Analisis Aromatizado

Ernesto Rodriguez - Juan Roberto Alvaro Saravia

Universidad Francisco Marroquin

ernestorodriguez@ufm.edu - juanalvarado@ufm.edu

Ejemplo: Pila

- Se tienen dos operaciones: `Push` y `Pop`
- Cada operación toma un tiempo constante c
- Por lo tanto una secuencia de N operaciones toma cN
- Por lo tanto el promedio es $\frac{cN}{N} = c$

Operación Multipop

- Operación que ejecuta pop k veces
- Si $k > n$, solo se ejecuta n veces
- Tiempo de ejecución es $\mathcal{O}(n)$

Algorithm 1 Multipop

```
1: procedure MULTIPOP(S, k)
2:     while len(S) > 0 and k > 0 do
3:         Pop(S)
4:         k ← k − 1
```

Tiempo Promedio

- Se considera una secuencia de N operaciones, las cuales pueden ser `Push`, `Pop` y `Multipop`
  - ¿Cual es el tiempo de ejecución de N operaciones?
  - ¿Que pasa si la pila esta vacia?

Analisis Aromatizado

- Consiste en calcular el tiempo promedio que tomaria una secuencia de operaciones.
- Difiere de la complejidad promedio:
  - Es el tiempo promedio que toma cada operación en una secuencia de operaciones
  - Siempre se considera el peor caso del input
  - Ie. la cantidad de ciclos es la peor, pero el tiempo que toma cada ciclo se promedia.
- Al final, en vez de tener n operaciones de costo diferente, se obtiene el promedio repetido n veces.
- El ejemplo anterior se llama *Analisis Acumulado*
- Applicaciones: Scheduling de procesos

Analisis Acumulado

- Se considera una secuencia de n operaciones
- Cada operación tiene un costo $T(n)$
- Por eso mismo, el costo promedio de cada operación seria $T(n)/n$

Ejemplo: Contador Binario

- Se tiene un numero c de k bits.
- Cada bit se almacena en un arreglo de tamaño k.
- En la posición 0 del arreglo se encuentra el bit de menor orden
- En la posición $k-1$ del arreglo se encuentra el bit de mayor orden
- Por eso mismo $c = \sum_{i=0}^{k-1} A[i] * 2^i$

Incrementando el Contador

Algorithm 2 Incrementar

```
1: procedure INCREMENTAR(A)
2:     i ← 0
3:     while i < len(A) and A[i] ≡ 1 do
4:         A[i] ← 0
5:         i ← i + 1
6:     if i < len(A) then
7:         A[i] ← 1
```

Analisis Asintotico

- Cada ejecución de `Incrementar` toma $\mathcal{O}(k)$ operaciones
- Se ejecuta `Incrementar` n veces.
- El tiempo de ejecución es $\mathcal{O}(nk)$

¿Podemos ser mas precisios?

Incrementando el Contador

Counter value	A[7]	A[6]	A[5]	A[4]	A[3]	A[2]	A[1]	A[0]	Total cost
0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	1	1
2	0	0	0	0	0	0	1	0	3
3	0	0	0	0	0	0	1	1	4
4	0	0	0	0	0	1	0	0	7
5	0	0	0	0	0	1	0	1	8
6	0	0	0	0	0	1	1	0	10
7	0	0	0	0	0	1	1	1	11
8	0	0	0	0	1	0	0	0	15
9	0	0	0	0	1	0	0	1	16
10	0	0	0	0	1	0	1	0	18
11	0	0	0	0	1	0	1	1	19
12	0	0	0	0	1	1	0	0	22
13	0	0	0	0	1	1	0	1	23
14	0	0	0	0	1	1	1	0	25
15	0	0	0	0	1	1	1	1	26
16	0	0	0	1	0	0	0	0	31

Analisis Asintotico

- El primer bit se debe cambiar en cada operación
- Sin embargo, el segundo bit se cambia cada dos operaciones
- El cuarto bit se cambia cada cuatro operaciones
- etc...