算法分析 | Algorithm Analysis

Time complexities $T_{avg}(N)$, $T_{worst}(N)$

N stands for the input. There may be more than one input.

Predict the **growth** in run time as the N change.

$T(N) = O(f(N))$, if $T(N) \leq cf(N), N \geq n_0$ (upper bound)

$T(N) = \Omega(f(N))$, if $T(N) \geq cf(N), N \geq n_0$ (lower bound)

$T(N) = \Theta(f(N)) \equiv T(N) = O(f(N)) \wedge T(N) = \Omega(f(N))$

$T(N) = o(f(N))$, if $T(N) = O(f(N)) \wedge T(N) \neq \Theta(f(N))$

$T_1(N) = O(f(N)), T_2 = O(g(N))$, then:

$T_1(N) + T_2(N) = O(\max(f(N), g(N)))$

$T_1(N) \times T_2(N) = O(f(N) \cdot g(N))$

Recursions:

```
long int Fib(int N)     /* T(N) */
{
    if(N <= 1) return 1;
    else return Fib(N - 1) + Fib(N - 2);
                /* T(N-1)     T(N-2)   */
}
/* T(N) = T(N-1) + T(N-2) + 2 >= Fib(N) >= 1.5^N */
```