Explain why sorting all the buckets in bucket sort using an O(n2) algorithm like insertion sort results in an O(n) sorting algorithm.

If nb is number of elements per bucket, and n is the number of buckets, then the runtime for any individual bucket utilizing insertion sort would be:

O(nb^2)

The probability of any one element falling into a bucket is 1/n. Extrapolating an expected value from the statistical concept of a binomial distribution we get 2-1/n for nb^2.

The variance is 1-1/n. The mean of samples is 1.

expected value = variance + (mean)^2

so nb^2 = 2-1/n

So now out runtime becomes:

n x O(2n-1)

which the largest factor in there is n, so it simplifies to O(n) regardless of using an sorting algorithm of O(n^2) runtime.

In parlance that I did not have to thoroughly pore over to understand, the buckets are relatively small and the values are evenly distributed (as best as we can get with rand()). Any call to insertion sort does not have to take that many steps to complete its task.