

Análise amortizada da complexidade

Estudo da complexidade com base no comportamento temporal de uma sequência de operações sobre uma estrutura de dados, no pior caso

Análise amortizada da complexidade

Técnicas

Análise agregada

Se uma sequência de n operações demora tempo $T(n)$, cada operação demora

$$\frac{T(n)}{n}$$

Método da contabilidade

A cada operação é associado um **custo**, cuja diferença para o **custo real** da operação pode ser usada como **crédito** para **pagar** operações posteriores ou ser **abatida** ao crédito existente

Método do potencial

...

Método do potencial (1)

O potencial $\Phi(D_i)$ do estado D_i de uma estrutura de dados representa energia que pode ser usada por operações futuras

- ▶ D_0 é o estado inicial da estrutura de dados
- ▶ D_i é o estado depois da i -ésima operação
- ▶ Φ é a função potencial
- ▶ $\Phi(D_0)$ é o potencial inicial, em geral 0
- ▶ $\Phi(D_i) - \Phi(D_j)$, $i > j$, é a diferença de potencial entre os estados D_j e D_i
- ▶ c_i é o custo real da i -ésima operação
- ▶ O custo amortizado da i -ésima operação, relativo a Φ , é

$$\hat{c}_i = c_i + \Phi(D_i) - \Phi(D_{i-1})$$

Método do potencial (2)

Pretende-se obter um **majorante** do custo da sequência de operações

$$\begin{aligned}\sum_{i=1}^n \hat{c}_i &= \sum_{i=1}^n (c_i + \Phi(D_i) - \Phi(D_{i-1})) \\ &= \sum_{i=1}^n c_i + \Phi(D_n) - \Phi(D_0) \\ &\geq \sum_{i=1}^n c_i\end{aligned}$$

Logo, a função Φ tem de ser tal que, para qualquer $0 < i \leq n$,

$$\Phi(D_i) \geq \Phi(D_0)$$

Pilha com operação MULTIPOP (1)

Exemplo

Operações $PUSH(S,x)$, $POP(S)$ e $STACK-EMPTY(S)$ com complexidade temporal $O(1)$

MULTIPOP(S, k)

```
1 while not STACK-EMPTY(S) and k > 0 do
2     POP(S)
3     k <- k - 1
```

Custo (real) das operações

PUSH 1

POP 1

MULTIPOP $\min(|S|, k)$

Pilha com operação MULTIPOP (2)

Exemplo

Função potencial

$\Phi(D_i) = s_i$, onde s_i é o número de elementos na pilha

$$\Phi(D_i) = s_i \geq 0 = \Phi(D_0) \quad (\text{pilha inicialmente vazia})$$

Custo amortizado das operações

$$\begin{aligned}\hat{c}_i &= c_i + \Phi(D_i) - \Phi(D_{i-1}) \\ &= c_i + s_i - s_{i-1}\end{aligned}$$

Pilha com operação MULTIPOP (3)

Exemplo

Custo amortizado das operações

	Custo real (c_i)	Dif. potencial ($s_i - s_{i-1}$)	Custo amortizado (\hat{c}_i)	
PUSH	1	1	2	$O(1)$
POP	1	-1	0	$O(1)$
MULTIPOP	$\min(s_{i-1}, k)$	$-\min(s_{i-1}, k)^*$	0	$O(1)$

- * Se $s_{i-1} \geq k$, são desempilhados k elementos, se $s_{i-1} < k$, são desempilhados s_{i-1} elementos, donde $s_i = s_{i-1} - \min(s_{i-1}, k)$ e

$$\begin{aligned}\Phi(D_i) - \Phi(D_{i-1}) &= s_i - s_{i-1} \\ &= s_{i-1} - \min(s_{i-1}, k) - s_{i-1} \\ &= -\min(s_{i-1}, k)\end{aligned}$$

Pilha com operação MULTIPOP (4)

Exemplo

Uma sequência de operações

i	Operação	Estado			Custo real		Custo amortizado	
		Pilha	s_i	$s_i - s_{i-1}$	c_i	Total	\hat{c}_i	Total
		\emptyset	0			0		0
1	PUSH(a)	[a]	1	1	1	1	2	2
2	PUSH(b)	[a b]	2	1	1	2	2	4
3	PUSH(c)	[a b c]	3	1	1	3	2	6
4	PUSH(d)	[a b c d]	4	1	1	4	2	8
5	MULTIPOP(2)	[a b]	2	-2	2	6	0	8
6	POP()	[a]	1	-1	1	7	0	8
7	MULTIPOP(2)	\emptyset	0	-1	1	8	0	8

O custo amortizado total **não é inferior** ao custo real **em nenhum momento**

Contador binário (1)

Exemplo

Contador binário com k bits, $C[0..k-1]$

- ▶ Bit menos significativo na posição 0
- ▶ Operação INCREMENT

INCREMENT(C)

```
1 i ← 0
2 while i < |C| and C[i] = 1 do
3   C[i] ← 0
4   i ← i + 1
5 if i < |C| then
6   C[i] ← 1
```

Bit					Custo		
	3	2	1	0	Oper.	c_i	Total
	0	0	0	0			0
+1	0	0	0	1	1	1	1
+1	0	0	1	0	2	2	3
+1	0	0	1	1	3	1	4
+1	0	1	0	0	4	3	7
+1	0	1	0	1	5	1	8
+1	0	1	1	0	6	2	10

O custo de uma operação é o número de bits que mudam de valor

No pior caso, todos os k bits passam de 1 a 0

Contador binário (2)

Exemplo

Numa sequência de n operações

- ▶ O bit 0 muda a cada operação
- ▶ O bit 1 muda a cada 2 operações (*i.e.*, em metade)
- ▶ O bit 2 muda a cada 4 operações (*i.e.*, num quarto)
- ▶ ...
- ▶ O bit i muda a cada i operações (*i.e.*, em $n \times 2^{-i}$)

Partindo de todos os bits a 0, o número **total** de mudanças de valor de um bit é

$$n + \left\lfloor \frac{n}{2} \right\rfloor + \left\lfloor \frac{n}{4} \right\rfloor + \dots = \sum_{i=0}^{k-1} \left\lfloor \frac{n}{2^i} \right\rfloor < n \sum_{i=0}^{\infty} \frac{1}{2^i} = 2n$$

Contador binário (3)

Exemplo

Sejam C_i o estado do contador a seguir à i -ésima operação e b_i o número de bits a 1 em C_i

Função potencial

$$\Phi(C_i) = b_i$$

$$\Phi(C_0) = 0$$

Inicialmente, todos os bits estão a 0

$$\Phi(C_i) = b_i \geq \Phi(C_0)$$

O número de bits a 1 nunca é negativo

Seja t_i o número de bits que mudam de 1 para 0 no i -ésimo incremento

Se $b_{i-1} < k$, então $b_i = b_{i-1} - t_i + 1$ (há um bit que passa de 0 a 1)

Se $b_{i-1} = k$, então $t_i = k$ e $b_i = b_{i-1} - t_i = 0$ (todos os bits passam a 0)

Contador binário (4)

Exemplo

Diferença de potencial

$$\begin{aligned}\Phi(C_i) - \Phi(C_{i-1}) &= b_i - b_{i-1} \\ &= \begin{cases} b_{i-1} - t_i + 1 - b_{i-1} = 1 - t_i & \text{se } b_{i-1} < k \\ b_{i-1} - t_i - b_{i-1} = -t_i = -k & \text{se } b_{i-1} = k \end{cases}\end{aligned}$$

Custo amortizado de INCREMENT

$$\hat{c}_i = c_i + \Phi(C_i) - \Phi(C_{i-1})$$

	Custo real (c_i)	Dif. potencial ($b_i - b_{i-1}$)	Custo amortizado (\hat{c}_i)	
$b_{i-1} < k$	$t_i + 1$	$1 - t_i$	2	$O(1)$
$b_{i-1} = k$	k	$-k$	0	$O(1)$

Contador binário (5)

Exemplo

Bit	3	2	1	0	Operação	Custo real		Custo amortizado	
						c_i	Total	\hat{c}_i	Total
	0	0	0	0			0		0
+1	0	0	0	1	1	1	1	2	2
+1	0	0	1	0	2	2	3	2	4
+1	0	0	1	1	3	1	4	2	6
+1	0	1	0	0	4	3	7	2	8
+1	0	1	0	1	5	1	8	2	10
+1	0	1	1	0	6	2	10	2	12
+1	0	1	1	1	7	1	11	2	14
+1	1	0	0	0	8	4	15	2	16
	...								
+1	1	1	1	1	15	1	26	2	30
+1	0	0	0	0	16	4	30	0	30

O custo amortizado total **nunca é inferior** ao custo real