# Q1

## 1) A_is an ordered integer array with 10 elements from small to large.

- ***Applying Shell sort:***

*A = {1,3,4,6,7,9,11,17,18,20}*

*n = 10;*

| [0] | [1] | [2] | [3] | [4] | [5] | [6] | [7] | [8] | [9] |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 |

*Our first gap is n/2.*

*Gap = n / 2 = 10 / 2 = 5.*

*İnitial array:*

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 |

***When gap = 5;***

| Subarray | Comparing elements | [0] | [1] | [2] | [3] | [4] | [5] | [6] | [7] | [8] | [9] | isChanged |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 - 5 | 0 - 5 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 1 – 6 | 1 – 6 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 2 - 7 | 2 - 7 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 3 – 8 | 3 – 8 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 4 - 9 | 4 - 9 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |

***Note:*** ***The blue boxes are the array's current state after the comparison process.***

***When gap is equal 5, 5 comparison and 0 displacement done.***

***gap = (int) (gap / 2.2);***

***gap = (int) (5 / 2.2) = 2.***

***When gap = 2;***

***İnitial array:***

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 |

| Subarray | Comparing elements | [0] | [1] | [2] | [3] | [4] | [5] | [6] | [7] | [8] | [9] | isChanged |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 - 2 | 0 - 2 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 1 - 3 | 1 - 3 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 0 - 2 - 4 | 2 - 4 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 0 - 2 - 4 | 0 – 2 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 1 - 3 - 5 | 3 - 5 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 1 - 3 - 5 | 1 - 3 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 0 - 2 - 4 - 6 | 4 - 6 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 0 - 2 - 4 - 6 | 2 - 4 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 0 - 2 - 4 - 6 | 0 - 2 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 1 - 3 - 5 - 7 | 5 - 7 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 1 - 3 - 5 - 7 | 3 - 5 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 1 - 3 - 5 - 7 | 1 - 3 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 0 - 2 - 4 - 6 - 8 | 6 - 8 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 0 - 2 - 4 - 6 – 8 | 4 - 6 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 0 - 2 - 4- 6 - 8 | 2 - 4 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 - 2 - 4 - 6 - 8 | 0 - 2 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 1 - 3 - 5 - 7 - 9 | 7 - 9 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 1 - 3 - 5 - 7 – 9 | 5 - 7 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 1 - 3 - 5 - 7 - 9 | 3 - 5 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 1 - 3 - 5 - 7 - 9 | 1 - 3 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |

***Note:*** ***The blue boxes are the array's current state after the comparison process.***

***When gap is equal 2, 20 comparison and 0 displacement done.***

***When gap is 2, we divide by 2.***

***gap = (int) (gap / 2);***

***gap = (int) (2 / 2) = 1.***

***When gap = 1;***

***İnitial array:***

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 |

***As we can see, array is already sorted but we continue the sorting with gap = 1.***

| Subarray | Comparing elements | [0] | [1] | [2] | [3] | [4] | [5] | [6] | [7] | [8] | [9] | isChanged |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 - 1 | 0 - 1 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 0 - 1 - 2 | 1 - 2 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 0 - 1 - 2 | 0 - 1 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 0 - 1 - 2 -3 | 2 -3 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 0 - 1 - 2 -3 | 1 - 2 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 0 - 1 - 2 -3 | 0 - 1 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 0 - 1 - 2 -3 - 4 | 3 - 4 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 - 1 - 2 -3 - 4 | 2 -3 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 0 - 1 - 2 -3 - 4 | 1 - 2 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 0 - 1 - 2 -3 - 4 | 0 - 1 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 0 - 1 - 2 -3 - 4 - 5 | 4 - 5 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 0 - 1 - 2 -3 - 4 – 5 | 3 - 4 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 0 - 1 - 2 -3 - 4 – 5 | 2 -3 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 0 - 1 - 2 -3 - 4 – 5 | 1 - 2 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 0 - 1 - 2 -3 - 4 - 5 | 0 - 1 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 0 - 1 - 2 -3 - 4 - 5 - 6 | 5 - 6 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 | 4 - 5 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 | 3 - 4 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 0 - 1 - 2 -3 - 4 - 5 - 6 | 2 -3 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 | 1 - 2 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 0 - 1 - 2 -3 - 4 - 5 - 6 | 0 - 1 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 | 6 -7 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 | 5 - 6 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 | 4 - 5 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 | 3 - 4 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 | 2 -3 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 | 1 - 2 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 | 0 - 1 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 - 8 | 7 - 8 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 | 6 -7 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 | 5 - 6 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 | 4 - 5 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 | 3 - 4 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 | 2 -3 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 | 1 - 2 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 - 8 | 0 - 1 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 - 9 | 8 - 9 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 – 9 | 7 - 8 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 – 9 | 6 -7 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 – 9 | 5 - 6 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 – 9 | 4 - 5 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 – 9 | 3 - 4 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 – 9 | 2 -3 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 – 9 | 1 - 2 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 - 9 | 0 - 1 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |

***Note:** The blue boxes are the array's current state after the comparison process.*

***When gap is equal 1, 45 comparison and 0 displacement done.***

***In total, 70 comparison and 0 displacement done.***

- ***Applying Merge sort:***

***A = {1,3,4,6,7,9,11,17,18,20}***

***n = 10;***

| [0] | [1] | [2] | [3] | [4] | [5] | [6] | [7] | [8] | [9] |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 |

***n = n / 2*** ***n = n / 2***

| | | | | |
|---|---|---|---|---|
| 1 | 3 | 4 | 6 | 7 |

| | | | | |
|---|---|---|---|---|
| 9 | 11 | 17 | 18 | 20 |

| | |
|---|---|
| 1 | 3 |

| | | |
|---|---|---|
| 4 | 6 | 7 |

| |
|---|
| 1 |

| |
|---|
| 3 |

***1) Compare 1 and 3. And associate as sorted version.***

***Output:***

| | |
|---|---|
| 1 | 3 |

***2)***

| | | |
|---|---|---|
| 4 | 6 | 7 |

| |
|---|
| 4 |

| | |
|---|---|
| 6 | 7 |

| |
|---|
| 6 |

| |
|---|
| 7 |

3) *Compare 6 and 7. And associate as sorted version.*

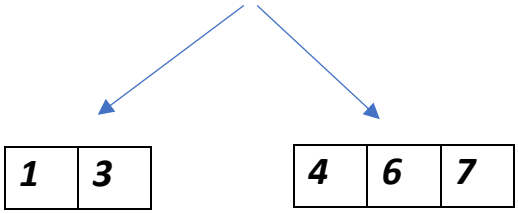
**Output**

| | |
|---|---|
| 6 | 7 |

4) *Compare* | 4 | *and* | 6 | 7 |

- *Compare 4 – 6 →* | 4 |
- *Add 6 →* | 4 | 6 |
- *Add 7 →* | 4 | 6 | 7 |

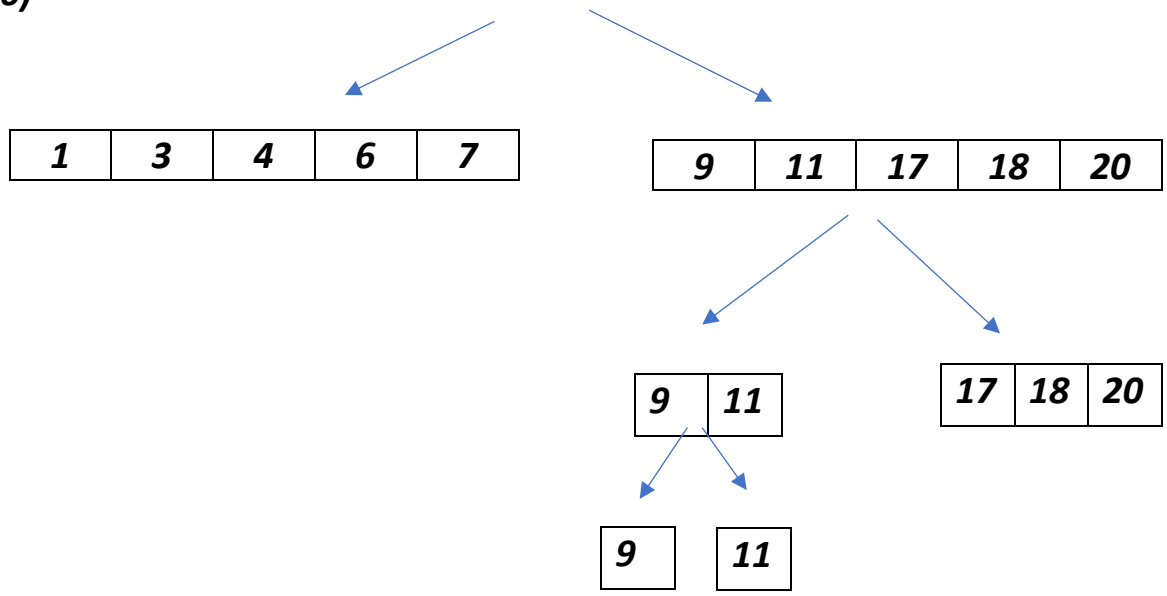
5)




| | | | | |
|---|---|---|---|---|
| 1 | 3 | | 4 | 6 | 7 |

- *Compare 1 – 4 →* | 1 |
- *Compare 3 – 4 →* | 1 | 3 |
- *Add 4 →* | 1 | 3 | 4 |
- *Add 6 →* | 1 | 3 | 4 | 6 |
- *Add 7 →* | 1 | 3 | 4 | 6 | 7 |

6)

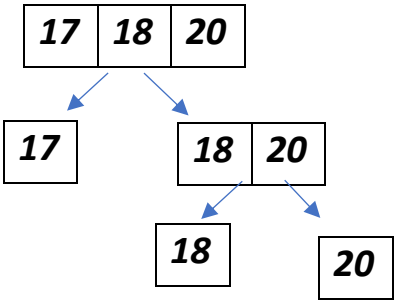



7) Compare 9 and 11. And associate as sorted version.

**Output:**

| | |
|---|---|
| 9 | 11 |

8)




9) Compare 18 and 20. And associate as sorted version.

**Output:**

| | |
|---|---|
| 18 | 20 |

***10)*** ***Compare*** | 17 | ***and*** | 18 | 20 |

- ***Compare 17 – 18 →*** | 17 |
- ***Add 18 →*** | 17 | 18 |
- ***Add 20 →*** | 17 | 18 | 20 |

***11)***

| 9 | 11 |

| 17 | 18 | 20 |

| 9 | 11 |

| 17 | 18 | 20 |

- ***Compare 9 – 17 →*** | 9 |
- ***Compare 11 – 17 →*** | 9 | 11 |
- ***Add 17 →*** | 9 | 11 | 17 |
- ***Add 18 →*** | 9 | 11 | 17 | 18 |
- ***Add 20 →*** | 9 | 11 | 17 | 18 | 20 |

12)

| | | | | |
|---|---|---|---|---|
| 1 | 3 | 4 | 6 | 7 |

| | | | | |
|---|---|---|---|---|
| 9 | 11 | 17 | 18 | 20 |

- Compare 1 - 9 → | 1 |
- Compare 3 – 9 → | 1 | 3 |
- Compare 4 – 9 → | 1 | 3 | 4 |
- Compare 6 – 9 → | 1 | 3 | 4 | 6 |
- Compare 7 – 9 → | 1 | 3 | 4 | 6 | 7 |
- Add 9 → | 1 | 3 | 4 | 6 | 7 | 9 |
- Add 11 → | 1 | 3 | 4 | 6 | 7 | 9 | 11 |
- Add 17 → | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 |
- Add 18 → | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 |
- Add 20 → | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 |

Sorted array:

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 |

**In total, 15 comparison and 0 displacement done.**

- ***Applying heap sort:***

***A = {1,3,4,6,7,9,11,17,18,20}***

***n = 10;***

| [0] | [1] | [2] | [3] | [4] | [5] | [6] | [7] | [8] | [9] |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 |

***We have to do two steps for converting that array to the increasing order array.***

1) ***Build a max heap from array.***
2) ***Shrink the heap.***

***<u>Build process</u>***

```
private <T extends Comparable<T>> void buildHeap(T[] table) {
        int n = 1;
        // Invariant: table[0 . . . n - 1] is a heap.
        while (n < table.length) {
                n++; // Add a new item to the heap and reheap.
                int child = n - 1;
                int parent = (child - 1) / 2; // Find parent.
                while (parent >= 0
                && table[parent].compareTo(table[child]) < 0) {
                swap(table, parent, child);
                child = parent;
                parent = (child - 1) / 2;
        }
}
```

***I start to control of child parent relation from index 1. Every index which I controled, I looked at its parent up to root. If necessary I swap the values. Parent = (child-1)/2***

***İnital array:***

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 |

| Current child index | Comparing elements | [0] | [1] | [2] | [3] | [4] | [5] | [6] | [7] | [8] | [9] | isChanged |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 – 0 | 3 | 1 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | True |
| 2 | 2 – 0 | 4 | 1 | 3 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | True |
| 3 | 3 – 1 | 4 | 6 | 3 | 1 | 7 | 9 | 11 | 17 | 18 | 20 | True |
| 3 | 1 – 0 | 6 | 4 | 3 | 1 | 7 | 9 | 11 | 17 | 18 | 20 | True |
| 4 | 4 – 1 | 6 | 7 | 3 | 1 | 4 | 9 | 11 | 17 | 18 | 20 | True |
| 4 | 1 – 0 | 7 | 6 | 3 | 1 | 4 | 9 | 11 | 17 | 18 | 20 | True |
| 5 | 5 – 2 | 7 | 6 | 9 | 1 | 4 | 3 | 11 | 17 | 18 | 20 | True |
| 5 | 2 – 0 | 9 | 6 | 7 | 1 | 4 | 3 | 11 | 17 | 18 | 20 | True |
| 6 | 6 – 2 | 9 | 6 | 11 | 1 | 4 | 3 | 7 | 17 | 18 | 20 | True |
| 6 | 2 – 0 | 11 | 6 | 9 | 1 | 4 | 3 | 7 | 17 | 18 | 20 | True |
| 7 | 7 – 3 | 11 | 6 | 9 | 17 | 4 | 3 | 7 | 1 | 18 | 20 | True |
| 7 | 3 – 1 | 11 | 17 | 9 | 6 | 4 | 3 | 7 | 1 | 18 | 20 | True |
| 7 | 1 – 0 | 17 | 11 | 9 | 6 | 4 | 3 | 7 | 1 | 18 | 20 | True |
| 8 | 8 – 3 | 17 | 11 | 9 | 18 | 4 | 3 | 7 | 1 | 6 | 20 | True |
| 8 | 3 – 1 | 17 | 18 | 9 | 11 | 4 | 3 | 7 | 1 | 6 | 20 | True |
| 8 | 1 – 0 | 18 | 17 | 9 | 11 | 4 | 3 | 7 | 1 | 6 | 20 | True |
| 9 | 9 – 4 | 18 | 17 | 9 | 11 | 20 | 3 | 7 | 1 | 6 | 4 | True |
| 9 | 4 – 1 | 18 | 20 | 9 | 11 | 17 | 3 | 7 | 1 | 6 | 4 | True |

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ***9*** | ***1 – 0*** | ***20*** | ***18*** | ***9*** | ***11*** | ***17*** | ***3*** | ***7*** | ***1*** | ***6*** | ***4*** | ***True*** |

***In build process, 19 comparison and 19 displacement done***

***Max heap:***

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| ***20*** | ***18*** | ***9*** | ***11*** | ***17*** | ***3*** | ***7*** | ***1*** | ***6*** | ***4*** |

## ***Shrink process***

```
private <T extends Comparable<T>> void shrinkHeap(T[] table) {
	int n = table.length;
	// Invariant: table[0 . . . n - 1] forms a heap.
	// table[n . . . table.length - 1] is sorted.
	while (n > 0) {
		n--;
		swap(table, 0, n);
		// table[1 . . . n - 1] form a heap.
		// table[n . . . table.length - 1] is sorted.
		int parent = 0;
		while (true) {
			int leftChild = 2 * parent + 1;
			if (leftChild >= n) {
				break; // No more children.
			}
			int rightChild = leftChild + 1;
			// Find the larger of the two children.
			int maxChild = leftChild;
			if (rightChild < n // There is a right child.
			&& table[leftChild].compareTo(table[rightChild]) < 0) {
			maxChild = rightChild;
			}
			// If the parent is smaller than the larger child,
			if (table[parent].compareTo(table[maxChild]) < 0){
				// Swap the parent and child.
				swap(table, parent, maxChild);
				// Continue at the child level.
				parent = maxChild;
	} else { // Heap property is restored.
		break; // Exit the loop.
	}
```

```
        }
        }
}
```

***Left child = 2 * parent + 1***

***Right child = 2 * parent + 2***

***While the blue box represents the current index, the green box represents the sorted section.***

| Left child index | Right child index | Compare notes | [0] | [1] | [2] | [3] | [4] | [5] | [6] | [7] | [8] | [9] | isChanged |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| - | - | Swap the first value and last value | 4 | 18 | 9 | 11 | 17 | 3 | 7 | 1 | 6 | 20 | True |
| 1 | 2 | [1] > [2] Swap [0] and [1] | 18 | 4 | 9 | 11 | 17 | 3 | 7 | 1 | 6 | 20 | True |
| 3 | 4 | [4] > [3] Swap [1] and [4] | 18 | 17 | 9 | 11 | 4 | 3 | 7 | 1 | 6 | 20 | True |
| 9 | 10 | There is no 9th and 10th element. | 18 | 17 | 9 | 11 | 4 | 3 | 7 | 1 | 6 | 20 | False |
| - | - | Swap the first value and last value | 6 | 17 | 9 | 11 | 4 | 3 | 7 | 1 | 18 | 20 | True |
| 1 | 2 | [1] > [2] Swap [0] and [1] | 17 | 6 | 9 | 11 | 4 | 3 | 7 | 1 | 18 | 20 | True |

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 3 | 4 | [3] > [4] Swap [1] and [3] | 17 | 11 | 9 | 6 | 4 | 3 | 7 | 1 | 18 | 20 | True |
| 7 | 8 | There is no 8th element | 17 | 11 | 9 | 6 | 4 | 3 | 7 | 1 | 18 | 20 | True |
| - | - | Swap the first value and last value | 1 | 11 | 9 | 6 | 4 | 3 | 7 | 17 | 18 | 20 | True |
| 1 | 2 | [1] > [2] Swap [0] and [1] | 11 | 1 | 9 | 6 | 4 | 3 | 7 | 17 | 18 | 20 | True |
| 3 | 4 | [3] > [4] Swap [1] and [3] | 11 | 6 | 9 | 1 | 4 | 3 | 7 | 17 | 18 | 20 | True |
| 7 | 8 | There is no 7th and 8th element | 11 | 6 | 9 | 1 | 4 | 3 | 7 | 17 | 18 | 20 | False |
| - | - | Swap the first value and last value | 7 | 6 | 9 | 1 | 4 | 3 | 11 | 17 | 18 | 20 | True |
| 1 | 2 | [2] > [1] Swap [0] and [2] | 9 | 6 | 7 | 1 | 4 | 3 | 11 | 17 | 18 | 20 | True |
| 5 | 6 | There is no 6th element | 9 | 6 | 7 | 1 | 4 | 3 | 11 | 17 | 18 | 20 | False |
| - | - | Swap the first value and last value | 3 | 6 | 7 | 1 | 4 | 9 | 11 | 17 | 18 | 20 | True |

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | [2] > [1] Swap [0] and [2] | 7 | 6 | 3 | 1 | 4 | 9 | 11 | 17 | 18 | 20 | True |
| 5 | 6 | There is no 6th element | 7 | 6 | 3 | 1 | 4 | 9 | 11 | 17 | 18 | 20 | False |
| - | - | Swap the first value and last value | 4 | 6 | 3 | 1 | 7 | 9 | 11 | 17 | 18 | 20 | True |
| 1 | 2 | [1] > [2] Swap [0] and [1] | 6 | 4 | 3 | 1 | 7 | 9 | 11 | 17 | 18 | 20 | True |
| 3 | 4 | There is no 4 the element | 6 | 4 | 3 | 1 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| - | - | Swap the first value and last value | 1 | 4 | 3 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | True |
| 1 | 2 | [1] > [2] Swap [0] and [1] | 4 | 1 | 3 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | True |
| 3 | 4 | There is no 3th and 4th the element | 4 | 1 | 3 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| - | - | Swap the first value and last value | 3 | 1 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | True |

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *1* | *2* | *There is no 2th element* | *3* | *1* | *4* | *6* | *7* | *9* | *11* | *17* | *18* | *20* | *False* |
| *-* | *-* | *Swap the first value and last value* | *1* | *3* | *4* | *6* | *7* | *9* | *11* | *17* | *18* | *20* | *True* |

***In shrink process, 27 comparison and 20 displacement done***

***Sorted array:***

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| *2* | *3* | *7* | *9* | *11* | *13* | *17* | *21* | *29* | *31* |

***In total, 46 comparison and 39 displacement done.***

- ***Applying QuickSort:***

***A = {1,3,4,6,7,9,11,17,18,20}***

***n = 10;***

| [0] | [1] | [2] | [3] | [4] | [5] | [6] | [7] | [8] | [9] |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 |

```
protected <T extends Comparable<T>> void quickSort(T[] table, int first, int last) {
        if (first < last) {
                int pivIndex = partition(table, first, last);
                quickSort(table, first, pivIndex - 1);
                quickSort(table, pivIndex + 1, last);
        }
}

protected <T extends Comparable<T>> int partition(T[] table, iint first, int last) {
        T pivot = table[first];
        int up = first; int down = last;
        do {
        while ((up < last) && (pivot.compareTo(table[up]) >= 0)) {
                up++;
        }
        while (pivot.compareTo(table[down]) < 0) {
                down--;
        }
        if (up < down) {
                swap(table, up, down);
        }
        } while (up < down); // Repeat while up is left of down.
        swap(table, first, down);
        return down;
}
```

➔ *Sort(table,0,9)*

*Pivot value = c[0] = 1*

| Compare Pivot-current | Compare Result | Up | down | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | isChanged |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 0 | 9 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | - |
| 1 – [0] | = | 1 | 9 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 1 – [1] | < | 1 | 9 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 1 – [9] | < | 1 | 8 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 1 – [8] | < | 1 | 7 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 1 – [7] | < | 1 | 6 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 1 – [6] | < | 1 | 5 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 1 – [5] | < | 1 | 4 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 1 – [4] | < | 1 | 3 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 1 – [3] | < | 1 | 2 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 1 – [2] | < | 1 | 1 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 1 – [1] | < | 1 | 0 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 1 – [0] | = | 1 | 0 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| | Swap(f,d) | 1 | 0 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |

*13 comparison and 0 displacement done and array did not change.*
*Because of f=0 and d=0*

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 |

→ *Sort(table,1,9)*
*Pivot value = c[1] = 3*

| Compare Pivot-current | Compare Result | Up | down | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | isChanged |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 1 | 9 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | - |
| 3 – [1] | = | 2 | 9 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 3 – [2] | < | 2 | 9 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 3 – [9] | < | 2 | 8 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 3 – [8] | < | 2 | 7 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 3 – [7] | < | 2 | 6 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 3 – [6] | < | 2 | 5 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 3 - [5] | < | 2 | 4 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 3 – [4] | < | 2 | 3 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 3 – [3] | < | 2 | 2 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 3 – [2] | < | 2 | 1 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 3 – [1] | = | 2 | 1 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| | Swap(f,d) | 2 | 1 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |

12 comparison and 0 displacement done and array did not change.
Because of f=1 and d=1

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 |

→ Sort(table,2,9)
Pivot value = c[2] = 4

| Compare Pivot-current | Compare Result | Up | down | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | isChanged |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 2 | 9 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | - |
| 4 – [2] | = | 3 | 9 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 4 – [3] | < | 3 | 9 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 4 – [9] | < | 3 | 8 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 4 – [8] | < | 3 | 7 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 4 – [7] | < | 3 | 6 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 4 – [6] | < | 3 | 5 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 4 - [5] | < | 3 | 4 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 4 – [4] | < | 3 | 3 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 4 – [3] | < | 3 | 2 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 4 – [2] | = | 3 | 2 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| | Swap(f,d) | 3 | 2 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |

*11 comparison and 0 displacement done and array did not change.*
*Because of f=2 and d=2*

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 |

→ *Sort(table,3,9)*
*Pivot value = c[3] = 6*

| Compare Pivot-current | Compare Result | Up | down | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | isChanged |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 3 | 9 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | - |
| 6 – [3] | = | 4 | 9 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 6 – [4] | < | 4 | 9 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 6 – [9] | < | 4 | 8 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 6 – [8] | < | 4 | 7 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 6 – [7] | < | 4 | 6 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 6 – [6] | < | 4 | 5 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 6 - [5] | < | 4 | 4 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 6 – [4] | < | 4 | 3 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 6 – [3] | = | 4 | 3 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| | Swap(f,d) | 4 | 3 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |

*10 comparison and 0 displacement done and array did not change.*
*Because of f=3 and d=3*

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 |

→ *Sort(table,4,9)*
*Pivot value = c[4] = 7*

| Compare Pivot-current | Compare Result | Up | down | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | isChanged |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 4 | 9 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | - |
| 7 – [4] | = | 5 | 9 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 7 – [5] | < | 5 | 9 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 7 – [9] | < | 5 | 8 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 7 – [8] | < | 5 | 7 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 7 – [7] | < | 5 | 6 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 7 – [6] | < | 5 | 5 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 7 - [5] | < | 5 | 4 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 7 – [4] | = | 5 | 4 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| | Swap(f,d) | 5 | 4 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |

9 comparison and 0 displacement done and array did not change.
Because of f=4 and d=4

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 |

→ Sort(table,5,9)
Pivot value = c[5] = 9

| Compare Pivot-current | Compare Result | Up | down | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | isChanged |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 5 | 9 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | - |
| 9 – [5] | = | 6 | 9 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 9 – [6] | < | 6 | 9 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 9 – [9] | < | 6 | 8 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 9 – [8] | < | 6 | 7 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 9 – [7] | < | 6 | 6 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 9 – [6] | < | 6 | 5 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 9 – [5] | = | 6 | 5 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| | Swap(f,d) | 6 | 5 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |

*8 comparison and 0 displacement done and array did not change.*
*Because of f=5 and d=5*

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 |

→ *Sort(table,6,9)*
*Pivot value = c[6] = 11*

| Compare Pivot-current | Compare Result | Up | down | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | isChanged |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 6 | 9 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | - |
| 11 – [6] | = | 7 | 9 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 11 – [7] | < | 7 | 9 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 11 – [9] | < | 7 | 8 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 11 – [8] | < | 7 | 7 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 11 – [7] | < | 7 | 6 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 11 – [6] | = | 7 | 6 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| | Swap(f,d) | 7 | 6 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |

*7 comparison and 0 displacement done and array did not change.*
*Because of f=6 and d=6*

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 |

➔ *Sort(table,7,9)*

*Pivot value = c[7] = 17*

| Compare Pivot-current | Compare Result | Up | down | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | isChanged |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 7 | 9 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | - |
| 17 – [7] | = | 8 | 9 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 17 – [8] | < | 8 | 9 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 17 – [9] | < | 8 | 8 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 17 – [8] | < | 8 | 7 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 17 – [7] | = | 8 | 7 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| | Swap(f,d) | 8 | 7 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |

*6 comparison and 0 displacement done and array did not change.*

*Because of f=7 and d=7*

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 |

➔ *Sort(table,8,9)*

*Pivot value = c[8] = 18*

| Compare Pivot-current | Compare Result | Up | down | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | isChanged |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 8 | 9 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | - |
| 18 – [8] | = | 9 | 9 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 18 – [9] | < | 8 | 8 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| 18 – [8] | = | 8 | 8 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |
| | Swap(f,d) | 8 | 8 | 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 | False |

*4 comparison and 0 displacement done and array did not change.*

*Because of f=8 and d=8*

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 |

*Sorted array:*

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 3 | 4 | 6 | 7 | 9 | 11 | 17 | 18 | 20 |

***In total, 80 comparison and 0 displacement done.***

## 2) B is an ordered integer array with 10 elements from large to small

B = {31,29,21,17,13,11,9,7,3,2}

n = 10;

| [0] | [1] | [2] | [3] | [4] | [5] | [6] | [7] | [8] | [9] |
|---|---|---|---|---|---|---|---|---|---|
| 31 | 29 | 21 | 17 | 13 | 11 | 9 | 7 | 3 | 2 |

- **Applying Shell Sort**

Our first gap is n/2.

Gap = n / 2 = 10 / 2 = 5.

İnitial array:

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 31 | 29 | 21 | 17 | 13 | 11 | 9 | 7 | 3 | 2 |

When gap = 5;

| Subarray | Comparing elements | [0] | [1] | [2] | [3] | [4] | [5] | [6] | [7] | [8] | [9] | isChanged |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 - 5 | 0 - 5 | 11 | 29 | 21 | 17 | 13 | 31 | 9 | 7 | 3 | 2 | True |
| 1 – 6 | 1 – 6 | 11 | 9 | 21 | 17 | 13 | 31 | 29 | 7 | 3 | 2 | True |
| 2 - 7 | 2 - 7 | 11 | 9 | 7 | 17 | 13 | 31 | 29 | 21 | 3 | 2 | True |
| 3 – 8 | 3 – 8 | 11 | 9 | 7 | 3 | 13 | 31 | 29 | 21 | 17 | 2 | True |
| 4 - 9 | 4 - 9 | 11 | 9 | 7 | 3 | 2 | 31 | 29 | 21 | 17 | 13 | True |

**Note:** The blue boxes are the array's current state after the comparison process.

When gap is equal 5, 5 comparison and 5 displacement done.

***gap = (int) (gap / 2.2);***

***gap = (int) (5 / 2.2) = 2.***

***When gap = 2;***

***İnitial array:***

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 11 | 9 | 7 | 3 | 2 | 31 | 29 | 21 | 17 | 13 |

| Subarray | Comparing elements | [0] | [1] | [2] | [3] | [4] | [5] | [6] | [7] | [8] | [9] | isChanged |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 - 2 | 0 - 2 | 7 | 9 | 11 | 3 | 2 | 31 | 29 | 21 | 17 | 13 | True |
| 1 - 3 | 1 - 3 | 7 | 3 | 11 | 9 | 2 | 31 | 29 | 21 | 17 | 13 | True |
| 0 - 2 - 4 | 2 - 4 | 7 | 3 | 2 | 9 | 11 | 31 | 29 | 21 | 17 | 13 | True |
| 0 - 2 - 4 | 0 – 2 | 2 | 3 | 7 | 9 | 11 | 31 | 29 | 21 | 17 | 13 | True |
| 1 - 3 - 5 | 3 - 5 | 2 | 3 | 7 | 9 | 11 | 31 | 29 | 21 | 17 | 13 | False |
| 1 - 3 - 5 | 1 - 3 | 2 | 3 | 7 | 9 | 11 | 31 | 29 | 21 | 17 | 13 | False |
| 0 - 2 - 4 - 6 | 4 - 6 | 2 | 3 | 7 | 9 | 11 | 31 | 29 | 21 | 17 | 13 | False |
| 0 - 2 - 4 - 6 | 2 - 4 | 2 | 3 | 7 | 9 | 11 | 31 | 29 | 21 | 17 | 13 | False |
| 0 - 2 - 4 - 6 | 0 - 2 | 2 | 3 | 7 | 9 | 11 | 31 | 29 | 21 | 17 | 13 | False |
| 1 - 3 - 5 - 7 | 5 - 7 | 2 | 3 | 7 | 9 | 11 | 21 | 29 | 31 | 17 | 13 | True |
| 1 - 3 - 5 - 7 | 3 - 5 | 2 | 3 | 7 | 9 | 11 | 21 | 29 | 31 | 17 | 13 | False |
| 1 - 3 - 5 - 7 | 1 - 3 | 2 | 3 | 7 | 9 | 11 | 21 | 29 | 31 | 17 | 13 | False |
| 0 - 2 - 4 - 6 - 8 | 6 - 8 | 2 | 3 | 7 | 9 | 11 | 21 | 17 | 31 | 29 | 13 | True |
| 0 - 2 - 4 - 6 – 8 | 4 - 6 | 2 | 3 | 7 | 9 | 11 | 21 | 17 | 31 | 29 | 13 | False |
| 0 - 2 - 4- 6 - 8 | 2 - 4 | 2 | 3 | 7 | 9 | 11 | 21 | 17 | 31 | 29 | 13 | False |
| 0 - 2 - 4 - 6 - 8 | 0 - 2 | 2 | 3 | 7 | 9 | 11 | 21 | 17 | 31 | 29 | 13 | False |
| 1 - 3 - 5 - 7 - 9 | 7 - 9 | 2 | 3 | 7 | 9 | 11 | 21 | 17 | 13 | 29 | 31 | True |
| 1 - 3 - 5 - 7 – 9 | 5 - 7 | 2 | 3 | 7 | 9 | 11 | 13 | 17 | 21 | 29 | 31 | True |

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 - 3 - 5 - 7 - 9 | 3 - 5 | 2 | 3 | 7 | 9 | 11 | 13 | 17 | 21 | 29 | 31 | False |
| 1 - 3 - 5 - 7 - 9 | 1 - 3 | 2 | 3 | 7 | 9 | 11 | 13 | 17 | 21 | 29 | 31 | False |

***Note:*** ***The blue boxes are the array's current state after the comparison process.***

***When gap is equal 2, 20 comparison and 8 displacement done.***

***When gap is 2, we divide by 2.***

***gap = (int) (gap / 2);***

***gap = (int) (2 / 2) = 1.***

***When gap = 1;***

***İnitial array:***

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 2 | 3 | 7 | 9 | 11 | 13 | 17 | 21 | 29 | 31 |

***As we can see, array is already sorted but we continue the sorting with gap = 1.***

| Subarray | Comparing elements | [0] | [1] | [2] | [3] | [4] | [5] | [6] | [7] | [8] | [9] | isChanged |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 - 1 | 0 - 1 | 2 | 3 | 7 | 9 | 11 | 13 | 17 | 21 | 29 | 31 | False |
| 0 - 1 - 2 | 1 - 2 | 2 | 3 | 7 | 9 | 11 | 13 | 17 | 21 | 29 | 31 | False |
| 0 - 1 - 2 | 0 - 1 | 2 | 3 | 7 | 9 | 11 | 13 | 17 | 21 | 29 | 31 | False |
| 0 - 1 - 2 -3 | 2 -3 | 2 | 3 | 7 | 9 | 11 | 13 | 17 | 21 | 29 | 31 | False |
| 0 - 1 - 2 -3 | 1 - 2 | 2 | 3 | 7 | 9 | 11 | 13 | 17 | 21 | 29 | 31 | False |
| 0 - 1 - 2 -3 | 0 - 1 | 2 | 3 | 7 | 9 | 11 | 13 | 17 | 21 | 29 | 31 | False |
| 0 - 1 - 2 -3 - 4 | 3 - 4 | 2 | 3 | 7 | 9 | 11 | 13 | 17 | 21 | 29 | 31 | False |
| 0 - 1 - 2 -3 - 4 | 2 -3 | 2 | 3 | 7 | 9 | 11 | 13 | 17 | 21 | 29 | 31 | False |
| 0 - 1 - 2 -3 - 4 | 1 - 2 | 2 | 3 | 7 | 9 | 11 | 13 | 17 | 21 | 29 | 31 | False |
| 0 - 1 - 2 -3 - 4 | 0 - 1 | 2 | 3 | 7 | 9 | 11 | 13 | 17 | 21 | 29 | 31 | False |
| 0 - 1 - 2 -3 - 4 - 5 | 4 - 5 | 2 | 3 | 7 | 9 | 11 | 13 | 17 | 21 | 29 | 31 | False |

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0 - 1 - 2 -3 - 4 – 5** | **3 - 4** | 2 | 3 | 7 | 9 | 11 | 13 | 17 | 21 | 29 | 31 | **False** |
| **0 - 1 - 2 -3 - 4 – 5** | **2 -3** | 2 | 3 | 7 | 9 | 11 | 13 | 17 | 21 | 29 | 31 | **False** |
| **0 - 1 - 2 -3 - 4 – 5** | **1 - 2** | 2 | 3 | 7 | 9 | 11 | 13 | 17 | 21 | 29 | 31 | **False** |
| **0 - 1 - 2 -3 - 4 - 5** | **0 - 1** | 2 | 3 | 7 | 9 | 11 | 13 | 17 | 21 | 29 | 31 | **False** |
| **0 - 1 - 2 -3 - 4 - 5 - 6** | **5 - 6** | 2 | 3 | 7 | 9 | 11 | 13 | 17 | 21 | 29 | 31 | **False** |
| **0 - 1 - 2 -3 - 4 - 5 – 6** | **4 - 5** | 2 | 3 | 7 | 9 | 11 | 13 | 17 | 21 | 29 | 31 | **False** |
| **0 - 1 - 2 -3 - 4 - 5 – 6** | **3 - 4** | 2 | 3 | 7 | 9 | 11 | 13 | 17 | 21 | 29 | 31 | **False** |
| **0 - 1 - 2 -3 - 4 - 5 - 6** | **2 -3** | 2 | 3 | 7 | 9 | 11 | 13 | 17 | 21 | 29 | 31 | **False** |
| **0 - 1 - 2 -3 - 4 - 5 – 6** | **1 - 2** | 2 | 3 | 7 | 9 | 11 | 13 | 17 | 21 | 29 | 31 | **False** |
| **0 - 1 - 2 -3 - 4 - 5 - 6** | **0 - 1** | 2 | 3 | 7 | 9 | 11 | 13 | 17 | 21 | 29 | 31 | **False** |
| **0 - 1 - 2 -3 - 4 - 5 – 6 -7** | **6 -7** | 2 | 3 | 7 | 9 | 11 | 13 | 17 | 21 | 29 | 31 | **False** |
| **0 - 1 - 2 -3 - 4 - 5 – 6 -7** | **5 - 6** | 2 | 3 | 7 | 9 | 11 | 13 | 17 | 21 | 29 | 31 | **False** |
| **0 - 1 - 2 -3 - 4 - 5 – 6 -7** | **4 - 5** | 2 | 3 | 7 | 9 | 11 | 13 | 17 | 21 | 29 | 31 | **False** |
| **0 - 1 - 2 -3 - 4 - 5 – 6 -7** | **3 - 4** | 2 | 3 | 7 | 9 | 11 | 13 | 17 | 21 | 29 | 31 | **False** |
| **0 - 1 - 2 -3 - 4 - 5 – 6 -7** | **2 -3** | 2 | 3 | 7 | 9 | 11 | 13 | 17 | 21 | 29 | 31 | **False** |
| **0 - 1 - 2 -3 - 4 - 5 – 6 -7** | **1 - 2** | 2 | 3 | 7 | 9 | 11 | 13 | 17 | 21 | 29 | 31 | **False** |
| **0 - 1 - 2 -3 - 4 - 5 – 6 -7** | **0 - 1** | 2 | 3 | 7 | 9 | 11 | 13 | 17 | 21 | 29 | 31 | **False** |
| **0 - 1 - 2 -3 - 4 - 5 – 6 -7 - 8** | **7 - 8** | 2 | 3 | 7 | 9 | 11 | 13 | 17 | 21 | 29 | 31 | **False** |
| **0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8** | **6 -7** | 2 | 3 | 7 | 9 | 11 | 13 | 17 | 21 | 29 | 31 | **False** |
| **0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8** | **5 - 6** | 2 | 3 | 7 | 9 | 11 | 13 | 17 | 21 | 29 | 31 | **False** |
| **0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8** | **4 - 5** | 2 | 3 | 7 | 9 | 11 | 13 | 17 | 21 | 29 | 31 | **False** |
| **0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8** | **3 - 4** | 2 | 3 | 7 | 9 | 11 | 13 | 17 | 21 | 29 | 31 | **False** |

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8** | **2 -3** | 2 | 3 | 7 | 9 | 11 | 13 | 17 | 21 | 29 | 31 | **False** |
| **0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8** | **1 - 2** | 2 | 3 | 7 | 9 | 11 | 13 | 17 | 21 | 29 | 31 | **False** |
| **0 - 1 - 2 -3 - 4 - 5 – 6 -7 - 8** | **0 - 1** | 2 | 3 | 7 | 9 | 11 | 13 | 17 | 21 | 29 | 31 | **False** |
| **0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 - 9** | **8 - 9** | 2 | 3 | 7 | 9 | 11 | 13 | 17 | 21 | 29 | 31 | **False** |
| **0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 – 9** | **7 - 8** | 2 | 3 | 7 | 9 | 11 | 13 | 17 | 21 | 29 | 31 | **False** |
| **0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 – 9** | **6 -7** | 2 | 3 | 7 | 9 | 11 | 13 | 17 | 21 | 29 | 31 | **False** |
| **0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 – 9** | **5 - 6** | 2 | 3 | 7 | 9 | 11 | 13 | 17 | 21 | 29 | 31 | **False** |
| **0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 – 9** | **4 - 5** | 2 | 3 | 7 | 9 | 11 | 13 | 17 | 21 | 29 | 31 | **False** |
| **0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 – 9** | **3 - 4** | 2 | 3 | 7 | 9 | 11 | 13 | 17 | 21 | 29 | 31 | **False** |
| **0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 – 9** | **2 -3** | 2 | 3 | 7 | 9 | 11 | 13 | 17 | 21 | 29 | 31 | **False** |
| **0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 – 9** | **1 - 2** | 2 | 3 | 7 | 9 | 11 | 13 | 17 | 21 | 29 | 31 | **False** |
| **0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 - 9** | **0 - 1** | 2 | 3 | 7 | 9 | 11 | 13 | 17 | 21 | 29 | 31 | **False** |

***Note:*** ***The blue boxes are the array's current state after the comparison process.***

***When gap is equal 1, 45 comparison and 0 displacement done.***

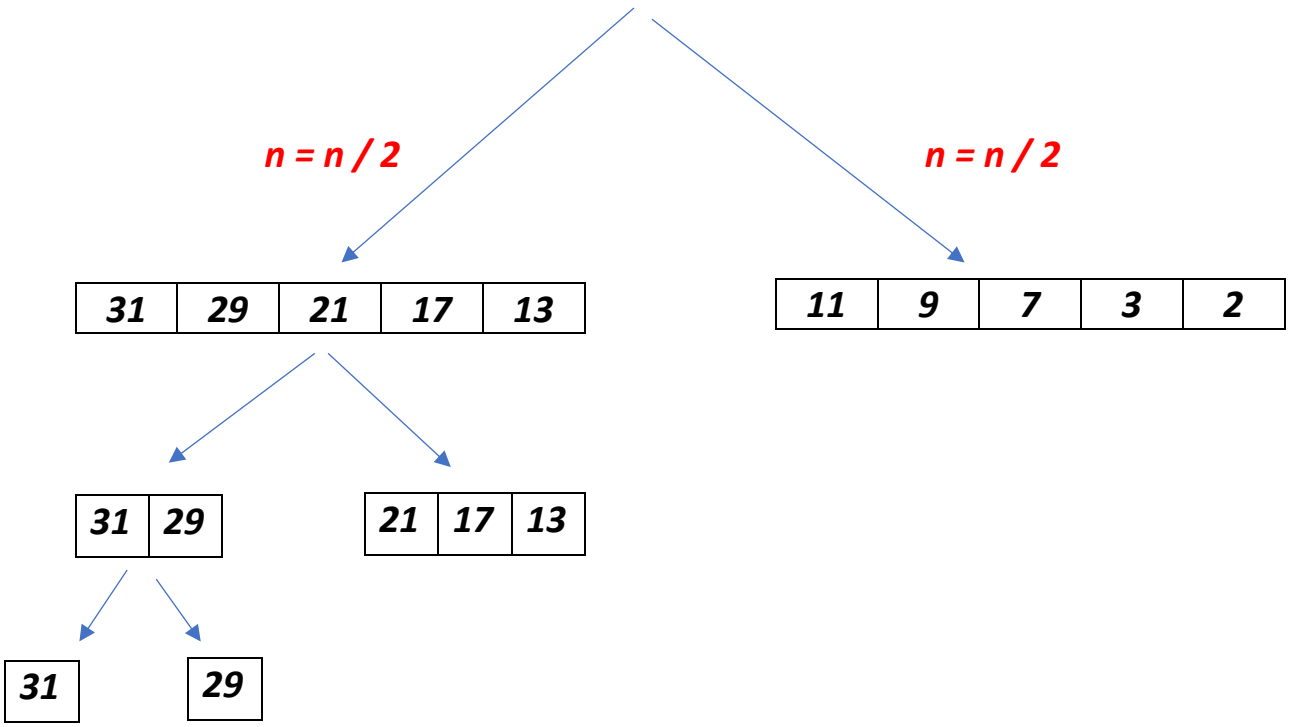
## *In total 70 comparison and 13 displacement done.*

- ***Applying Merge Sort***

***B = {31,29,21,17,13,11,9,7,3,2}***

***n = 10;***

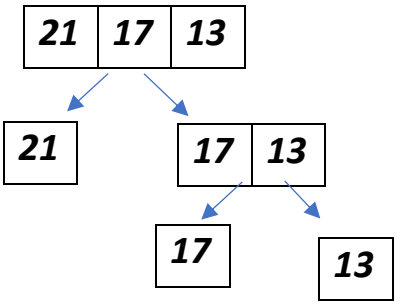
| [0] | [1] | [2] | [3] | [4] | [5] | [6] | [7] | [8] | [9] |
|---|---|---|---|---|---|---|---|---|---|
| 31 | 29 | 21 | 17 | 13 | 11 | 9 | 7 | 3 | 2 |




***1) Compare 31 and 29. And associate as sorted version.***

***Output:*** 29 | 31

***2)***



3) *Compare 17 and 13. And associate as sorted version.*

**Output**

| | |
|---|---|
| 13 | 17 |

4) *Compare* | 21 | *and*

| | |
|---|---|
| 13 | 17 |

- *Compare 21 – 13 →* | 13 |
- *Compare 21 – 17 →*

| | |
|---|---|
| 13 | 17 |

- *Add 21 →*

| | | |
|---|---|---|
| 13 | 17 | 21 |

5)

| | |
|---|---|
| 29 | 31 |

| | | |
|---|---|---|
| 13 | 17 | 21 |

| | |
|---|---|
| 29 | 31 |

| | | |
|---|---|---|
| 13 | 17 | 21 |

- *Compare 29 – 13 →* | 13 |
- *Compare 29 – 17 →*

| | |
|---|---|
| 13 | 17 |

- *Compare 29 – 21 →*

| | | |
|---|---|---|
| 13 | 17 | 21 |

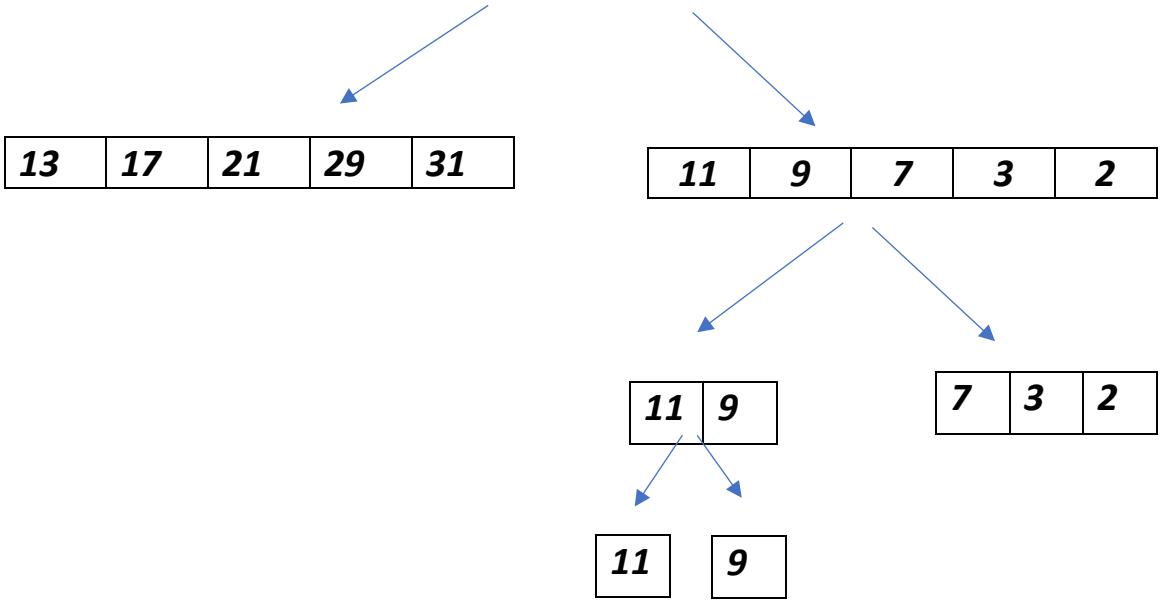
- *Add 29 →*

| | | | |
|---|---|---|---|
| 13 | 17 | 21 | 29 |

- *Add 31 →*

| | | | | |
|---|---|---|---|---|
| 13 | 17 | 21 | 29 | 31 |

6)

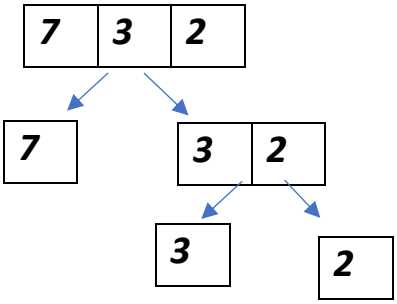



7) Compare 11 and 9. And associate as sorted version.

**Output:**

| | |
|---|---|
| 9 | 11 |

8)




9) Compare 3 and 2. And associate as sorted version.

**Output:**

| | |
|---|---|
| 2 | 3 |

***10)*** ***Compare*** | 7 | ***and*** | 2 | 3 |

- ***Compare 7 – 2*** → | 2 |
- ***Compare 7 – 3*** → | 2 | 3 |
- ***Add 7*** → | 2 | 3 | 7 |

***11)***

| 9 | 11 |

| 2 | 3 | 7 |

| 9 | 11 |

| 2 | 3 | 7 |

- ***Compare 9 – 2*** → | 2 |
- ***Compare 9 – 3*** → | 2 | 3 |
- ***Compare 9 – 7*** → | 2 | 3 | 7 |
- ***Add 9*** → | 2 | 3 | 7 | 9 |
- ***Add 11*** → | 2 | 3 | 7 | 9 | 11 |

*12)*

| | | | | |
|---|---|---|---|---|
| 13 | 17 | 21 | 29 | 31 |

| | | | | |
|---|---|---|---|---|
| 2 | 3 | 7 | 9 | 11 |

- *Compare 13 - 2 →* | 2 |
- *Compare 13 – 3 →* | 1 | 3 |
- *Compare 13 – 7 →* | 1 | 3 | 7 |
- *Compare 13 – 9 →* | 1 | 3 | 7 | 9 |
- *Compare 13 – 11 →* | 1 | 3 | 7 | 9 | 11 |
- *Add 13 →* | 1 | 3 | 7 | 9 | 11 | 13 |
- *Add 17 →* | 1 | 3 | 7 | 9 | 11 | 13 | 17 |
- *Add 21 →* | 1 | 3 | 7 | 9 | 11 | 13 | 17 | 21 |
- *Add 29 →* | 1 | 3 | 7 | 9 | 11 | 13 | 17 | 21 | 29 |
- *Add 31 →* | 1 | 3 | 7 | 9 | 11 | 13 | 17 | 21 | 29 | 31 |

***Sorted array:***

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 3 | 7 | 9 | 11 | 13 | 17 | 21 | 29 | 31 |

***In total 21 comparison 30 displacement.***

***B = {31,29,21,17,13,11,9,7,3,2}***

***n = 10;***

| [0] | [1] | [2] | [3] | [4] | [5] | [6] | [7] | [8] | [9] |
|---|---|---|---|---|---|---|---|---|---|
| 31 | 29 | 21 | 17 | 13 | 11 | 9 | 7 | 3 | 2 |

***We have to do two steps for converting that array to the increasing order array.***

1) ***Build a max heap from array.***
2) ***Shrink the heap.***

## ***Build process***

```
private <T extends Comparable<T>> void buildHeap(T[] table) {
        int n = 1;
        // Invariant: table[0 . . . n - 1] is a heap.
        while (n < table.length) {
                n++; // Add a new item to the heap and reheap.
                int child = n - 1;
                int parent = (child - 1) / 2; // Find parent.
                while (parent >= 0
                && table[parent].compareTo(table[child]) < 0) {
                swap(table, parent, child);
                child = parent;
                parent = (child - 1) / 2;
        }
}
```

***I start to control of child parent relation from index 1. Every index which I controled, I looked at its parent up to root. If necessary I swap the values. Parent = (child-1)/2***

***İnital array:***

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 31 | 29 | 21 | 17 | 13 | 11 | 9 | 7 | 3 | 2 |

| Current child index | Comparing elements | [0] | [1] | [2] | [3] | [4] | [5] | [6] | [7] | [8] | [9] | isChanged |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 – 0 | 31 | 29 | 21 | 17 | 13 | 11 | 9 | 7 | 3 | 2 | False |
| 2 | 2 – 0 | 31 | 29 | 21 | 17 | 13 | 11 | 9 | 7 | 3 | 2 | False |
| 3 | 3 – 1 | 31 | 29 | 21 | 17 | 13 | 11 | 9 | 7 | 3 | 2 | False |
| 3 | 1 – 0 | 31 | 29 | 21 | 17 | 13 | 11 | 9 | 7 | 3 | 2 | False |
| 4 | 4 – 1 | 31 | 29 | 21 | 17 | 13 | 11 | 9 | 7 | 3 | 2 | False |
| 4 | 1 – 0 | 31 | 29 | 21 | 17 | 13 | 11 | 9 | 7 | 3 | 2 | False |
| 5 | 5 – 2 | 31 | 29 | 21 | 17 | 13 | 11 | 9 | 7 | 3 | 2 | False |
| 5 | 2 – 0 | 31 | 29 | 21 | 17 | 13 | 11 | 9 | 7 | 3 | 2 | False |
| 6 | 6 – 2 | 31 | 29 | 21 | 17 | 13 | 11 | 9 | 7 | 3 | 2 | False |
| 6 | 2 – 0 | 31 | 29 | 21 | 17 | 13 | 11 | 9 | 7 | 3 | 2 | False |
| 7 | 7 – 3 | 31 | 29 | 21 | 17 | 13 | 11 | 9 | 7 | 3 | 2 | False |
| 7 | 3 – 1 | 31 | 29 | 21 | 17 | 13 | 11 | 9 | 7 | 3 | 2 | False |
| 7 | 1 – 0 | 31 | 29 | 21 | 17 | 13 | 11 | 9 | 7 | 3 | 2 | False |
| 8 | 8 – 3 | 31 | 29 | 21 | 17 | 13 | 11 | 9 | 7 | 3 | 2 | False |
| 8 | 3 – 1 | 31 | 29 | 21 | 17 | 13 | 11 | 9 | 7 | 3 | 2 | False |
| 8 | 1 – 0 | 31 | 29 | 21 | 17 | 13 | 11 | 9 | 7 | 3 | 2 | False |
| 9 | 9 – 4 | 31 | 29 | 21 | 17 | 13 | 11 | 9 | 7 | 3 | 2 | False |
| 9 | 4 – 1 | 31 | 29 | 21 | 17 | 13 | 11 | 9 | 7 | 3 | 2 | False |

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ***9*** | ***1 – 0*** | ***31*** | ***29*** | ***21*** | ***17*** | ***13*** | ***11*** | ***9*** | ***7*** | ***3*** | ***2*** | ***False*** |

***That array is already sorted by descending order. For this reason array is already act as a max heap.***

***In build process, 19 comparison and 0 displacement done***

***Max heap:***

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| ***31*** | ***29*** | ***21*** | ***17*** | ***13*** | ***11*** | ***9*** | ***7*** | ***3*** | ***2*** |

***Shrink process***

```
private <T extends Comparable<T>> void shrinkHeap(T[] table) {
        int n = table.length;
        // Invariant: table[0 . . . n - 1] forms a heap.
        // table[n . . . table.length - 1] is sorted.
        while (n > 0) {
                n--;
                swap(table, 0, n);
                // table[1 . . . n - 1] form a heap.
                // table[n . . . table.length - 1] is sorted.
                int parent = 0;
                while (true) {
                        int leftChild = 2 * parent + 1;
                        if (leftChild >= n) {
                                break; // No more children.
                        }
                        int rightChild = leftChild + 1;
                        // Find the larger of the two children.
                        int maxChild = leftChild;
                        if (rightChild < n // There is a right child.
                        && table[leftChild].compareTo(table[rightChild]) < 0) {
                        maxChild = rightChild;
                        }
                        // If the parent is smaller than the larger child,
                        if (table[parent].compareTo(table[maxChild]) < 0){
                                // Swap the parent and child.
                                swap(table, parent, maxChild);
                                // Continue at the child level.
                                parent = maxChild;
```

```
        } else { // Heap property is restored.
                break; // Exit the loop.
        }
        }
        }
}
```

***Left child = 2 * parent + 1***

***Right child = 2 * parent + 2***

***While the blue box represents the current index, the green box represents the sorted section.***

| Left child index | Right child index | Compare notes | [0] | [1] | [2] | [3] | [4] | [5] | [6] | [7] | [8] | [9] | isChanged |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| - | - | Swap the first value and last value | 2 | 29 | 21 | 17 | 13 | 11 | 9 | 7 | 3 | 31 | True |
| 1 | 2 | [1] > [2] Swap [0] and [1] | 29 | 2 | 21 | 17 | 13 | 11 | 9 | 7 | 3 | 31 | True |
| 3 | 4 | [3] > [4] Swap [1] and [3] | 29 | 17 | 21 | 2 | 13 | 11 | 9 | 7 | 3 | 31 | True |
| 7 | 8 | [7] > [8] Swap [3] and [7] | 29 | 17 | 21 | 7 | 13 | 11 | 9 | 2 | 3 | 31 | True |
| 15 | 16 | There is no 15 th and 16 and index in our array | 29 | 17 | 21 | 7 | 13 | 11 | 9 | 2 | 3 | 31 | False |

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| - | - | Swap the first value and last value | 3 | 17 | 21 | 7 | 13 | 11 | 9 | 2 | 29 | 31 | True |
| 1 | 2 | [2] > [1] Swap [0] and [2] | 21 | 17 | 3 | 7 | 13 | 11 | 9 | 2 | 29 | 31 | True |
| 4 | 5 | [4] > [5] Swap [1] and [4] | 21 | 17 | 13 | 7 | 3 | 11 | 9 | 2 | 29 | 31 | True |
| 9 | 10 | There is no 9 th and 10 th index in our array | 21 | 17 | 13 | 7 | 3 | 11 | 9 | 2 | 29 | 31 | False |
| - | - | Swap the first value and last value | 2 | 17 | 13 | 7 | 3 | 11 | 9 | 21 | 29 | 31 | True |
| 1 | 2 | [1] > [2] Swap [0] and [1] | 17 | 2 | 13 | 7 | 3 | 11 | 9 | 21 | 29 | 31 | True |
| 3 | 4 | [3] > [4] Swap [1] and [3] | 17 | 7 | 13 | 2 | 3 | 11 | 9 | 21 | 29 | 31 | True |
| 7 | 8 | There is no 7 th and 8 th index in our array] | 17 | 7 | 13 | 2 | 3 | 11 | 9 | 21 | 29 | 31 | False |
| - | - | Swap the first value and last value | 9 | 7 | 13 | 2 | 3 | 11 | 17 | 21 | 29 | 31 | True |

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | [2] > [1] Swap [0] and [2] | 13 | 7 | 9 | 2 | 3 | 11 | 17 | 21 | 29 | 31 | True |
| 5 | 6 | There is no 6th index in our array | 13 | 7 | 9 | 2 | 3 | 11 | 17 | 21 | 29 | 31 | False |
| - | - | Swap the first value and last value | 11 | 7 | 9 | 2 | 3 | 13 | 17 | 21 | 29 | 31 | True |
| 1 | 2 | [2] > [1] But [0]>[2] No swap | 11 | 7 | 9 | 2 | 3 | 13 | 17 | 21 | 29 | 31 | False |
| - | - | Swap the first value and last value | 3 | 7 | 9 | 2 | 11 | 13 | 17 | 21 | 29 | 31 | True |
| 1 | 2 | [2] > [1] Swap [0] and [2] | 9 | 7 | 3 | 2 | 11 | 13 | 17 | 21 | 29 | 31 | True |
| 5 | 6 | There is no 5 th and 6 index in our array | 9 | 7 | 3 | 2 | 11 | 13 | 17 | 21 | 29 | 31 | False |
| - | - | Swap the first value and last value | 2 | 7 | 3 | 9 | 11 | 13 | 17 | 21 | 29 | 31 | True |

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | [1] > [2] Swap [0] and [1] | 7 | 2 | 3 | 9 | 11 | 13 | 17 | 21 | 29 | 31 | True |
| 3 | 4 | There is no 3 th and 4 index in our array | 7 | 2 | 3 | 9 | 11 | 13 | 17 | 21 | 29 | 31 | True |
| - | - | Swap the first value and last value | 3 | 2 | 7 | 9 | 11 | 13 | 17 | 21 | 29 | 31 | True |
| 1 | 2 | There is no 2th index in our array | 3 | 2 | 7 | 9 | 11 | 13 | 17 | 21 | 29 | 31 | False |
| - | - | Swap the first value and last value | 2 | 3 | 7 | 9 | 11 | 13 | 17 | 21 | 29 | 31 | True |

*In shrink process, 27 comparison and 20 displacement done*

*Sorted array:*

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 2 | 3 | 7 | 9 | 11 | 13 | 17 | 21 | 29 | 31 |

***In total 46 comparison 20 displacement.***

- ## ***Applying QuickSort:***

***B = {31,29,21,17,13,11,9,7,3,2}***

***n = 10;***

| [0] | [1] | [2] | [3] | [4] | [5] | [6] | [7] | [8] | [9] |
|---|---|---|---|---|---|---|---|---|---|
| 31 | 29 | 21 | 17 | 13 | 11 | 9 | 7 | 3 | 2 |

```
protected <T extends Comparable<T>> void quickSort(T[] table, int first, int last) {
        if (first < last) {
                int pivIndex = partition(table, first, last);
                quickSort(table, first, pivIndex - 1);
                quickSort(table, pivIndex + 1, last);
        }
}

protected <T extends Comparable<T>> int partition(T[] table, iint first, int last) {
        T pivot = table[first];
        int up = first; int down = last;
        do {
        while ((up < last) && (pivot.compareTo(table[up]) >= 0)) {
                up++;
        }
        while (pivot.compareTo(table[down]) < 0) {
                down--;
        }
        if (up < down) {
                swap(table, up, down);
        }
        } while (up < down); // Repeat while up is left of down.
        swap(table, first, down);
        return down;
}
```

➔ *Sort(table,0,9)*

*Pivot value = c[0] = 31*

| Compare Pivot-current | Compare Result | Up | down | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | isChanged |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 0 | 9 | 31 | 29 | 21 | 17 | 13 | 11 | 9 | 7 | 3 | 2 | - |
| 31 – [0] | = | 1 | 9 | 31 | 29 | 21 | 17 | 13 | 11 | 9 | 7 | 3 | 2 | False |
| 31 – [1] | > | 2 | 9 | 31 | 29 | 21 | 17 | 13 | 11 | 9 | 7 | 3 | 2 | False |
| 31 – [2] | > | 3 | 9 | 31 | 29 | 21 | 17 | 13 | 11 | 9 | 7 | 3 | 2 | False |
| 31 – [3] | > | 4 | 9 | 31 | 29 | 21 | 17 | 13 | 11 | 9 | 7 | 3 | 2 | False |
| 31 – [4] | > | 5 | 9 | 31 | 29 | 21 | 17 | 13 | 11 | 9 | 7 | 3 | 2 | False |
| 31 – [5] | > | 6 | 9 | 31 | 29 | 21 | 17 | 13 | 11 | 9 | 7 | 3 | 2 | False |
| 31 – [6] | > | 7 | 9 | 31 | 29 | 21 | 17 | 13 | 11 | 9 | 7 | 3 | 2 | False |
| 31 – [7] | > | 8 | 9 | 31 | 29 | 21 | 17 | 13 | 11 | 9 | 7 | 3 | 2 | False |
| 31 – [8] | > | 9 | 9 | 31 | 29 | 21 | 17 | 13 | 11 | 9 | 7 | 3 | 2 | False |
| 31 – [9] | > | 9 | 9 | 31 | 29 | 21 | 17 | 13 | 11 | 9 | 7 | 3 | 2 | False |
| | Swap(f,d) | 9 | 9 | 2 | 29 | 21 | 17 | 13 | 11 | 9 | 7 | 3 | 31 | True |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 2 | 29 | 21 | 17 | 13 | 11 | 9 | 7 | 3 | 31 |

*11 comparison and 1 displacement done.*

→ Sort(table,0,8)

Pivot value = c[0] = 2

| Compare Pivot-current | Compare Result | Up | down | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | isChanged |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 0 | 8 | 2 | 29 | 21 | 17 | 13 | 11 | 9 | 7 | 3 | 31 | - |
| 2 – [0] | = | 1 | 8 | 2 | 29 | 21 | 17 | 13 | 11 | 9 | 7 | 3 | 31 | False |
| 2 – [1] | < | 1 | 8 | 2 | 29 | 21 | 17 | 13 | 11 | 9 | 7 | 3 | 31 | False |
| 2 – [8] | < | 1 | 7 | 2 | 29 | 21 | 17 | 13 | 11 | 9 | 7 | 3 | 31 | False |
| 2 – [7] | < | 1 | 6 | 2 | 29 | 21 | 17 | 13 | 11 | 9 | 7 | 3 | 31 | False |
| 2 – [6] | < | 1 | 5 | 2 | 29 | 21 | 17 | 13 | 11 | 9 | 7 | 3 | 31 | False |
| 2 – [5] | < | 1 | 4 | 2 | 29 | 21 | 17 | 13 | 11 | 9 | 7 | 3 | 31 | False |
| 2 – [4] | < | 1 | 3 | 2 | 29 | 21 | 17 | 13 | 11 | 9 | 7 | 3 | 31 | False |
| 2 – [3] | < | 1 | 2 | 2 | 29 | 21 | 17 | 13 | 11 | 9 | 7 | 3 | 31 | False |
| 2 – [2] | < | 1 | 1 | 2 | 29 | 21 | 17 | 13 | 11 | 9 | 7 | 3 | 31 | False |
| 2 – [1] | < | 1 | 0 | 2 | 29 | 21 | 17 | 13 | 11 | 9 | 7 | 3 | 31 | False |
| 2 – [0] | = | 1 | 0 | 2 | 29 | 21 | 17 | 13 | 11 | 9 | 7 | 3 | 31 | False |
| | Swap(f,d) | 1 | 0 | 2 | 29 | 21 | 17 | 13 | 11 | 9 | 7 | 3 | 31 | False |

Array did not change. Because of f=0 and d=0

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 2 | 29 | 21 | 17 | 13 | 11 | 9 | 7 | 3 | 31 |

12 comparison and 0 displacement done.

→ *Sort(table,1,8)*

*Pivot value = c[1] = 29*

| Compare Pivot-current | Compare Result | Up | down | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | isChanged |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 1 | 8 | 2 | 29 | 21 | 17 | 13 | 11 | 9 | 7 | 3 | 31 | - |
| 29 – [1] | = | 2 | 8 | 2 | 29 | 21 | 17 | 13 | 11 | 9 | 7 | 3 | 31 | False |
| 29 – [2] | > | 3 | 8 | 2 | 29 | 21 | 17 | 13 | 11 | 9 | 7 | 3 | 31 | False |
| 29 – [3] | > | 4 | 8 | 2 | 29 | 21 | 17 | 13 | 11 | 9 | 7 | 3 | 31 | False |
| 29 – [4] | > | 5 | 8 | 2 | 29 | 21 | 17 | 13 | 11 | 9 | 7 | 3 | 31 | False |
| 29 – [5] | > | 6 | 8 | 2 | 29 | 21 | 17 | 13 | 11 | 9 | 7 | 3 | 31 | False |
| 29 – [6] | > | 7 | 8 | 2 | 29 | 21 | 17 | 13 | 11 | 9 | 7 | 3 | 31 | False |
| 29 – [7] | > | 8 | 8 | 2 | 29 | 21 | 17 | 13 | 11 | 9 | 7 | 3 | 31 | False |
| 29 – [8] | > | 8 | 8 | 2 | 29 | 21 | 17 | 13 | 11 | 9 | 7 | 3 | 31 | False |
| | Swap(f,d) | 8 | 8 | 2 | 3 | 21 | 17 | 13 | 11 | 9 | 7 | 29 | 31 | True |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 2 | 3 | 21 | 17 | 13 | 11 | 9 | 7 | 29 | 31 |

*9 comparison and 1 displacement done.*

→ Sort(table,1,7)

Pivot value = c[1] = 3

| Compare Pivot-current | Compare Result | Up | down | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | isChanged |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 1 | 7 | 2 | 3 | 21 | 17 | 13 | 11 | 9 | 7 | 29 | 31 | - |
| 3 – [1] | = | 2 | 7 | 2 | 3 | 21 | 17 | 13 | 11 | 9 | 7 | 29 | 31 | False |
| 3 – [2] | < | 2 | 7 | 2 | 3 | 21 | 17 | 13 | 11 | 9 | 7 | 29 | 31 | False |
| 3 – [7] | < | 2 | 6 | 2 | 3 | 21 | 17 | 13 | 11 | 9 | 7 | 29 | 31 | False |
| 3 – [6] | < | 2 | 5 | 2 | 3 | 21 | 17 | 13 | 11 | 9 | 7 | 29 | 31 | False |
| 3 – [5] | < | 2 | 4 | 2 | 3 | 21 | 17 | 13 | 11 | 9 | 7 | 29 | 31 | False |
| 3 – [4] | < | 2 | 3 | 2 | 3 | 21 | 17 | 13 | 11 | 9 | 7 | 29 | 31 | False |
| 3 – [3] | < | 2 | 2 | 2 | 3 | 21 | 17 | 13 | 11 | 9 | 7 | 29 | 31 | False |
| 3 – [2] | < | 2 | 1 | 2 | 3 | 21 | 17 | 13 | 11 | 9 | 7 | 29 | 31 | False |
| 3 – [1] | = | 2 | 1 | 2 | 3 | 21 | 17 | 13 | 11 | 9 | 7 | 29 | 31 | False |
| | Swap(f,d) | 2 | 1 | 2 | 3 | 21 | 17 | 13 | 11 | 9 | 7 | 29 | 31 | False |

Array did not change. Because of f=1 and d=1

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 2 | 3 | 21 | 17 | 13 | 11 | 9 | 7 | 29 | 31 |

10 comparison and 0 displacement done.

➔ *Sort(table,2,7)*

*Pivot value = c[2] = 21*

| Compare Pivot-current | Compare Result | Up | down | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | isChanged |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 2 | 7 | 2 | 3 | 21 | 17 | 13 | 11 | 9 | 7 | 29 | 31 | - |
| 21 – [2] | = | 3 | 7 | 2 | 3 | 21 | 17 | 13 | 11 | 9 | 7 | 29 | 31 | False |
| 21 – [3] | > | 4 | 7 | 2 | 3 | 21 | 17 | 13 | 11 | 9 | 7 | 29 | 31 | False |
| 21 – [4] | > | 5 | 7 | 2 | 3 | 21 | 17 | 13 | 11 | 9 | 7 | 29 | 31 | False |
| 21 – [5] | > | 6 | 7 | 2 | 3 | 21 | 17 | 13 | 11 | 9 | 7 | 29 | 31 | False |
| 21 – [6] | > | 7 | 7 | 2 | 3 | 21 | 17 | 13 | 11 | 9 | 7 | 29 | 31 | False |
| 21 – [7] | > | 7 | 7 | 2 | 3 | 21 | 17 | 13 | 11 | 9 | 7 | 29 | 31 | False |
| | Swap(f,d) | 7 | 7 | 2 | 3 | 7 | 17 | 13 | 11 | 9 | 21 | 29 | 31 | True |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 2 | 3 | 7 | 17 | 13 | 11 | 9 | 21 | 29 | 31 |

*7 comparison and 1 displacement done.*

➔ *Sort(table,2,6)*

*Pivot value = c[2] = 7*

| Compare Pivot-current | Compare Result | Up | down | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | isChanged |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 2 | 6 | 2 | 3 | 7 | 17 | 13 | 11 | 9 | 21 | 29 | 31 | - |
| 7 – [2] | = | 3 | 6 | 2 | 3 | 7 | 17 | 13 | 11 | 9 | 21 | 29 | 31 | False |
| 7 – [3] | < | 3 | 6 | 2 | 3 | 7 | 17 | 13 | 11 | 9 | 21 | 29 | 31 | False |
| 7 – [6] | < | 3 | 5 | 2 | 3 | 7 | 17 | 13 | 11 | 9 | 21 | 29 | 31 | False |
| 7 – [5] | < | 3 | 4 | 2 | 3 | 7 | 17 | 13 | 11 | 9 | 21 | 29 | 31 | False |
| 7 – [4] | < | 3 | 3 | 2 | 3 | 7 | 17 | 13 | 11 | 9 | 21 | 29 | 31 | False |
| 7 – [3] | < | 3 | 2 | 2 | 3 | 7 | 17 | 13 | 11 | 9 | 21 | 29 | 31 | False |
| 7 – [2] | = | 3 | 2 | 2 | 3 | 7 | 17 | 13 | 11 | 9 | 21 | 29 | 31 | False |
| | Swap(f,d) | 3 | 2 | 2 | 3 | 7 | 17 | 13 | 11 | 9 | 21 | 29 | 31 | False |

*Array did not change. Because of f=2 and d=2*

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 2 | 3 | 7 | 17 | 13 | 11 | 9 | 21 | 29 | 31 |

*8 comparison and 0 displacement done.*

➔ *Sort(table,3,6)*

*Pivot value = c[3] = 17*

| Compare Pivot-current | Compare Result | Up | down | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | isChanged |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 3 | 6 | 2 | 3 | 7 | 17 | 13 | 11 | 9 | 21 | 29 | 31 | - |
| 17 – [3] | = | 4 | 6 | 2 | 3 | 7 | 17 | 13 | 11 | 9 | 21 | 29 | 31 | False |
| 17 – [4] | > | 5 | 6 | 2 | 3 | 7 | 17 | 13 | 11 | 9 | 21 | 29 | 31 | False |
| 17 – [5] | > | 6 | 6 | 2 | 3 | 7 | 17 | 13 | 11 | 9 | 21 | 29 | 31 | False |
| 17 – [6] | > | 6 | 6 | 2 | 3 | 7 | 17 | 13 | 11 | 9 | 21 | 29 | 31 | False |
| | Swap(f,d) | 6 | 6 | 2 | 3 | 7 | 9 | 13 | 11 | 17 | 21 | 29 | 31 | True |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 2 | 3 | 7 | 9 | 13 | 11 | 17 | 21 | 29 | 31 |

*5 comparison and 1 displacement done.*

➔ *Sort(table,3,5)*

*Pivot value = c[3] = 9*

| Compare Pivot-current | Compare Result | Up | down | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | isChanged |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 3 | 5 | 2 | 3 | 7 | 9 | 13 | 11 | 17 | 21 | 29 | 31 | - |
| 9 – [3] | = | 4 | 5 | 2 | 3 | 7 | 9 | 13 | 11 | 17 | 21 | 29 | 31 | False |
| 9 – [4] | < | 4 | 5 | 2 | 3 | 7 | 9 | 13 | 11 | 17 | 21 | 29 | 31 | False |
| 9 – [5] | < | 4 | 4 | 2 | 3 | 7 | 9 | 13 | 11 | 17 | 21 | 29 | 31 | False |
| 9 – [4] | < | 4 | 3 | 2 | 3 | 7 | 9 | 13 | 11 | 17 | 21 | 29 | 31 | False |
| 9 – [3] | = | 4 | 3 | 2 | 3 | 7 | 9 | 13 | 11 | 17 | 21 | 29 | 31 | False |
| | Swap(f,d) | 4 | 3 | 2 | 3 | 7 | 9 | 13 | 11 | 17 | 21 | 29 | 31 | False |

*Array did not change. Because of f=3 and d=3*

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 2 | 3 | 7 | 9 | 13 | 11 | 17 | 21 | 29 | 31 |

*6 comparison and 0 displacement done.*

➔ *Sort(table,4,5)*

*Pivot value = c[4] = 13*

| Compare Pivot-current | Compare Result | Up | down | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | isChanged |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 4 | 5 | 2 | 3 | 7 | 9 | 13 | 11 | 17 | 21 | 29 | 31 | - |
| 13 – [4] | = | 5 | 5 | 2 | 3 | 7 | 9 | 13 | 11 | 17 | 21 | 29 | 31 | False |
| 13 – [5] | > | 5 | 5 | 2 | 3 | 7 | 9 | 13 | 11 | 17 | 21 | 29 | 31 | False |
| | Swap(f,d) | 5 | 5 | 2 | 3 | 7 | 9 | 11 | 13 | 17 | 21 | 29 | 31 | True |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 2 | 3 | 7 | 9 | 11 | 13 | 17 | 21 | 29 | 31 |

*3 comparison and 1 displacement done.*

*Sorted array:*

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 2 | 3 | 7 | 9 | 11 | 13 | 17 | 21 | 29 | 31 |

## *In total 71 comparison 5 displacement.*

3) C = {5, 2, 13, 9, 1, 7, 6, 8, 1, 15, 4, 11}

- **Applying Shell sort:**

n = 12

| [0] | [1] | [2] | [3] | [4] | [5] | [6] | [7] | [8] | [9] | [10] | [11] |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 5 | 2 | 13 | 9 | 1 | 7 | 6 | 8 | 1 | 15 | 4 | 11 |

Our first gap is n/2.

Gap = n / 2 = 12 / 2 = 6.

İnitial array:

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 5 | 2 | 13 | 9 | 1 | 7 | 6 | 8 | 1 | 15 | 4 | 11 |

| Subarray | Comparing elements | [0] | [1] | [2] | [3] | [4] | [5] | [6] | [7] | [8] | [9] | [10] | [11] | isChanged |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 - 6 | 0 - 6 | 5 | 2 | 13 | 9 | 1 | 7 | 6 | 8 | 1 | 15 | 4 | 11 | False |
| 1 - 7 | 1 - 7 | 5 | 2 | 13 | 9 | 1 | 7 | 6 | 8 | 1 | 15 | 4 | 11 | False |
| 2 - 8 | 2 - 8 | 5 | 2 | 1 | 9 | 1 | 7 | 6 | 8 | 13 | 15 | 4 | 11 | True |
| 3 – 9 | 3 – 9 | 5 | 2 | 1 | 9 | 1 | 7 | 6 | 8 | 13 | 15 | 4 | 11 | False |
| 4 – 10 | 4 – 10 | 5 | 2 | 1 | 9 | 1 | 7 | 6 | 8 | 13 | 15 | 4 | 11 | False |
| 5 – 11 | 5 - 11 | 5 | 2 | 1 | 9 | 1 | 7 | 6 | 8 | 13 | 15 | 4 | 11 | False |

**Note:** *The blue boxes are the array's current state after the comparison process.*

*When gap is equal 6, 6 comparison and 1 displacement done.*

*gap = (int) (gap / 2.2);*

*gap = (int) (6 / 2.2) = 2.*

***When gap = 2;***

***İnitial array:***

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 5 | 2 | 1 | 9 | 1 | 7 | 6 | 8 | 13 | 15 | 4 | 11 |

| Subarray | Comparing elements | [0] | [1] | [2] | [3] | [4] | [5] | [6] | [7] | [8] | [9] | [10] | [11] | isChanged |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 - 2 | 0 - 2 | 1 | 2 | 5 | 9 | 1 | 7 | 6 | 8 | 13 | 15 | 4 | 11 | True |
| 1 - 3 | 1 - 3 | 1 | 2 | 5 | 9 | 1 | 7 | 6 | 8 | 13 | 15 | 4 | 11 | False |
| 0 - 2 - 4 | 2 - 4 | 1 | 2 | 1 | 9 | 5 | 7 | 6 | 8 | 13 | 15 | 4 | 11 | True |
| 0 - 2 - 4 | 0 – 2 | 1 | 2 | 1 | 9 | 5 | 7 | 6 | 8 | 13 | 15 | 4 | 11 | False |
| 1 - 3 - 5 | 3 - 5 | 1 | 2 | 1 | 7 | 5 | 9 | 6 | 8 | 13 | 15 | 4 | 11 | True |
| 1 - 3 - 5 | 1 - 3 | 1 | 2 | 1 | 7 | 5 | 9 | 6 | 8 | 13 | 15 | 4 | 11 | False |
| 0 - 2 - 4 - 6 | 4 - 6 | 1 | 2 | 1 | 7 | 5 | 9 | 6 | 8 | 13 | 15 | 4 | 11 | False |
| 0 - 2 - 4 - 6 | 2 - 4 | 1 | 2 | 1 | 7 | 5 | 9 | 6 | 8 | 13 | 15 | 4 | 11 | False |
| 0 - 2 - 4 - 6 | 0 - 2 | 1 | 2 | 1 | 7 | 5 | 9 | 6 | 8 | 13 | 15 | 4 | 11 | False |
| 1 - 3 - 5 - 7 | 5 - 7 | 1 | 2 | 1 | 7 | 5 | 8 | 6 | 9 | 13 | 15 | 4 | 11 | True |
| 1 - 3 - 5 - 7 | 3 - 5 | 1 | 2 | 1 | 7 | 5 | 8 | 6 | 9 | 13 | 15 | 4 | 11 | False |
| 1 - 3 - 5 - 7 | 1 - 3 | 1 | 2 | 1 | 7 | 5 | 8 | 6 | 9 | 13 | 15 | 4 | 11 | False |
| 0 - 2 - 4 - 6 - 8 | 6 - 8 | 1 | 2 | 1 | 7 | 5 | 8 | 6 | 9 | 13 | 15 | 4 | 11 | False |
| 0 - 2 - 4 - 6 – 8 | 4 - 6 | 1 | 2 | 1 | 7 | 5 | 8 | 6 | 9 | 13 | 15 | 4 | 11 | False |
| 0 - 2 - 4- 6 - 8 | 2 - 4 | 1 | 2 | 1 | 7 | 5 | 8 | 6 | 9 | 13 | 15 | 4 | 11 | False |
| 0 - 2 - 4 - 6 - 8 | 0 - 2 | 1 | 2 | 1 | 7 | 5 | 8 | 6 | 9 | 13 | 15 | 4 | 11 | False |
| 1 - 3 - 5 - 7 - 9 | 7 - 9 | 1 | 2 | 1 | 7 | 5 | 8 | 6 | 9 | 13 | 15 | 4 | 11 | False |

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 - 3 - 5 - 7 – 9 | 5 - 7 | 1 | 2 | 1 | 7 | 5 | 8 | 6 | 9 | 13 | 15 | 4 | 11 | False |
| 1 - 3 - 5 - 7 - 9 | 3 - 5 | 1 | 2 | 1 | 7 | 5 | 8 | 6 | 9 | 13 | 15 | 4 | 11 | False |
| 1 - 3 - 5 - 7 - 9 | 1 - 3 | 1 | 2 | 1 | 7 | 5 | 8 | 6 | 9 | 13 | 15 | 4 | 11 | False |
| 0 - 2 - 4 - 6 – 8 - 10 | 8 - 10 | 1 | 2 | 1 | 7 | 5 | 8 | 6 | 9 | 4 | 15 | 13 | 11 | True |
| 0 - 2 - 4 - 6 – 8 – 10 | 6 - 8 | 1 | 2 | 1 | 7 | 5 | 8 | 4 | 9 | 6 | 15 | 13 | 11 | True |
| 0 - 2 - 4 - 6 – 8 – 10 | 4 - 6 | 1 | 2 | 1 | 7 | 4 | 8 | 5 | 9 | 6 | 15 | 13 | 11 | True |
| 0 - 2 - 4 - 6 – 8 - 10 | 2 - 4 | 1 | 2 | 1 | 7 | 4 | 8 | 5 | 9 | 6 | 15 | 13 | 11 | False |
| 0 - 2 - 4 - 6 – 8 - 10 | 0 - 2 | 1 | 2 | 1 | 7 | 4 | 8 | 5 | 9 | 6 | 15 | 13 | 11 | False |
| 1 - 3 - 5 - 7 – 9 - 11 | 9 - 11 | 1 | 2 | 1 | 7 | 4 | 8 | 5 | 9 | 6 | 11 | 13 | 15 | True |
| 1 - 3 - 5 - 7 – 9 – 11 | 7 - 9 | 1 | 2 | 1 | 7 | 4 | 8 | 5 | 9 | 6 | 11 | 13 | 15 | False |
| 1 - 3 - 5 - 7 – 9 – 11 | 5 - 7 | 1 | 2 | 1 | 7 | 4 | 8 | 5 | 9 | 6 | 11 | 13 | 15 | False |
| 1 - 3 - 5 - 7 – 9 – 11 | 3 - 5 | 1 | 2 | 1 | 7 | 4 | 8 | 5 | 9 | 6 | 11 | 13 | 15 | False |
| 1 - 3 - 5 - 7 – 9 - 11 | 1 - 3 | 1 | 2 | 1 | 7 | 4 | 8 | 5 | 9 | 6 | 11 | 13 | 15 | False |

***Note:*** ***The blue boxes are the array's current state after the comparison process.***

***When gap is equal 2, 30 comparison and 8 displacement done.***

***gap = (int) (gap / 2.2);***

***gap = (int) (2 / 2) = 1.***

***When gap = 1;***

***İnitial array:***

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 1 | 7 | 4 | 8 | 5 | 9 | 6 | 11 | 13 | 15 |

| Subarray | Comparing elements | [0] | [1] | [2] | [3] | [4] | [5] | [6] | [7] | [8] | [9] | [10] | [11] | isChanged |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 - 1 | 0 - 1 | 1 | 2 | 1 | 7 | 4 | 8 | 5 | 9 | 6 | 11 | 13 | 15 | False |
| 0 - 1 - 2 | 1 - 2 | 1 | 1 | 2 | 7 | 4 | 8 | 5 | 9 | 6 | 11 | 13 | 15 | True |
| 0 - 1 - 2 | 0 - 1 | 1 | 1 | 2 | 7 | 4 | 8 | 5 | 9 | 6 | 11 | 13 | 15 | False |
| 0 - 1 - 2 -3 | 2 -3 | 1 | 1 | 2 | 7 | 4 | 8 | 5 | 9 | 6 | 11 | 13 | 15 | False |
| 0 - 1 - 2 -3 | 1 - 2 | 1 | 1 | 2 | 7 | 4 | 8 | 5 | 9 | 6 | 11 | 13 | 15 | False |
| 0 - 1 - 2 -3 | 0 - 1 | 1 | 1 | 2 | 7 | 4 | 8 | 5 | 9 | 6 | 11 | 13 | 15 | False |
| 0 - 1 - 2 -3 - 4 | 3 - 4 | 1 | 1 | 2 | 4 | 7 | 8 | 5 | 9 | 6 | 11 | 13 | 15 | True |
| 0 - 1 - 2 -3 - 4 | 2 -3 | 1 | 1 | 2 | 4 | 7 | 8 | 5 | 9 | 6 | 11 | 13 | 15 | False |
| 0 - 1 - 2 -3 - 4 | 1 - 2 | 1 | 1 | 2 | 4 | 7 | 8 | 5 | 9 | 6 | 11 | 13 | 15 | False |
| 0 - 1 - 2 -3 - 4 | 0 - 1 | 1 | 1 | 2 | 4 | 7 | 8 | 5 | 9 | 6 | 11 | 13 | 15 | False |
| 0 - 1 - 2 -3 - 4 - 5 | 4 - 5 | 1 | 1 | 2 | 4 | 7 | 8 | 5 | 9 | 6 | 11 | 13 | 15 | False |
| 0 - 1 - 2 -3 - 4 – 5 | 3 - 4 | 1 | 1 | 2 | 4 | 7 | 8 | 5 | 9 | 6 | 11 | 13 | 15 | False |
| 0 - 1 - 2 -3 - 4 – 5 | 2 -3 | 1 | 1 | 2 | 4 | 7 | 8 | 5 | 9 | 6 | 11 | 13 | 15 | False |
| 0 - 1 - 2 -3 - 4 – 5 | 1 - 2 | 1 | 1 | 2 | 4 | 7 | 8 | 5 | 9 | 6 | 11 | 13 | 15 | False |
| 0 - 1 - 2 -3 - 4 - 5 | 0 - 1 | 1 | 1 | 2 | 4 | 7 | 8 | 5 | 9 | 6 | 11 | 13 | 15 | False |
| 0 - 1 - 2 -3 - 4 - 5 - 6 | 5 - 6 | 1 | 1 | 2 | 4 | 7 | 5 | 8 | 9 | 6 | 11 | 13 | 15 | True |
| 0 - 1 - 2 -3 - 4 - 5 – 6 | 4 - 5 | 1 | 1 | 2 | 4 | 5 | 7 | 8 | 9 | 6 | 11 | 13 | 15 | True |
| 0 - 1 - 2 -3 - 4 - 5 – 6 | 3 - 4 | 1 | 1 | 2 | 4 | 5 | 7 | 8 | 9 | 6 | 11 | 13 | 15 | False |
| 0 - 1 - 2 -3 - 4 - 5 - 6 | 2 -3 | 1 | 1 | 2 | 4 | 5 | 7 | 8 | 9 | 6 | 11 | 13 | 15 | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 | 1 - 2 | 1 | 1 | 2 | 4 | 5 | 7 | 8 | 9 | 6 | 11 | 13 | 15 | False |
| 0 - 1 - 2 -3 - 4 - 5 - 6 | 0 - 1 | 1 | 1 | 2 | 4 | 5 | 7 | 8 | 9 | 6 | 11 | 13 | 15 | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 | 6 -7 | 1 | 1 | 2 | 4 | 5 | 7 | 8 | 9 | 6 | 11 | 13 | 15 | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 | 5 - 6 | 1 | 1 | 2 | 4 | 5 | 7 | 8 | 9 | 6 | 11 | 13 | 15 | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 | 4 - 5 | 1 | 1 | 2 | 4 | 5 | 7 | 8 | 9 | 6 | 11 | 13 | 15 | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 | 3 - 4 | 1 | 1 | 2 | 4 | 5 | 7 | 8 | 9 | 6 | 11 | 13 | 15 | False |

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 | 2 -3 | 1 | 1 | 2 | 4 | 5 | 7 | 8 | 9 | 6 | 11 | 13 | 15 | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 | 1 - 2 | 1 | 1 | 2 | 4 | 5 | 7 | 8 | 9 | 6 | 11 | 13 | 15 | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 | 0 - 1 | 1 | 1 | 2 | 4 | 5 | 7 | 8 | 9 | 6 | 11 | 13 | 15 | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 - 8 | 7 - 8 | 1 | 1 | 2 | 4 | 5 | 7 | 8 | 6 | 9 | 11 | 13 | 15 | True |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 | 6 -7 | 1 | 1 | 2 | 4 | 5 | 7 | 6 | 8 | 9 | 11 | 13 | 15 | True |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 | 5 - 6 | 1 | 1 | 2 | 4 | 5 | 6 | 7 | 8 | 9 | 11 | 13 | 15 | True |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 | 4 - 5 | 1 | 1 | 2 | 4 | 5 | 6 | 7 | 8 | 9 | 11 | 13 | 15 | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 | 3 - 4 | 1 | 1 | 2 | 4 | 5 | 6 | 7 | 8 | 9 | 11 | 13 | 15 | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 | 2 -3 | 1 | 1 | 2 | 4 | 5 | 6 | 7 | 8 | 9 | 11 | 13 | 15 | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 | 1 - 2 | 1 | 1 | 2 | 4 | 5 | 6 | 7 | 8 | 9 | 11 | 13 | 15 | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 - 8 | 0 - 1 | 1 | 1 | 2 | 4 | 5 | 6 | 7 | 8 | 9 | 11 | 13 | 15 | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 - 9 | 8 - 9 | 1 | 1 | 2 | 4 | 5 | 6 | 7 | 8 | 9 | 11 | 13 | 15 | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 – 9 | 7 - 8 | 1 | 1 | 2 | 4 | 5 | 6 | 7 | 8 | 9 | 11 | 13 | 15 | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 – 9 | 6 -7 | 1 | 1 | 2 | 4 | 5 | 6 | 7 | 8 | 9 | 11 | 13 | 15 | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 – 9 | 5 - 6 | 1 | 1 | 2 | 4 | 5 | 6 | 7 | 8 | 9 | 11 | 13 | 15 | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 – 9 | 4 - 5 | 1 | 1 | 2 | 4 | 5 | 6 | 7 | 8 | 9 | 11 | 13 | 15 | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 – 9 | 3 - 4 | 1 | 1 | 2 | 4 | 5 | 6 | 7 | 8 | 9 | 11 | 13 | 15 | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 – 9 | 2 -3 | 1 | 1 | 2 | 4 | 5 | 6 | 7 | 8 | 9 | 11 | 13 | 15 | False |

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 – 9 | 1 - 2 | 1 | 1 | 2 | 4 | 5 | 6 | 7 | 8 | 9 | 11 | 13 | 15 | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 - 9 | 0 - 1 | 1 | 1 | 2 | 4 | 5 | 6 | 7 | 8 | 9 | 11 | 13 | 15 | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 – 9 - 10 | 9 - 10 | 1 | 1 | 2 | 4 | 5 | 6 | 7 | 8 | 9 | 11 | 13 | 15 | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 – 9 - 10 | 8 - 9 | 1 | 1 | 2 | 4 | 5 | 6 | 7 | 8 | 9 | 11 | 13 | 15 | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 – 9 – 10 | 7 - 8 | 1 | 1 | 2 | 4 | 5 | 6 | 7 | 8 | 9 | 11 | 13 | 15 | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 – 9 - 10 | 6 -7 | 1 | 1 | 2 | 4 | 5 | 6 | 7 | 8 | 9 | 11 | 13 | 15 | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 – 9 - 10 | 5 - 6 | 1 | 1 | 2 | 4 | 5 | 6 | 7 | 8 | 9 | 11 | 13 | 15 | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 – 9 - 10 | 4 - 5 | 1 | 1 | 2 | 4 | 5 | 6 | 7 | 8 | 9 | 11 | 13 | 15 | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 – 9 - 10 | 3 - 4 | 1 | 1 | 2 | 4 | 5 | 6 | 7 | 8 | 9 | 11 | 13 | 15 | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 – 9 - 10 | 2 -3 | 1 | 1 | 2 | 4 | 5 | 6 | 7 | 8 | 9 | 11 | 13 | 15 | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 – 9 - 10 | 1 - 2 | 1 | 1 | 2 | 4 | 5 | 6 | 7 | 8 | 9 | 11 | 13 | 15 | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 – 9 - 10 | 0 - 1 | 1 | 1 | 2 | 4 | 5 | 6 | 7 | 8 | 9 | 11 | 13 | 15 | False |
| 0 - 1 - 2 -3 - 4 - 5 - 6 -7 - 8 - 9 - 10 - 11 | 10 - 11 | 1 | 1 | 2 | 4 | 5 | 6 | 7 | 8 | 9 | 11 | 13 | 15 | False |
| 0 - 1 - 2 -3 - 4 - 5 - 6 -7 - 8 - 9 - 10 - 11 | 9 - 10 | 1 | 1 | 2 | 4 | 5 | 6 | 7 | 8 | 9 | 11 | 13 | 15 | False |
| 0 - 1 - 2 -3 - 4 - 5 - 6 -7 - 8 - 9 - 10 - 11 | 8 - 9 | 1 | 1 | 2 | 4 | 5 | 6 | 7 | 8 | 9 | 11 | 13 | 15 | False |
| 0 - 1 - 2 -3 - 4 - 5 - 6 -7 - | 7 - 8 | 1 | 1 | 2 | 4 | 5 | 6 | 7 | 8 | 9 | 11 | 13 | 15 | False |

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 8 - 9 - 10 - 11 | | | | | | | | | | | | | | |
| 0 - 1 - 2 -3 - 4 - 5 - 6 -7 - 8 - 9 - 10 - 11 | 6 -7 | 1 | 1 | 2 | 4 | 5 | 6 | 7 | 8 | 9 | 11 | 13 | 15 | False |
| 0 - 1 - 2 -3 - 4 - 5 - 6 -7 - 8 - 9 - 10 - 11 | 5 - 6 | 1 | 1 | 2 | 4 | 5 | 6 | 7 | 8 | 9 | 11 | 13 | 15 | False |
| 0 - 1 - 2 -3 - 4 - 5 - 6 -7 - 8 - 9 - 10 - 11 | 4 - 5 | 1 | 1 | 2 | 4 | 5 | 6 | 7 | 8 | 9 | 11 | 13 | 15 | False |
| 0 - 1 - 2 -3 - 4 - 5 - 6 -7 - 8 - 9 - 10 - 11 | 3 - 4 | 1 | 1 | 2 | 4 | 5 | 6 | 7 | 8 | 9 | 11 | 13 | 15 | False |
| 0 - 1 - 2 -3 - 4 - 5 - 6 -7 - 8 - 9 - 10 - 11 | 2 -3 | 1 | 1 | 2 | 4 | 5 | 6 | 7 | 8 | 9 | 11 | 13 | 15 | False |
| 0 - 1 - 2 -3 - 4 - 5 - 6 -7 - 8 - 9 - 10 - 11 | 1 - 2 | 1 | 1 | 2 | 4 | 5 | 6 | 7 | 8 | 9 | 11 | 13 | 15 | False |
| 0 - 1 - 2 -3 - 4 - 5 - 6 -7 - 8 - 9 - 10 – 11 | 0 - 1 | 1 | 1 | 2 | 4 | 5 | 6 | 7 | 8 | 9 | 11 | 13 | 15 | False |

***Note:** The blue boxes are the array's current state after the comparison process.*

***When gap is equal 1, 66 comparison and 7 displacement done.***

## *In total 102 comparison and 16 displacement*

- ***Applying merge sort***

***C = {5, 2, 13, 9, 1, 7, 6, 8, 1, 15, 4, 11}***

***n = 12***

| [0] | [1] | [2] | [3] | [4] | [5] | [6] | [7] | [8] | [9] | [10] | [11] |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 5 | 2 | 13 | 9 | 1 | 7 | 6 | 8 | 1 | 15 | 4 | 11 |

***n = n / 2*** ***n = n / 2***

| | | | | | |
|---|---|---|---|---|---|
| 5 | 2 | 13 | 9 | 1 | 7 |

| | | | | | |
|---|---|---|---|---|---|
| 6 | 8 | 1 | 15 | 4 | 11 |

***n = n / 2*** ***n = n / 2***

| | | |
|---|---|---|
| 5 | 2 | 13 |

| | | |
|---|---|---|
| 9 | 1 | 7 |

***n = n / 2***

| |
|---|
| 5 |

| | |
|---|---|
| 2 | 3 |

| |
|---|
| 2 |

| |
|---|
| 3 |

***1) Compare 2 and 13. And associate as sorted version.***

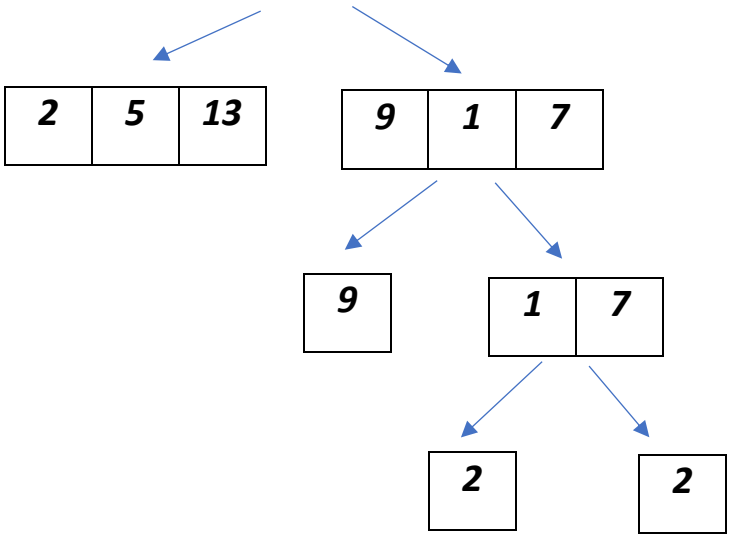
***Output:***

| | |
|---|---|
| 2 | 13 |

2) Compare | 5 | and | 2 | 13 |

- Compare 5 - 2 → | 2 |
- Compare 5 - 13 → | 2 | 5 |
- Add 13 → | 2 | 5 | 13 |

3)




4) Compare 1 and 7. And associate as sorted version.

**Output:** | 1 | 7 |

5) Compare | 9 | and | 1 | 7 |

- Compare 9 - 1 → | 1 |
- Compare 9 - 7 → | 1 | 7 |

6)

- *Add 9 →* 

| | | |
|---|---|---|
| 1 | 7 | 9 |

| | | |
|---|---|---|
| 2 | 5 | 13 |

| | | |
|---|---|---|
| 1 | 7 | 9 |

- *Compare 2 - 1 →*

| |
|---|
| 1 |

- *Compare 2 - 7 →*

| | |
|---|---|
| 1 | 2 |

- *Compare 5 - 7 →*

| | | |
|---|---|---|
| 1 | 2 | 5 |

- *Compare 13 - 7 →*

| | | | |
|---|---|---|---|
| 1 | 2 | 5 | 7 |

- *Compare 13 - 9 →*

| | | | | |
|---|---|---|---|---|
| 1 | 2 | 5 | 7 | 9 |

- *Add 13 →*

| | | | | | |
|---|---|---|---|---|---|
| 1 | 2 | 5 | 7 | 9 | 13 |

7)

| | | | | | |
|---|---|---|---|---|---|
| 1 | 2 | 5 | 7 | 9 | 13 |

| | | | | | |
|---|---|---|---|---|---|
| 6 | 8 | 1 | 15 | 4 | 11 |

| | | |
|---|---|---|
| 6 | 8 | 1 |

| | | |
|---|---|---|
| 15 | 4 | 11 |

| |
|---|
| 6 |

| | |
|---|---|
| 8 | 1 |

| |
|---|
| 8 |

| |
|---|
| 1 |

8) *Compare 8 and 1. And associate as sorted version.*

**Output:**

| | |
|---|---|
| 1 | 8 |

9) *Compare* | 6 | *and* | 1 | 8 |

- *Compare 6 – 1 →* | 1 |
- *Compare 6 – 8 →*

| | |
|---|---|
| 1 | 6 |

- *Add 8 →*

| | | |
|---|---|---|
| 1 | 6 | 8 |

10)

| | | |
|---|---|---|
| 1 | 6 | 8 |

| | | |
|---|---|---|
| 15 | 4 | 11 |

| | |
|---|---|
| 15 | 4 11 |

| | |
|---|---|
| 4 | 11 |

11) *Compare 4 and 11. And associate as sorted version.*

**Output:**

| | |
|---|---|
| 4 | 11 |

12) *Compare* | 15 | *and*

| | |
|---|---|
| 4 | 11 |

- *Compare 15 – 4 →* | 4 |
- *Compare 15 – 11 →*

| | |
|---|---|
| 4 | 11 |

- *Add 15 →*

| | | |
|---|---|---|
| 4 | 11 | 15 |

***13)***

| | | |
|---|---|---|
| 1 | 6 | 8 |

| | | |
|---|---|---|
| 4 | 11 | 15 |

- ***Compare 1 - 4 →*** | 1 |
- ***Compare 6 – 4 →*** | 1 | 4 |
- ***Compare 6 – 11 →*** | 1 | 4 | 6 |
- ***Compare 8 – 11 →*** | 1 | 4 | 6 | 8 |
- ***Add 11 →*** | 1 | 4 | 6 | 8 | 11 |
- ***Add 15 →*** | 1 | 4 | 6 | 8 | 11 | 15 |

***14 )***

| | | | | | |
|---|---|---|---|---|---|
| 1 | 2 | 5 | 7 | 9 | 13 |

| | | | | | |
|---|---|---|---|---|---|
| 1 | 4 | 6 | 8 | 11 | 15 |

- ***Compare 1 - 1 →*** | 1 |
- ***Compare 1 – 4 →*** | 1 | 1 |
- ***Compare 2 - 4 →*** | 1 | 1 | 2 |
- ***Compare 5 - 4 →*** | 1 | 1 | 2 | 4 |

- ***Compare 5 - 6 →***

| | | | | |
|---|---|---|---|---|
| 1 | 1 | 2 | 4 | 5 |

- ***Compare 7 - 6 →***

| | | | | | |
|---|---|---|---|---|---|
| 1 | 1 | 2 | 4 | 5 | 6 |

- ***Compare 7 - 8 →***

| | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 1 | 2 | 4 | 5 | 6 | 7 |

- ***Compare 9 - 8 →***

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

- ***Compare 9 - 11 →***

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 2 | 4 | 5 | 6 | 7 | 8 | 9 |

- ***Compare 13 - 11 →***

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 2 | 4 | 5 | 6 | 7 | 8 | 9 | 11 |

- ***Compare 13 - 15 →***

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 2 | 4 | 5 | 6 | 7 | 8 | 9 | 11 | 13 |

- ***Add 15 →***

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 2 | 4 | 5 | 6 | 7 | 8 | 9 | 11 | 13 | 15 |

***Sorted array:***

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 2 | 4 | 5 | 6 | 7 | 8 | 9 | 11 | 13 | 15 |

***Result:***

***In this sorting algorithm, 32 comparison done.***

- ***Applying heap sort***

***C = {5, 2, 13, 9, 1, 7, 6, 8, 1, 15, 4, 11}***

***n = 12***

| [0] | [1] | [2] | [3] | [4] | [5] | [6] | [7] | [8] | [9] | [10] | [11] |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 5 | 2 | 13 | 9 | 1 | 7 | 6 | 8 | 1 | 15 | 4 | 11 |

***We have to do two steps for converting that array to the increasing order array.***

1) ***Build a max heap from array.***
2) ***Shrink the heap.***

***Build process***

```
private <T extends Comparable<T>> void buildHeap(T[] table) {
        int n = 1;
        // Invariant: table[0 . . . n - 1] is a heap.
        while (n < table.length) {
                n++; // Add a new item to the heap and reheap.
                int child = n - 1;
                int parent = (child - 1) / 2; // Find parent.
                while (parent >= 0
                && table[parent].compareTo(table[child]) < 0) {
                swap(table, parent, child);
                child = parent;
                parent = (child - 1) / 2;
        }
}
```

***I start to control of child parent relation from index 1. Every index which I controled, I looked at its parent up to root. If necessary I swap the values. Parent = (child-1)/2***

***İnital array:***

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 5 | 2 | 13 | 9 | 1 | 7 | 6 | 8 | 1 | 15 | 4 | 11 |

| Current child index | Comparing elements | [0] | [1] | [2] | [3] | [4] | [5] | [6] | [7] | [8] | [9] | [10] | [11] | isChanged |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 – 0 | 5 | 2 | 13 | 9 | 1 | 7 | 6 | 8 | 1 | 15 | 4 | 11 | False |
| 2 | 2 – 0 | 13 | 2 | 5 | 9 | 1 | 7 | 6 | 8 | 1 | 15 | 4 | 11 | True |
| 3 | 3 – 1 | 13 | 9 | 5 | 2 | 1 | 7 | 6 | 8 | 1 | 15 | 4 | 11 | True |
| 3 | 1 – 0 | 13 | 9 | 5 | 2 | 1 | 7 | 6 | 8 | 1 | 15 | 4 | 11 | False |
| 4 | 4 – 1 | 13 | 9 | 5 | 2 | 1 | 7 | 6 | 8 | 1 | 15 | 4 | 11 | False |
| 4 | 1 – 0 | 13 | 9 | 5 | 2 | 1 | 7 | 6 | 8 | 1 | 15 | 4 | 11 | False |
| 5 | 5 – 2 | 13 | 9 | 7 | 2 | 1 | 5 | 6 | 8 | 1 | 15 | 4 | 11 | True |
| 5 | 2 – 0 | 13 | 9 | 7 | 2 | 1 | 5 | 6 | 8 | 1 | 15 | 4 | 11 | False |
| 6 | 6 – 2 | 13 | 9 | 7 | 2 | 1 | 5 | 6 | 8 | 1 | 15 | 4 | 11 | False |
| 6 | 2 – 0 | 13 | 9 | 7 | 2 | 1 | 5 | 6 | 8 | 1 | 15 | 4 | 11 | False |
| 7 | 7 – 3 | 13 | 9 | 7 | 8 | 1 | 5 | 6 | 2 | 1 | 15 | 4 | 11 | True |
| 7 | 3 – 1 | 13 | 9 | 7 | 8 | 1 | 5 | 6 | 2 | 1 | 15 | 4 | 11 | False |
| 7 | 1 – 0 | 13 | 9 | 7 | 8 | 1 | 5 | 6 | 2 | 1 | 15 | 4 | 11 | False |
| 8 | 8 – 3 | 13 | 9 | 7 | 8 | 1 | 5 | 6 | 2 | 1 | 15 | 4 | 11 | False |
| 8 | 3 – 1 | 13 | 9 | 7 | 8 | 1 | 5 | 6 | 2 | 1 | 15 | 4 | 11 | False |
| 8 | 1 – 0 | 13 | 9 | 7 | 8 | 1 | 5 | 6 | 2 | 1 | 15 | 4 | 11 | False |
| 9 | 9 – 4 | 13 | 9 | 7 | 8 | 15 | 5 | 6 | 2 | 1 | 1 | 4 | 11 | True |
| 9 | 4 – 1 | 13 | 15 | 7 | 8 | 9 | 5 | 6 | 2 | 1 | 1 | 4 | 11 | True |

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 9 | 1 – 0 | 15 | 13 | 7 | 8 | 9 | 5 | 6 | 2 | 1 | 1 | 4 | 11 | True |
| 10 | 10 – 4 | 15 | 13 | 7 | 8 | 9 | 5 | 6 | 2 | 1 | 1 | 4 | 11 | False |
| 10 | 4 – 1 | 15 | 13 | 7 | 8 | 9 | 5 | 6 | 2 | 1 | 1 | 4 | 11 | False |
| 10 | 1 – 0 | 15 | 13 | 7 | 8 | 9 | 5 | 6 | 2 | 1 | 1 | 4 | 11 | False |
| 11 | 11 – 5 | 15 | 13 | 7 | 8 | 9 | 11 | 6 | 2 | 1 | 1 | 4 | 5 | True |
| 11 | 5 – 2 | 15 | 13 | 11 | 8 | 9 | 7 | 6 | 2 | 1 | 1 | 4 | 5 | True |
| 11 | 2 – 0 | 15 | 13 | 11 | 8 | 9 | 7 | 6 | 2 | 1 | 1 | 4 | 5 | True |

***In build process, 25 comparison and 10 displacement done***

***Max heap:***

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 15 | 13 | 11 | 8 | 9 | 7 | 6 | 2 | 1 | 1 | 4 | 5 |

## ***Shrink process***

```
private <T extends Comparable<T>> void shrinkHeap(T[] table) {
        int n = table.length;
        // Invariant: table[0 . . . n - 1] forms a heap.
        // table[n . . . table.length - 1] is sorted.
        while (n > 0) {
                n--;
                swap(table, 0, n);
                // table[1 . . . n - 1] form a heap.
                // table[n . . . table.length - 1] is sorted.
                int parent = 0;
                while (true) {
                        int leftChild = 2 * parent + 1;
                        if (leftChild >= n) {
                                break; // No more children.
                        }
                        int rightChild = leftChild + 1;
                        // Find the larger of the two children.
                        int maxChild = leftChild;
                        if (rightChild < n // There is a right child.
                        && table[leftChild].compareTo(table[rightChild]) < 0) {
                        maxChild = rightChild;
```

```
                    }
                    // If the parent is smaller than the larger child,
                    if (table[parent].compareTo(table[maxChild]) < 0){
                        // Swap the parent and child.
                        swap(table, parent, maxChild);
                        // Continue at the child level.
                        parent = maxChild;
            } else { // Heap property is restored.
                break; // Exit the loop.
            }
            }
            }
}
```

***Left child = 2 * parent + 1***

***Right child = 2 * parent + 2***

***While the blue box represents the current index, the green box represents the sorted section.***

| Left child index | Right child index | Compare notes | [0] | [1] | [2] | [3] | [4] | [5] | [6] | [7] | [8] | [9] | [10] | [11] | isChanged |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| - | - | Swap the first value and last value | 5 | 13 | 11 | 8 | 9 | 7 | 6 | 2 | 1 | 1 | 4 | 15 | True |
| 1 | 2 | [1] > [2] Swap [0] and [1] | 13 | 5 | 11 | 8 | 9 | 7 | 6 | 2 | 1 | 1 | 4 | 15 | True |
| 3 | 4 | [4] > [3] Swap [1] and [4] | 13 | 9 | 11 | 8 | 5 | 7 | 6 | 2 | 1 | 1 | 4 | 15 | True |
| 9 | 10 | [8] = [9] < [4]. | 13 | 9 | 11 | 8 | 5 | 7 | 6 | 2 | 1 | 1 | 4 | 15 | False |

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | No swap | | | | | | | | | | | | | |
| - | - | Swap the first value and last value | 4 | 9 | 11 | 8 | 5 | 7 | 6 | 2 | 1 | 1 | 13 | 15 | True |
| 1 | 2 | [2] > [1] Swap [0] and [2] | 11 | 9 | 4 | 8 | 5 | 7 | 6 | 2 | 1 | 1 | 13 | 15 | True |
| 5 | 6 | [5] > [6] Swap [2] and [5] | 11 | 9 | 7 | 8 | 5 | 4 | 6 | 2 | 1 | 1 | 13 | 15 | True |
| 11 | 12 | There is no 11 th and 12 th index in our array. | 11 | 9 | 7 | 8 | 5 | 4 | 6 | 2 | 1 | 1 | 13 | 15 | False |
| - | - | Swap the first value and last value | 1 | 9 | 7 | 8 | 5 | 4 | 6 | 2 | 1 | 11 | 13 | 15 | True |
| 1 | 2 | [1] > [2] Swap [0] and [1] | 9 | 1 | 7 | 8 | 5 | 4 | 6 | 2 | 1 | 11 | 13 | 15 | True |
| 3 | 4 | [3] > [4] Swap [1] and [3] | 9 | 8 | 7 | 1 | 5 | 4 | 6 | 2 | 1 | 11 | 13 | 15 | True |
| 7 | 8 | [7] > [8] | 9 | 8 | 7 | 2 | 5 | 4 | 6 | 1 | 1 | 11 | 13 | 15 | True |

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Swap [3] and [7] | | | | | | | | | | | | | |
| 15 | 16 | There is no 15 th and 16 th index in our array. | 9 | 8 | 7 | 2 | 5 | 4 | 6 | 1 | 1 | 11 | 13 | 15 | False |
| - | - | Swap the first value and last value | 1 | 8 | 7 | 2 | 5 | 4 | 6 | 1 | 9 | 11 | 13 | 15 | True |
| 1 | 2 | [1] > [2] Swap [0] and [1] | 8 | 1 | 7 | 2 | 5 | 4 | 6 | 1 | 9 | 11 | 13 | 15 | True |
| 3 | 4 | [4] > [3] Swap [1] and [4] | 8 | 5 | 7 | 2 | 1 | 4 | 6 | 1 | 9 | 11 | 13 | 15 | True |
| 9 | 10 | There is no 9 th and 10 th index in our array. | 8 | 5 | 7 | 2 | 1 | 4 | 6 | 1 | 9 | 11 | 13 | 15 | False |
| - | - | Swap the first value and last value | 1 | 5 | 7 | 2 | 1 | 4 | 6 | 8 | 9 | 11 | 13 | 15 | True |
| 1 | 2 | [2] >[1] Swap [0] and [2] | 7 | 5 | 1 | 2 | 1 | 4 | 6 | 8 | 9 | 11 | 13 | 15 | True |

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 5 | 6 | [6] > [5] Swap [2] and [6] | 7 | 5 | 6 | 2 | 1 | 4 | 1 | 8 | 9 | 11 | 13 | 15 | True |
| 13 | 14 | There is no 13 th and 14 th index in our array. | 7 | 5 | 6 | 2 | 1 | 4 | 1 | 8 | 9 | 11 | 13 | 15 | False |
| - | - | Swap the first value and last value | 1 | 5 | 6 | 2 | 1 | 4 | 7 | 8 | 9 | 11 | 13 | 15 | True |
| 1 | 2 | [2] > [1] Swap [0] and [2] | 6 | 5 | 1 | 2 | 1 | 4 | 7 | 8 | 9 | 11 | 13 | 15 | True |
| 5 | 6 | There is no 6 th index in our array. | 6 | 5 | 1 | 2 | 1 | 4 | 7 | 8 | 9 | 11 | 13 | 15 | False |
| - | - | Swap the first value and last value | 4 | 5 | 1 | 2 | 1 | 6 | 7 | 8 | 9 | 11 | 13 | 15 | True |
| 1 | 2 | [1] > [2] Swap [0] and [1] | 5 | 4 | 1 | 2 | 1 | 6 | 7 | 8 | 9 | 11 | 13 | 15 | True |

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 3 | 4 | [3] > [4] but [3] < [1]. No swap | 5 | 4 | 1 | 2 | 1 | 6 | 7 | 8 | 9 | 11 | 13 | 15 | False |
| - | - | Swap the first value and last value | 1 | 4 | 1 | 2 | 5 | 6 | 7 | 8 | 9 | 11 | 13 | 15 | True |
| 1 | 2 | [1] > [2] Swap [0] and [1] | 4 | 1 | 1 | 2 | 5 | 6 | 7 | 8 | 9 | 11 | 13 | 15 | True |
| 3 | 4 | There is no 4 th index in our array. | 4 | 1 | 1 | 2 | 5 | 6 | 7 | 8 | 9 | 11 | 13 | 15 | False |
| - | - | Swap the first value and last value | 2 | 1 | 1 | 4 | 5 | 6 | 7 | 8 | 9 | 11 | 13 | 15 | True |
| 1 | 2 | [1] > [2] But [0] > all of them. | 2 | 1 | 1 | 4 | 5 | 6 | 7 | 8 | 9 | 11 | 13 | 15 | False |
| - | - | Swap the first value and last value | 1 | 1 | 2 | 4 | 5 | 6 | 7 | 8 | 9 | 11 | 13 | 15 | True |

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *1* | *2* | *There is no 2 th index in our array.* | *1* | *1* | *2* | *4* | *5* | *6* | *7* | *8* | *9* | *11* | *13* | *15* | *False* |
| *-* | *-* | *Swap the first value and last value* | *1* | *1* | *2* | *4* | *5* | *6* | *7* | *8* | *9* | *11* | *13* | *15* | *True* |

***In shrink process, 35 comparison and 25 displacement done***

***Sorted array:***

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| *1* | *1* | *2* | *4* | *5* | *6* | *7* | *8* | *9* | *11* | *13* | *15* |

## *In total 60 comparison and 35 displacement*

- ***Applying QuickSort:***

***C = {5,2,13,9,1,7,6,8,1,15,4,11}***

| [0] | [1] | [2] | [3] | [4] | [5] | [6] | [7] | [8] | [9] | [10] | [11] |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 5 | 2 | 13 | 9 | 1 | 7 | 6 | 8 | 1 | 15 | 4 | 11 |

```
protected <T extends Comparable<T>> void quickSort(T[] table, int first, int last) {
	if (first < last) {
		int pivIndex = partition(table, first, last);
		quickSort(table, first, pivIndex - 1);
		quickSort(table, pivIndex + 1, last);
	}
}

protected <T extends Comparable<T>> int partition(T[] table, iint first, int last) {
	T pivot = table[first];
	int up = first; int down = last;
	do {
	while ((up < last) && (pivot.compareTo(table[up]) >= 0)) {
		up++;
	}
	while (pivot.compareTo(table[down]) < 0) {
		down--;
	}
	if (up < down) {
		swap(table, up, down);
	}
	} while (up < down); // Repeat while up is left of down.
	swap(table, first, down);
	return down;
}
```

→ Sort(table,0,11)

Pivot value = c[0] = 5

| Compare Pivot-current | Compare Result | Up | down | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | isChanged |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 0 | 11 | 5 | 2 | 13 | 9 | 1 | 7 | 6 | 8 | 1 | 15 | 4 | 11 | - |
| 5- [0] | = | 1 | 11 | 5 | 2 | 13 | 9 | 1 | 7 | 6 | 8 | 1 | 15 | 4 | 11 | False |
| 5 – [1] | > | 2 | 11 | 5 | 2 | 13 | 9 | 1 | 7 | 6 | 8 | 1 | 15 | 4 | 11 | False |
| 5 – [2] | < | 2 | 11 | 5 | 2 | 13 | 9 | 1 | 7 | 6 | 8 | 1 | 15 | 4 | 11 | False |
| 5 – [11] | < | 2 | 10 | 5 | 2 | 13 | 9 | 1 | 7 | 6 | 8 | 1 | 15 | 4 | 11 | False |
| 5 –[10] | > | 2 | 10 | 5 | 2 | 13 | 9 | 1 | 7 | 6 | 8 | 1 | 15 | 4 | 11 | False |
| | Swap(u,d) | 2 | 10 | 5 | 2 | 4 | 9 | 1 | 7 | 6 | 8 | 1 | 15 | 13 | 11 | True |
| 5 – [2] | > | 3 | 10 | 5 | 2 | 4 | 9 | 1 | 7 | 6 | 8 | 1 | 15 | 13 | 11 | False |
| 5 – [3] | < | 3 | 10 | 5 | 2 | 4 | 9 | 1 | 7 | 6 | 8 | 1 | 15 | 13 | 11 | False |
| 5 – [10] | < | 3 | 9 | 5 | 2 | 4 | 9 | 1 | 7 | 6 | 8 | 1 | 15 | 13 | 11 | False |
| 5 – [9] | < | 3 | 8 | 5 | 2 | 4 | 9 | 1 | 7 | 6 | 8 | 1 | 15 | 13 | 11 | False |
| 5 – [8] | > | 3 | 8 | 5 | 2 | 4 | 9 | 1 | 7 | 6 | 8 | 1 | 15 | 13 | 11 | False |
| | Swap(u,d) | 3 | 8 | 5 | 2 | 4 | 1 | 1 | 7 | 6 | 8 | 9 | 15 | 13 | 11 | True |
| 5 – [3] | > | 4 | 8 | 5 | 2 | 4 | 1 | 1 | 7 | 6 | 8 | 9 | 15 | 13 | 11 | False |
| 5 – [4] | > | 5 | 8 | 5 | 2 | 4 | 1 | 1 | 7 | 6 | 8 | 9 | 15 | 13 | 11 | False |
| 5 – [5] | < | 5 | 8 | 5 | 2 | 4 | 1 | 1 | 7 | 6 | 8 | 9 | 15 | 13 | 11 | False |
| 5 – [8] | < | 5 | 7 | 5 | 2 | 4 | 1 | 1 | 7 | 6 | 8 | 9 | 15 | 13 | 11 | False |
| 5 – [7] | < | 5 | 6 | 5 | 2 | 4 | 1 | 1 | 7 | 6 | 8 | 9 | 15 | 13 | 11 | False |
| 5 – [6] | < | 5 | 5 | 5 | 2 | 4 | 1 | 1 | 7 | 6 | 8 | 9 | 15 | 13 | 11 | False |

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 5 – [5] | < | 5 | 4 | 5 | 2 | 4 | 1 | 1 | 7 | 6 | 8 | 9 | 15 | 13 | 11 | False |
| 5 – [4] | > | 5 | 4 | 5 | 2 | 4 | 1 | 1 | 7 | 6 | 8 | 9 | 15 | 13 | 11 | False |
| | Swap(f,d) | 5 | 4 | 1 | 2 | 4 | 1 | 5 | 7 | 6 | 8 | 9 | 15 | 13 | 11 | True |

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 4 | 1 | 5 | 7 | 6 | 8 | 9 | 15 | 13 | 11 |

21 comparison and 3 displacement done.

→ Sort(table,0,3)

Pivot value = c[0] = 1

| Compare Pivot-current | Compare Result | Up | down | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | isChanged |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 0 | 3 | 1 | 2 | 4 | 1 | 5 | 7 | 6 | 8 | 9 | 15 | 13 | 11 | - |
| 1 – [0] | = | 1 | 3 | 1 | 2 | 4 | 1 | 5 | 7 | 6 | 8 | 9 | 15 | 13 | 11 | False |
| 1 – [1] | < | 1 | 3 | 1 | 2 | 4 | 1 | 5 | 7 | 6 | 8 | 9 | 15 | 13 | 11 | False |
| 1 – [3] | = | 1 | 3 | 1 | 2 | 4 | 1 | 5 | 7 | 6 | 8 | 9 | 15 | 13 | 11 | False |
| | Swap(u,d) | 1 | 3 | 1 | 1 | 4 | 2 | 5 | 7 | 6 | 8 | 9 | 15 | 13 | 11 | True |
| 1 – [1] | = | 2 | 3 | 1 | 1 | 4 | 2 | 5 | 7 | 6 | 8 | 9 | 15 | 13 | 11 | False |
| 1 – [2] | < | 2 | 3 | 1 | 1 | 4 | 2 | 5 | 7 | 6 | 8 | 9 | 15 | 13 | 11 | False |
| 1 – [3] | < | 2 | 2 | 1 | 1 | 4 | 2 | 5 | 7 | 6 | 8 | 9 | 15 | 13 | 11 | False |
| 1 – [2] | < | 2 | 1 | 1 | 1 | 4 | 2 | 5 | 7 | 6 | 8 | 9 | 15 | 13 | 11 | False |
| 1 – [1] | = | 2 | 1 | 1 | 1 | 4 | 2 | 5 | 7 | 6 | 8 | 9 | 15 | 13 | 11 | False |
| | Swap(f,d) | 2 | 1 | 1 | 1 | 4 | 2 | 5 | 7 | 6 | 8 | 9 | 15 | 13 | 11 | True |

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 4 | 2 | 5 | 7 | 6 | 8 | 9 | 15 | 13 | 11 |

*10 comparison and 2 displacement done.*

➔ *Sort(table,2,3)*

*Pivot value = c[2] = 4*

| Compare Pivot-current | Compare Result | Up | down | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | isChanged |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 2 | 3 | 1 | 1 | 4 | 2 | 5 | 7 | 6 | 8 | 9 | 15 | 13 | 11 | - |
| 4 – [2] | = | 3 | 3 | 1 | 1 | 4 | 2 | 5 | 7 | 6 | 8 | 9 | 15 | 13 | 11 | False |
| 4 – [3] | > | 3 | 3 | 1 | 1 | 4 | 2 | 5 | 7 | 6 | 8 | 9 | 15 | 13 | 11 | False |
| | Swap(f,d) | 3 | 3 | 1 | 1 | 2 | 4 | 5 | 7 | 6 | 8 | 9 | 15 | 13 | 11 | True |

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 2 | 4 | 5 | 7 | 6 | 8 | 9 | 15 | 13 | 11 |

*3 comparison and 1 displacement done.*

➔ *Sort(table,5,11)*

*Pivot value = c[5] = 7*

| Compare Pivot-current | Compare Result | Up | down | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | isChanged |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 5 | 11 | 1 | 1 | 2 | 4 | 5 | 7 | 6 | 8 | 9 | 15 | 13 | 11 | - |
| 7 – [5] | = | 6 | 11 | 1 | 1 | 2 | 4 | 5 | 7 | 6 | 8 | 9 | 15 | 13 | 11 | False |

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 7 – [6] | > | 7 | 11 | 1 | 1 | 2 | 4 | 5 | 7 | 6 | 8 | 9 | 15 | 13 | 11 | False |
| 7 – [7] | < | 7 | 11 | 1 | 1 | 2 | 4 | 5 | 7 | 6 | 8 | 9 | 15 | 13 | 11 | False |
| 7 – [11] | < | 7 | 10 | 1 | 1 | 2 | 4 | 5 | 7 | 6 | 8 | 9 | 15 | 13 | 11 | False |
| 7 – [10] | < | 7 | 9 | 1 | 1 | 2 | 4 | 5 | 7 | 6 | 8 | 9 | 15 | 13 | 11 | False |
| 7 – [9] | < | 7 | 8 | 1 | 1 | 2 | 4 | 5 | 7 | 6 | 8 | 9 | 15 | 13 | 11 | False |
| 7 – [8] | < | 7 | 7 | 1 | 1 | 2 | 4 | 5 | 7 | 6 | 8 | 9 | 15 | 13 | 11 | False |
| 7 – [7] | < | 7 | 6 | 1 | 1 | 2 | 4 | 5 | 7 | 6 | 8 | 9 | 15 | 13 | 11 | False |
| 7 – [6] | > | 7 | 6 | 1 | 1 | 2 | 4 | 5 | 7 | 6 | 8 | 9 | 15 | 13 | 11 | False |
| | Swap(f,d) | 7 | 6 | 1 | 1 | 2 | 4 | 5 | 6 | 7 | 8 | 9 | 15 | 13 | 11 | True |

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 2 | 4 | 5 | 6 | 7 | 8 | 9 | 15 | 13 | 11 |

*10 comparison and 1 displacement done.*

*➔ Sort(table,7,11)*

*Pivot value = c[7] = 8*

| Compare Pivot-current | Compare Result | Up | down | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | isChanged |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 7 | 11 | 1 | 1 | 2 | 4 | 5 | 6 | 7 | 8 | 9 | 15 | 13 | 11 | - |
| 8 – [7] | = | 8 | 11 | 1 | 1 | 2 | 4 | 5 | 6 | 7 | 8 | 9 | 15 | 13 | 11 | False |

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 8 – [8] | < | 8 | 11 | 1 | 1 | 2 | 4 | 5 | 6 | 7 | 8 | 9 | 15 | 13 | 11 | False |
| 8 – [11] | < | 8 | 10 | 1 | 1 | 2 | 4 | 5 | 6 | 7 | 8 | 9 | 15 | 13 | 11 | False |
| 8 – [10] | < | 8 | 9 | 1 | 1 | 2 | 4 | 5 | 6 | 7 | 8 | 9 | 15 | 13 | 11 | False |
| 8 – [9] | < | 8 | 8 | 1 | 1 | 2 | 4 | 5 | 6 | 7 | 8 | 9 | 15 | 13 | 11 | False |
| 8 – [8] | < | 8 | 7 | 1 | 1 | 2 | 4 | 5 | 6 | 7 | 8 | 9 | 15 | 13 | 11 | False |
| 8 – [7] | = | 8 | 7 | 1 | 1 | 2 | 4 | 5 | 6 | 7 | 8 | 9 | 15 | 13 | 11 | False |
| | Swap(f,d) | 8 | 7 | 1 | 1 | 2 | 4 | 5 | 6 | 7 | 8 | 9 | 15 | 13 | 11 | False |

***On last stage, in spite of swap, there is no displacement. Because of f =7 and d = 7, so it wont change.***

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 2 | 4 | 5 | 6 | 7 | 8 | 9 | 15 | 13 | 11 |

***8 comparison and 0 displacement done.***

***➔ Sort(table,8,11)***

***Pivot value = c[8] = 9***

| Compare Pivot-current | Compare Result | Up | down | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | isChanged |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 8 | 11 | 1 | 1 | 2 | 4 | 5 | 6 | 7 | 8 | 9 | 15 | 13 | 11 | - |
| 9 – [8] | = | 9 | 11 | 1 | 1 | 2 | 4 | 5 | 6 | 7 | 8 | 9 | 15 | 13 | 11 | False |

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 9 – [9] | < | 9 | 11 | 1 | 1 | 2 | 4 | 5 | 6 | 7 | 8 | 9 | 15 | 13 | 11 | False |
| 9 – [11] | < | 9 | 10 | 1 | 1 | 2 | 4 | 5 | 6 | 7 | 8 | 9 | 15 | 13 | 11 | False |
| 9 – [10] | < | 9 | 9 | 1 | 1 | 2 | 4 | 5 | 6 | 7 | 8 | 9 | 15 | 13 | 11 | False |
| 9 – [9] | < | 9 | 8 | 1 | 1 | 2 | 4 | 5 | 6 | 7 | 8 | 9 | 15 | 13 | 11 | False |
| 9 – [8] | = | 9 | 8 | 1 | 1 | 2 | 4 | 5 | 6 | 7 | 8 | 9 | 15 | 13 | 11 | False |
| | Swap(f,d) | 9 | 8 | 1 | 1 | 2 | 4 | 5 | 6 | 7 | 8 | 9 | 15 | 13 | 11 | False |

***On last stage, in spite of swap, there is no displacement. Because of f =8 and d = 8, so it wont change.***

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 2 | 4 | 5 | 6 | 7 | 8 | 9 | 15 | 13 | 11 |

***7 comparison and 0 displacement done.***

***➔ Sort(table,9,11)***

***Pivot value = c[9] = 15***

| Compare Pivot-current | Compare Result | Up | down | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | isChanged |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 9 | 11 | 1 | 1 | 2 | 4 | 5 | 6 | 7 | 8 | 9 | 15 | 13 | 11 | - |
| 15 – [9] | = | 10 | 11 | 1 | 1 | 2 | 4 | 5 | 6 | 7 | 8 | 9 | 15 | 13 | 11 | False |
| 15 –[10] | > | 11 | 11 | 1 | 1 | 2 | 4 | 5 | 6 | 7 | 8 | 9 | 15 | 13 | 11 | False |

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *15 –[11]* | *>* | *11* | *11* | *1* | *1* | *2* | *4* | *5* | *6* | *7* | *8* | *9* | *15* | *13* | *11* | *False* |
| | *Swap(f,d)* | *11* | *11* | *1* | *1* | *2* | *4* | *5* | *6* | *7* | *8* | *9* | *11* | *13* | *15* | *True* |

***4 comparison and 1 displacement done.***

***Sorted array:***

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| *1* | *1* | *2* | *4* | *5* | *6* | *7* | *8* | *9* | *11* | *13* | *15* |

## *In total 53 comparison and 8 displacement done.*

4) D = {'S', 'B', 'I', 'M', 'H', 'Q', 'C', 'L', 'R', 'E', 'P', 'K'}

- ***Applying Shell sort:***

*n = 12*

| [0] | [1] | [2] | [3] | [4] | [5] | [6] | [7] | [8] | [9] | [10] | [11] |
|---|---|---|---|---|---|---|---|---|---|---|---|
| S | B | I | M | H | Q | C | L | R | E | P | K |

*Our first gap is n/2.*

*Gap = n / 2 = 12 / 2 = 6.*

| Subarray | Comparing elements | [0] | [1] | [2] | [3] | [4] | [5] | [6] | [7] | [8] | [9] | [10] | [11] | isChanged |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 - 6 | 0 - 6 | C | B | I | M | H | Q | S | L | R | E | P | K | True |
| 1 - 7 | 1 - 7 | C | B | I | M | H | Q | S | L | R | E | P | K | False |
| 2 - 8 | 2 - 8 | C | B | I | M | H | Q | S | L | R | E | P | K | False |
| 3 – 9 | 3 – 9 | C | B | I | E | H | Q | S | L | R | M | P | K | True |
| 4 – 10 | 4 – 10 | C | B | I | E | H | Q | S | L | R | M | P | K | False |
| 5 – 11 | 5 - 11 | C | B | I | E | H | K | S | L | R | M | P | Q | True |

*İnitial array:*

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

***Note:*** ***The blue boxes are the array's current state after the comparison process.***

***When gap is equal 6, 6 comparison and 3 displacement done.***

*gap = (int) (gap / 2.2);*

*gap = (int) (6 / 2.2) = 2.*

***When gap = 2;***

***İnitial array:***

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

| Subarray | Comparing elements | [0] | [1] | [2] | [3] | [4] | [5] | [6] | [7] | [8] | [9] | [10] | [11] | isChanged |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 - 2 | 0 - 2 | C | B | I | E | H | K | S | L | R | M | P | Q | False |
| 1 - 3 | 1 - 3 | C | B | I | E | H | K | S | L | R | M | P | Q | False |
| 0 - 2 - 4 | 2 - 4 | C | B | H | E | I | K | S | L | R | M | P | Q | True |
| 0 - 2 - 4 | 0 – 2 | C | B | H | E | I | K | S | L | R | M | P | Q | False |
| 1 - 3 - 5 | 3 - 5 | C | B | H | E | I | K | S | L | R | M | P | Q | False |
| 1 - 3 - 5 | 1 - 3 | C | B | H | E | I | K | S | L | R | M | P | Q | False |
| 0 - 2 - 4 - 6 | 4 - 6 | C | B | H | E | I | K | S | L | R | M | P | Q | False |
| 0 - 2 - 4 - 6 | 2 - 4 | C | B | H | E | I | K | S | L | R | M | P | Q | False |
| 0 - 2 - 4 - 6 | 0 - 2 | C | B | H | E | I | K | S | L | R | M | P | Q | False |
| 1 - 3 - 5 - 7 | 5 - 7 | C | B | H | E | I | K | S | L | R | M | P | Q | False |
| 1 - 3 - 5 - 7 | 3 - 5 | C | B | H | E | I | K | S | L | R | M | P | Q | False |
| 1 - 3 - 5 - 7 | 1 - 3 | C | B | H | E | I | K | S | L | R | M | P | Q | False |
| 0 - 2 - 4 - 6 - 8 | 6 - 8 | C | B | H | E | I | K | R | L | S | M | P | Q | True |
| 0 - 2 - 4 - 6 – 8 | 4 - 6 | C | B | H | E | I | K | R | L | S | M | P | Q | False |
| 0 - 2 - 4- 6 - 8 | 2 - 4 | C | B | H | E | I | K | R | L | S | M | P | Q | False |
| 0 - 2 - 4 - 6 - 8 | 0 - 2 | C | B | H | E | I | K | R | L | S | M | P | Q | False |
| 1 - 3 - 5 - 7 - 9 | 7 - 9 | C | B | H | E | I | K | R | L | S | M | P | Q | False |
| 1 - 3 - 5 - 7 – 9 | 5 - 7 | C | B | H | E | I | K | R | L | S | M | P | Q | False |

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 - 3 - 5 - 7 - 9 | 3 - 5 | C | B | H | E | I | K | R | L | S | M | P | Q | False |
| 1 - 3 - 5 - 7 - 9 | 1 - 3 | C | B | H | E | I | K | R | L | S | M | P | Q | False |
| 0 - 2 - 4 - 6 – 8 - 10 | 8 - 10 | C | B | H | E | I | K | R | L | P | M | S | Q | True |
| 0 - 2 - 4 - 6 – 8 – 10 | 6 - 8 | C | B | H | E | I | K | P | L | R | M | S | Q | True |
| 0 - 2 - 4 - 6 – 8 – 10 | 4 - 6 | C | B | H | E | I | K | P | L | R | M | S | Q | False |
| 0 - 2 - 4 - 6 – 8 - 10 | 2 - 4 | C | B | H | E | I | K | P | L | R | M | S | Q | False |
| 0 - 2 - 4 - 6 – 8 - 10 | 0 - 2 | C | B | H | E | I | K | P | L | R | M | S | Q | False |
| 1 - 3 - 5 - 7 – 9 - 11 | 9 - 11 | C | B | H | E | I | K | P | L | R | M | S | Q | False |
| 1 - 3 - 5 - 7 – 9 – 11 | 7 - 9 | C | B | H | E | I | K | P | L | R | M | S | Q | False |
| 1 - 3 - 5 - 7 – 9 – 11 | 5 - 7 | C | B | H | E | I | K | P | L | R | M | S | Q | False |
| 1 - 3 - 5 - 7 – 9 – 11 | 3 - 5 | C | B | H | E | I | K | P | L | R | M | S | Q | False |
| 1 - 3 - 5 - 7 – 9 - 11 | 1 - 3 | C | B | H | E | I | K | P | L | R | M | S | Q | False |

***Note:*** ***The blue boxes are the array's current state after the comparison process.***

***When gap is equal 2, 30 comparison and 4 displacement done.***

***gap = (int) (gap / 2.2);***

***gap = (int) (2 / 2) = 1.***

***When gap = 1;***

***İnitial array:***

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

| Subarray | Comparing elements | [0] | [1] | [2] | [3] | [4] | [5] | [6] | [7] | [8] | [9] | [10] | [11] | isChanged |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 - 1 | 0 - 1 | B | C | H | E | I | K | P | L | R | M | S | Q | True |
| 0 - 1 - 2 | 1 - 2 | B | C | H | E | I | K | P | L | R | M | S | Q | False |
| 0 - 1 - 2 | 0 - 1 | B | C | H | E | I | K | P | L | R | M | S | Q | False |
| 0 - 1 - 2 -3 | 2 -3 | B | C | E | H | I | K | P | L | R | M | S | Q | True |
| 0 - 1 - 2 -3 | 1 - 2 | B | C | E | H | I | K | P | L | R | M | S | Q | False |
| 0 - 1 - 2 -3 | 0 - 1 | B | C | E | H | I | K | P | L | R | M | S | Q | False |
| 0 - 1 - 2 -3 - 4 | 3 - 4 | B | C | E | H | I | K | P | L | R | M | S | Q | False |
| 0 - 1 - 2 -3 - 4 | 2 -3 | B | C | E | H | I | K | P | L | R | M | S | Q | False |
| 0 - 1 - 2 -3 - 4 | 1 - 2 | B | C | E | H | I | K | P | L | R | M | S | Q | False |
| 0 - 1 - 2 -3 - 4 | 0 - 1 | B | C | E | H | I | K | P | L | R | M | S | Q | False |
| 0 - 1 - 2 -3 - 4 - 5 | 4 - 5 | B | C | E | H | I | K | P | L | R | M | S | Q | False |
| 0 - 1 - 2 -3 - 4 – 5 | 3 - 4 | B | C | E | H | I | K | P | L | R | M | S | Q | False |
| 0 - 1 - 2 -3 - 4 – 5 | 2 -3 | B | C | E | H | I | K | P | L | R | M | S | Q | False |
| 0 - 1 - 2 -3 - 4 – 5 | 1 - 2 | B | C | E | H | I | K | P | L | R | M | S | Q | False |
| 0 - 1 - 2 -3 - 4 - 5 | 0 - 1 | B | C | E | H | I | K | P | L | R | M | S | Q | False |
| 0 - 1 - 2 -3 - 4 - 5 - 6 | 5 - 6 | B | C | E | H | I | K | P | L | R | M | S | Q | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 | 4 - 5 | B | C | E | H | I | K | P | L | R | M | S | Q | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 | 3 - 4 | B | C | E | H | I | K | P | L | R | M | S | Q | False |
| 0 - 1 - 2 -3 - 4 - 5 - 6 | 2 -3 | B | C | E | H | I | K | P | L | R | M | S | Q | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 | 1 - 2 | B | C | E | H | I | K | P | L | R | M | S | Q | False |
| 0 - 1 - 2 -3 - 4 - 5 - 6 | 0 - 1 | B | C | E | H | I | K | P | L | R | M | S | Q | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 | 6 -7 | B | C | E | H | I | K | L | P | R | M | S | Q | True |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 | 5 - 6 | B | C | E | H | I | K | L | P | R | M | S | Q | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 | 4 - 5 | B | C | E | H | I | K | L | P | R | M | S | Q | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 | 3 - 4 | B | C | E | H | I | K | L | P | R | M | S | Q | False |

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 | 2 -3 | B | C | E | H | I | K | L | P | R | M | S | Q | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 | 1 - 2 | B | C | E | H | I | K | L | P | R | M | S | Q | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 | 0 - 1 | B | C | E | H | I | K | L | P | R | M | S | Q | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 - 8 | 7 - 8 | B | C | E | H | I | K | L | P | R | M | S | Q | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 | 6 -7 | B | C | E | H | I | K | L | P | R | M | S | Q | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 | 5 - 6 | B | C | E | H | I | K | L | P | R | M | S | Q | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 | 4 - 5 | B | C | E | H | I | K | L | P | R | M | S | Q | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 | 3 - 4 | B | C | E | H | I | K | L | P | R | M | S | Q | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 | 2 -3 | B | C | E | H | I | K | L | P | R | M | S | Q | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 | 1 - 2 | B | C | E | H | I | K | L | P | R | M | S | Q | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 - 8 | 0 - 1 | B | C | E | H | I | K | L | P | R | M | S | Q | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 - 9 | 8 - 9 | B | C | E | H | I | K | L | P | M | R | S | Q | True |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 – 9 | 7 - 8 | B | C | E | H | I | K | L | M | P | R | S | Q | True |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 – 9 | 6 -7 | B | C | E | H | I | K | L | M | P | R | S | Q | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 – 9 | 5 - 6 | B | C | E | H | I | K | L | M | P | R | S | Q | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 – 9 | 4 - 5 | B | C | E | H | I | K | L | M | P | R | S | Q | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 – 9 | 3 - 4 | B | C | E | H | I | K | L | M | P | R | S | Q | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 – 9 | 2 -3 | B | C | E | H | I | K | L | M | P | R | S | Q | False |

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 – 9 | 1 - 2 | B | C | E | H | I | K | L | M | P | R | S | Q | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 - 9 | 0 - 1 | B | C | E | H | I | K | L | M | P | R | S | Q | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 – 9 - 10 | 9 - 10 | B | C | E | H | I | K | L | M | P | R | S | Q | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 – 9 - 10 | 8 - 9 | B | C | E | H | I | K | L | M | P | R | S | Q | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 – 9 – 10 | 7 - 8 | B | C | E | H | I | K | L | M | P | R | S | Q | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 – 9 - 10 | 6 -7 | B | C | E | H | I | K | L | M | P | R | S | Q | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 – 9 - 10 | 5 - 6 | B | C | E | H | I | K | L | M | P | R | S | Q | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 – 9 - 10 | 4 - 5 | B | C | E | H | I | K | L | M | P | R | S | Q | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 – 9 - 10 | 3 - 4 | B | C | E | H | I | K | L | M | P | R | S | Q | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 – 9 - 10 | 2 -3 | B | C | E | H | I | K | L | M | P | R | S | Q | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 – 9 - 10 | 1 - 2 | B | C | E | H | I | K | L | M | P | R | S | Q | False |
| 0 - 1 - 2 -3 - 4 - 5 – 6 -7 – 8 – 9 - 10 | 0 - 1 | B | C | E | H | I | K | L | M | P | R | S | Q | False |
| 0 - 1 - 2 -3 - 4 - 5 - 6 -7 - 8 - 9 - 10 - 11 | 10 - 11 | B | C | E | H | I | K | L | M | P | R | Q | S | True |
| 0 - 1 - 2 -3 - 4 - 5 - 6 -7 - 8 - 9 - 10 - 11 | 9 - 10 | B | C | E | H | I | K | L | M | P | Q | R | S | True |

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 - 1 - 2 -3 - 4 - 5 - 6 -7 - 8 - 9 - 10 - 11 | 8 - 9 | B | C | E | H | I | K | L | M | P | Q | R | S | False |
| 0 - 1 - 2 -3 - 4 - 5 - 6 -7 - 8 - 9 - 10 - 11 | 7 - 8 | B | C | E | H | I | K | L | M | P | Q | R | S | False |
| 0 - 1 - 2 -3 - 4 - 5 - 6 -7 - 8 - 9 - 10 - 11 | 6 -7 | B | C | E | H | I | K | L | M | P | Q | R | S | False |
| 0 - 1 - 2 -3 - 4 - 5 - 6 -7 - 8 - 9 - 10 - 11 | 5 - 6 | B | C | E | H | I | K | L | M | P | Q | R | S | False |
| 0 - 1 - 2 -3 - 4 - 5 - 6 -7 - 8 - 9 - 10 - 11 | 4 - 5 | B | C | E | H | I | K | L | M | P | Q | R | S | False |
| 0 - 1 - 2 -3 - 4 - 5 - 6 -7 - 8 - 9 - 10 - 11 | 3 - 4 | B | C | E | H | I | K | L | M | P | Q | R | S | False |
| 0 - 1 - 2 -3 - 4 - 5 - 6 -7 - 8 - 9 - 10 - 11 | 2 -3 | B | C | E | H | I | K | L | M | P | Q | R | S | False |
| 0 - 1 - 2 -3 - 4 - 5 - 6 -7 - 8 - 9 - 10 - 11 | 1 - 2 | B | C | E | H | I | K | L | M | P | Q | R | S | False |
| 0 - 1 - 2 -3 - 4 - 5 - 6 -7 - 8 - 9 - 10 – 11 | 0 - 1 | B | C | E | H | I | K | L | M | P | Q | R | S | False |

***Note:*** ***The blue boxes are the array's current state after the comparison process.***

***When gap is equal 1, 66 comparison and 7 displacement done.***

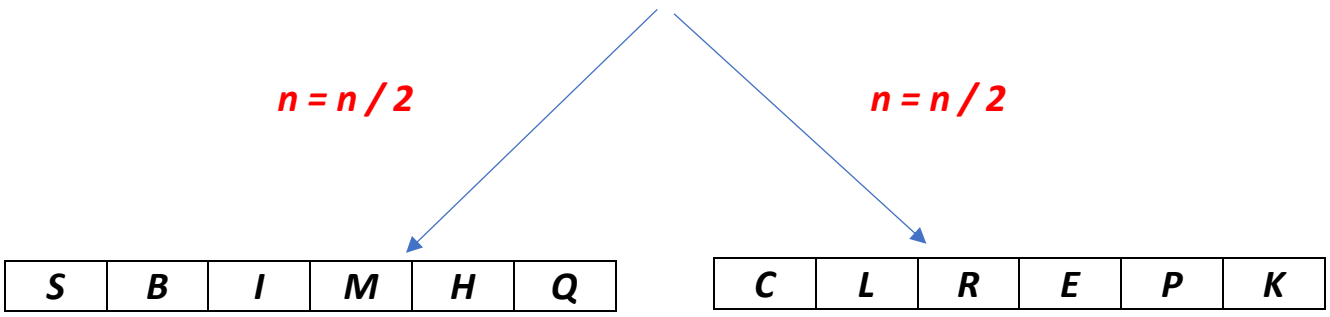
## *In total 102 comparison and 14 displacement done.*

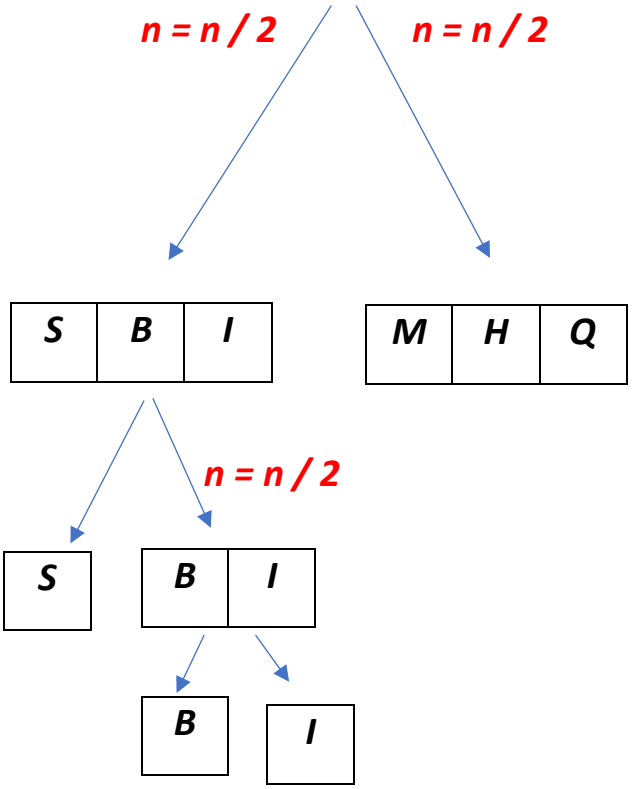
- ***Applying Merge sort:***

***D= {'S', 'B', 'I', 'M', 'H', 'Q', 'C', 'L', 'R', 'E', 'P', 'K'}***

***n = 12***

| [0] | [1] | [2] | [3] | [4] | [5] | [6] | [7] | [8] | [9] | [10] | [11] |
|---|---|---|---|---|---|---|---|---|---|---|---|
| S | B | I | M | H | Q | C | L | R | E | P | K |



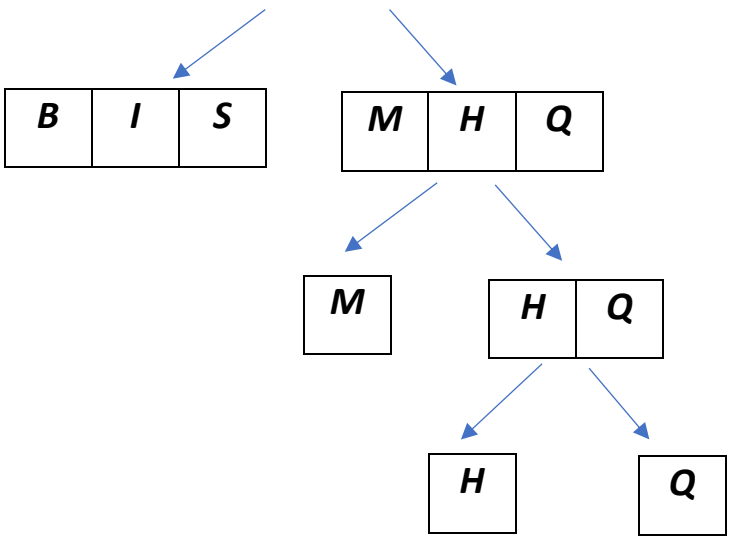
1) *Compare B and I. And associate as sorted version.*

**Output:** | B | I |

2) *Compare* | S | *and* | B | I |

- *Compare S - B →* | B |
- *Compare S – I →* | B | I |
- *Add S →* | B | I | S |

3)




4) *Compare H and Q. And associate as sorted version.*

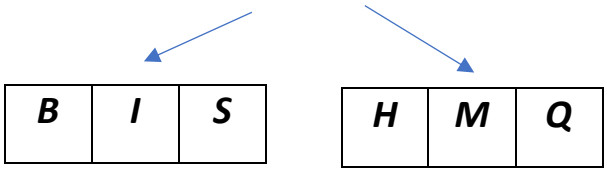
**Output:** | H | Q |

5) *Compare* | M | *and* | H | Q |

- *Compare M - H →* | H |
- *Compare M - Q →* | H | M |

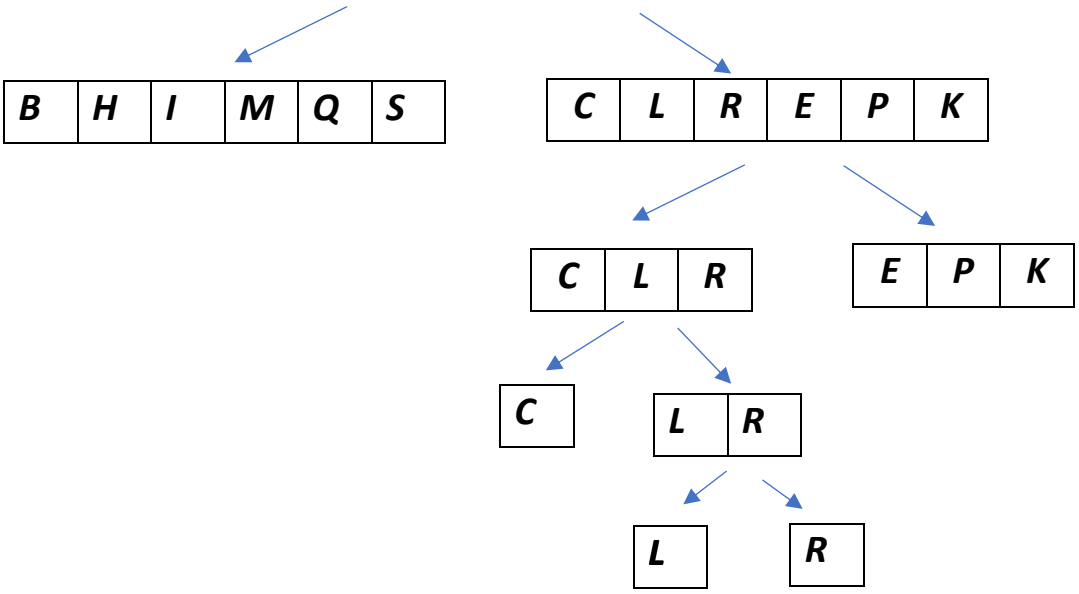
- *Add Q →* | H | M | Q |

*6)*




- *Compare B - H →* | B |
- *Compare I - H →* | B | H |
- *Compare I - M →* | B | H | I |
- *Compare S - M →* | B | H | I | M |
- *Compare S - Q →* | B | H | I | M | Q |
- *Add S →* | B | H | I | M | Q | S |

*7)*



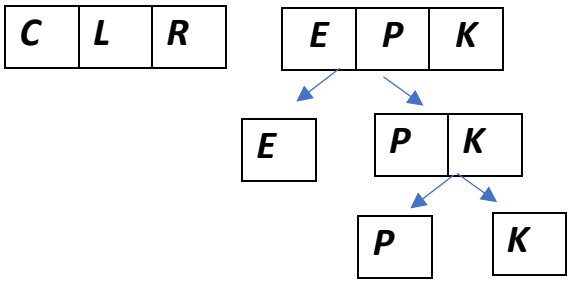
8) *Compare L and R. And associate as sorted version.*

**Output:** | L | R |

9) *Compare* | C | *and* | L | R |

- *Compare C – L →* | C |
- *Add L →* | C | L |
- *Add R →* | C | L | R |

10)




11) *Compare P and K. And associate as sorted version.*

**Output:** | K | P |

12) *Compare* | E | *and* | K | P |

- *Compare E – K →* | E |
- *Add K→* | E | K |
- *Add P →* | E | K | P |

***13)***

| | | |
|---|---|---|
| C | L | R |

| | | |
|---|---|---|
| E | K | P |

- ***Compare C - E →***

| |
|---|
| C |

- ***Compare L – E →***

| | |
|---|---|
| C | E |

- ***Compare L – K →***

| | | |
|---|---|---|
| C | E | K |

- ***Compare L – P →***

| | | | |
|---|---|---|---|
| C | E | K | L |

- ***Compare R – P →***

| | | | | |
|---|---|---|---|---|
| C | E | K | L | P |

- ***Add R →***

| | | | | | |
|---|---|---|---|---|---|
| C | E | K | L | P | R |

***14 )***

| | | | | | |
|---|---|---|---|---|---|
| B | H | I | M | Q | S |

| | | | | | |
|---|---|---|---|---|---|
| C | E | K | L | P | R |

- ***Compare B - C →***

| |
|---|
| B |

- ***Compare H – C →***

| | |
|---|---|
| B | C |

- ***Compare H - E →***

| | | |
|---|---|---|
| B | C | E |

- ***Compare H - K →***

| | | | |
|---|---|---|---|
| B | C | E | H |

- ***Compare I - K →***

| | | | | |
|---|---|---|---|---|
| B | C | E | H | I |

- ***Compare M - K →***

| | | | | | |
|---|---|---|---|---|---|
| B | C | E | H | I | K |

- ***Compare M - L →***

| | | | | | | |
|---|---|---|---|---|---|---|
| B | C | E | H | I | K | L |

- ***Compare M - P →***

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| B | C | E | H | I | K | L | M |

- ***Compare Q - P →***

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| B | C | E | H | I | K | L | M | P |

- ***Compare Q - R →***

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| B | C | E | H | I | K | L | M | P | Q |

- ***Compare S - R →***

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| B | C | E | H | I | K | L | M | P | Q | R |

- ***Add S →***

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

***Sorted array:***

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

***In total 30 comparison and 39 displacement done.***

- ***Applying heap sort:***

***D = {'S', 'B', 'I', 'M', 'H', 'Q', 'C', 'L', 'R', 'E', 'P', 'K'}***

***n = 12***

| [0] | [1] | [2] | [3] | [4] | [5] | [6] | [7] | [8] | [9] | [10] | [11] |
|---|---|---|---|---|---|---|---|---|---|---|---|
| S | B | I | M | H | Q | C | L | R | E | P | K |

***We have to do two steps for converting that array to the increasing order array.***

1) ***Build a max heap from array.***
2) ***Shrink the heap.***

***<u>Build process</u>***

```
private <T extends Comparable<T>> void buildHeap(T[] table) {
        int n = 1;
        // Invariant: table[0 . . . n - 1] is a heap.
        while (n < table.length) {
                n++; // Add a new item to the heap and reheap.
                int child = n - 1;
                int parent = (child - 1) / 2; // Find parent.
                while (parent >= 0
                && table[parent].compareTo(table[child]) < 0) {
                swap(table, parent, child);
                child = parent;
                parent = (child - 1) / 2;
        }
}
```

***I start to control of child parent relation from index 1. Every index which I controled, I looked at its parent up to root. If necessary I swap the values. Parent = (child-1)/2***

***İnital array:***

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

| Current child index | Comparing elements | [0] | [1] | [2] | [3] | [4] | [5] | [6] | [7] | [8] | [9] | [10] | [11] | isChanged |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 – 0 | S | B | I | M | H | Q | C | L | R | E | P | K | False |
| 2 | 2 – 0 | S | B | I | M | H | Q | C | L | R | E | P | K | False |
| 3 | 3 – 1 | S | M | I | B | H | Q | C | L | R | E | P | K | True |
| 3 | 1 – 0 | S | M | I | B | H | Q | C | L | R | E | P | K | False |
| 4 | 4 – 1 | S | M | I | B | H | Q | C | L | R | E | P | K | False |
| 4 | 1 – 0 | S | M | I | B | H | Q | C | L | R | E | P | K | False |
| 5 | 5 – 2 | S | M | Q | B | H | I | C | L | R | E | P | K | True |
| 5 | 2 – 0 | S | M | Q | B | H | I | C | L | R | E | P | K | False |
| 6 | 6 – 2 | S | M | Q | B | H | I | C | L | R | E | P | K | False |
| 6 | 2 – 0 | S | M | Q | B | H | I | C | L | R | E | P | K | False |
| 7 | 7 – 3 | S | M | Q | L | H | I | C | B | R | E | P | K | True |
| 7 | 3 – 1 | S | M | Q | L | H | I | C | B | R | E | P | K | False |
| 7 | 1 – 0 | S | M | Q | L | H | I | C | B | R | E | P | K | False |
| 8 | 8 – 3 | S | M | Q | R | H | I | C | B | L | E | P | K | True |
| 8 | 3 – 1 | S | R | Q | M | H | I | C | B | L | E | P | K | True |
| 8 | 1 – 0 | S | R | Q | M | H | I | C | B | L | E | P | K | False |
| 9 | 9 – 4 | S | R | Q | M | H | I | C | B | L | E | P | K | False |
| 9 | 4 – 1 | S | R | Q | M | H | I | C | B | L | E | P | K | False |

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ***9*** | ***1 – 0*** | ***S*** | ***R*** | ***Q*** | ***M*** | ***H*** | ***I*** | ***C*** | ***B*** | ***L*** | ***E*** | ***P*** | ***K*** | ***False*** |
| ***10*** | ***10 – 4*** | ***S*** | ***R*** | ***Q*** | ***M*** | ***P*** | ***I*** | ***C*** | ***B*** | ***L*** | ***E*** | ***H*** | ***K*** | ***True*** |
| ***10*** | ***4 – 1*** | ***S*** | ***R*** | ***Q*** | ***M*** | ***P*** | ***I*** | ***C*** | ***B*** | ***L*** | ***E*** | ***H*** | ***K*** | ***False*** |
| ***10*** | ***1 – 0*** | ***S*** | ***R*** | ***Q*** | ***M*** | ***P*** | ***I*** | ***C*** | ***B*** | ***L*** | ***E*** | ***H*** | ***K*** | ***False*** |
| ***11*** | ***11 – 5*** | ***S*** | ***R*** | ***Q*** | ***M*** | ***P*** | ***K*** | ***C*** | ***B*** | ***L*** | ***E*** | ***H*** | ***I*** | ***True*** |
| ***11*** | ***5 – 2*** | ***S*** | ***R*** | ***Q*** | ***M*** | ***P*** | ***K*** | ***C*** | ***B*** | ***L*** | ***E*** | ***H*** | ***I*** | ***False*** |
| ***11*** | ***2 – 0*** | ***S*** | ***R*** | ***Q*** | ***M*** | ***P*** | ***K*** | ***C*** | ***B*** | ***L*** | ***E*** | ***H*** | ***I*** | ***False*** |

***In build process, 25 comparison and 7 displacement done***

## ***Max heap:***

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ***S*** | ***R*** | ***Q*** | ***M*** | ***P*** | ***K*** | ***C*** | ***B*** | ***L*** | ***E*** | ***H*** | ***I*** |

## ***Shrink process***

```
private <T extends Comparable<T>> void shrinkHeap(T[] table) {
        int n = table.length;
        // Invariant: table[0 . . . n - 1] forms a heap.
        // table[n . . . table.length - 1] is sorted.
        while (n > 0) {
                n--;
                swap(table, 0, n);
                // table[1 . . . n - 1] form a heap.
                // table[n . . . table.length - 1] is sorted.
                int parent = 0;
                while (true) {
                        int leftChild = 2 * parent + 1;
                        if (leftChild >= n) {
                                break; // No more children.
                        }
                        int rightChild = leftChild + 1;
                        // Find the larger of the two children.
                        int maxChild = leftChild;
                        if (rightChild < n // There is a right child.
```

```
                && table[leftChild].compareTo(table[rightChild]) < 0) {
                maxChild = rightChild;
                }
                // If the parent is smaller than the larger child,
                if (table[parent].compareTo(table[maxChild]) < 0){
                        // Swap the parent and child.
                        swap(table, parent, maxChild);
                        // Continue at the child level.
                        parent = maxChild;
        } else { // Heap property is restored.
                break; // Exit the loop.
        }
        }
        }
}
```

***Left child = 2 * parent + 1***

***Right child = 2 * parent + 2***

***While the blue box represents the current index, the green box represents the sorted section.***

| Left child index | Right child index | Compare notes | [0] | [1] | [2] | [3] | [4] | [5] | [6] | [7] | [8] | [9] | [10] | [11] | isChanged |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| - | - | Swap the first value and last value | I | R | Q | M | P | K | C | B | L | E | H | S | True |
| 1 | 2 | [1] > [2] Swap [0] and [1] | R | I | Q | M | P | K | C | B | L | E | H | S | True |
| 3 | 4 | [4] > [3] Swap [1] and [4] | R | P | Q | M | I | K | C | B | L | E | H | S | True |

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 9 | 10 | [10] > [9] But [4] > [10] No swap | R | P | Q | M | I | K | C | B | L | E | H | S | False |
| - | - | Swap the first value and last value | H | P | Q | M | I | K | C | B | L | E | R | S | True |
| 1 | 2 | [2] > [1] Swap [0] and [2] | Q | P | H | M | I | K | C | B | L | E | R | S | True |
| 5 | 6 | [5] > [6] Swap [1] and [5] | Q | P | K | M | I | H | C | B | L | E | R | S | True |
| 11 | 12 | There is no 12 th index in our array. | Q | P | K | M | I | H | C | B | L | E | R | S | False |
| - | - | Swap the first value and last value | E | P | K | M | I | H | C | B | L | Q | R | S | True |
| 1 | 2 | [1] > [2] Swap [0] and [1] | P | E | K | M | I | H | C | B | L | Q | R | S | True |
| 3 | 4 | [1] > [2] Swap [1] and [3] | P | M | K | E | I | H | C | B | L | Q | R | S | True |
| 7 | 8 | [8] > [7] Swap [3] and [8] | P | M | K | L | I | H | C | B | E | Q | R | S | True |

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 17 | 18 | There is no 17 th and 18 and index in our array | P | M | K | L | I | H | C | B | E | Q | R | S | False |
| - | - | Swap the first value and last value | E | M | K | L | I | H | C | B | P | Q | R | S | True |
| 1 | 2 | [1] > [2] Swap [0] and [1] | M | E | K | L | I | H | C | B | P | Q | R | S | True |
| 3 | 4 | [3] > [4] Swap [0] and [3] | M | L | K | E | I | H | C | B | P | Q | R | S | True |
| 7 | 8 | There is no 8 th index in our array | M | L | K | E | I | H | C | B | P | Q | R | S | False |
| - | - | Swap the first value and last value | B | L | K | E | I | H | C | M | P | Q | R | S | True |
| 1 | 2 | [1] > [2] Swap [0] and [1] | L | B | K | E | I | H | C | M | P | Q | R | S | True |
| 3 | 4 | [4] > [3] Swap [1] and [4] | L | I | K | E | B | H | C | M | P | Q | R | S | True |

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 9 | 10 | There is no 9 th and 10 and index in our array | L | I | K | E | B | H | C | M | P | Q | R | S | False |
| - | - | Swap the first value and last value | C | I | K | E | B | H | L | M | P | Q | R | S | True |
| 1 | 2 | [2] > [1] Swap [0] and [2] | K | I | C | E | B | H | L | M | P | Q | R | S | True |
| 5 | 6 | There is no 6 th index in our array | K | I | C | E | B | H | L | M | P | Q | R | S | False |
| - | - | Swap the first value and last value | H | I | C | E | B | K | L | M | P | Q | R | S | True |
| 1 | 2 | [1] > [2] Swap [0] and [1] | I | H | C | E | B | K | L | M | P | Q | R | S | True |
| 3 | 4 | [3] > [4] But [1] > [3] No swap | I | H | C | E | B | K | L | M | P | Q | R | S | False |
| - | - | Swap the first value and last value | B | H | C | E | I | K | L | M | P | Q | R | S | True |
| 1 | 2 | [1] > [2] Swap [0] and [1] | H | B | C | E | I | K | L | M | P | Q | R | S | True |

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 3 | 4 | There is no 4 th index in our array | H | B | C | E | I | K | L | M | P | Q | R | S | False |
| - | - | Swap the first value and last value | E | B | C | H | I | K | L | M | P | Q | R | S | True |
| 1 | 2 | [2] > [1] But [0] > [2] No swap | E | B | C | H | I | K | L | M | P | Q | R | S | False |
| - | - | Swap the first value and last value | C | B | E | H | I | K | L | M | P | Q | R | S | True |
| 1 | 2 | There is no 2 th index in our array | C | B | E | H | I | K | L | M | P | Q | R | S | False |
| - | - | Swap the first value and last value | B | C | E | H | I | K | L | M | P | Q | R | S | True |

*In shrink process, 35 comparison and 25 displacement done*

*Sorted array:*

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

***In total 60 comparison and 32 displacement done.***

- ***Applying quicksort:***

***D = {S,B,I,M,H,Q,C,L,R,E,P,K}***

| [0] | [1] | [2] | [3] | [4] | [5] | [6] | [7] | [8] | [9] | [10] | [11] |
|---|---|---|---|---|---|---|---|---|---|---|---|
| S | B | I | M | H | Q | C | L | R | E | P | K |

```
protected <T extends Comparable<T>> void quickSort(T[] table, int first, int last) {
        if (first < last) {
                int pivIndex = partition(table, first, last);
                quickSort(table, first, pivIndex - 1);
                quickSort(table, pivIndex + 1, last);
        }
}

protected <T extends Comparable<T>> int partition(T[] table, iint first, int last) {
        T pivot = table[first];
        int up = first; int down = last;
        do {
        while ((up < last) && (pivot.compareTo(table[up]) >= 0)) {
                up++;
        }
        while (pivot.compareTo(table[down]) < 0) {
                down--;
        }
        if (up < down) {
                swap(table, up, down);
        }
        } while (up < down); // Repeat while up is left of down.
        swap(table, first, down);
        return down;
}
```

→ Sort(table,0,11)

Pivot value = c[0] = s

| Compare Pivot-current | Compare Result | Up | down | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | isChanged |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 0 | 11 | S | B | I | M | H | Q | C | L | R | E | P | K | - |
| S – [0] | = | 1 | 11 | S | B | I | M | H | Q | C | L | R | E | P | K | False |
| S – [1] | > | 2 | 11 | S | B | I | M | H | Q | C | L | R | E | P | K | False |
| S – [2] | > | 3 | 11 | S | B | I | M | H | Q | C | L | R | E | P | K | False |
| S – [3] | > | 4 | 11 | S | B | I | M | H | Q | C | L | R | E | P | K | False |
| S – [4] | > | 5 | 11 | S | B | I | M | H | Q | C | L | R | E | P | K | False |
| S – [5] | > | 6 | 11 | S | B | I | M | H | Q | C | L | R | E | P | K | False |
| S – [6] | > | 7 | 11 | S | B | I | M | H | Q | C | L | R | E | P | K | False |
| S – [7] | > | 8 | 11 | S | B | I | M | H | Q | C | L | R | E | P | K | False |
| S – [8] | > | 9 | 11 | S | B | I | M | H | Q | C | L | R | E | P | K | False |
| S – [9] | > | 10 | 11 | S | B | I | M | H | Q | C | L | R | E | P | K | False |
| S – [10] | > | 11 | 11 | S | B | I | M | H | Q | C | L | R | E | P | K | False |
| S – [11] | > | 11 | 11 | S | B | I | M | H | Q | C | L | R | E | P | K | False |
| | Swap(f,d) | 11 | 11 | K | B | I | M | H | Q | C | L | R | E | P | S | True |

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

*13 comparison and 1 displacement done.*

→ *Sort(table,0,10)*

*Pivot value = c[0] = K*

| Compare Pivot-current | Compare Result | Up | down | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | isChanged |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 0 | 10 | K | B | I | M | H | Q | C | L | R | E | P | S | - |
| K – [0] | = | 1 | 10 | K | B | I | M | H | Q | C | L | R | E | P | S | False |
| K – [1] | > | 2 | 10 | K | B | I | M | H | Q | C | L | R | E | P | S | False |
| K – [2] | > | 3 | 10 | K | B | I | M | H | Q | C | L | R | E | P | S | False |
| K – [3] | < | 3 | 10 | K | B | I | M | H | Q | C | L | R | E | P | S | False |
| K – [10] | < | 3 | 9 | K | B | I | M | H | Q | C | L | R | E | P | S | False |
| K – [9] | > | 3 | 9 | K | B | I | M | H | Q | C | L | R | E | P | S | False |
| Up < down | Swap(u,d) | 3 | 9 | K | B | I | E | H | Q | C | L | R | M | P | S | True |
| K – [3] | > | 4 | 9 | K | B | I | E | H | Q | C | L | R | M | P | S | False |
| K – [4] | > | 5 | 9 | K | B | I | E | H | Q | C | L | R | M | P | S | False |
| K – [5] | < | 5 | 9 | K | B | I | E | H | Q | C | L | R | M | P | S | False |
| K – [9] | < | 5 | 8 | K | B | I | E | H | Q | C | L | R | M | P | S | False |

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| K – [8] | < | 5 | 7 | K | B | I | E | H | Q | C | L | R | M | P | S | False |
| K – [7] | < | 5 | 6 | K | B | I | E | H | Q | C | L | R | M | P | S | False |
| K – [6] | > | 5 | 6 | K | B | I | E | H | Q | C | L | R | M | P | S | False |
| Up < down | Swap(u,d) | 5 | 6 | K | B | I | E | H | C | Q | L | R | M | P | S | True |
| K – [5] | > | 6 | 6 | K | B | I | E | H | C | Q | L | R | M | P | S | False |
| K – [6] | < | 6 | 5 | K | B | I | E | H | C | Q | L | R | M | P | S | False |
| K – [5] | > | 6 | 5 | K | B | I | E | H | C | Q | L | R | M | P | S | False |
| | Swap(f,d) | 6 | 5 | C | B | I | E | H | K | Q | L | R | M | P | S | True |

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

*19 comparison and 3 displacement done.*

➔ *Sort(table,0,4)*

*Pivot value = c[0] = C*

| Compare Pivot-current | Compare Result | Up | down | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | isChanged |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 0 | 4 | C | B | I | E | H | K | Q | L | R | M | P | S | - |
| C – [0] | = | 1 | 4 | C | B | I | E | H | K | Q | L | R | M | P | S | False |
| C – [1] | > | 2 | 4 | C | B | I | E | H | K | Q | L | R | M | P | S | False |
| C – [2] | < | 2 | 4 | C | B | I | E | H | K | Q | L | R | M | P | S | False |

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| C – [4] | < | 2 | 3 | C | B | I | E | H | K | Q | L | R | M | P | S | False |
| C – [3] | < | 2 | 2 | C | B | I | E | H | K | Q | L | R | M | P | S | False |
| C – [2] | < | 2 | 1 | C | B | I | E | H | K | Q | L | R | M | P | S | False |
| C – [1] | > | 2 | 1 | C | B | I | E | H | K | Q | L | R | M | P | S | False |
| | Swap(f,d) | 2 | 1 | B | C | I | E | H | K | Q | L | R | M | P | S | True |

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

8 comparison and 1 displacement done.

➔ Sort(table,2,4)

Pivot value = c[2] = I

| Compare Pivot-current | Compare Result | Up | down | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | isChanged |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 2 | 4 | B | C | I | E | H | K | Q | L | R | M | P | S | - |
| I – [2] | = | 3 | 4 | B | C | I | E | H | K | Q | L | R | M | P | S | False |
| I– [3] | > | 4 | 4 | B | C | I | E | H | K | Q | L | R | M | P | S | False |
| I – [4] | > | 4 | 4 | B | C | I | E | H | K | Q | L | R | M | P | S | False |
| | Swap(f,d) | 4 | 4 | B | C | H | E | I | K | Q | L | R | M | P | S | True |

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

4 comparison and 1 displacement done.

➔ *Sort(table,2,3)*

*Pivot value = c[2] = H*

| Compare Pivot-current | Compare Result | Up | down | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | isChanged |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 2 | 3 | B | C | H | E | I | K | Q | L | R | M | P | S | - |
| H – [2] | = | 3 | 3 | B | C | H | E | I | K | Q | L | R | M | P | S | False |
| H – [3] | > | 3 | 3 | B | C | H | E | I | K | Q | L | R | M | P | S | False |
| | Swap(f,d) | 3 | 3 | B | C | E | H | I | K | Q | L | R | M | P | S | True |

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

*3 comparison and 1 displacement done.*

➔ *Sort(table,6,10)*

*Pivot value = c[6] = Q*

| Compare Pivot-current | Compare Result | Up | down | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | isChanged |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 6 | 10 | B | C | H | E | I | K | Q | L | R | M | P | S | - |
| Q – [6] | = | 7 | 10 | B | C | H | E | I | K | Q | L | R | M | P | S | False |
| Q – [7] | > | 8 | 10 | B | C | H | E | I | K | Q | L | R | M | P | S | False |
| Q – [8] | < | 8 | 10 | B | C | H | E | I | K | Q | L | R | M | P | S | False |
| Q – [10] | > | 8 | 10 | B | C | H | E | I | K | Q | L | R | M | P | S | False |

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Swap(u,d) | 8 | 10 | B | C | H | E | I | K | Q | L | P | M | R | S | True |
| Q – [8] | > | 9 | 10 | B | C | H | E | I | K | Q | L | P | M | R | S | False |
| Q – [9] | > | 10 | 10 | B | C | H | E | I | K | Q | L | P | M | R | S | False |
| Q – [10] | < | 10 | 9 | B | C | H | E | I | K | Q | L | P | M | R | S | False |
| Q – [9] | > | 10 | 9 | B | C | H | E | I | K | Q | L | P | M | R | S | False |
| | Swap(f,d) | 10 | 9 | B | C | H | E | I | K | M | L | P | Q | R | S | True |

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

10 comparison and 2 displacement.

➔ Sort(table,6,8)
Pivot value = c[6] = M

| Compare Pivot-current | Compare Result | Up | down | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | isChanged |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 6 | 8 | B | C | H | E | I | K | M | L | P | Q | R | S | - |
| M –[6] | = | 7 | 8 | B | C | H | E | I | K | M | L | P | Q | R | S | False |
| M –[7] | > | 8 | 8 | B | C | H | E | I | K | M | L | P | Q | R | S | False |
| M –[8] | < | 8 | 7 | B | C | H | E | I | K | M | L | P | Q | R | S | False |
| M –[7] | > | 8 | 7 | B | C | H | E | I | K | M | L | P | Q | R | S | False |

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | *Swap(f,d)* | *8* | *7* | *B* | *C* | *H* | *E* | *I* | *K* | *L* | *M* | *P* | *Q* | *R* | *S* | *True* |

***5 comparison and 1 displacement.***

***Sorted array:***

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| *B* | *C* | *H* | *E* | *I* | *K* | *L* | *M* | *P* | *Q* | *R* | *S* |

## *In total 62 comparison and 10 displacement done.*