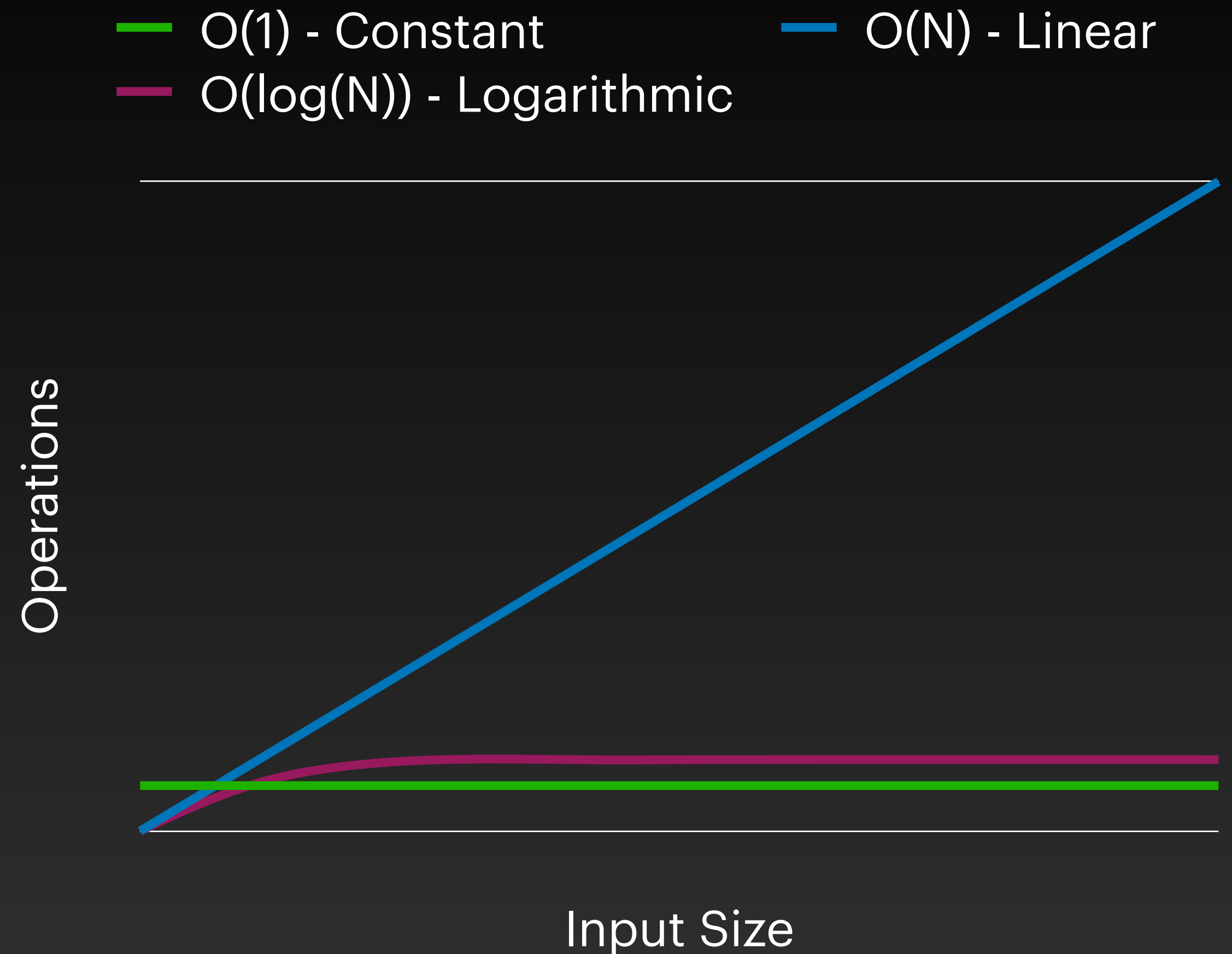


$O(\log(N))$

The big idea:

For each time the input grows the processing time required by the algorithm will increase by half what it previously increased by. N represents the size of the input.

$O(\log(N))$ is very efficient. By their nature these algorithms can solve complex problems very quickly.



$O(\log(N))$ - Logarithmic Complexity

Examples:

- Many kinds of tree data structures, such as a Binary Search Tree
- Looking up a number in a phone book by searching one half of it at a time.