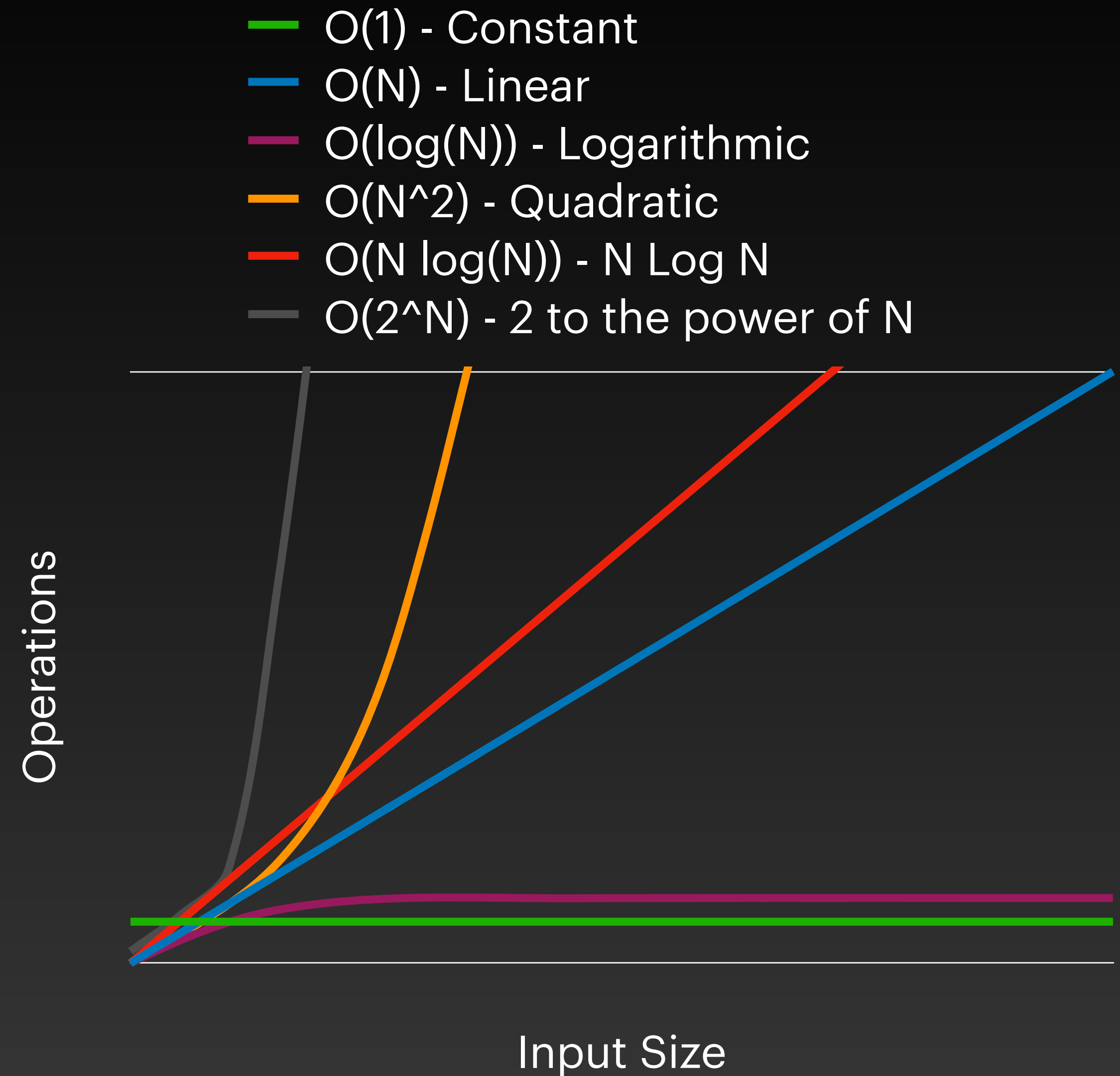# O(2^N)
The big idea:

For each time the input grows the processing time required by the algorithm will double.

Legend:
- O(1) - Constant
- O(N) - Linear
- O(log(N)) - Logarithmic
- O(N^2) - Quadratic
- O(N log(N)) - N Log N
- O(2^N) - 2 to the power of N

Operations (y-axis)

Input Size (x-axis)

O(2^N) - Power of 2 Complexity
Examples:

- A fibonacci sequence

- Calling a function recursively twice within a recursive function.