Average Case Analysis of Insertionsort

We want to analyze the average case number of comparisons performed by Insertionsort under the assumption that every one of the $n!$ different permutations of the n items $\{1, 2, \ldots, n\}$ are equally likely as input.

In what follows we let $\pi = \langle a_1, a_2, \ldots, a_n\rangle$ denote a permutation of $\{1, 2, \ldots, n\}$

Definition: Let (p, q) be such that $1 \le p < q \le n$. We say that (p, q) is an *inversion* of π if q appears before p in π.

Example: The inversions of $\pi = \langle 3, 4, 1, 2\rangle$ are $(1, 3)$, $(1, 4)$, $(2, 3)$,, $(2, 4)$.

Definition: For $p < q$ set

$$\begin{aligned} Z_{p,q} &= \begin{cases} 1 & \text{if } (p,q) \text{ is an inversion in } \pi \\ 0 & \text{if } (p,q) \text{ is not an inversion in } \pi \end{cases}. \\ I_p &= \sum_{q=p+1}^{n} Z_{p,q} \\ &= \text{the total number of inversions of the form } (p,q) \end{aligned}$$

Now suppose that $p = a_j$ in the original permutation π..
How many comparisons are performed by Insertionsort when p is compared to the items to its left? We will now see that the answer is $I_p + e_p$ where $e_p \in \{0, 1\}$

If $j = 1$ then no comparisons are performed since we're at the left of the array. On the other hand, since we're at the left of the array $p = a_1$ is not involved in *any* inversions, so $I_p = 0$. Setting $e_p = 0$ gives the result.

If $j > 1$ note that when it's time to process p the items to p's left are the items $a_a, a_2, \ldots, a_{j-1}$ from π, but now in sorted order.
So, the algorithm will compare p to all of the items $q \in \{a_a, a_2, \ldots, a_{j-1}\}$ such that $q > p$, each time shifting one item to the left. The algorithm stops either when it compares a_j to the largest $q \in \{a_a, a_2, \ldots, a_{j-1}\}$ such that $q < p$ or, if no such element exists, when it reaches the leftmost end of the array.
The important observation is that q has the property

$$q \in \{a_a, a_2, \ldots, a_{j-1}\} \text{ and } q > p$$

if and only if (p, q) is an inversion of π.
Thus, the number of comparisons performed by the algorithm when processing p is either (i) I_p or (ii) $1 + I_p$ We will write this as $I_p + e_p$ where $e_p \in \{0, 1\}$
Summing over all of the a_j (which are a permutation of $1, 2, \ldots, n$) we see that the total amount of work perfomed by the algorithm is exactly

$$\begin{aligned}\sum_{j=1}^{n} \left(I_{a_j} + e_{a_j}\right) &= \sum_{p=1}^{n} I_p + \sum_{p=1}^{n} e_p \\ &= \sum_{p,q\colon 1 \le p < q \le n} Z_{p,q} + \sum_{p=1}^{n} e_p\end{aligned}$$

The final thing to notice is that for any *fixed* p, q it is equally likely that in a random permutation p will appear before q and that p will appear after it. Thus

$$\forall p, q,\ \Pr(Z_{p,q} = 1) = \Pr(Z_{p,q} = 0) = \frac{1}{2}$$

and

$$E(Z_{p,q}) = 1 \cdot \Pr(Z_{p,q} = 1) + 0 \cdot \Pr(Z_{p,q} = 0) = \frac{1}{2}$$

$E()$ is the expectation operator. To finish we now recall the *Linearity of the expectation operator, i.e., that* $E(X+Y) = E(X) + E(Y)$ (see the appendix of CLRS for a review of this fact) to find that the expected amount of work done by Insertionsort is

$$\begin{aligned} E\left(\sum_{p,q:\,1\le p<q\le n} Z_{p,q} + \sum_{p=1}^{n} e_p\right) &= \sum_{p,q:\,1\le p<q\le n} E(Z_{p,q}) + \sum_{p=1}^{n} E(e_p) \\ &= \frac{n(n-1)}{2}\frac{1}{2} + \sum_{p=1}^{n} E(e_p) \\ &= \frac{n(n-1)}{4} + \sum_{p=1}^{n} E(e_p) \end{aligned}$$

Recalling that $e_p \in \{0,1\}$ we have that $\sum_{p=1}^{n} E(e_p) \le n$ so the average case of insertionsort runs in approximately $n^2/4$ time, half the worst case $n^2/2$ time needed.

Note: In order to simplify the analysis we did not analyze the value of $\sum_{p=1}^{n} E(e_p)$ *exactly. As an extra credit exercise, try doing this.*