2   7
8    9     10

Swap: 6<--->2
       2
    4      5
 1   3   6   7
8    9     10

RightChild: 4 [1]
LeftChild: 5 [2]
MaxChild: 5 [2]
Parent: 2 [0]
Swap: 2<--->5
       5
    4      2
 1   3   6   7
8    9     10
```

```
n = 5
        5
    4       2
 1   3   6   7
8     9     10

Swap: 5<--->3
        3
    4       2
 1   5   6   7
8     9     10

RightChild: 4 [1]
LeftChild: 2 [2]
MaxChild: 4 [1]
Parent: 3 [0]
Swap: 3<--->4
        4
    3       2
 1   5   6   7
8     9     10

RightChild: 1 [3]
LeftChild: 5 [4]
MaxChild: 1 [3]
Parent: 3 [1]
n = 4
        4
    3       2
 1   5   6   7
```

```
8     9     10

Swap: 4<--->1
        1
    3       2
 4   5   6   7
8     9     10

RightChild: 3 [1]
LeftChild: 2 [2]
MaxChild: 3 [1]
Parent: 1 [0]
Swap: 1<--->3
        3
    1       2
 4   5   6   7
8     9     10

n = 3
        3
    1       2
 4   5   6   7
8     9     10

Swap: 3<--->2
        2
    1       3
 4   5   6   7
8     9     10

RightChild: 1 [1]
```

```
LeftChild: 3 [2]

MaxChild: 1 [1]

Parent: 2 [0]

n = 2

       2

    1      3

 4   5   6   7

8     9     10


Swap: 2<--->1

       1

    2      3

 4   5   6   7

8     9     10


n = 1

       1

    2      3

 4   5   6   7

8     9     10


Swap: 1<--->1

       1

    2      3

 4   5   6   7

8     9     10
```

38 swaps

B:

Building a Max Heap:

10

9 8

7 6 5 4

3 2 1

Child: 9 [1]

Parent: 10 [0]

10

9 8

7 6 5 4

3 2 1

Child: 8 [2]

Parent: 10 [0]

10

9 8

7 6 5 4

3 2 1

Child: 7 [3]

Parent: 9 [1]

10

9 8

7 6 5 4

3 2 1

```
Child: 6 [4]

Parent: 9 [1]



        10

     9       8

 7   6   5   4

3    2    1

Child: 5 [5]

Parent: 8 [2]



        10

     9       8

 7   6   5   4

3    2    1

Child: 4 [6]

Parent: 8 [2]



        10

     9       8

 7   6   5   4

3    2    1

Child: 3 [7]

Parent: 7 [3]



        10

     9       8

 7   6   5   4
```

```
3   2    1
```

Child: 2 [8]

Parent: 7 [3]

```
    10
  9    8
 7  6  5  4
3   2    1
```

Child: 1 [9]

Parent: 6 [4]

```
    10
  9    8
 7  6  5  4
3   2    1
```

Shrinking the Max Heap:

n = 10

```
    10
  9    8
 7  6  5  4
3   2    1
```

Swap: 10<--->1

```
    1
  9    8
 7  6  5  4
3   2    10
```

```
RightChild: 9 [1]
LeftChild: 8 [2]
MaxChild: 9 [1]
Parent: 1 [0]
Swap: 1<--->9
      9
   1     8
 7  6  5  4
3    2    10

RightChild: 7 [3]
LeftChild: 6 [4]
MaxChild: 7 [3]
Parent: 1 [1]
Swap: 1<--->7
      9
   7     8
 1  6  5  4
3    2    10

RightChild: 3 [7]
LeftChild: 2 [8]
MaxChild: 3 [7]
Parent: 1 [3]
Swap: 1<--->3
      9
   7     8
 3  6  5  4
1    2    10

n = 9
```

```
         9

     7       8

 3   6   5   4

1     2     10


Swap: 9<--->2

         2

     7       8

 3   6   5   4

1     9     10


RightChild: 7 [1]

LeftChild: 8 [2]

MaxChild: 8 [2]

Parent: 2 [0]

Swap: 2<--->8

         8

     7       2

 3   6   5   4

1     9     10


RightChild: 5 [5]

LeftChild: 4 [6]

MaxChild: 5 [5]

Parent: 2 [2]

Swap: 2<--->5

         8

     7       5

 3   6   2   4

1     9     10
```

```
n = 8
      8
   7     5
 3  6  2  4
1    9    10

Swap: 8<--->1
      1
   7     5
 3  6  2  4
8    9    10

RightChild: 7 [1]
LeftChild: 5 [2]
MaxChild: 7 [1]
Parent: 1 [0]
Swap: 1<--->7
      7
   1     5
 3  6  2  4
8    9    10

RightChild: 3 [3]
LeftChild: 6 [4]
MaxChild: 6 [4]
Parent: 1 [1]
Swap: 1<--->6
      7
   6     5
 3  1  2  4
8    9    10
```

```
n = 7
     7
   6     5
 3  1  2  4
8   9   10

Swap: 7<--->4
     4
   6     5
 3  1  2  7
8   9   10

RightChild: 6 [1]
LeftChild: 5 [2]
MaxChild: 6 [1]
Parent: 4 [0]
Swap: 4<--->6
     6
   4     5
 3  1  2  7
8   9   10

RightChild: 3 [3]
LeftChild: 1 [4]
MaxChild: 3 [3]
Parent: 4 [1]
n = 6
     6
   4     5
 3  1  2  7
```

```
8    9    10

Swap: 6<--->2
       2
   4     5
 3  1  6  7
8    9    10

RightChild: 4 [1]
LeftChild: 5 [2]
MaxChild: 5 [2]
Parent: 2 [0]
Swap: 2<--->5
       5
   4     2
 3  1  6  7
8    9    10

n = 5
       5
   4     2
 3  1  6  7
8    9    10

Swap: 5<--->1
       1
   4     2
 3  5  6  7
8    9    10

RightChild: 4 [1]
```

```
LeftChild: 2 [2]

MaxChild: 4 [1]

Parent: 1 [0]

Swap: 1<--->4

        4

    1       2

 3   5   6   7

8     9     10


RightChild: 3 [3]

LeftChild: 5 [4]

MaxChild: 3 [3]

Parent: 1 [1]

Swap: 1<--->3

        4

    3       2

 1   5   6   7

8     9     10


n = 4

        4

    3       2

 1   5   6   7

8     9     10


Swap: 4<--->1

        1

    3       2

 4   5   6   7

8     9     10
```

```
RightChild: 3 [1]
LeftChild: 2 [2]
MaxChild: 3 [1]
Parent: 1 [0]
Swap: 1<--->3
       3
    1      2
 4   5   6   7
8     9    10

n = 3
       3
    1      2
 4   5   6   7
8     9    10

Swap: 3<--->2
       2
    1      3
 4   5   6   7
8     9    10

RightChild: 1 [1]
LeftChild: 3 [2]
MaxChild: 1 [1]
Parent: 2 [0]
n = 2
       2
    1      3
 4   5   6   7
8     9    10
```

```
Swap: 2<--->1
         1
     2       3
 4   5   6   7
8     9     10

n = 1
         1
     2       3
 4   5   6   7
8     9     10

Swap: 1<--->1
         1
     2       3
 4   5   6   7
8     9     10
```

21 swaps

C:

Building a Max Heap:

```
          5
       2        13
   9    1    7    6
8    1     15    4     11
```

Child: 2 [1]

Parent: 5 [0]

```
          5
       2        13
   9    1    7    6
8    1     15    4     11
```

Child: 13 [2]

Parent: 5 [0]

Swap: 13<--->5

```
          13
       2        5
   9    1    7    6
8    1     15    4     11
```

Child: 9 [3]

Parent: 2 [1]

Swap: 9<--->2

```
      13

    9     5

  2   1   7   6

8   1   15  4   11

Child: 1 [4]

Parent: 9 [1]



      13

    9     5

  2   1   7   6

8   1   15  4   11

Child: 7 [5]

Parent: 5 [2]

Swap: 7<--->5



      13

    9     7

  2   1   5   6

8   1   15  4   11

Child: 6 [6]

Parent: 7 [2]



      13

    9     7
  2    1    5    6
8    1    15   4    11
Child: 8 [7]
Parent: 2 [3]
Swap: 8<--->2


        13
     9       7
  8    1    5    6
2    1    15   4    11
Child: 1 [8]
Parent: 8 [3]


        13
     9       7
  8    1    5    6
2    1    15   4    11
Child: 15 [9]
Parent: 1 [4]
Swap: 15<--->1

Child: 15 [4]
Parent: 9 [1]
Swap: 15<--->9

Child: 15 [1]
```

```
Parent: 13 [0]
Swap: 15<--->13


        15
     13      7
  8    9    5    6
2    1    1    4    11
Child: 4 [10]
Parent: 9 [4]


        15
     13      7
  8    9    5    6
2    1    1    4    11
Child: 11 [11]
Parent: 5 [5]
Swap: 11<--->5

Child: 11 [5]
Parent: 7 [2]
Swap: 11<--->7


        15
     13      11
  8    9    7    6
```

```
2    1    1    4    5

Shrinking the Max Heap:
n = 12
          15
      13       11
   8    9    7    6
2    1    1    4    5

Swap: 15<--->5
          5
      13       11
   8    9    7    6
2    1    1    4    15

RightChild: 13 [1]
LeftChild: 11 [2]
MaxChild: 13 [1]
Parent: 5 [0]
Swap: 5<--->13
          13
      5       11
   8    9    7    6
2    1    1    4    15

RightChild: 8 [3]
LeftChild: 9 [4]
MaxChild: 9 [4]
```

```
Parent: 5 [1]

Swap: 5<--->9

      13

    9     11

  8   5   7   6

2   1   1   4   15


RightChild: 1 [9]

LeftChild: 4 [10]

MaxChild: 4 [10]

Parent: 5 [4]

n = 11

      13

    9     11

  8   5   7   6

2   1   1   4   15


Swap: 13<--->4

      4

    9     11

  8   5   7   6

2   1   1   13   15


RightChild: 9 [1]

LeftChild: 11 [2]

MaxChild: 11 [2]

Parent: 4 [0]

Swap: 4<--->11
```

```
      11
    9      4
  8   5   7    6
2   1    1   13   15

RightChild: 7 [5]
LeftChild: 6 [6]
MaxChild: 7 [5]
Parent: 4 [2]
Swap: 4<--->7
      11
    9      7
  8   5   4    6
2   1    1   13   15

n = 10
      11
    9      7
  8   5   4    6
2   1    1   13   15

Swap: 11<--->1
      1
    9      7
  8   5   4    6
2   1    11   13   15

RightChild: 9 [1]
```

```
LeftChild: 7 [2]
MaxChild: 9 [1]
Parent: 1 [0]
Swap: 1<--->9
       9
     1      7
  8    5   4    6
2    1   11   13   15

RightChild: 8 [3]
LeftChild: 5 [4]
MaxChild: 8 [3]
Parent: 1 [1]
Swap: 1<--->8
       9
     8      7
  1    5   4    6
2    1   11   13   15

RightChild: 2 [7]
LeftChild: 1 [8]
MaxChild: 2 [7]
Parent: 1 [3]
Swap: 1<--->2
       9
     8      7
  2    5   4    6
1    1   11   13   15
```

```
n = 9
         9
      8       7
  2     5     4     6
1     1     11    13    15

Swap: 9<--->1
         1
      8       7
  2     5     4     6
1     9     11    13    15

RightChild: 8 [1]
LeftChild: 7 [2]
MaxChild: 8 [1]
Parent: 1 [0]
Swap: 1<--->8
         8
      1       7
  2     5     4     6
1     9     11    13    15

RightChild: 2 [3]
LeftChild: 5 [4]
MaxChild: 5 [4]
Parent: 1 [1]
Swap: 1<--->5
```

```
          8
      5       7
  2    1    4    6
1    9    11    13    15

n = 8
          8
      5       7
  2    1    4    6
1    9    11    13    15

Swap: 8<--->1
          1
      5       7
  2    1    4    6
8    9    11    13    15

RightChild: 5 [1]
LeftChild: 7 [2]
MaxChild: 7 [2]
Parent: 1 [0]
Swap: 1<--->7
          7
      5       1
  2    1    4    6
8    9    11    13    15

RightChild: 4 [5]
```

```
LeftChild: 6 [6]

MaxChild: 6 [6]

Parent: 1 [2]

Swap: 1<--->6

          7

      5        6

   2     1     4     1

8     9     11    13    15


n = 7

          7

      5        6

   2     1     4     1

8     9     11    13    15


Swap: 7<--->1

          1

      5        6

   2     1     4     7

8     9     11    13    15


RightChild: 5 [1]

LeftChild: 6 [2]

MaxChild: 6 [2]

Parent: 1 [0]

Swap: 1<--->6

          6

      5        1
```

```
  2    1    4    7
8    9    11    13    15

RightChild: 4 [5]
LeftChild: 7 [6]
MaxChild: 4 [5]
Parent: 1 [2]
Swap: 1<--->4
         6
      5       4
  2    1    1    7
8    9    11    13    15

n = 6
         6
      5       4
  2    1    1    7
8    9    11    13    15

Swap: 6<--->1
         1
      5       4
  2    1    6    7
8    9    11    13    15

RightChild: 5 [1]
LeftChild: 4 [2]
MaxChild: 5 [1]
```

```
Parent: 1 [0]
Swap: 1<--->5
          5
      1        4
  2    1    6    7
8    9    11   13   15

RightChild: 2 [3]
LeftChild: 1 [4]
MaxChild: 2 [3]
Parent: 1 [1]
Swap: 1<--->2
          5
      2        4
  1    1    6    7
8    9    11   13   15

n = 5
          5
      2        4
  1    1    6    7
8    9    11   13   15

Swap: 5<--->1
          1
      2        4
  1    5    6    7
8    9    11   13   15
```

```
RightChild: 2 [1]

LeftChild: 4 [2]

MaxChild: 4 [2]

Parent: 1 [0]

Swap: 1<--->4

          4

      2         1

   1     5     6     7

8     9     11    13    15


n = 4

          4

      2         1

   1     5     6     7

8     9     11    13    15


Swap: 4<--->1

          1

      2         1

   4     5     6     7

8     9     11    13    15


RightChild: 2 [1]

LeftChild: 1 [2]

MaxChild: 2 [1]

Parent: 1 [0]

Swap: 1<--->2
```

```
        2
     1       1
  4    5    6    7
8    9    11    13    15

n = 3
        2
     1       1
  4    5    6    7
8    9    11    13    15

Swap: 2<--->1
        1
     1       2
  4    5    6    7
8    9    11    13    15

RightChild: 1 [1]
LeftChild: 2 [2]
MaxChild: 1 [1]
Parent: 1 [0]
n = 2
        1
     1       2
  4    5    6    7
8    9    11    13    15

Swap: 1<--->1
```

```
       1
    1      2
  4    5    6    7
8    9    11    13    15
```

n = 1

```
       1
    1      2
  4    5    6    7
8    9    11    13    15
```

Swap: 1<--->1

```
       1
    1      2
  4    5    6    7
8    9    11    13    15
```

36 swaps

D:

Building a Max Heap:

```
      S
    B     I
  M   H   Q   C
L   R   E   P   K
```

Child: B [1]

Parent: S [0]

```
      S
    B     I
  M   H   Q   C
L   R   E   P   K
```

Child: I [2]

Parent: S [0]

```
      S
    B     I
  M   H   Q   C
L   R   E   P   K
```

Child: M [3]

Parent: B [1]

Swap: M<--->B

```
      S
    M     I
  B   H   Q   C
```

L R E P K

Child: H [4]

Parent: M [1]

S

M I

B H Q C

L R E P K

Child: Q [5]

Parent: I [2]

Swap: Q<--->I

S

M Q

B H I C

L R E P K

Child: C [6]

Parent: Q [2]

S

M Q

B H I C

L R E P K

Child: L [7]

Parent: B [3]

Swap: L<--->B

S

M Q

L H I C

B R E P K

Child: R [8]

Parent: L [3]

Swap: R<--->L

Child: R [3]

Parent: M [1]

Swap: R<--->M

S

R Q

M H I C

B L E P K

Child: E [9]

Parent: H [4]

S

R Q

M H I C

B L E P K

Child: P [10]

Parent: H [4]

Swap: P<--->H

S

```
      R         Q
   M      P     I     C
B     L     E     H     K
```

Child: K [11]

Parent: I [5]

Swap: K<--->I

```
          S
      R         Q
   M      P     K     C
B     L     E     H     I
```

Shrinking the Max Heap:

n = 12

```
          S
      R         Q
   M      P     K     C
B     L     E     H     I
```

Swap: S<--->I

```
          I
      R         Q
   M      P     K     C
B     L     E     H     S
```

RightChild: R [1]

LeftChild: Q [2]

MaxChild: R [1]

Parent: I [0]

Swap: I<--->R

R

I Q

M P K C

B L E H S

RightChild: M [3]

LeftChild: P [4]

MaxChild: P [4]

Parent: I [1]

Swap: I<--->P

R

P Q

M I K C

B L E H S

RightChild: E [9]

LeftChild: H [10]

MaxChild: H [10]

Parent: I [4]

n = 11

R

P Q

M I K C

B L E H S

Swap: R<--->H

H

P Q

M I K C

B L E R S

RightChild: P [1]

LeftChild: Q [2]

MaxChild: Q [2]

Parent: H [0]

Swap: H<--->Q

```
       Q
    P     H
  M   I  K   C
B   L   E   R   S
```

RightChild: K [5]

LeftChild: C [6]

MaxChild: K [5]

Parent: H [2]

Swap: H<--->K

```
       Q
    P     K
  M   I  H   C
B   L   E   R   S
```

n = 10

```
       Q
    P     K
  M   I  H   C
B   L   E   R   S
```

Swap: Q<--->E

```
       E
    P     K
  M   I  H   C
B   L   Q   R   S
```

RightChild: P [1]

LeftChild: K [2]

MaxChild: P [1]

Parent: E [0]

Swap: E<--->P

```
       P
     E      K
  M    I    H    C
B    L    Q    R    S
```

RightChild: M [3]

LeftChild: I [4]

MaxChild: M [3]

Parent: E [1]

Swap: E<--->M

```
       P
     M       K
  E    I    H    C
B    L    Q    R    S
```

RightChild: B [7]

LeftChild: L [8]

MaxChild: L [8]

Parent: E [3]

Swap: E<--->L

```
       P
     M       K
  L    I    H    C
B    E    Q    R    S
```

```
n = 9
      P
    M      K
  L   I   H   C
B   E   Q   R   S

Swap: P<--->E
      E
    M      K
  L   I   H   C
B   P   Q   R   S

RightChild: M [1]
LeftChild: K [2]
MaxChild: M [1]
Parent: E [0]
Swap: E<--->M
      M
    E     K
  L   I   H   C
B   P   Q   R   S

RightChild: L [3]
LeftChild: I [4]
MaxChild: L [3]
Parent: E [1]
Swap: E<--->L
      M
    L     K
  E   I   H   C
B   P   Q   R   S
```

RightChild: B [7]

LeftChild: P [8]

MaxChild: B [7]

Parent: E [3]

n = 8

```
        M
     L      K
  E   I   H   C
B   P   Q   R   S
```

Swap: M<--->B

```
        B
     L      K
  E   I   H   C
M   P   Q   R   S
```

RightChild: L [1]

LeftChild: K [2]

MaxChild: L [1]

Parent: B [0]

Swap: B<--->L

```
        L
     B      K
  E   I   H   C
M   P   Q   R   S
```

RightChild: E [3]

LeftChild: I [4]

MaxChild: I [4]

Parent: B [1]

Swap: B<--->I

```
        L
     I      K
  E    B    H    C
M    P    Q    R    S
```

n = 7

```
        L
     I      K
  E    B    H    C
M    P    Q    R    S
```

Swap: L<--->C

```
        C
     I      K
  E    B    H    L
M    P    Q    R    S
```

RightChild: I [1]

LeftChild: K [2]

MaxChild: K [2]

Parent: C [0]

Swap: C<--->K

```
        K
     I      C
  E    B    H    L
M    P    Q    R    S
```

RightChild: H [5]

LeftChild: L [6]

MaxChild: H [5]

Parent: C [2]

Swap: C<--->H

```
      K
    I     H
  E   B   C   L
M    P   Q   R   S
```

n = 6

```
      K
    I     H
  E   B   C   L
M    P   Q   R   S
```

Swap: K<--->C

```
      C
    I     H
  E   B   K   L
M    P   Q   R   S
```

RightChild: I [1]

LeftChild: H [2]

MaxChild: I [1]

Parent: C [0]

Swap: C<--->I

```
      I
    C     H
  E   B   K   L
M    P   Q   R   S
```

RightChild: E [3]

LeftChild: B [4]

MaxChild: E [3]

Parent: C [1]

Swap: C<--->E

```
      I
    E     H
  C   B   K   L
M   P   Q   R   S
```

n = 5

```
      I
    E     H
  C   B   K   L
M   P   Q   R   S
```

Swap: I<--->B

```
      B
    E     H
  C   I  K   L
M   P   Q   R   S
```

RightChild: E [1]

LeftChild: H [2]

MaxChild: H [2]

Parent: B [0]

Swap: B<--->H

```
      H
    E     B
  C   I  K   L
M   P   Q   R   S
```

n = 4

```
    H
  E    B
 C  I  K  L
M  P  Q  R  S
```

Swap: H<--->C

```
    C
  E    B
 H  I  K  L
M  P  Q  R  S
```

RightChild: E [1]

LeftChild: B [2]

MaxChild: E [1]

Parent: C [0]

Swap: C<--->E

```
    E
  C    B
 H  I  K  L
M  P  Q  R  S
```

n = 3

```
    E
  C    B
 H  I  K  L
M  P  Q  R  S
```

Swap: E<--->B

```
    B
  C    E
 H  I  K  L
```

```
M    P    Q    R    S
```

RightChild: C [1]

LeftChild: E [2]

MaxChild: C [1]

Parent: B [0]

Swap: B<--->C

```
       C
    B      E
  H    I    K    L
M    P    Q    R    S
```

n = 2

```
       C
    B      E
  H    I    K    L
M    P    Q    R    S
```

Swap: C<--->B

```
       B
    C      E
  H    I    K    L
M    P    Q    R    S
```

n = 1

```
       B
    C      E
  H    I    K    L
M    P    Q    R    S
```

Swap: B<--->B

B

C E

H I K L

M P Q R S

36 swaps

Quick Sort:

A:

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]


Up Border: 0
Up Border: 1
Down Border: 9
Down Border: 8
Down Border: 7
Down Border: 6
Down Border: 5
Down Border: 4
Down Border: 3
Down Border: 2
Down Border: 1
Down Border: 0
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
Swap: 1 <---> 1
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]


[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
Up Border: 1
Up Border: 2
Down Border: 9
Down Border: 8
Down Border: 7
Down Border: 6
Down Border: 5
Down Border: 4
Down Border: 3

Down Border: 2

Down Border: 1

[2, 3, 4, 5, 6, 7, 8, 9, 10]

Swap: 2 <---> 2

[2, 3, 4, 5, 6, 7, 8, 9, 10]


[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

Up Border: 2

Up Border: 3

Down Border: 9

Down Border: 8

Down Border: 7

Down Border: 6

Down Border: 5

Down Border: 4

Down Border: 3

Down Border: 2

[3, 4, 5, 6, 7, 8, 9, 10]

Swap: 3 <---> 3

[3, 4, 5, 6, 7, 8, 9, 10]


[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

Up Border: 3

Up Border: 4

Down Border: 9

Down Border: 8

Down Border: 7

Down Border: 6
Down Border: 5

Down Border: 4

Down Border: 3

[4, 5, 6, 7, 8, 9, 10]

Swap: 4 <---> 4

[4, 5, 6, 7, 8, 9, 10]


[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

Up Border: 4

Up Border: 5

Down Border: 9

Down Border: 8

Down Border: 7

Down Border: 6

Down Border: 5

Down Border: 4

[5, 6, 7, 8, 9, 10]

Swap: 5 <---> 5

[5, 6, 7, 8, 9, 10]


[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

Up Border: 5

Up Border: 6

Down Border: 9

Down Border: 8

Down Border: 7

Down Border: 6

Down Border: 5

[6, 7, 8, 9, 10]
```

```
Swap: 6 <---> 6
[6, 7, 8, 9, 10]

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
Up Border: 6
Up Border: 7
Down Border: 9
Down Border: 8
Down Border: 7
Down Border: 6
[7, 8, 9, 10]
Swap: 7 <---> 7
[7, 8, 9, 10]

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
Up Border: 7
Up Border: 8
Down Border: 9
Down Border: 8
Down Border: 7
[8, 9, 10]
Swap: 8 <---> 8
[8, 9, 10]

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
Up Border: 8
Up Border: 9
Down Border: 9
Down Border: 8
```

```
[9, 10]
Swap: 9 <---> 9
[9, 10]

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

0 swaps
```

B:

```
[10, 9, 8, 7, 6, 5, 4, 3, 2, 1]

Up Border: 0
Up Border: 1
Up Border: 2
Up Border: 3
Up Border: 4
Up Border: 5
Up Border: 6
Up Border: 7
Up Border: 8
Up Border: 9
Down Border: 9
[10, 9, 8, 7, 6, 5, 4, 3, 2, 1]
Swap: 10 <---> 1
[1, 9, 8, 7, 6, 5, 4, 3, 2, 10]

[1, 9, 8, 7, 6, 5, 4, 3, 2, 10]
Up Border: 0
Up Border: 1
Down Border: 8
Down Border: 7
Down Border: 6
Down Border: 5
Down Border: 4
Down Border: 3
Down Border: 2
Down Border: 1
Down Border: 0
[1, 9, 8, 7, 6, 5, 4, 3, 2]
Swap: 1 <---> 1
[1, 9, 8, 7, 6, 5, 4, 3, 2]

[1, 9, 8, 7, 6, 5, 4, 3, 2, 10]
Up Border: 1
Up Border: 2
Up Border: 3
Up Border: 4
Up Border: 5
Up Border: 6
Up Border: 7
Up Border: 8
Down Border: 8
[9, 8, 7, 6, 5, 4, 3, 2]
Swap: 9 <---> 2
[2, 8, 7, 6, 5, 4, 3, 9]

[1, 2, 8, 7, 6, 5, 4, 3, 9, 10]
Up Border: 1
Up Border: 2
```

```
Down Border: 7
Down Border: 6
Down Border: 5
Down Border: 4
Down Border: 3
Down Border: 2
Down Border: 1
[2, 8, 7, 6, 5, 4, 3]
Swap: 2 <---> 2
[2, 8, 7, 6, 5, 4, 3]

[1, 2, 8, 7, 6, 5, 4, 3, 9, 10]
Up Border: 2
Up Border: 3
Up Border: 4
Up Border: 5
Up Border: 6
Up Border: 7
Down Border: 7
[8, 7, 6, 5, 4, 3]
Swap: 8 <---> 3
[3, 7, 6, 5, 4, 8]

[1, 2, 3, 7, 6, 5, 4, 8, 9, 10]
Up Border: 2
Up Border: 3
Down Border: 6
Down Border: 5
Down Border: 4
Down Border: 3
Down Border: 2
[3, 7, 6, 5, 4]
Swap: 3 <---> 3
[3, 7, 6, 5, 4]

[1, 2, 3, 7, 6, 5, 4, 8, 9, 10]
Up Border: 3
Up Border: 4
Up Border: 5
Up Border: 6
Down Border: 6
[7, 6, 5, 4]
Swap: 7 <---> 4
[4, 6, 5, 7]

[1, 2, 3, 4, 6, 5, 7, 8, 9, 10]
Up Border: 3
Up Border: 4
Down Border: 5
Down Border: 4
Down Border: 3
[4, 6, 5]
Swap: 4 <---> 4
[4, 6, 5]

[1, 2, 3, 4, 6, 5, 7, 8, 9, 10]
Up Border: 4
Up Border: 5
```

```
Down Border: 5
[6, 5]
Swap: 6 <---> 5
[5, 6]

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

5 swaps
```

C:

```
[5, 2, 13, 9, 1, 7, 6, 8, 1, 15, 4, 11]
Up Border: 0
Up Border: 1
Up Border: 2
Down Border: 11
Down Border: 10
[13, 9, 1, 7, 6, 8, 1, 15]
Swap: 13 <---> 4
[4, 9, 1, 7, 6, 8, 1, 15]

[5, 2, 4, 9, 1, 7, 6, 8, 1, 15, 13, 11]
Up Border: 2
Up Border: 3
Down Border: 10
Down Border: 9
Down Border: 8
[9, 1, 7, 6, 8]
Swap: 9 <---> 1
[1, 1, 7, 6, 8]
```

```
[5, 2, 4, 1, 1, 7, 6, 8, 9, 15, 13, 11]
Up Border: 3
Up Border: 4
Up Border: 5
Down Border: 8
Down Border: 7
Down Border: 6
Down Border: 5
Down Border: 4
[5, 2, 4, 1, 1, 7, 6, 8, 9, 15, 13, 11]
Swap: 5 <---> 1
[1, 2, 4, 1, 5, 7, 6, 8, 9, 15, 13, 11]

[1, 2, 4, 1, 5, 7, 6, 8, 9, 15, 13, 11]
Up Border: 0
Up Border: 1
Down Border: 3
[2, 4]
Swap: 2 <---> 1
[1, 4]

[1, 1, 4, 2, 5, 7, 6, 8, 9, 15, 13, 11]
Up Border: 1
Up Border: 2
Down Border: 3
Down Border: 2
Down Border: 1
[1, 1, 4, 2]
Swap: 1 <---> 1
```

```
[1, 1, 4, 2]


[1, 1, 4, 2, 5, 7, 6, 8, 9, 15, 13, 11]
Up Border: 2
Up Border: 3
Down Border: 3
[4, 2]
Swap: 4 <---> 2
[2, 4]


[1, 1, 2, 4, 5, 7, 6, 8, 9, 15, 13, 11]
Up Border: 5
Up Border: 6
Up Border: 7
Down Border: 11
Down Border: 10
Down Border: 9
Down Border: 8
Down Border: 7
Down Border: 6
[7, 6, 8, 9, 15, 13, 11]
Swap: 7 <---> 6
[6, 7, 8, 9, 15, 13, 11]


[1, 1, 2, 4, 5, 6, 7, 8, 9, 15, 13, 11]
Up Border: 7
Up Border: 8
Down Border: 11
Down Border: 10
Down Border: 9
Down Border: 8
Down Border: 7
[8, 9, 15, 13, 11]
Swap: 8 <---> 8
[8, 9, 15, 13, 11]

[1, 1, 2, 4, 5, 6, 7, 8, 9, 15, 13, 11]
Up Border: 8
Up Border: 9
Down Border: 11
Down Border: 10
Down Border: 9
Down Border: 8
[9, 15, 13, 11]
Swap: 9 <---> 9
[9, 15, 13, 11]

[1, 1, 2, 4, 5, 6, 7, 8, 9, 15, 13, 11]
Up Border: 9
Up Border: 10
Up Border: 11
Down Border: 11
[15, 13, 11]
Swap: 15 <---> 11
[11, 13, 15]

[1, 1, 2, 4, 5, 6, 7, 8, 9, 11, 13, 15]
Up Border: 9
Up Border: 10
Down Border: 10
Down Border: 9
[11, 13]
Swap: 11 <---> 11
[11, 13]

[1, 1, 2, 4, 5, 6, 7, 8, 9, 11, 13, 15]
7 swaps
```

D:

```
[S, B, I, M, H, Q, C, L, R, E, P, K]
Up Border: 0
Up Border: 1
Up Border: 2
Up Border: 3
Up Border: 4
Up Border: 5
Up Border: 6
Up Border: 7
Up Border: 8
Up Border: 9
Up Border: 10
Up Border: 11
Down Border: 11
[S, B, I, M, H, Q, C, L, R, E, P, K]
Swap: S <---> K
[K, B, I, M, H, Q, C, L, R, E, P, S]

[K, B, I, M, H, Q, C, L, R, E, P, S]
```

```
Up Border: 0
Up Border: 1
Up Border: 2
Up Border: 3
Down Border: 10
Down Border: 9
[M, H, Q, C, L, R]
Swap: M <---> E
[E, H, Q, C, L, R]

[K, B, I, E, H, Q, C, L, R, M, P, S]
Up Border: 3
Up Border: 4
Up Border: 5
Down Border: 9
Down Border: 8
Down Border: 7
Down Border: 6
[Q]
Swap: Q <---> C
[C]

[K, B, I, E, H, C, Q, L, R, M, P, S]
Up Border: 5
Up Border: 6
Down Border: 6
Down Border: 5
[K, B, I, E, H, C, Q, L, R, M, P]
Swap: K <---> C
```

```
[C, B, I, E, H, K, Q, L, R, M, P]

[C, B, I, E, H, K, Q, L, R, M, P, S]
Up Border: 0
Up Border: 1
Up Border: 2
Down Border: 4
Down Border: 3
Down Border: 2
Down Border: 1
[C, B, I, E, H]
Swap: C <---> B
[B, C, I, E, H]

[B, C, I, E, H, K, Q, L, R, M, P, S]
Up Border: 2
Up Border: 3
Up Border: 4
Down Border: 4
[I, E, H]
Swap: I <---> H
[H, E, I]

[B, C, H, E, I, K, Q, L, R, M, P, S]
Up Border: 2
Up Border: 3
Down Border: 3
[H, E]
Swap: H <---> E
```

```
[E, H]

[B, C, E, H, I, K, Q, L, R, M, P, S]
Up Border: 6
Up Border: 7
Up Border: 8
Down Border: 10
[R, M]
Swap: R <---> P
[P, M]

[B, C, E, H, I, K, Q, L, P, M, R, S]
Up Border: 8
Up Border: 9
Up Border: 10
Down Border: 10
Down Border: 9
[Q, L, P, M, R]
Swap: Q <---> M
[M, L, P, Q, R]

[B, C, E, H, I, K, M, L, P, Q, R, S]
Up Border: 6
Up Border: 7
Up Border: 8
Down Border: 8
Down Border: 7
[M, L, P]
Swap: M <---> L
```

[L, M, P]

[B, C, E, H, I, K, L, M, P, Q, R, S]

10 swaps