Capítulo 4 ANÁLISIS DE ALGORTIMOS

Nombre: Jean Carlos Iñiguez

* **ANÁLISIS DE LAS ESTRUCTURAS DE CONTROL:**

El análisis de algoritmos se realiza usualmente desde adentro hacia afuera, empezando por las instrucciones individuales (que generalmente consumen tiempo constante) y luego considerando cómo se combinan mediante las estructuras de control.

**4.2.1 Secuencias**

* Si se tienen dos fragmentos de algoritmo P₁ y P₂, con tiempos de ejecución t₁ y t₂, entonces el tiempo total de ejecutar P₁; P₂ es t₁ + t₂.
* Por la regla del máximo:

**Advertencia:** no siempre se pueden analizar P₁ y P₂ por separado, ya que el comportamiento de P₂ puede depender del resultado de P₁.

**4.2.2 Bucles “para” (for)**

* Los bucles "para" son más fáciles de analizar.
* Ejemplo:

para i = 1 hasta m hacer P(i)

* Ejemplo:
* Si P(i) toma un tiempo constante t (aunque dependa del tamaño del ejemplar), y se ejecuta m veces, el tiempo total es:



* Ejemplo:
* Nota: se ignora temporalmente el **tiempo de control del bucle**, aunque en realidad existe (como en su equivalente mientras).
* **Convención importante**: si m = 0, el bucle no se ejecuta ninguna vez, y eso no es un error.
* **USO DE BARÓMETRO:**

El método del barómetro es una técnica útil para simplificar el análisis del tiempo de ejecución de un algoritmo.

🔹 ¿Qué es una instrucción barómetro?

* Es una instrucción (o comprobación) que se ejecuta al menos tantas veces como cualquier otra del algoritmo.
* Puede haber otras instrucciones que se ejecuten un número constante de veces más, pero eso no afecta el análisis asintótico.
* Permite estimar el tiempo total del algoritmo simplemente contando cuántas veces se ejecuta el barómetro.

🔹 Ventajas del uso del barómetro:

* Simplifica el análisis: no es necesario analizar cada instrucción individualmente.
* Elimina las constantes específicas de implementación (como el tiempo de suma, comparación, etc.).
* Ayuda a centrarse en la estructura general del algoritmo y no en los detalles de bajo nivel.

🔹 Ejemplo:

* En el algoritmo Fibiter (iterativo para la sucesión de Fibonacci), si contamos operaciones aritméticas como de costo unitario:
  + Podríamos contar cuántas veces se ejecuta j ← i + j.
  + Como se ejecuta n veces, entonces el algoritmo es de orden O(n).

🔹 Uso en bucles anidados:

* En bucles anidados, la instrucción más interna suele ser un buen candidato a barómetro.
* Cuidado con los bucles que a veces no se ejecutan:
  + Aunque no ejecuten el barómetro, sí consumen tiempo de control.
  + Si se repite muchas veces, ese tiempo de control puede ser relevante.

🔹 Advertencia:

* Si un algoritmo tiene muchos bucles vacíos, usar como barómetro una instrucción del interior puede subestimar el tiempo total.
* Ejemplo: clasificación por casillas (bucket sort) con intervalos muy amplios (como valores entre 1 y s):
  + Muchos "casilleros" pueden estar vacíos, y eso influye en el análisis, aunque no se ejecuten instrucciones dentro.
* **ANÁLISIS DEL CASO MEDIO:**

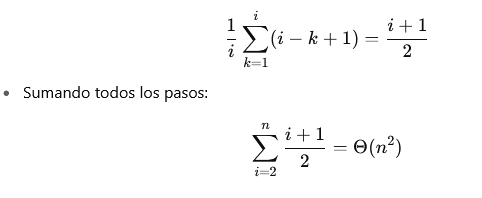
**¿Qué es el "caso medio"?**

* Es el tiempo promedio que tarda un algoritmo, suponiendo que todas las entradas posibles son igual de probables.
* Para ordenación, se asume que:
  + Todos los elementos son distintos.
  + Todas las n! permutaciones son igualmente probables.

**Aplicación a la inserción**

* En cada paso i del algoritmo, el elemento T[i] se inserta en el subarreglo ordenado T[1..i-1].
* El rango parcial de T[i] (su futura posición ordenada) puede tomar cualquier valor entre 1 e i con igual probabilidad.

**Cálculo del número medio de comparaciones:**

* Si el rango parcial de T[i] es k, entonces el número de comparaciones es i - k + 1.
* ****El promedio de comparaciones por paso i es: