Describe an algorithm that, given n integers in the range 0 to k, preprocesses its input and then answers any query about how many of the n integers fall into the range [a … b] in O(1) time. Your algorithm should use Θ(n+k) preprocessing time.

Explanation:

The algorithm I would use to accomplish this would be the counting sort algorithm. Assuming an array containing the integers, the counting sort algorithm works by initialising a second array of length k (since range of the integers is 0 to k). Each element of the second array will be initialised to 0.

Looping through every element of the input array (containing all the integers), the second array will be incremented at index array2[inputArray[i]] by 1 (i.e. array2[inputArray[i]]++). This will ensure that the second array will contain the number of elements of the input array having the value inputArray[i] (i.e. the count of each of the unique values in the input array).

Next, a third array will be initialised. The second array will be looped and every element plus the previous element will be inserted into the 3rd array at the given index (i.e. array3[i] = array2[i] + array2[i − 1]).

Finally, to answer any query about how many of the n integers fall into the range [a ... b], you would simply calculate array3[b] − array3[a] + array2[a] for the final result.

Pseuodocode:

```
def getRangeCount(inputArray, a, b, k):
  array2 = [0]*k
  array3 = [0]*k

  for num in inputArray:
    array2[num]++

  for i to len(array2):
    array3[i] = array2[i] + array2[i - 1]

  return array3[b] − array3[a] + array2[a]
```

Example:

input: [1, 4, 1, 2, 7, 5, 2]

[a ... b]: [4 ... 7]

index: 0 1 2 3 4 5 6 7 8 9

array2 [0, 2, 2, 0, 1, 1, 0, 1, 0, 0]

array3: [0, 2, 4, 4, 5, 6, 6, 7, 7, 7]

final result = array3[b] − array3[a] + array2[a] = 7 - 5 + 1 = 3