Introducción al Análisis de Algoritmos

1. Algoritmo fácil de entender y codificado.
2. Algoritmos que hagan un uso eficiente de los recursos ( lo más rápido posible ).

* “ Medir ” el tiempo de ejecución.
* El tiempo de ejecución depende de :

1. Tamaño de la entrada.
2. Calidad del código generado por el compilador.
3. Rapidez de las instrucciones de “ máquina ”.
4. Complejidad en el tiempo de ejecución dada por el algoritmo.
5. Tiempo de ejecución.

* Mejor caso : 1
* Peor caso :
* Caso promedio :

1. Búsqueda lineal :

* Mejor caso : cantidad mínima de tiempo de ejecución.
* Peor caso : cantidad máxima de tiempo de ejecución. [ n ]
* Caso Promedio: cantidad de tiempo promedio que ejecuta.

1. Búsqueda binaria :

* Mejor caso : una sola separación.
* Pero caso : separe pero no encuentre. [ Log2 (n) ]

[ logb a = c ←→ bc = a ]

* Caso promedio : máxima cantidad de separaciones promediadas.

Tipos de Orden : [ O ][ Ω ][ Ө ]

1. [ O ] → T(n) = O(f(n)) si existen constantes positivas c y n0 / T(n) ≤ c.f(n) ∀ n ≥ n0
2. [ Ω ] → T(n) = Ω(f(n)) si existen constantes positivas c y n0 / T(n) ≥ c.f(n) ∀ n ≥ n0
3. [ Ө ] → Tiempo exacto de ejecución.

Regla de la suma : [ cons = O(1) ][ snoc = O(n) ]

1. A → T(n) = O(f(n))
2. B → T1(n) = O(g(n))
3. ¿Cuál es el orden de T(n) + T1(n) ? → O(max(f(n), g(n)))
4. Algoritmos de ordenamiento(Sorting).

* Ordenamiento Interno.
* Estabilidad : Un algoritmo es estable si mantiene el orden relativo que tenían los elementos con claves iguales.