SHUBHAM GUPTA
180742

theo - ass 1

8) pseudocode

```
q8 (A, n) {
    t[n][log2 n + 1]
    for (i = 0, n-1) {
        t[i][0] ← i
    }
    for (i = 1, log2 n) {
        for (j = 0, n - 2^i) {
            a ← t[j][i-1]
            b ← t[j + 2^(i-1)][j-1];
            if (A[a] > A[b]) {
                t[j][i] ← a
            } else if (A[a] ≤ A[b]) {
                t[j][i] ← b;
            }
        }
    }
    return t;
}
q8-l (A, x) {
    l ← 0, r ← n-1
    while (l < r) {
        mid = (l + r) / 2
        if (A[mid] >= x) {
            r ← mid;
        } else {
            l ← mid + 1;
        }
    }
    return r;
}
```

```
q.s-u(A, n) {
    l ← 0      r ← n-1
    while (l ≤ r) {
        mid = (l+r)/2
        if (A[mid] > x) {
            r ← mid;
        } else {
            l ← mid + 1
        }
    }
    return r;
}
```

algorithm :- we will first sort A and then store unique values in B and frequency of each unique value in C (index)
we define 2 funcns for upper & lower bound for array, using them we can find the no. of elements in A b/w (a, b].
now we create a 2D matrix, matrix element (i)(j) store index of maximum occuring element from our array C
now find indices such that B(i) ≥ a & B(j) ≤ b
answer can be found in table.

time complexity :→ on the → O(n log n)