## Answers of Tutorial 8

1. **Let A be the following array.**

| | | | | |
|---|---|---|---|---|
| 16 | 23 | 31 | 20 | 4 |

**(i) Show the array A after calling heapify on it to produce a max-heap.**

| | | | | |
|---|---|---|---|---|
| 31 | 23 | 16 | 20 | 4 |

**(ii) Starting from the array in (i), trace the steps of the heapsort algorithm on A.**

**Swap A[1] to end:**

| | | | | |
|---|---|---|---|---|
| 4 | 23 | 16 | 20 | 31 |

**Siftdown:**

| | | | | |
|---|---|---|---|---|
| 23 | 4 | 16 | 20 | 31 |

| | | | | |
|---|---|---|---|---|
| 23 | 20 | 16 | 4 | 31 |

**Swap A[1] to end:**

| | | | | |
|---|---|---|---|---|
| 4 | 20 | 16 | 23 | 31 |

**Siftdown:**

| | | | | |
|---|---|---|---|---|
| 20 | 4 | 16 | 23 | 31 |

**Swap A[1] to end:**

| | | | | |
|---|---|---|---|---|
| 16 | 4 | 20 | 23 | 31 |

**Swap A[1] to end:**

| | | | | |
|---|---|---|---|---|
| 4 | 16 | 20 | 23 | 31 |

2. **Trace the execution of the partition algorithm to show how it partitions the array: 'N', 'A', 'N', 'Y', 'A', 'N', 'G', 'U', 'N', 'I' in the alphabetical order.**

**Answer:**

```
partition(a,i,j) {
    val = a[i]
    h = i
    for k = i + 1 to j   // Scan for a smaller value
        if (a[k] < val) {        // YES
            h = h + 1
            swap(a[h],a[k])
        }
    swap (a[i],a[h])
    return h
}
```

The trace is as follows:

Scan for a smaller value

is a[k] = A smaller than val = N ?
YES A is smaller

increase h =1
Swap a[h] = A with a[k] = A
Thus, we have:
N A N Y A N G U N I

Scan for a smaller value

is a[k] = N smaller than val = N ?

Scan for a smaller value

is a[k] = Y smaller than val = N ?

Scan for a smaller value

is a[k] = A smaller than val = N ?
YES A is smaller

increase h =2
Swap a[h] = N with a[k] = A
Thus, we have:
N A A Y N N G U N I

Scan for a smaller value

is a[k] = N smaller than val = N ?

Scan for a smaller value

is a[k] = G smaller than val = N ?
YES G is smaller

increase h =3
Swap a[h] = Y with a[k] = G
Thus, we have:
N A A G N N Y U N I

Scan for a smaller value

is a[k] = U smaller than val = N ?

Scan for a smaller value

is a[k] = N smaller than val = N ?

Scan for a smaller value

is a[k] = I smaller than val = N ?
YES I is smaller

increase h =4
Swap a[h] = N with a[k] = I
Thus, we have:
N A A G I N Y U N N

Swap a[h] = I with pivot a[0] = N

Partition lement N is rank 5; and finally:

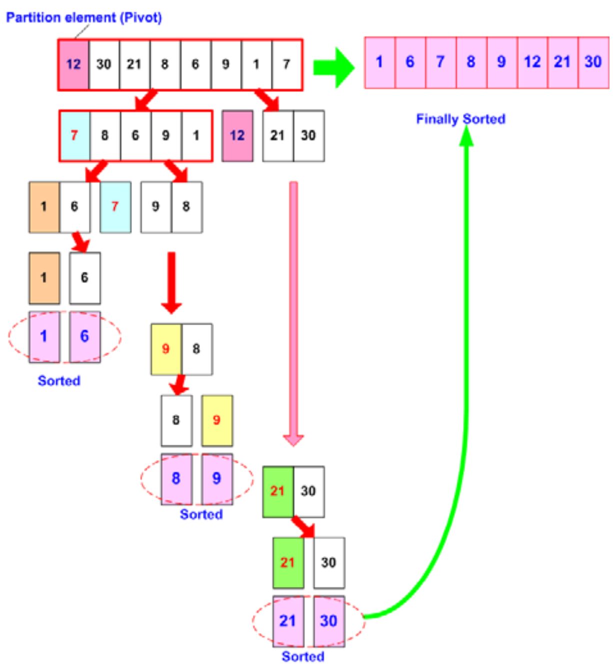
I A A G **N** N Y U N N

3. **Show the working of Quick Sort algorithms that sorts the array:**

**12, 30, 21, 8, 6, 9, 1, 7**

**Answer:**

The visualization of an execution of Quick-sort is shown below.



4. **Consider performing counting sort on the following array:**

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 5 | 7 | 5 | 1 | 3 | 7 | 6 | 3 | 1 | 6 | 6 | 5 | 3 |

**i. What does the final count array look like?**

```
for j = 1 to n {
        count[A[j]] = count[A[j]] + 1
    } // count[i] now contains the no. of elements = i
```

| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|
| 2 | 0 | 3 | 0 | 3 | 3 | 2 |

```
for i = 1 to k {
        count[i] = count[i-1] + count[i]
} // count[i] now contains no. of elements ≤ i
```

| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|
| 2 | 2 | 5 | 5 | 8 | 11 | 13 |

**ii. Use the count array to determine the sorted array.**

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 3 | 3 | 3 | 5 | 5 | 5 | 6 | 6 | 6 | 7 | 7 |