When it comes to developing different kinds of algorithms many factors and concepts are brought up. From that list there is one that stands out the most, asymptotic analysis. The concept deals with calculating the time an algorithm takes through the understanding of the input size it receives. The reason why this concept is huge when it comes to designing algorithms is due to the sole reason why algorithms are useful. They allow computers to make advanced calculations that proved results in a timely manner. It also allows for predictions of resource allocation. To do this, there are three different case analysis’ that can be done.

The first case being the Big-O Notation or the worse-case analysis. Generally, this is the most used method for predicting the runtime of an algorithm since it results in the providing the upper bound in the prediction model. This allows for programmers to understand the maximum time it would take to crunch through large sets of data. Along with providing the programmer with feedback to decide on whether the algorithm needs to be optimized to increase efficiency or not.

The second case being the Theta Notation or the average-case analysis. This is used to find the typical runtime for any set of data. You must know (or predict) the different cases to accurately average the case runtimes together. Due to that reason this case only used on occasion.

The third and last case is the Omega Notation or the best-case analysis. This case is barely ever used since it only provides the minimum time an algorithm will take. Thus, resulting in information that doesn’t provide enough case data to determine if the algorithm needs to be optimized to handle the extremes. Due to that must, if not all, use the Big-O Notation to determine efficiency and the need to optimize their algorithms.