

Princípios de Análise de Algoritmos

Marcelo K. Albertini

10 de Março de 2020

Avaliação matemática

A avaliação matemática

- uso de algoritmo em um **modelo computacional teórico**
 - Exemplo de modelo: “Máquina de Acesso Aleatório”
- simplificação para considerar somente os custos dominantes

Exemplo: avaliação de custos em um algoritmo

```
1 // Entrada: inteiro positivo "n"
2 // Saida: imprimir multiplicacoes de 0 a "n"
3 if (n > 10) // 1 vez, custo c1
4   System.out.println("Isto pode demorar..."); c2
5
6 for (int i = 0; i < n; i++) // n vezes, c3 + nc1 + nc4
7   for (int j = 0; j < i; j++) // m =  $\sum_{j=1}^n j$  vezes,
8     System.out.println(i * j); // c2 + c5
```

Custos

comparação c_1 , impressão c_2 , atribuição c_3 , incremento c_4 ,
multiplicação c_5

Exemplo: avaliação de custos em um algoritmo

```
1 // Entrada: inteiro positivo "n"
2 // Saida: imprimir multiplicacoes de 0 a "n"
3 if (n > 10) // 1 vez, custo c1
4   System.out.println("Isto pode demorar..."); c2
5
6 for (int i = 0; i < n; i++) // n vezes, c3 + nc1 + nc4
7   for (int j = 0; j < i; j++) // m = ∑j=1n j vezes,
8     System.out.println(i * j); // c2 + c5
```

Custos: pior caso – quando o algoritmo executar tudo

$$T(n) = c_1 + c_2 + (c_3 + nc_1 + nc_4) + (c_3 + mc_1 + mc_4) + m * (c_2 + c_5)$$

Exemplo: avaliação de custos em um algoritmo

Custos: pior caso – quando o algoritmo executar tudo

- $T(n) = c_1 + c_2 + (c_3 + nc_1 + nc_4) + (c_3 + mc_1 + mc_4) + m * (c_2 + c_5)$
- se todos custos forem considerados iguais a “c”:
- $T(n) = c + c + (c + nc + nc) + (c + mc + mc) + m * (c + c)$
- $T(n) = 2c + (c + 2nc) + (c + 2mc) + 2mc$
- $T(n) = 4c + 2nc + 4mc$

Exemplo: avaliação de custos em um algoritmo

Custo do algoritmo é

$$T(n) = 4c + 2nc + 4mc$$

Quanto é m ?

$$m = \sum_{j=1}^n j =$$

- $1 + 2 + 3 + 4 + 5 + \dots + (n - 3) + (n - 2) + (n - 1) + n =$
- $(n + 1) + (2 + (n - 1)) + (3 + (n - 2)) + \dots + (n + 1)$
- Somamos $n/2$ pares de números $(n + 1)$
 - $m = \frac{n}{2}(n + 1)$

Custo do algoritmo é

$$T(n) = 4c + 2nc + 4\left(\frac{n}{2}(n + 1)\right)c$$

Exemplo: avaliação de custos em um algoritmo

Custo do algoritmo é

- $T(n) = 4c + 2nc + 4\left(\frac{n}{2}(n+1)\right)c$
- $T(n) = 4c + 2nc + 2n(n+1)c$
- $T(n) = 4c + 2nc + 2n^2c + 2nc$
- $T(n) = 4c + 4cn + 2cn^2$

Qual é o custo dominante do algoritmo?

Complexidade de tempo

Custo do algoritmo de multiplicação de números

A função que descreve o custo do nosso algoritmo é

$$T(n) = 4c + 4cn + 2cn^2.$$

Complexidade de tempo

Para descrever a complexidade usamos somente o custo dominante: $2cn^2$.

Porquê?

Existe um número $n = a$ a partir do qual o custo $2ca^2$ é sempre superior a $4ca$, tornando-o pouco importante.

Ordem de crescimento

Dizemos que a complexidade de *pior caso* do nosso algoritmo é $O(n^2)$.

Exemplo: ordenação de transações bancárias

Problema

- Um banco tem 10 milhões clientes.
- Cada cliente faz em média 4 transações diárias.

Relatório

O banco precisa entregar um relatório diário para o Banco Central, com uma relação ordenada de todas as transações.

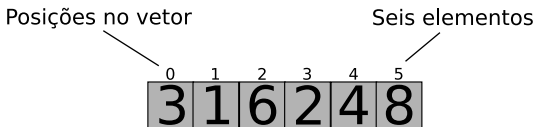
Infraestrutura

O computador do banco é dedicado ao problema e resolve operações lógico-aritméticas na taxa de 2^{30} (~ 1 bilhão) operações por segundo.

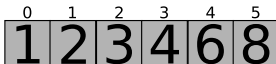
Exemplo: o problema da ordenação interna

ordenar em memória

- Pré-condições: vetor em **memória principal**, inicializado com elementos
- Pós-condições: vetor com elementos em ordem crescente (ou decrescente)



Como fica ordenado?



Exemplo: análise de um algoritmo de ordenação

Definição de ordenação

Sequência de comparações e trocas de posição entre elementos para obter vetor ordenado.

Análise

Quantas trocas, comparações (complexidade de **tempo**) e variáveis auxiliares (de espaço) são necessárias?

Exemplo: algoritmo de ordenação interna bubblesort

Como programar um algoritmo de ordenação simples?

Ideia

Comparar pares consecutivos de elementos e trocá-los de posição caso o primeiro seja maior que o segundo.

```
1 if (vetor[i] > vetor[i+1]) {  
2   aux = vetor[i];  
3   vetor[i] = vetor[i+1];  
4   vetor[i+1] = aux;  
5 }
```

Variável auxiliar **aux** é custo extra de memória .

Exemplo: algoritmo bubblesort simplificado

“Um dos piores algoritmos de ordenação desde o tempo dos Faraós.”

```
1 bubblesort (int vetor[], int nelem) {
2     int i, iteracao, aux;
3
4     /*controle do numero de iteracoes (n - 1)*/
5     for (iteracao = 0; iteracao < nelem-1; iteracao++){
6         /*repeticao interna, percorre vetor (n - 1)*/
7         for (i = 0; i < nelem - 1; i++){
8             if (vetor[i] > vetor[i+1]){
9                 /*necessaria uma troca*/
10                aux = vetor[i];
11                vetor[i] = vetor[i+1];
12                vetor[i+1] = aux;
13            }
14        }
15    }
16 }
```

Exemplo: análise de uma solução para o problema de ordenação de transações

Se o algoritmo é o Bubblesort, o banco:

- a) conseguirá tranquilamente fazer o relatório a tempo;
- b) deverá comprar um computador melhor;
- c) deverá contratar um programador melhor.

Exemplo: aplicação de análise de algoritmos ao caso bancário

Avaliação:

- Ordem do Bubblesort: $O(n^2)$
- Elementos a ordenar: número de transações $n = 4 \times 10^7$
- Operações necessárias: $(4 \times 10^7)^2$
- Capacidade do computador: 2^{30} operações por segundo
- Tempo necessário em minutos: $(4 \times 10^7)^2 / 2^{30}$

= 17 dias, 4 horas, 55 minutos e 16 segundos

Notação de complexidade

Função de complexidade (de tempo ou espaço)

Para um problema de tamanho n , buscamos uma função $f(n)$ que descreve o componente principal do nosso custo

Conflito de interesses

Frequentemente, quando a complexidade de tempo diminui, a complexidade de espaço aumenta

Exemplos de uso de notação assintótica

Casos

- Para “o custo pelo menos é...”, usar notação Ω
 - Se $t(n) = 2$ operações, então $t(n) = \Omega(1)$
i.e., $t(n) \geq c * 1$ para $n \geq 1$ e $c = 2$
- Para “o custo é até ...”, usar notação O
 - Se $t(n) = 2n + 1$ operações
então $t(n) = O(n)$, i.e., $t(n) \leq c * n$ para $c = 3$ e $n \geq 1$
- Para “o custo cresce na ordem de magnitude de ...”, usar notação Θ :
 - Se $t(n) = 3n^2 - n + 1$,
então $t(n) = \Theta(n^2)$
pois $t(n) = O(n^2)$, $t(n) \geq c_1 n^2$ e $t(n) \leq c_2 n^2$ para $c = 3$ e $n \geq 1$

