Análise Amortizada
(Cap. 17 CLRS)

Prof. Jussara Almeida

Adaptado do material de Prof. Luiz Chaimowicz

Análise Amortizada

- Útil quando se tem uma sequência de operações sobre alguma estrutura de dados, sendo algumas "caras" e outras "baratas"
- Em uma análise de complexidade simples, a operação cara pode elevar erroneamente o custo do pior caso
- Com a análise amortizada é possível mostrar que o **custo amortizado da operação é menor, quando consideradas todas as operações**

Análise Amortizada

- A análise amortizada não é a análise do caso médio que, como vimos, envolve a análise de probabilidades sobre diferentes entradas

- A análise amortizada é a **análise do pior caso,** mas considerando que em uma sequência de operações os **custos diferentes se compensam**

Análise Amortizada

- Atribuir custos às operações mantendo o custo total (logo, limite superior).

- Para qualquer sequência de operações:

  $$\sum \text{custo amortizado} \geq \sum \text{custo real}$$

- Objetivo: simplificar a apresentação da soma dos custos

- Três métodos principais:
  - Análise Agregada
  - Método Contábil (*accounting*)
  - Método do Potencial

Análise Agregada

- Na análise agregada mostra-se que uma sequência de n operações tem o custo $T(n)$
- No pior caso, o custo médio (ou amortizado) de cada operação é $T(n)/n$
- Nessa análise o custo amortizado de todas as operações é igual, mesmo sendo o custo real diferente

Exemplo: operações sobre uma pilha

- Operações comuns, custo O(1):
  - Push(S,x): empilha o item x na pilha S
  - Pop(S): desempilha o top da pilha S
- Nova operação
  - **Multipop(S,k)**: desempilha k itens da pilha
  - Custo: min(n,k), para *n* itens na pilha O(n)

```
Multipop(S,k)
    while not Vazia(S) and K>0
        pop(S)
        k=k-1
```

Exemplo: operações sobre uma pilha

- No pior caso, qual o custo de uma sequência de *n* operações PUSH, POP e MULTIPOP sobre uma pilha inicialmente vazia ?
  - Cada push ou pop é O(1), logo: O(n)
  - Cada multipop é O(n), logo: n.O(n) = O(n^2). Correto?
- Correto, mas esse pior caso não é firme!
- Usando a análise agregada, pode-se obter um **limite mais firme**, considerando que as outras operações "compensam" essa.

Exemplo: operações sobre uma pilha

- Apesar de uma operação multipop ser O(n), uma sequência qualquer de n operações push, pop e multipop em uma pilha vazia é O(n)
  - Só podemos desempilhar um item que foi empilhado.
  - Portanto para *n* operações, temos no máximo *n* itens: O(n).

- Fazendo-se a análise agregada, o **custo amortizado** de cada operação é **O(n)/n = O(1)**

Exemplo: Contador Binário

A

5	4	3	2	1	0
0	0	0	0	0	0
0	0	0	0	0	1
0	0	0	0	1	0
0	0	0	0	1	1
0	0	0	1	0	0
0	0	0	1	0	1
0	0	0	1	1	0
0	0	0	1	1	1
0	0	1	0	0	0
0	0	1	0	0	1

$k-1$		3	2	1	0

INCREMENT (A, k)

$i \leftarrow 0$
enquanto $i < k$ e $A[i] = 1$
    faça $A[i] \leftarrow 0$
        $i \leftarrow i + 1$
se $i < k$
    então $A[i] \leftarrow 1$

Custo:
número de bits invertidos = O(k)

Exemplo: Contador Binário

A 5	4	3	2	1	0	Custo Total
0	0	0	0	0	0	0
0	0	0	0	0	1	1
0	0	0	0	1	0	3
0	0	0	0	1	1	4
0	0	0	1	0	0	7
0	0	0	1	0	1	8
0	0	0	1	1	0	10
0	0	0	1	1	1	11
0	0	1	0	0	0	15
0	0	1	0	0	1	16

$k-1$		3	2	1	0

INCREMENT (A, k)

```
i ← 0
enquanto i < k e A[i] = 1
    faça A[i] ← 0
         i ← i + 1
se i < k
    então A[i] ← 1
```

Custo de uma chamada de INCREMENT: número de bits invertidos = O(k)

Exemplo: Contador Binário

- n chamadas do procedimento increment
  - n.O(k) = O(n.k).
- Mas quando se analisa a sequência de operações, observa-se que o custo total é menor:
  - A[0] é invertido n vezes
  - A[1] é invertido n/2 vezes
  - A[2] é invertido n/4 vezes …

$$\sum_{i=0}^{k-1} \left\lfloor \frac{n}{2^i} \right\rfloor < n \sum_{i=0}^{\infty} \frac{1}{2^i} = 2n = O(n)$$

Custo amortizado por operação é O(n) / n = O(1)

Método Contábil

- No método contábil atribui-se um custo fictício (amortizado) a cada operação, que pode ser maior ou menor que o custo real
  - $\hat{c}_i$: custo amortizado da operação i
  - c_i: real da operação i
- Em uma sequência de n operações a seguinte condição deve ser satisfeita: $$\sum_{i=1}^{n} \hat{c}_i \geq \sum_{i=1}^{n} c_i$$
- A estrutura de dados armazena o “saldo” das operações, que ajuda a “pagar” operações mais caras

Método Contábil

- Analogia:
  - Banco : para armazenar tempo
  - Cada operação: deposita ou armazena tempo

Custo amortizado = custo real + deposito - saque

∑ custo amortizado = ∑ custo real + (∑ depositos - ∑ saques)

= ∑ custo real + saldo do banco ≥ ∑ custo real

desde que saldo do banco ≥ 0

Exemplo: pilha

	Custo Real	Custo Amortizado
Push	1	2
Pop	1	0
Multipop	*min(n,k)*	0

- Cada operação push deixa um "crédito" de 1 que será usado pela operação pop ou multipop.
- Comecando com uma pilha vazia, a condição

$$\sum_{i=1}^{n} \hat{c}_i \geq \sum_{i=1}^{n} c_i$$

é satisfeita

Exemplo: pilha

- Para qualquer sequência de *n* operações Push, Pop e Multipop o **custo amortizado total é *2n = O(n)***
  - Assumindo pop/multipop após push

- Como o **custo amortizado é um limite superior do custo real**, considerando-se *n* operações, temos que **o custo real das operações também é *O(n)***

Exemplo: Contador Binário

A 5	4	3	2	1	0	Custo Total
0	0	0	0	0	0	0
0	0	0	0	0	1	1
0	0	0	0	1	0	3
0	0	0	0	1	1	4
0	0	0	1	0	0	7
0	0	0	1	0	1	8
0	0	0	1	1	0	10
0	0	0	1	1	1	11
0	0	1	0	0	0	15
0	0	1	0	0	1	16

$k-1$		3	2	1	0

INCREMENT (A, k)

```
i ← 0
enquanto i < k e A[i] = 1
    faça A[i] ← 0
         i ← i + 1
se i < k
    então A[i] ← 1
```

	Custo real:	Custo amortizado
0->1	1	2
1-> 0	1	0

Quanto é o custo de n operações?

Método Potencial

- Define-se uma função que representa a “Energia Potencial” acumulada na estrutura
  - Paralelo com método contábil
    (crédito associado a elemento na estrutura)
- Seja c_i uma operação e D_i o estado da estrutura após a aplicação de c_i em D_{i-1}
- A função Φ representa o potencial, que pode ser liberado para operações futuras
- Definimos o custo amortizado de uma operação como o custo real acrescido da mudança de potencial

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

Método Potencial

- Para uma sequência de n operações:

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

- Definindo um potencial Φ tal que $\Phi(D_n) \geq \Phi(D_o)$, temos que o custo amortizado vai ser um limite superior para o custo real

Exemplo: Pilha

- Vamos definir uma função potencial tal que Φ seja o número de itens na pilha. Logo
  - $\Phi(D_0) = 0$
  - $\Phi(D_i) \geq 0$, para todo i pois o número de elementos da pilha nunca é negativo
- Portanto o custo amortizado é um limite superior para o custo real

Exemplo: Pilha

Analisando o custo amortizado de cada operação:

- Push: a diferença de potencial causada pela operação em uma pilha com *s* elementos é $\Phi(D_i) - \Phi(D_{i-1}) = (s + 1) - s = 1$

  O custo da operação é portanto:

  $\hat{c}_i = c_i + \Phi(D_i) - \Phi(D_{i-1}) = 1 + 1 = 2$
- Pop: fazendo um raciocínio similar

  $\Phi(D_i) - \Phi(D_{i-1}) = (s - 1) - s = -1$

  $\hat{c}_i = c_i + \Phi(D_i) - \Phi(D_{i-1}) = 1 - 1 = 0$

Exemplo: Pilha

- <u>Multipop</u>: a operação multipop em uma pilha com s items remove $k' = min\ (s,k)$ items da pilha. Logo a diferença de potencial é
  $\Phi(D_i) - \Phi(D_{i-1}) = -k'$
  O custo da operação é portanto:
  $\hat{c}_i = c_i + \Phi(D_i) - \Phi(D_{i-1}) = k' - k' = 0$

- Logo, **o custo amortizado das 3 operações é O(1)**

Exemplo Pilha

- Em uma sequência de n operações quaisquer sobre a pilha, temos que o custo total amortizado é $n.O(1) = O(n)$
- Como o custo amortizado é um limite superior para o custo real, temos que o custo real também é O(n).