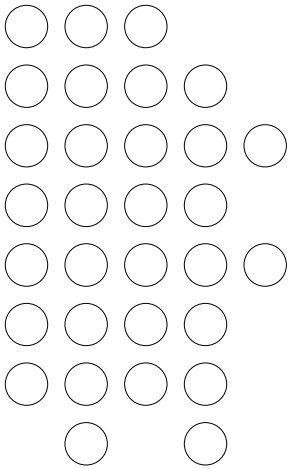
Análisis Amortizado



Diseño y Análisis de Algoritmos

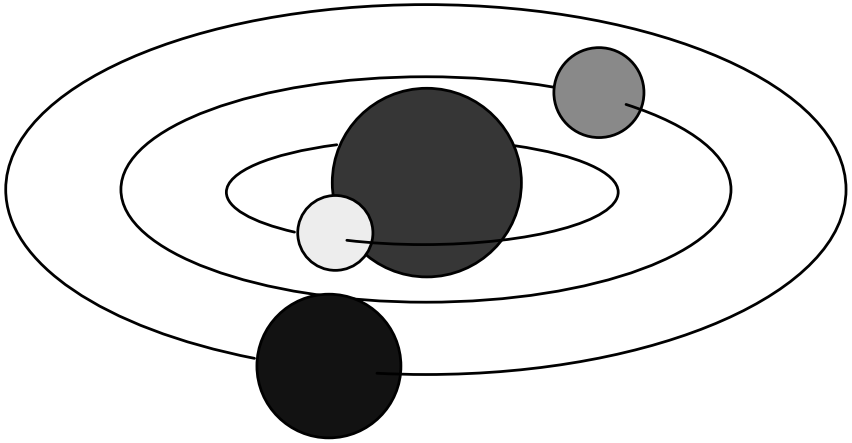
Cátedra de Programación

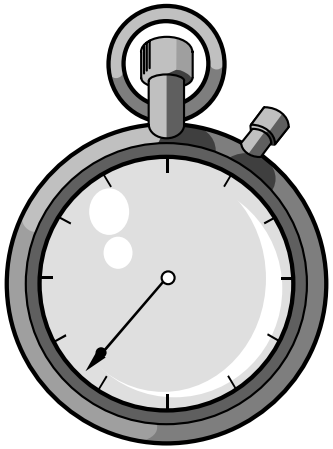
Carrera de Ingeniería de Sistemas

Prof. Isabel Besembel Carrera

UNIVERSIDAD DE LOS ANDES

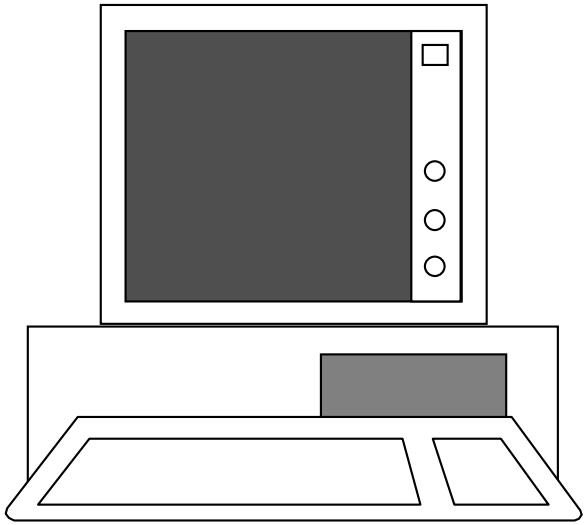
Análisis de algoritmos





- La eficiencia de un programa tiene dos ingredientes fundamentales: espacio y tiempo.
- La eficiencia en espacio es una medida de la cantidad de memoria requerida por un programa.
- La eficiencia en tiempo se mide en términos de la cantidad de tiempo de ejecución del programa.

Análisis de algoritmos



Eficiencia de los algoritmos: Enfoques
Empírico (a posteriori): programar las soluciones competidoras, probarlas en máquina y seleccionar
Teórico (a priori): determinar matemáticamente la cantidad de recursos necesarios para cada algoritmo en función del tamaño de las entradas

- Ambas dependen del tipo de computador y compilador, por lo que **no** se estudiará aquí la eficiencia de los programas, sino la eficiencia de los algoritmos.
- Asimismo, este análisis dependerá de si trabajamos con máquinas de un solo procesador o de varios de ellos.
- Centraremos nuestra atención en los algoritmos para máquinas de un solo procesador que ejecutan una instrucción luego de otra.

Operaciones características y complejidad de cálculo

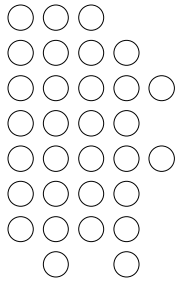
- La eficiencia de los algoritmos está basada en una operación característica que el algoritmo repite y que define su **complejidad en Tiempo (T(n))**.
- **T(n)** es el número de operaciones características que el algoritmo desarrolla para una entrada **n** dada.
- El máximo tiempo de ejecución de un algoritmo para todas las instancias de tamaño **n**, se denomina la complejidad en tiempo para el peor caso W(n). Asimismo, la complejidad promedio en tiempo es A(n), donde pj es la probabilidad de que esta instancia ocurra.

$$W(n) = \max_{1 \le j \le k} T_j(n) \qquad A(n) = \sum_{j=1}^{k} p_j T_j(n)$$

- Normalmente se tendrán muchos algoritmos diferentes para resolver un mismo problema, por lo que debe existir un criterio para seleccionar el mejor.

UNIVERSIDAD DE LOS ANDES

Notaciones



- El interés principal del análisis de algoritmos radica en saber cómo crece el tiempo de ejecución, cuando el tamaño de la entrada crece.
- Esto es la **eficiencia asintótica** del algoritmo.
- La notación asintótica se describe por medio de una función cuyo dominio es los números naturales ($\aleph$).
- Se consideran las funciones asintóticamente no negativas.

1.- **Notación Θ**, límite asintóticamente estrecho

Para una función dada g(n),

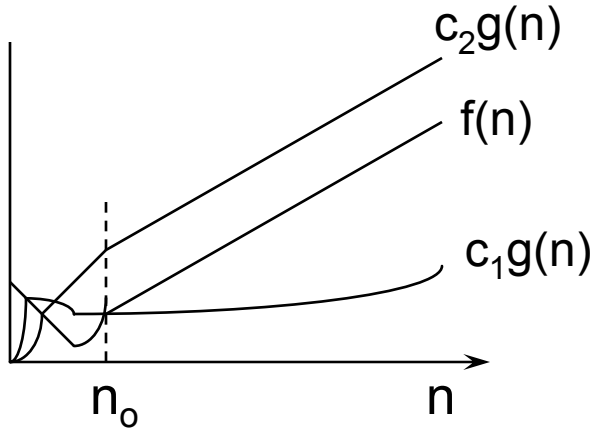
$\Theta(g(n)) = \{ f(n) / \exists$ las constantes positivas c_1, c_2 y n_o /

$0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n) \; \forall \; n \geq n_o\}$

Ejemplo: $n^2/2 - 3n = \Theta(n^2) \Rightarrow c_1 n^2 \leq n^2/2 - 3n \leq c_2 n^2 \Rightarrow c_1 \leq 1/2 - 3 \leq c_2$

UNIVERSIDAD DE LOS ANDES

Notaciones



Se denota
$f(n) = \Theta(g(n)) \equiv f(n) \in \Theta(g(n))$,
y se dice que g(n) es el límite asintóticamente estrecho de f(n).

- 2.- **Notación O**, límite asintótico superior

  $O(g(n)) = \{ f(n) / \exists$ las constantes positivas c y n_o /

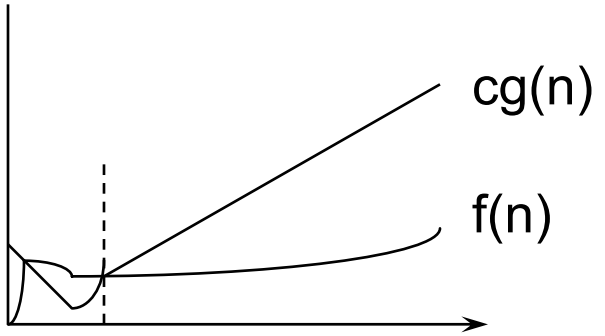
  $0 \leq f(n) \leq c\, g(n)\ \forall\ n \geq n_o\}$

Ejemplo: $an^2 + bn + c$ con $a > 0$ tiene $O(n^2)$

UNIVERSIDAD DE LOS ANDES

Notaciones

- El hecho que f(n) = **Θ**(g(n)) implica f(n) = O(g(n)), por lo que **Θ**(g(n)) ≤ O(g(n)).
- Se dice que g(n) es el límite asintótico superior de f(n).

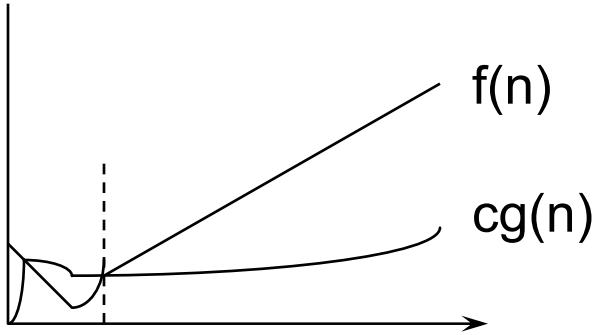


3.- **Notación Ω**, límite asintótico inferior

Ω(g(n)) = { f(n) / ∃ las constantes positivas c y n_o /

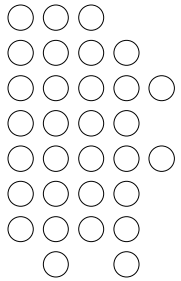
$0 \leq c\, g(n) \leq f(n)\ \forall\ n \geq n_o$}

Ejemplo: $an^2 + bn + c$ con $a > 0$ tiene $O(n^2) = \mathbf{\Omega}(n^2) = \mathbf{\Theta}(n^2)$



UNIVERSIDAD DE LOS ANDES

Notaciones



- Teorema: Para cualquier función f(n) y g(n), f(n) = Θ(g(n)) si y solo si f(n) = O(g(n)) y f(n) = Ω(g(n))
  - Ejemplo:

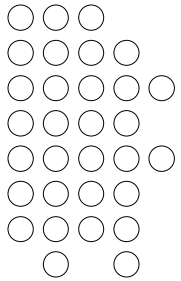
$an^2 + bn + c = \Theta(n^2)$ para cualquiera de las constantes a, b, c donde a > 0, según el teorema anterior se obtiene inmediatamente que

$an^2 + bn + c = O(n^2)$

$an^2 + bn + c = \Omega(n^2)$

UNIVERSIDAD DE LOS ANDES

Notaciones



- Comparación de las funciones: f(n) y g(n) son asintóticamente +

1.- Transitividad
   - Si $f(n) = \Theta(g(n))$ y $g(n) = \Theta(h(n))$ entonces $f(n) = \Theta(h(n))$
   - Si $f(n) = O(g(n))$ y $g(n) = O(h(n))$ entonces $f(n) = O(h(n))$
   - Si $f(n) = \Omega(g(n))$ y $g(n) = \Omega(h(n))$ entonces $f(n) = \Omega(h(n))$

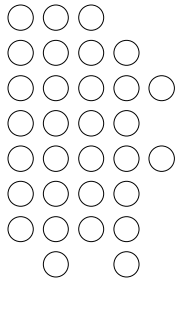
2.- Reflexibidad
   - $f(n) = \Theta(f(n))$, $f(n) = O(f(n))$, $f(n) = \Omega(f(n))$

3.- Simetría
   - $f(n) = \Theta(g(n))$ si y solo si $g(n) = \Theta(f(n))$

4.- Simetría transpuesta
   - $f(n) = O(g(n))$ si y solo si $g(n) = \Omega(g(n))$

UNIVERSIDAD DE LOS ANDES

Notaciones



- Analogía entre la comparación asintótica de dos funciones f y g y la comparación entre dos números reales a y b:
  - $f(n) = O(g(n)) \Rightarrow a \leq b$
  - $f(n) = \Omega(g(n)) \Rightarrow a \geq b$
  - $f(n) = \Theta(g(n)) \Rightarrow a = b$
- Para dos números reales cualesquiera una de las siguientes comparaciones es cierta: $a < b$, $a = b$ o $a > b$.
- No todas las funciones son comparables asintóticamente.
- Ejemplo: $f(n) = n$ y $g(n) = n^{1+\text{sen}(n)}$, no pueden ser comparadas usando la notación asintótica pues el valor del exponente 1+sen(n) oscila entre 0 y 2 tomando todos los valores en ese rango.

UNIVERSIDAD DE LOS ANDES

Técnicas de demostración

1. Contradicción o reducción a lo absurdo:

Demostrar la veracidad de una sentencia, demostrando que su negación da lugar a una contradicción

Teorema: Existen dos números irracionales X e Y tales que X^Y es racional

Demostración:

Suposición: X^Y es irracional siempre que X e Y son irracionales

$\sqrt{2}$ es irracional y $Z = X^Y$ es irracional, $X = \sqrt{2}$, $Y = \sqrt{2}$

$W = Z^{\sqrt{2}}$, W es irracional dada la suposición hecha que Z y $\sqrt{2}$ son irracionales

$$W = Z^{\sqrt{2}} = (\sqrt{2}^{\sqrt{2}})^{\sqrt{2}} = (\sqrt{2})^{(\sqrt{2} \times \sqrt{2})} = (\sqrt{2})^2 = 2$$

Se concluye que 2 es irracional, lo cual es claramente falso.

Por tanto, se concluye que la suposición es falsa y por ello, es posible obtener un número racional cuando se eleva un irracional a una potencia irracional

Técnicas de demostración

2. Inducción matemática:

- Inducción: inferir una ley general de casos particulares
- Deducción: inferencia de lo general a lo particular

Principio:

Enero/05 **cuadrado(Entero k):Entero** **{ pre: k ≥ 0 }**		**{ pos: p ≥ 0 }**
1	p = si (k = 0) entonces 0 sino 2 * k + cuadrado(k - 1) - 1 fsi	**p:** Entero. Resultado.
2	regrese p	
1	k = 0 => p = 0	
2	k = 1 => p = 1	
3	k = 2 => p = 4	
4	k = 8 => p = 64	

Técnicas de demostración

Etapa básica: k = 0, el resultado es 0

Etapa inductiva: Se supone k ≥ 1 y que por el momento el algoritmo tiene éxito en k - 1. Por definición cuadrado(k) = 2k+cuadrado(k-1)-1.

Por la suposición, cuadrado(k-1)=$(k-1)^2$, por lo tanto

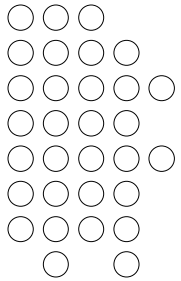
cuadrado(n)=$2k+(k-1)^2-1=2k+(k^2-2k+1)-1=k^2$

Se ha demostrado que el algoritmo tiene éxito en k siempre que tenga éxito para k-1, siempre y cuando k ≥ 1.

análisis

UNIVERSIDAD DE LOS ANDES

Técnicas de demostración



- Errores en las demostraciones:

Teorema: Todos los toros son del mismo color

Demostración: Sea H={toros}, n=|H|

Caso base: n=0 => H=∅ cierto, n=1 => cierto pues hay un solo toro

Paso de inducción: n>1, h1, h2, h3, …, hn son los toros

Hipótesis: todo conjunto de n-1 toros contiene toros del mismo color

Sea H1 = H - {h1} y H2 = H - {h2}

H1: h2 h3 h4 h5 toros del mismo color (n=5)

H2: h1 h3 h4 h5

Como |H1| = |H2| = n-1, la hipótesis de inducción es aplicable

Todos los toros de H1 son del mismo color c1 y todos los de H2 son del mismo color c2

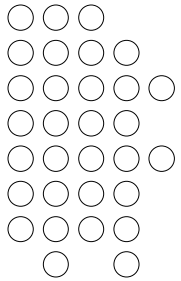
?c1<>c2? No, ya que hn ∈ H1 y hn ∈ H2 => c1 = c2

Como todos los toros del conjunto H pertenecen a cualquier H1 o H2 o ambos, se concluye que c1 = c2 = c

Dónde está la falacia? En que hn ∈ H1 y hn ∈ H2, ya que para n=2, h2 ∉ H2

UNIVERSIDAD DE LOS ANDES

Técnicas de demostración



- Inducción matemática generalizada

  Considere cualquier P de los Z y a, b ∈ Z / a ≤ b, si

  1. P(n) es válido ∀ a ≤ n ≤ b y
  2. Para cualquier n ∈ Z, n ≥ b, el hecho consistente en que P(n) es válido se sigue de la suposición de que P(m) es válido ∀ m, a ≤ m ≤ n

  ⇒ P(n) es válida ∀ n, n ≥ a

Teorema: la multiplicación "a la rusa" multiplica correctamente cualquier pareja de números enteros positivos

UNIVERSIDAD DE LOS ANDES

Multiplicación a la rusa

981	1234	1234
490	2468	
245	4936	4936
122	9872	
61	19744	19744
30	39488	
15	78976	78976
7	157952	157952
3	315904	315904
1	631808	631808
		1210554

UNIVERSIDAD DE LOS ANDES

Multiplicación a la rusa

Demostración: m x n sobre m

Caso base: m = 1 => solo 1 fila 1 x n, no se tacha por ser par => el resultado es n

Paso de inducción: m >= 2 y n ∈ Z+

Hipótesis: la multiplicación a la rusa multiplica correctamente s x t para s ∈ Z+, s < m, t ∈ Z+.

Caso 1: Si m es par, la 2da. Fila de la tabla contiene m/2 en la columna izquierda y 2n en la derecha => la fila se tachará antes de la suma final => el resultado final m x n a la rusa = m/2 x n/2, pero m/2 es positivo y menor que m => la hipótesis es aplicable.

El resultado m/2 x n/2 a la rusa es (m/2) x (n/2) = m x n.

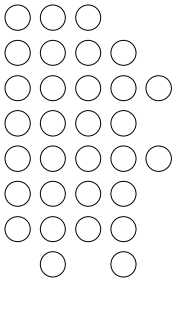
Multiplicación a la rusa

Demostración (continuación)

Caso 2: Si m es impar, similar sustituyendo m/2 por (m-1)/2 en todas las partes => el resultado final de m x n a la rusa = n + resultado de (m-1)/2 x 2n a la rusa => por la hipótesis ((m-1)/2) x (2n) => n + ((m-1)/2) x (2n) = m x n.

UNIVERSIDAD DE LOS ANDES

Sumatorias



- Los métodos más usados para describir el tiempo de corrida de los algoritmos se basan en encontrar una limitación para la suma de una secuencia mediante:

1.- Inducción matemática: Ejemplo: la serie geométrica dada es

$$\sum_{k=0}^{n+1} 3^k = \sum_{k=0}^{n} 3^k + 3^{n+1} \le c3^n + 3^{n+1} = \left(\tfrac{1}{3} + \tfrac{1}{c}\right)c3^{n+1} \le c3^{n+1}$$ y está limitada con $O(3^n)$

2.- Limitando los términos: Normalmente se escoge el término más grande para limitar a los otros, en general si $a_{max}=\max 1<k<n\ a_k$ entonces $\sum_{k=0}^{n} a_k \le n a_{max}$. Este método no es muy recomendable si la serie puede ser limitada por una serie geométrica, en cuyo caso se prefiere el límite de esta última.

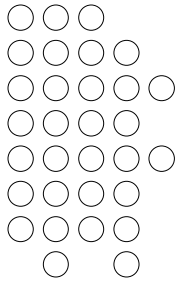
3.- Fisión: Se expresa la sumatoria en dos o más series separando el rango del índice y encontrando la limitación de cada serie.

En general, $\sum_{k=0}^{n} a_k = \sum_{k=0}^{k_0-1} a_k + \sum_{k=k_0}^{n} a_k = \Theta(1) + \sum_{k=k_0}^{n} a_k$

Ejemplo: $\sum_{k=1}^{n} k = \sum_{k=1}^{n/2} k + \sum_{k=n/2+1}^{n} k \ge \sum_{k=1}^{n/2} 0 + \sum_{k=n/2+1}^{n} (n/2) \ge (n/2)^2 = \Omega(n^2)$

UNIVERSIDAD DE LOS ANDES

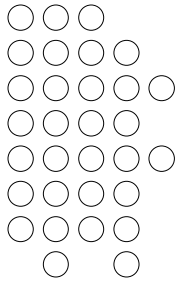
Recurrencias



- Algoritmo recursivo: su tiempo de ejecución se expresa normalmente con una recurrencia.
- Recurrencia: ecuación o desigualdad que describe una función en términos de sus valores para sus entradas más pequeñas.
- Para encontrar los límites asintóticos (Θ, O) se presentan tres métodos:
  - Método por substitución
  - Método por iteración
  - Método maestro
- **Método por substitución**: Se enuncia una forma de solución y se prueba por inducción que dicha solución asociada a una constante es buena.
  - Este método se aplica cuando se puede enunciar la solución posible de la recurrencia.

UNIVERSIDAD DE LOS ANDES

Recurrencias



- Ejemplo: Encontrar el límite superior de la recurrencia

$$T(n) = 2\,T(\lfloor n/2 \rfloor) + n$$

Suponemos que la solución es $T(n) = O(n \lg n)$.

Se probará que $T(n) \leq cn \lg n$ para una constante apropiada $c > 0$.

Se inicia asumiendo que el límite se da para $\lfloor n/2 \rfloor$.

Substituyendo,

$$T(n) \leq 2(c \lfloor n/2 \rfloor \lg (\lfloor n/2 \rfloor)) + n$$
$$T(n) \leq c\,n \lg (n/2) + n$$
$$T(n) = c\,n \lg n - c\,n \lg 2 + n$$
$$T(n) = c\,n \lg n - c\,n + n$$
$$T(n) \leq c\,n \lg n \qquad \text{se cumple para } c \geq 1$$

UNIVERSIDAD DE LOS ANDES

Recurrencias

- La inducción matemática requiere que se pruebe la solución encontrada para condiciones límites.

Si se asume $T(1) = 1$, no se puede escoger c grande, ya que $T(1) \leq c\ 1 \lg 1 = 0$.

Esta dificultad se resuelve tomando $n = 2$ o $n = 3$.

Así, $T(2)=4$, $T(3)=5$ y la prueba inductiva $T(n) \leq c\ n \lg n$ para $c \geq 2$ es suficiente

- Se pueden usar cambios de variables para simplificar, ejemplo: $T(n) = 2\ T(\lfloor \sqrt{n} \rfloor) + \lg n$.

Se hace $m = \lg n$ y $T(2^m) = 2T(2^{m/2}) + m$ se hace $S(m) = T(2^m)$ y se obtiene $S(m) = 2\ S(m/2) + m$ que es parecida a la anterior, por lo cual

$$T(n) = T(2^m) = S(m) = O(m \lg m) = O(\lg n \lg \lg n)$$

UNIVERSIDAD DE LOS ANDES

Recurrencias

- **Método iterativo:** Se expande la recurrencia y se expresa como una sumatoria de términos que depende solo de n y de condiciones iniciales.
- Ejemplo:

$T(n) = 3\,T(\lfloor n/4 \rfloor) + n$

$T(n) = n + 3\,(\lfloor n/4 \rfloor + 3\,T(\lfloor n/16 \rfloor))$

$T(n) = n + 3\,(\lfloor n/4 \rfloor + 3\,(\lfloor n/16 \rfloor + 3\,T(\lfloor n/64 \rfloor)))$

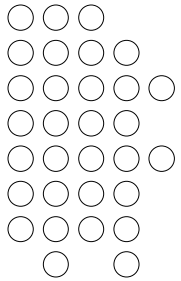
$T(n) = n + 3\lfloor n/4 \rfloor + 9\lfloor n/16 \rfloor + 27\,T(\lfloor n/64 \rfloor)$

$T(n) = n + 3\,n/4 + 9\,n/16 + 27\,n/64 + \ldots + 3\log_4 n\,\Theta(1)$

$$T(n) \le n\sum_{i=0}^{\infty}\left(\tfrac{3}{4}\right)^i + \Theta\left(n^{\log_4 3}\right) \qquad T(n) = 4n + O(n) = O(n)$$

UNIVERSIDAD DE LOS ANDES

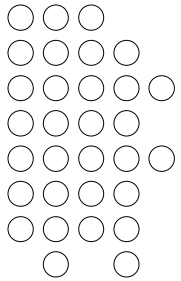
Recurrencias



- **Método maestro:** Se usa para resolver recurrencias del tipo $T(n) = \mathbf{a}\ T(n/b) + f(n)$, donde f(n) es una función asintóticamente positiva y las constantes $a \geq 1$, $b > 1$.
- Ella expresa que el problema se divide en a subproblemas de tamaño n/b.
- Los **a** subproblemas se resuelven recursivamente en tiempo T(n/b).
- El costo de dividir el problema y combinar sus soluciones es descrito por f(n).
- Normalmente en estos casos no se consideran los techos y pisos.

UNIVERSIDAD DE LOS ANDES

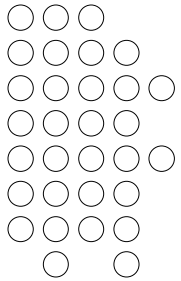
Teorema maestro



- Teorema maestro: Sean $a \geq 1$, $b > 1$ constantes, $f(n)$ una función y $T(n)$ una recurrencia para números enteros no negativos, $T(n) = a\,T(n/b) + f(n)$, donde n/b puede ser $\lfloor n/b \rfloor$ o $\lceil n/b \rceil$ entonces
- $T(n)$ está limitado asintóticamente según:
- 1.- Si $f(n) = O(n^{\log_b a - e})$ para alguna constante $e > 0$, entonces $T(n) = Q(n^{\log_b a})$
- 2.- Si $f(n) = Q(n^{\log_b a})$, entonces $T(n) = Q(n^{\log_b a} \lg n)$
- 3.- Si $f(n) = W(n^{\log_b a + e})$ para alguna constante $e > 0$, y si $a\,f(n/b) \leq c\,f(n)$ para alguna constante $c < 1$ y n suficientemente grande, entonces $T(n) = Q(f(n))$.

UNIVERSIDAD DE LOS ANDES

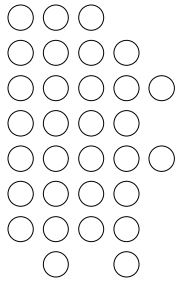
Análisis amortizado



- Garantiza el rendimiento promedio de cada operación en el peor de los casos.
- Es el tiempo requerido para desarrollar una secuencia de operaciones de un TAD promediado sobre todas sus operaciones.
- Métodos:
  - **Agregado:** En el peor de los casos el costo promedio o costo amortizado por operación es T(n)/n, donde T(n) es el costo total de una secuencia de n operaciones.
  - Este costo amortizado se aplica a cada operación, así existan varios tipos de operaciones en la secuencia. Los costos de cada operación son iguales.

UNIVERSIDAD DE LOS ANDES

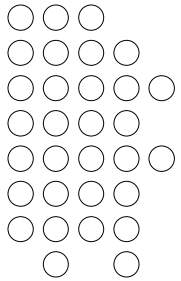
Análisis amortizado



- Ejemplo: Pila[TipoEle: e]
- meterElePila(Pila, TipoEle) y sacarElePila(Pila)
  cada una con O(1)
- El costo total de una secuencia de n operaciones de ellas es $\theta(n)$
- Si se agrega al TAD Pila una operación
  sacarElePila(Pila, Entero) que elimina k elementos del tope de la pila o saca todos los elementos si k > |Pila|

UNIVERSIDAD DE LOS ANDES

Análisis amortizado

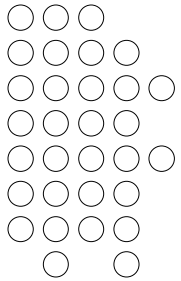


Oct.98	sacarElePila(Entero: k)	
1	(¬ vaciaPila() ∧ k ≠ 0) [sacarElePila() k = k – 1]	**-vaciaPila(), sacarElePila().** Definidos en Pila.
2	regrese	

Pila con n elementos $\Rightarrow$ costo total = min(n, k) $\Rightarrow$ W(n) = θ(n)
W(n) se mantiene en θ(n), pero en una secuencia
de n operaciones donde sacarElePila(k) está incluida
$\Rightarrow$ W(n) = $\theta(n^2)$. El costo amortizado del TAD W(n) = $\theta(n^2)/n = \theta(n)$.

UNIVERSIDAD DE LOS ANDES

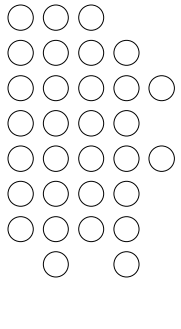
Análisis amortizado



- **Contable:** Se asignan costos amortizados a cada operación, colocándole una cantidad mayor o menor a lo que cada operación cuesta actualmente. Cuando se le asigna una cantidad mayor a lo que ella cuesta actualmente, la diferencia se asigna al TAD como su *crédito*, el cual puede ser utilizado después para *pagar* por aquellas operaciones con un costo menor o igual al crédito actual. Los costos de cada operación pueden ser diferentes.
- El costo total amortizado de una secuencia de operaciones debe ser un límite superior del costo total actual de la secuencia y esta relación debe darse para cualquier secuencia.
- El crédito total del TAD **no** puede ser negativo para cualquier estado del TAD.

UNIVERSIDAD DE LOS ANDES

Análisis amortizado



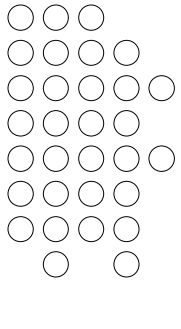
- Ejemplo:

Operación	Costo actual	Costo amortizado
meterElePila(Pila, TipoEle)	1	2
sacarElePila(Pila)	1	0
sacarElePila(Pila, Entero)	min(n, k)	0

- Cada inserción paga 2, 1 por el costo actual y 1 más para completar el costo amortizado
- Cada eliminación no paga, pues cada elemento insertado tiene un crédito de 1 que es utilizado para pagar el costo actual de la eliminación.
- Para cualquier secuencia de n operaciones, el costo total amortizado es un límite superior del costo total actual y los créditos no son negativos.
- El costo total amortizado en O(n) que es el costo total actual.

UNIVERSIDAD DE LOS ANDES

Análisis amortizado



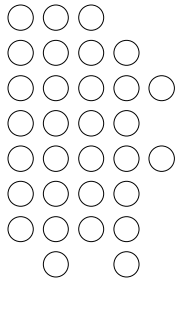
- **Potencial:** Se asignan potenciales a cada operación. Se comienza con un TAD vacío y sea ai el costo actual de la i-ésima operación y D_i el estado del TAD luego de la i-ésima operación que estaba en el estado D_{i-1}. La función potencial Φ va de D_i a un número real $\Phi(D_i)$, que es el potencial asociado a D_i. El costo amortizado $\hat{a}_i = a_i + \Phi(D_i) - \Phi(D_{i-1})$.
- El costo total amortizado de n operaciones es:

$$\sum_{i=1}^{n} \hat{a}_i = \sum_{i=1}^{n} (a_i + \Phi(D_i) - \Phi(D_{i-1})) = \sum_{i=1}^{n} a_i + \Phi(D_n) - \Phi(D_0)$$

- Definiendo una función potencial que haga $\Phi(D_n) \geq \Phi(D_0)$ entonces el costo total amortizado es un límite superior para el costo total actual. Normalmente se escoge $\Phi(D_0) = 0$ para que los potenciales $\Phi(D_i) \geq 0 \ \forall$ i.

UNIVERSIDAD DE LOS ANDES

Análisis amortizado



- Ejemplo: Se escoge el tamaño de la pila como la función potencial

i	operación	D	Φ(D)	a_i	$\hat{a}_i$
0	creaPila()		0		
1	meterElePila(a)	a	1	1	2
2	meterElePila(e)	a e	2	1	2
3	meterElePila(f)	a e f	3	1	2
4	sacarElePila(2)	a	1	2	0

- Si cada secuencia comienza con la pila vacía, su potencial será cero en ese estado.
- La complejidad total de cualquier secuencia de **m** inserciones y **q** eliminaciones es O(2**m**+0**q**) = O(**m**). El costo en el peor de los casos sigue siendo O(n).