Problem 7.1)

Ans c.)

```
count_numbers (A, a, b, k):   where A is the array in which
    C = new int [k+1]          the numbers are stored.
    For i = 0 to k:
        C [i] = 0
    for i = 1 to len(A):
        C[A[i]] = C[A[i]] + 1
    For i = 2 to len(C):
        C[i] = C[i] + C[i-1]
    if (a < 0 or b < 0 or a > k or b > k or a > b):
        Print ("invalid range")
        exit(1)
    N = C[b] - C[a-1]
    return N
```

Ans 7e.)

The expected time complexity of bucket sort which uses insertion sort to sort the buckets is $\theta(n)$. To get this result, it is assumed that the distribution of the numbers across the buckets is uniform. However, if all the numbers of the array ends up falling in a single bucket, then the insertion sort, which has the time complexity of n^2, will be executed for the whole array, leading the complexity of bucket sort to be $\theta(n^2)$. An example of such a sequence would be.

0.49, 0.48, 0.47, 0.46, 0.45, 0.44, 0.43, 0.42, 0.41, 0.40.

In the above sequence all the elements will fall into a single bucket and they will be sorted at once, leading to the time complexity being $\theta(n^2)$

Ans 7.2.5.)

Deriving the time complexity:

Now, we know that the number of arrays in which the Hollerith's sort is called, is the maximum when all the elements have the same order of magnitude. (i.e. if $a \in Array$. $\lfloor \log_{10}(a) \rfloor = C$, $C \in \mathbb{N}$) and when the most significant digits of all numbers are distinct. During such a scenario, the number of recursive calls is equal to the number of elements in the array.

However, the number of recursive calls is also dependent on the number of digits of each number. Since, the number of digits of a number (when it is expressed in decimal system is $\log_{10}(n)$ where, (n is any number in the array, since it has been assumed that all numbers have the same order of magnitude).

Therefore: $T(n, K) = O(n \cdot K)$ [since we are dealing with worse-case scenario]

$= O(n \log_{10}(n))$ [since K is the number of digits of the numbers in the array)