

Mais sobre Algoritmos Iterativos

Prof. André Vignatti

1 Resumo

atribuição	$\Theta(1)$
entrada da função	$\Theta(1)$
saída da função	$\Theta(1)$
desvio condicional (if)	tempo do teste mais $\Theta(\text{máximo dos dois branches})$
laço	soma de todas iterações do tempo de cada iteração

Juntar tudo isso usando a Regra da Soma e Regra do Produto.

Exceção - algoritmos recursivos.

1.1 Multiplicação

```
1 Algoritmo multiplica( $y, z$ )
2    $x \leftarrow 0$ 
3   enquanto  $z > 0$  faça
4     se  $z$  é ímpar então  $x \leftarrow x + y$ 
5      $y \leftarrow 2y$ 
6      $z \leftarrow \lfloor z/2 \rfloor$ 
7   retorna  $x$ 
```

Suponha que y e z tenham n bits.

- entrada e saída da função : $\Theta(1)$
- linha 2: $\Theta(1)$
- linhas 4, 5 e 6: $\Theta(1)$ cada vez que são executados
- laço custa $O(n)$ (é executado no máximo n vezes)

Portanto, a multiplicação leva tempo $O(n)$ (pela regra da soma e do produto)

1.2 Bubblesort

```
1 Algoritmo bubblesortA[1..n]
2   para  $i \leftarrow 1$  até  $n - 1$  faça
3     para  $j \leftarrow 1$  até  $n - i$  faça
4       se  $A[j] > A[j + 1]$  então
5         Troca  $A[j]$  com  $A[j + 1]$ 
```

- entrada e saída da função : $\Theta(1)$
- linha 5: $O(1)$
- if: $O(1)$ cada vez que é executado
- laço interno: $O(n)$
- laço externo: $O(\sum_{i=1}^{n-1} (n - i))$

$$O\left(\sum_{i=1}^{n-1} (n - i)\right) = O\left(n(n - 1) - \sum_{i=1}^{n-1} i\right) = O(n^2).$$

Portanto, no pior caso, o Bubblesort leva tempo $O(n^2)$.

De maneira parecida, pode-se provar que leva tempo $\Omega(n^2)$, concluindo então $\Theta(n^2)$.

1.3 Truque de Análise

Ao invés de usar o método passo-a-passo de análise:

- Identificar a **operação fundamental** usada no algoritmo, e observar que o tempo de execução é uma constante múltipla do número de operações fundamentais executadas (Vantagem: não precisa fazer uma análise linha-a-linha)
- Analisar o **número exato** dessa operação (Vantagem: trabalhar com números ao invés de notação assintótica)

1.4 Exemplo

No Bubblesort, a operação fundamental é a comparação (if).

Assim, tempo de execução será O do número de comparações.

- A linha 4 usa 1 comparação

- O laço interno usa $n - i$ comparações
- O laço externo usa $\sum_{i=1}^{n-1} (n - i)$ comparações

Então

$$\begin{aligned}\sum_{i=1}^{n-1} (n - i) &= n(n - 1) - \sum_{i=1}^{n-1} i \\ &= n(n - 1) - n(n - 1)/2 \\ &= n(n - 1)/2.\end{aligned}$$

1.5 Observações e Pegadinhas

O algoritmo de multiplicação leva tempo $O(n)$.

O que isso significa? Tenha cuidado com

- Suposições escondidas: n é o número de bits. Modelo RAM: +, -, * e / custam $\Theta(1)$.
- O algoritmo de multiplicação leva tempo $O(n^2)$ não é uma afirmação falsa!
- As constantes multiplicativas escondidas: podem fazer o algoritmo tornar-se ruim na prática.