## Problem 8.1:

c) idea: once we have created an array C holding, at each index, all elements from A smaller or equal than index, to get the number of elements in [a,b] we calculate difference of value of helper array in index b and a, then we add 2 since [a,b] is closed.

* Pseudocode:

```
ELEMENTS-IN-Interval (A, n).
    for i := 1 to k
        C[i] = 0
    for j := 1 to n
        C[A[j]] = C[A[j]] + 1
    for i := 2 to k.
        C[i] = C[i] + C[i-1]
    number_of_elements = (C[b] - C[a]) + 2.
```

e) Worst case scenario of Bucket sort occure if elements are not uniformally distributed and all elements fall in one single bucket. So we will have to only sort that bucket with n elements using a stable sorting alg.

Hence, worst case scenario will be worst case scenario of the sorting (helper) alg.

If we choose insertion sort. time Complexity will be $\Theta(n^2)$

so $T(n) = \Theta(n) + \Theta(n^2) = \Theta(n^2)$.

a) + b) + d) See implementation.