

1	2	3	4	5	6	7	8	9
10	15	30	40	50	60			

Quick sort :-

Quick sort (A, p, r)

if (p < r)

{

q = partition (A, p, r)

Quick sort (A, p, q-1)

Quick sort (A, q+1, r)

}

}

Partition (A, p, r)

```
x = A[r]
r = p - 1
for j = p to r - 1 do
    if A[j] ≤ x then
        i = i + 1
        A[i] <→ A[j]
        A[i + 1] <→ A[r]
    return i + 1
}
```

~~pivot~~
~~x = A[r]~~
~~i = p - 1~~

```
for (j = p; j ≤ r - 1; j++)
```

```
{
    if (arr[j] < pivot)
```

```
    i++;
```

```
    swap arr[i] and arr[j]
}
```

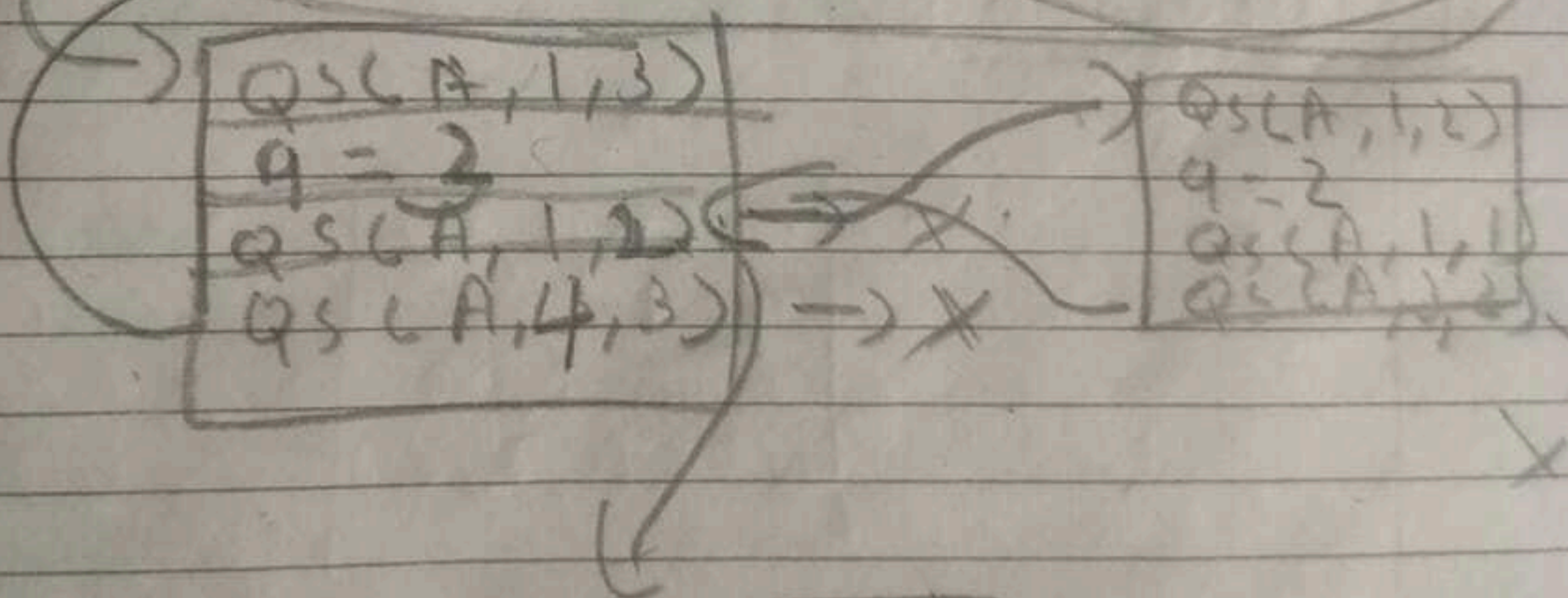
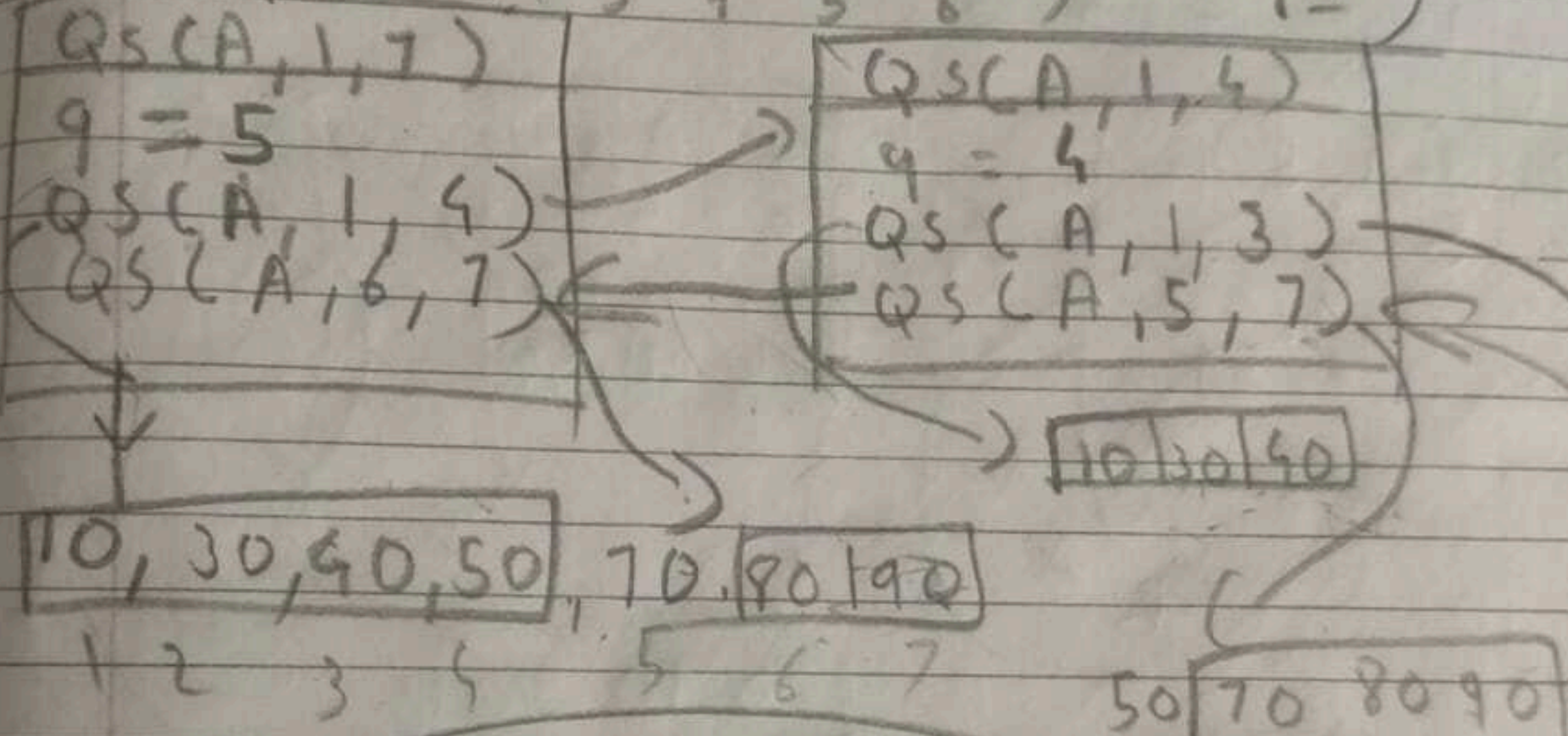
```
}
```

```
swap arr[i + 1] and arr[r]
return i + 1
```

```
}
```

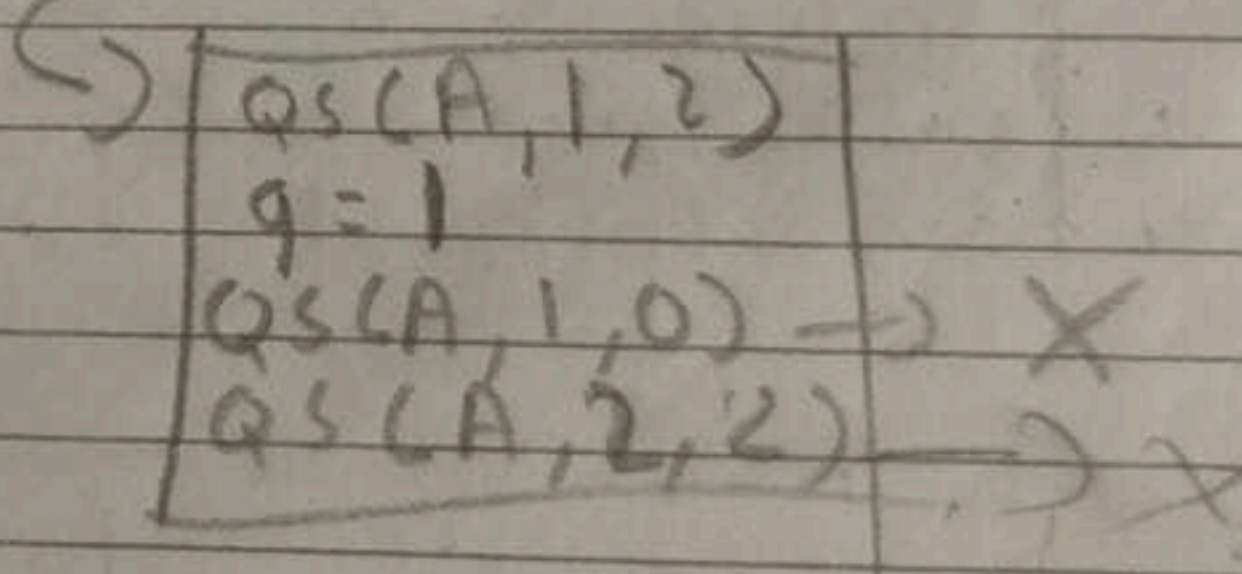
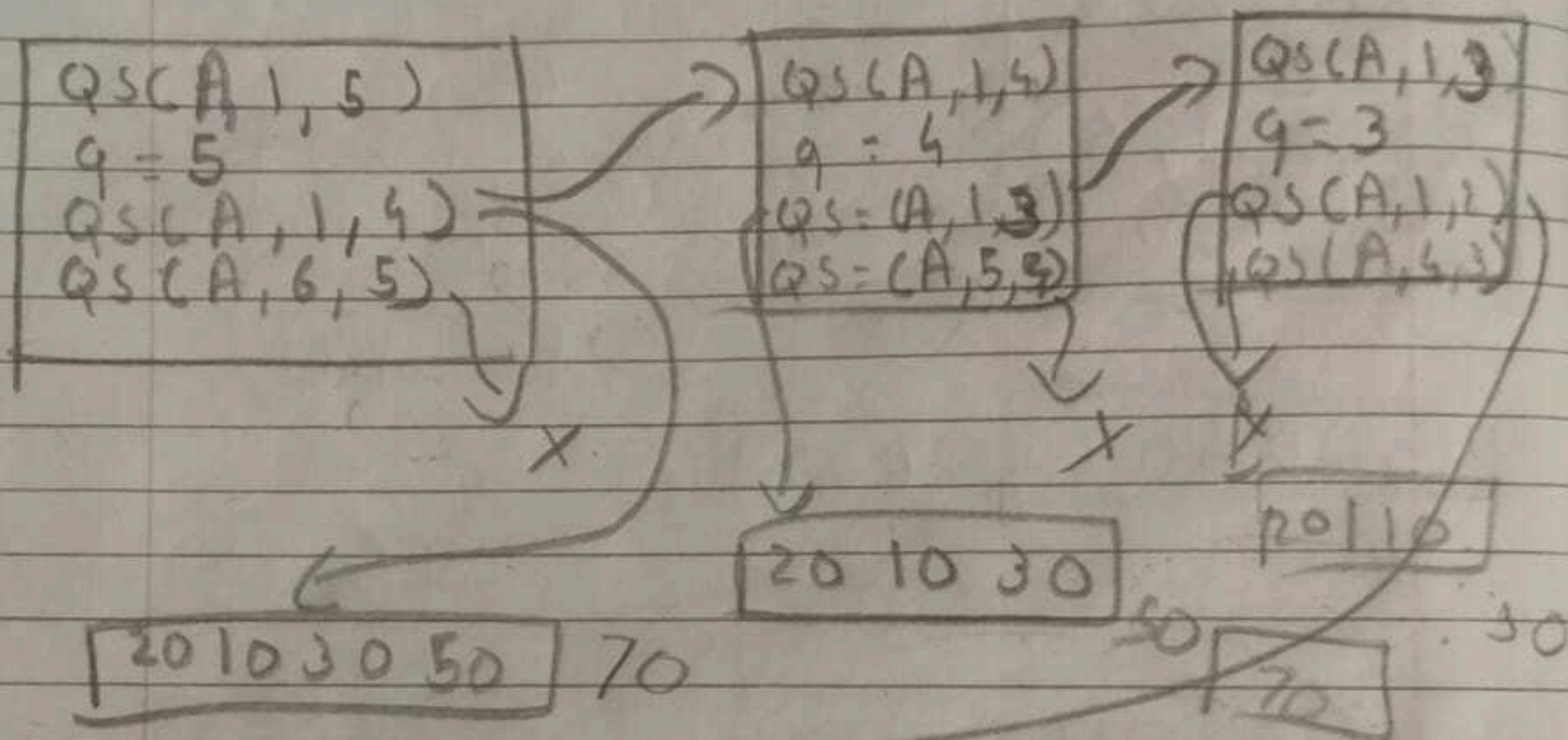

IV. N
 III. N
 II. N
 I. N
 Conflict and view serializable
 Not conflict but view serializable
 Conflict but not view serializable
 Page 1 of 3

example: 10, 30, 30, 40, 40, 50, 70
 P: 1 2 3 4 5 6 7



10, 30

20, 10, 30, 50, 70
1 2 3 4 5



Note: If data is ascending, descending and pivot element is first one or last one then time complexity $O(n)^2$ worst case.