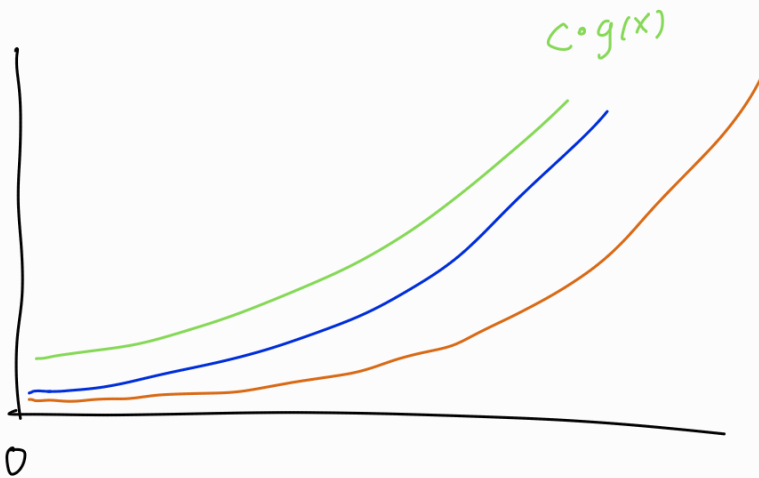


Complejidad

$$\underbrace{O(f(x))}_{\substack{\nearrow \Omega(x) \rightarrow \text{lim inferiores} \\ \nearrow \Theta(x) \rightarrow \text{Exactitud}}}$$

↳ todas las funciones $g(x)$ t.q. $g(x)$ está por debajo de $c \cdot f(x)$ para $c > 0$



$f(x) \rightarrow$ Algoritmo

$g(x) \rightarrow$ Limite superior más cerrado

$$f(x) \Rightarrow O(g(x)) \Rightarrow \exists c \text{ t.q. } \forall x > n \quad c \cdot g(x) > f(x)$$

Si $g(x)$ acota a $f(x)$ Superiormente

↓
a partir de un punto hacia el ∞ se cumple.

$$f(x) = x^2 + 5x + \dots + x^a \quad a \leq 2$$

$$O(x^2)$$

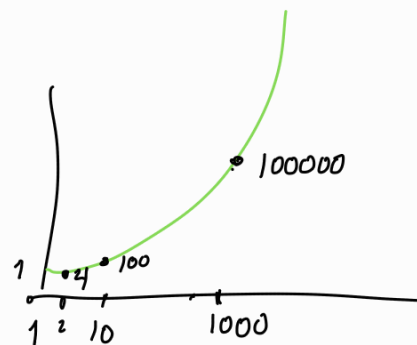
$$\underline{O(N^2)}$$

$$\Rightarrow$$

$$g(x) = \boxed{x^2}$$

$$\Rightarrow$$

$$\lim_{x \rightarrow \infty} = \text{igual}$$



$$10 \Rightarrow 150 \neq 100$$

$O(N) \rightarrow$ Modelar un crecimiento \rightarrow de un verurso respecto a una variable.
 Acotar superiormente el comportamiento de mi algoritmo

Recursos.

Tiempo y Memoria.

#operaciones

#variables

Ejemplo:

```
int suma = 0;
for (int i = 0; i < K * M; i++)
    suma = suma + 2;
cout << suma << endl;
```

Tiempo $O(n)$ x
 memoria $O(K)$
 $O(1)$

Siempre que escribamos una complejidad.

Respecto a algo visible y conocido en el código.

• Asociarlo a una variable.

Cuando no es importante definir la variable.

$O(n)$ donde $n = m \cdot K$

$O(KM)$

Tiempo.

Operación aritmética sencilla
 $+$, $-$, \times , \div

• definin, Asignación, acceso, modificación de variable

$a = 5$ \uparrow \uparrow $\text{int } b;$
 $b = v[i]$

• Entrada, Salida

$\text{cin} >>$ $\text{read}()$
 $\text{cout} <<$ $\text{print}()$

Memoria

• Variables primitivas \rightarrow int , double , char , bool

$\Rightarrow O(n)$

$\text{String } s$

$a = \text{"Hola mundo"}$ $\Rightarrow 10$

$\text{vector} < \text{int} >$

$x = [1, 2, 3, 4, 5]$ $\Rightarrow 5$

$V = []$
 $\text{for } (\text{int } i = 0; i < n; i++)$
 $V.append(i)$

$\hookrightarrow |V| \cdot n$

Tarea: Calcular complejidad en tiempo y memoria de los códigos enviados.