Result Analysis of Assignment-1:

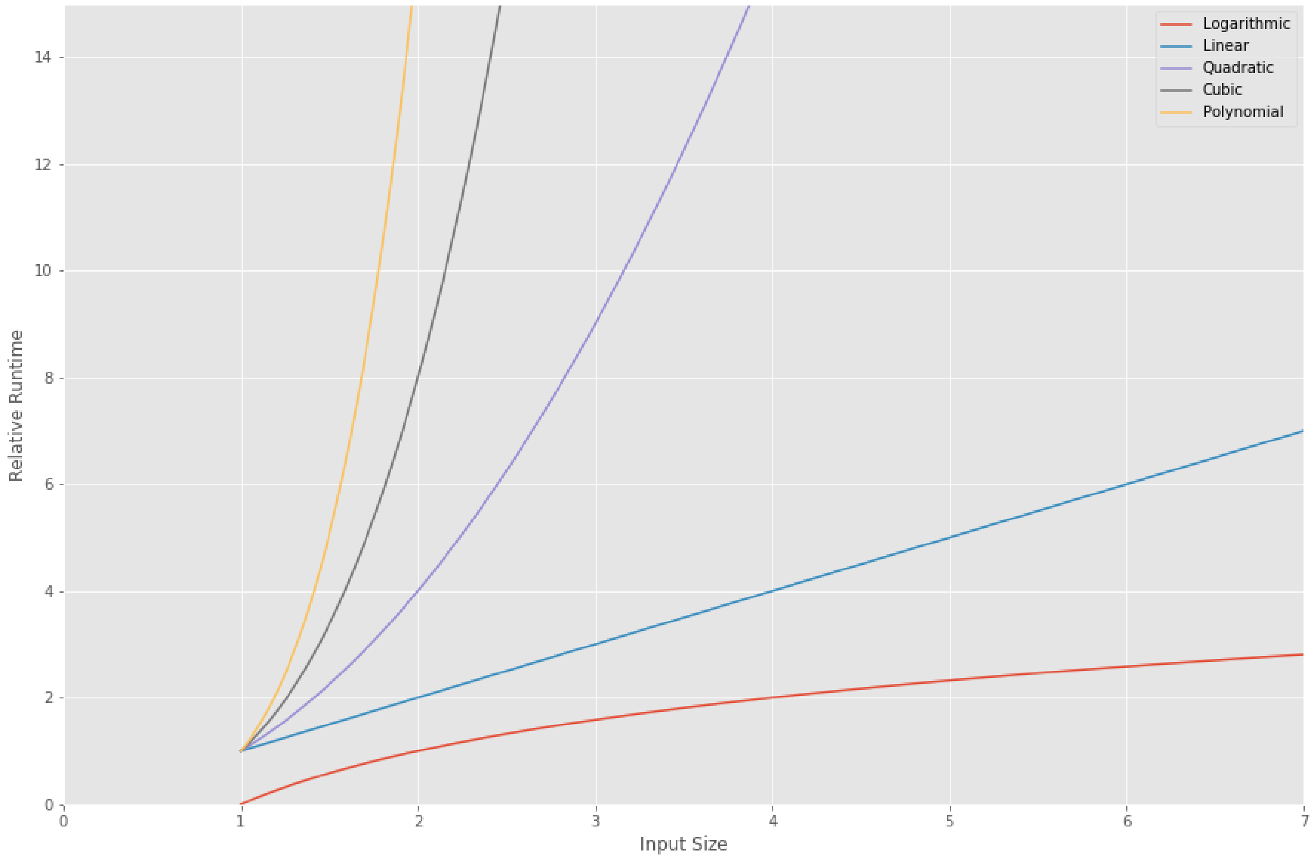

Figure-1

Big-O Notation is about finding an asymptotic upper bound. It also indicates towards the time and space complexity. Here, n is the input size and k is a positive constant. In case of complexity, we can say the complexity increases when the exponent increases.

In our task, we were given logarithmic, linear, Quadratic, Cubic and Polynomial equations to find out the best performing case in terms of time efficiency. The complexity and performance of running time depends on n that is the size of the input or the number of operations is required for each input item. From Fig-1, we can say the fastest possible running time for any algorithm is O(logn) in our given case. The time efficient sequence follows as:

$$\mathbf{O(logn)> O(n)> O(n^{2})> O(n^{3})> O(n^{k})}$$

The complexity of algorithms increase as the order increases and hence makes it time and computation inefficient. In terms of complexity, we can express the sequence as:

$$\mathbf{O(n^{k})> O(n^{3})> O(n^{2})> O(n)> O(logn)}$$