Analysis of the average time complexity of the insertion sort algorithm using the Kolmogorov complexity method.

Preliminary assumptions: let an n-element array be given. Example (n = 5):

5 1 3 4 2

Let's think about how this sorting method works. We start with the element with index 1. Move left with this element, until it is in the right place (if the predecessor is smaller than it - stop). Thanks to this, gradually from the left we get a sorted array at the end.

Let T - be all transpositions performed by the insertion sort algorithm. Note that in the analyzed algorithm, one transposition changes the position of only one element - in another sort algorithm (bubble sort) it is different.

For $i \in [0, n)$ let's define $dist_i$, this will be distance of a given element to the correct position in the array (for the example above, dist [1] = 1 etc.). In turn, let M be the sum of these distances. For the considered algorithm we have:

$$|T| \geqslant M = \sum_{i=1}^{n} dist_i$$

To uniquely code ("heart" of Kolmogorov method) insertion sort algorithm is enough for us the code number of solutions M. Among the natural numbers we have $\binom{M+n-1}{n-1}$ solutions to the equation $M = \sum_{i=1}^{n} dist_i$. So, when we give each solution a sequential number, we need $\lceil \log_2 \binom{M+n-1}{n-1} \rceil$ bits to write a specific solution. The whole encoded permutation has the form:

$$num\ of\ sol.\ M = \sum_{i=1}^{n} dist_i$$

In the insertion sort algorithm from this code, we can recreate the initial permutation. So we need $\lceil \log_2 \binom{M+n-1}{n-1} \rceil$ bits to code the output permutation. By the "code length theorem" for at least $(1-c)n!$ permutations we have:

$$\lceil \log_2 \binom{M+n-1}{n-1} \rceil \geqslant \log_2(cn!)$$

We will now use the Stirling formula: $n! \approx \left(\frac{n}{e}\right)^n \sqrt{2\pi n}$:

$$\lceil \log_2 \binom{M+n-1}{n-1} \rceil \geqslant \log_2 c + \log_2((\frac{n}{e})^n \cdot \sqrt{2\pi n}) = \log_2 c + (n \log_2 n - n \log_2 e + \frac{1}{2} \log_2 n + \frac{1}{2} \log_2 2\pi)$$

$$\lceil \log_2 \binom{M+n-1}{n-1} \rceil \geqslant \log_2 c + n \log_2 n - n \log_2 e$$

There is also a (easy to prove) inequality: $(\frac{m \cdot e}{k})^k > \binom{m}{k}$, from which we have:

$$\log_2(\frac{(M+n-1)e}{n-1})^{n-1} = (n-1)\log_2(M+n-1) + (n-1)\log_2 e - (n-1)\log_2(n-2) \geqslant \log_2 \binom{M+n-1}{n-1}$$

$$n \log_2(M+n) + n \log_2 e - n \log_2(n-1) + \log_2(n-1) \geqslant \log_2 \binom{M+n-1}{n-1}$$

On both sides, let's join the inequalities:

$$n \log_2(M+n) + n \log_2 e - n \log_2(n-1) + \log_2(n-1) \geqslant \log_2 c + n \log_2 n - n \log_2 e$$

$$n \log_2(M+n) \geqslant \log_2 c + n \log_2(n(n-1)) - n \log_2 e^2$$

It's time to substitute something for c, in this method most often we substitute full lowercase values, sometimes depending on another letter. So let's substitute for example $c = \frac{1}{n}$. Then let's divide the inequality on both sides by positive n.

$$\log_2(M+n) \geqslant -\frac{\log_2 n}{n} + \log_2(n(n-1)) - \log_2 e^2$$

$$\log_2(M+n) \geqslant \log_2(n(n-1)) - \log_2 e^2 = \log_2 \frac{n^2-n}{e^2}$$

Therefore:

$$M \geqslant \frac{n^2-n}{e^2}$$

What the conclusion implies:

$$|T| \geqslant \frac{n^2-n}{e^2} = \frac{n^2}{e^2} - O(n)$$

Finally:

$$\frac{cn! \cdot 0 + (1-c)n! \cdot (\frac{n^2}{e^2} - O(n))}{n!} = (1-\frac{1}{n})(\frac{n^2}{e^2} - O(n)) = \frac{n(n-1)}{e^2} - O(n)$$

We got a fairly accurate "bottom-up" estimate of the average time complexity of the insertion sort algorithm.