Aula 03: Análise de algoritmos — melhor caso, pior caso e caso médio

David Déharbe
Programa de Pós-graduação em Sistemas e Computação
Universidade Federal do Rio Grande do Norte
Centro de Ciências Exatas e da Terra
Departamento de Informática e Matemáica Aplicada

PPgSC

Plano da aula

Pior caso, melhor caso, caso médio, análise amortizada

Conclusão

Pior caso, melhor caso, caso médio

- A complexidade do algoritmo de *ordenação por inserção* depende não só do tamanho da entrada, mas também do valor desta entrada.
- A complexidade do algoritmo de *busca linear* depende não só do tamanho da entrada, mas também do valor desta entrada.

pior caso é a função que relaciona o tamanho da entrada n ao maior tempo de execução possível para tratar uma entrada de tamanho n.

melhor caso é a função que relaciona o tamanho da entrada n ao menor tempo de execução possível para tratar uma entrada de tamanho n.

caso médio é a função que relaciona o tamanho da entrada n ao tempo médio de execução possível para tratar uma entrada de tamanho n, *assumindo uma distribuição probabilística* das entradas possíveis.

PPgSC

Pior caso

Como calcular?

Determinar uma entrada, de tamanho n, tal que a operação básica é executada a maior quantidade de vezes possível.

Exemplo

LINEAR-SEARCH(A, v)

```
1  j = 1
2  while A[j] ≠ v and j ≤ length(A)
3        j = j + 1
4  if j ≤ length(A)
5        return j
6  else return NIL
```

No pior caso, $v \notin A$ e o número de vezes que a operação básica é executada é $n + 1$.

Melhor caso

Como calcular?

Determinar uma entrada, de tamanho n, tal que a operação básica é executada a menor quantidade de vezes possível.

Exemplo

LINEAR-SEARCH(A, v)

```
1  j = 1
2  while A[j] ≠ v and j ≤ length(A)
3        j = j + 1
4  if j ≤ length(A)
5        return j
6  else return NIL
```

No melhor caso, $v = A[1]$ e o número de vezes que a operação básica é executada é 1.

Complexidade média

Como calcular?

O cálculo é feito assumindo uma distribuição probabilística das entradas (ou classes de entradas) possíveis de tamanho n. É então realizada uma média ponderada do custo para aquela entrada (ou classe de entrada) com a probabilidade correspondente.

Exemplo

LINEAR-SEARCH(A, v)

```
1  j = 1
2  while A[j] ≠ v and j ≤ length(A)
3        j = j + 1
4  if j ≤ length(A)
5        return j
6  else return NIL
```

PPgSC

Complexidade média (busca linear)

Assume:

1. $0 \leq p \leq 1$ é a probabilidade de v estar em A,
2. p/n é a probabilidade de v estar em cada $A[i]$.

- O custo do algoritmo quando v está em $A[i]$ é i.
- O custo do algoritmo quando v não está em A é $n+1$.

A complexidade média é:

$$
\begin{aligned}
(1-p).(n+1) + \sum_{i=1}^{n} i.\frac{p}{n} &= (1-p).(n+1) + \frac{p}{n} \times (\sum_{i=1}^{n} i) \\
&= (1-p).(n+1) + \frac{p}{n} \times \frac{n.(n+1)}{2} \\
&= (1-p).(n+1) + \frac{p.(n+1)}{2} \\
&= \frac{(n+1).(2-p)}{2}
\end{aligned}
$$

PPgSC

Complexidade média (busca linear)

$$\frac{(n+1).(2-p)}{2}$$

Note que

- se $p = 1$, a busca é sempre bem-sucedida, e o valor $\frac{n+1}{2}$ faz sentido.
- se $p = 0$, a busca é sempre mal-sucedida, e o valor $n + 1$ faz sentido.

PPgSC

Considerações sobre essas medidas de complexidade

Pontos chaves

- A complexidade média é geralmente a mais importante, e a mais difícil de ser analisada.
- A complexidade no pior caso é também muito importante pois não raramente se aproxima da complexidade em média. Ela é essencial
- A complexidade no melhor caso não é tão importante, mas pode ser suficiente para *descartar* um algoritmo.
- A complexidade média não é a média da complexidade no pior caso e da complexidade no melhor caso.

PPgSC

Exercício

Como um algoritmo de ordenação qualquer pode ser alterado de tal forma que seu custo no melhor caso seja $n - 1$ para ordenar uma sequência de tamanho n?

Análise amortizada

- A complexidade amortizada considera uma série de execuções de um algoritmo sobre alguma *estrutura de dados*.
- Tem como objetivo determinar o custo médio por operação.
- Não necessita de nenhuma distribuição probabilística.

Métodos de resolução

- método agregado ←
- método do contador
- método do potencial

Método agregado

- Considere uma série de n operações em alguma estrutura de dados, n qualquer.
- Determina o custo $T(n)$ desta série de operações.
- O custo médio de cada operação é $T(n)/n$.
- Exemplo: contador binário

Método agregado: um exemplo

```
INCREMENT(A)
    // A é um arranjo de k bits
1   i = 0
2   while i < length[A] and A[i] = 1
3         A[i] = 0
4         i = i + 1
5   if i < length[A]
6
7         A[i] = 1
```

- O tamanho da entrada é k;
- O custo da execução da INCREMENT é proporcional ao número de bits invertidos.
- No pior caso, a complexidade é proporcional a k;
- Logo, uma sequência de n operações é proporcional a $n \times k$.
- Podemos prover uma análise mais precisa?

PPgSC

Método agregado: um exemplo

Em uma série de n execuções de INCREMENT:

- $A[0]$ é invertido n vezes;
- $A[1]$ é invertido $n/2$ vezes;
- $A[2]$ é invertido $n/4$ vezes;
- $A[k-1]$ é invertido $n/2^{k-1}$ vezes;

O custo total para n operações é

$$\begin{aligned} T(n) &= \sum_{i=1}^{\lfloor \lg n \rfloor} n/2^{i-1} \\ T(n) &< \sum_{i=1}^{\infty} n/2^{i-1} \\ T(n) &< n \times \sum_{i=1}^{\infty} 1/2^{i-1} = 2n \end{aligned}$$

Logo o custo amortizado de INCREMENT é $T(n)/n = 2$ constante.

Síntese

- A complexidade temporal tanto quanto a complexidade espacial são funções do tamanho da entrada.
- A complexidade temporal, ou complexidade, é determinado pelo número de vezes que a operação básica do algoritmo é executada.
- A complexidade de um algoritmo pode variar consideravelmente para entradas de mesmo tamanho. Nestes casos, deve-se distinguir melhor caso, pior caso, e complexidade média.
- O arcabouço teórico da análise de algoritmos baseia-se na noção de crescimento assintótico de funções.

PPgSC