

Non Trivial Runtime

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

We'll cover the following ^

- Sum of powers

Sum of powers

Take the code sample below:

```
for (int i = 1; i <= N; i *= 2)
    for (int j = 1; j <= i; j++)
        x++;
```

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
- ...
- Iteration $(\log N + 1)$: Inner loop runs for $2^{(\log N + 1)}$ 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:

$$1 + 2 + 4 + \dots + 2^{(\log N + 1)}$$

$$= \frac{1 * (2^{\log N + 2} - 1)}{2 - 1}$$

$$= 2^{\log N + 2} - 1$$

$$= 4 * 2^{\log N} - 1$$

$$= 4 * N - 1$$

So, the run-time complexity is actually ***linear*** - $O(N)$.

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