Lesson 2

"Think Big"

Jorge F., Leonidas G., Giordano A.

UTEC

Asymptotic analysis

Motivation

- In 2019, Twitter users send more than 500,000 tweets every minute.
- In 2019, 8.1 billion internet users.
- In 2018, 2.375 billion monthly active users in Facebook.

This numbers are increasing every day! How can services and pages sustain this? How can we compare different solutions?

Scalability

First Approach: Sampling & Extrapolation

- We can measure how long a program takes to run for different input sizes.
- This will give us a function we can extrapolate.
- Using this method we can experimentally compare different algorithms.

Is there a better and simpler way to estimate the complexity of algorithms?

Asymptotic Analysis - Definition

A method for defining the mathematical boundaries of the run-time performance or space usage of programs as the input size increases. Useful for estimating the time and space complexity in function of the input size.

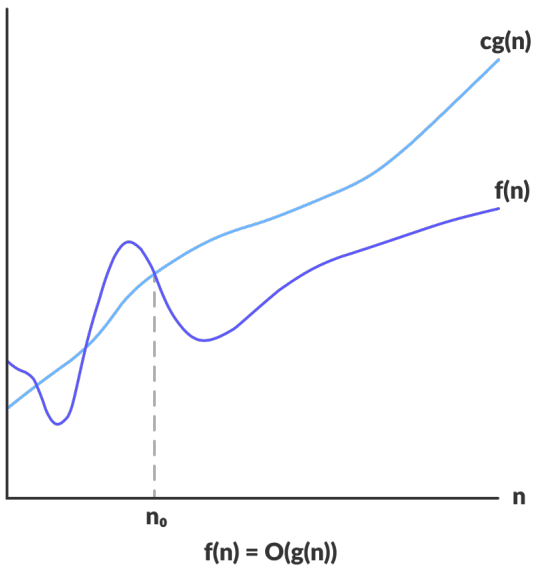
Using the asymptotic analysis we can easily estimate:

- Lower Bound - $\Omega(n)$ Omega notation
- Upper Bound - $O(n)$ Big oh notation
- *Tight Bound* - $\Theta(n)$ Theta notation

Big O Notation

The formal way to express the upper bound of an algorithm running time.

- Omit lower terms and coefficients.
- $f(n) = O(g(n)) \rightarrow f(n) \leq kg(n)$ for $n > n_0$



Space Usage vs Running Time

There exists almost always a **trade-off** in an algorithm between space-usage, run-time performance and brain power. Sometimes, we need to save computed values for future actions in order to improve the running time.

Example

Fibonacci Algorithm

Simplification examples

Drop lower terms and multiplicative constants.

$$n^3 + n^2 + 1$$

$$log(n) + n^2 + 100$$

$$0.001 * n^3 + 1$$

$$n * log(n) + n^2$$

$$n * log(n) + n + 10^{100}$$

$$n * log(n) + 500 * n + 10^{100}$$

$$n * log(n) + n + 2^n$$

How do we define lower terms?

Examples Linear Algorithm

```
1 int main() {
2     int ans = 0;
3     for (int i = 0; i < 100; i++) {
4         if (i % 2 == 0)
5             ans += i;
6     }
7     cout << ans << endl;
8     return 0;
9 }
```

Examples Quadratic Algorithm

```
1  int main(){
2      int ans=0;
3      for (int i = 0; i < 100; i++){
4          for (int j = 0; j < 100; j++) {
5              if (i * i == j)
6                  ans += i;
7              if (j - i == i)
8                  ans -= i;
9          }
10     }
11     cout << ans << endl;
12     return 0;
13 }
```

Examples of Time Complexity