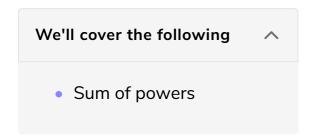
Non Trivial Runtime

In this lesson, we'll discuss an example with a non-trivial runtime.



Sum of powers

Take the code sample below:

As discussed earlier, the outer loop runs (log N + 1) times. The number of times the inner loop runs is equal to the first loop variable i.

- Iteration 1: Inner loop runs for 1 time
- Iteration 2: Inner loop runs for 2 times
- Iteration 3: Inner loop runs for 4 times
- •
- ullet Iteration (logN+1): Inner loop runs for $2^{(logN+1)}$ times

The number of operations forms a *Geometric Progression* which we will cover in the Number Theory chapter later on. We will use the sum formula directly here:

$$egin{aligned} 1+2+4+\ldots+2^{(logN+1)} \ &= rac{1*(2^{logN+2}-1)}{2-1} \ &= 2^{logN+2}-1 \ &= 4*2^{logN}-1 \ &= 4*N-1 \end{aligned}$$

So, the run-time complexity is actually $\operatorname{\it linear}$ - O(N).

In the next lesson, we'll learn about the amortized analysis.