

TEMA: Análisis de algoritmos

input \rightarrow Procedimiento bien definido \rightarrow output \rightarrow secuencia de pasos computacionales

Algoritmo \rightarrow Elegir cuidadosamente

Es correcto
Para todo input
genera el
output

Especificado de forma precisa
sin ambigüedad

Análisis de Algoritmos

predecir el uso de recursos

Tiempo computacional

debe ser \rightarrow Calculado independientemente del hardware

Conteo de instrucciones elementales

Eficiencia en tiempo y espacio

$f(n)$

Crece según el tamaño del input(n)

\rightarrow representa el orden de crecimiento

Instrucciones elementales

- Toman el mismo tiempo, independientemente del tamaño del input.
- El tiempo de ejecución de una instrucción elemental se denota mediante T.

- Aritméticas: +, -, /, *, ^
- Bit operations: <, >, &, or
- lógicas: ==, !=, ' ', &, "
- Jumps (return, llamadas o métodos)
- Asignaciones, acceso a arrays

Conteo de instrucciones

```
var M = A[0]  $\rightarrow 3T$  (asignación de variable)  
for (var i = 0; i < A.size; i++) {  
    if (A[i] > M) {  
        M = A[i]  $\rightarrow 2TN$   
    }  
}
```

$$f(n) = 7NT + 3T$$

bueno!

Análisis de Algoritmos - continuación

-> Análisis asintótico

-> Dada cada una función obtenida mediante cambio de instrucciones el análisis asintótico se enfoca en encontrar el término que determine el crecimiento de la función

$$f(n) = 2n^3 + 3n^2 + \log_{10}(n) + 3.33$$

↓
más importante

↑ de menor a mayor

-> Según el análisis asintótico, $f(n) = n^3$

-> El análisis asintótico se puede hacer enfocándose en

o Peor caso -> suele ser suficiente

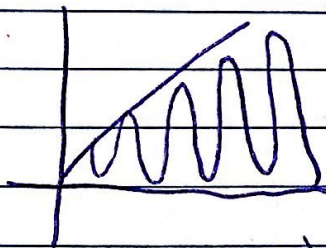
o Caso promedio

o Mejor caso

-> Big O

=> Encontrar el peor caso

=> Encontrar la cota superior

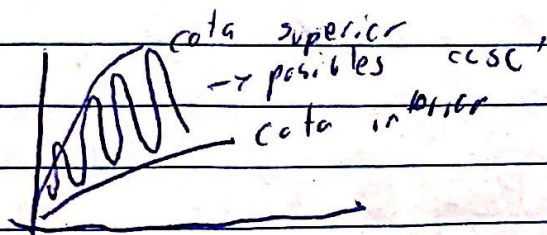


-> Si el peor caso es $O(n)$, o sea se comporta linealmente, cualquier caso va a ser igual o mejor que este.

espacio de posibles comportamientos

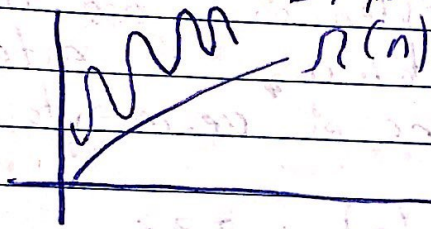
Big - O (theta)

=> Encontrar la cota superior o inferior a partir de cierta cota.



→ Big Ω (omega)

→ Es encontrar el límite inferior
→ posibles casos



→ Ordenes de crecimiento comunes

= > constante

= > logarítmico $\log(n)$ → base 2. Sin base base 2

= > $\log N$

= > lineal logarítmico $N \log(N)$

= > cuadrático N^2

= > cúbico N^3

= > exponencial 2^N

= > Factorial $N!$