

Complexidade Assintótica

Bruno César Ribas

Universidade Tecnológica Federal do Paraná - Campus Pato Branco

21 de setembro de 2018

Sumário

- 1 A Análise...
 - Análise de Complexidade de Algoritmos
- 2 Complexidade Assintótica
 - Notações
 - Notação O
- 3 E para o futuro?

O que é Análise de Complexidade de Algoritmos?

A Análise...

- 1 É o estudo dos comportamentos dos algoritmos para definir sua eficiência
- 2 Sendo que os aspectos mais analisados são o tempo de execução (eficiência do algoritmo) e espaço ocupado (eficiência da estrutura de dados)
- 3 É importante para a escolha de qual algoritmo usar dentre uma classe (família de algoritmos que resolvem um problema específico)

O que é Análise de Complexidade de Algoritmos?

A Análise...

- 1 O algoritmo mais imediato pode estar longe de ser o mais razoável em termos de eficiência;
- 2 O crescente avanço tecnológico ofusca a complexidade de tempo de um algoritmos?
- 3 Como se comportam os algoritmos de ordenação já estudados?

O que é Análise de Complexidade de Algoritmos?

A Análise...

- 1 É medido através da função de custo ou complexidade $f(n)$:
 - ▶ Sendo n uma variável que descreve o tamanho de instâncias do problema, exemplos:
 - ★ Problemas em grafos: número de vértices ou arestas
 - ★ Problemas em ordenação de vetores: tamanho do vetor
 - ★ Problemas de pesquisa em memória: número de chaves/registros
 - ★ Busca em texto: número de caracteres ou padrão de busca
 - ▶ É calculado para três cenários dependentes da entrada:
 - ★ Melhor caso: menor tempo de execução
 - ★ Pior caso: maior tempo de execução
 - ★ Caso médio: média dos tempos de execução

Exemplo: Análise de Complexidade de Algoritmos

A Análise...

- Considere a busca sequencial de um elemento de em um vetor
 - ▶ Melhor caso $f(n) = 1$: o elemento procurado é o primeiro consultado
 - ▶ Pior caso $f(n) = n$: o elemento procurado é o último consultado
 - ▶ Caso médio $f(n) = (n + 1)/2$
 - ★ Sendo p_i a probabilidade de encontrar o elemento na posição i
 - ★ E que cada posição do vetor tem o mesmo $p_i = 1/n$, $1 \leq i \leq n$
 - ★ $f(n)$ = soma do número de comparações vezes probabilidade

$$\begin{aligned}f(n) &= 1\left(\frac{1}{n}\right) + 2\left(\frac{1}{n}\right) + \dots + n\left(\frac{1}{n}\right) \\ &= \frac{1}{n}(1 + 2 + \dots + n) \\ &= \frac{1}{n}\left(\frac{n(n+1)}{2}\right) = \frac{(n+1)}{2}\end{aligned}$$

- ★ Ou seja, uma pesquisa bem-sucedida examina aproximadamente metade dos registros.

Tamanho de n x Análise de Complexidade de Algoritmos?

E o tamanho de n , pode interferir na complexidade de algoritmos?

Tamanho de $n \times$ Análise de Complexidade de Algoritmos?

Imagine a seguinte situação:

- Você se lembra de um amigo de infância e resolve procurá-lo no Facebook
- Insere o nome e encontra ele!
- Daí você pensa: como o Facebook achou tão rápido o meu amigo? Será que ele usa um vetor com todos os usuários?
- Será?! Quanto custaria a busca em um vetor com bilhões de elementos?
 - ▶ No pior caso $f(n) = n$, e n é bilhões!
 - ▶ Mas no caso médio é $f(n) = \frac{(n+1)}{2}$, ah!, é a metade! Mas é a metade de bilhões!
 - ▶ Nestes casos, estuda-se a complexidade assintótica da função custo.
- E o que é complexidade assintótica?!

Sumário

- 1 A Análise...
 - Análise de Complexidade de Algoritmos
- 2 Complexidade Assintótica
 - Notações
 - Notação O
- 3 E para o futuro?

Complexidade Assintótica

- É uma medição formal (matematicamente consistente) de se **calcular aproximadamente** a eficiência de algoritmos
- Medidas objetivas não são recomendadas pois
 - ▶ São dependentes de implementação e de fatores ambientais (cpu, barramentos, memória, etc.)
- Para algoritmos cuja entradas tendem para valores grandes
 - ▶ Descreve o crescimento de funções
 - ▶ Mede como $f(n)$ se comporta conforme n aumenta indefinidamente
- E como medimos/representamos esse comportamento assintótico?

Notações

Complexidade Assintótica

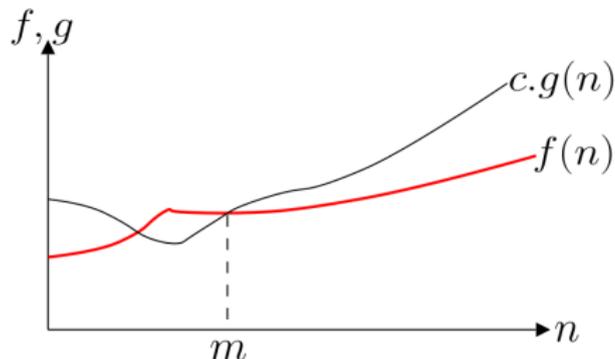
- Para representar a relação assintótica surgiu diversas notações
- A mais utilizada é a notação O
 - ▶ Para $f(n) = n^2 + 2n + 1$, sua complexidade em $O(n^2)$
 - ▶ A relação assintótica entre duas funções distintas $f(n)$ e $g(n)$ é:
 - ★ **Comparar a tendência de crescimento de $f(n)$ e $g(n)$**
 - ★ Para poder concluir que $f(n) = O(g(n))$
 - ★ Na notação O , $g(n)$ é o limite superior de $f(n)$
 - ★ $f(n)$ cresce, no máximo, tão rapidamente quanto $g(n)$
 - ★ $g(n)$ domina assintoticamente $f(n)$
 - ★ $g(n)$ é o **limite superior** para a taxa de crescimento de $f(n)$
 - ▶ Exemplo: busca sequencial
 - ★ $O(n)$, custo cresce no máximo conforme n cresce

Notação O

Complexidade Assintótica

- Formalmente, define-se:

- ▶ Uma função $f(n) = O(g(n))$
- ▶ Se $f(n) \leq c.g(n)$
- ▶ Para algum c positivo e para todo n suficiente grande
- ▶ Ou seja,
- ▶ Existem duas constantes positivas c e m
- ▶ Tais que, $f(n) \leq c.g(n)$
- ▶ Para todo $n \geq m$



Notação O: Valores Comuns

Complexidade Assintótica

- $f(n) = O(1)$:
 - ▶ Complexidade constante
 - ▶ Independem do tamanho de n
 - ▶ Possui instruções executadas um número fixo de vezes
- $f(n) = O(n)$:
 - ▶ Complexidade linear
 - ▶ Caracterizam-se por realizar um pequeno trabalho sobre cada elemento da entrada
 - ▶ Vetor

Notação O: Valores Comuns

Complexidade Assintótica

- $f(n) = O(\log n)$:
 - ▶ Complexidade logarítmica
 - ▶ Tipicamente resolvem um problema transformando-o em problemas menores
 - ▶ Busca Binária
- $f(n) = O(n \log n)$:
 - ▶ Complexidade logarítmica
 - ▶ Caracterizam-se por resolver um problema quebrando em problemas menores, resolvendo cada um deles independentemente e depois juntando as soluções
 - ▶ MergeSort

Notação O: Valores Comuns

Complexidade Assintótica

- $f(n) = O(n^2)$:
 - ▶ Complexidade quadrática
 - ▶ Caracterizam-se pelo processamento dos dados em pares, muitas vezes com vários aninhamentos
 - ▶ BubbleSort, ou simplesmente...

Notação O: Valores Comuns

Complexidade Assintótica

- $f(n) = O(n^2)$:
 - ▶ Complexidade quadrática
 - ▶ Caracterizam-se pelo processamento dos dados em pares, muitas vezes com vários aninhamentos
 - ▶ BubbleSort, ou simplesmente...

Entrada: dimensão da matriz n

Saída: variável m com a matriz lida

```
1 para ( $i=0; i<n; i++$ ) faça
2   | para ( $j=0; j<n; j++$ ) faça
3   |   |  $m[i,j] \leftarrow lernumero()$ 
```

Notação O: Valores Comuns

Complexidade Assintótica

- $f(n) = O(n^2)$:
 - ▶ Complexidade quadrática
 - ▶ Caracterizam-se pelo processamento dos dados em pares, muitas vezes com vários aninhamentos
 - ▶ BubbleSort, ou simplesmente...
- $f(n) = O(n^3)$:
 - ▶ Complexidade cúbica
 - ▶ Eficientes apenas para pequenos problemas
 - ▶ Floyd Warshall : distância entre nós, de todos para todos, de um grafo
 - ▶ Algoritmo Trivial de Multiplicação de Matrizes

Entrada: Matriz $m1$ e $m2$

Saída: Matriz $m3$ com multiplicação $m1 \times m2$

```
1 para (i=0;i<n;i++) faça
2   para (j=0;j<n;j++) faça
3     para (k=0;k<n;k++) faça
4       |  m3[i,j] ← m3[i,j] + m1[i,k] x m2[k,j]
```

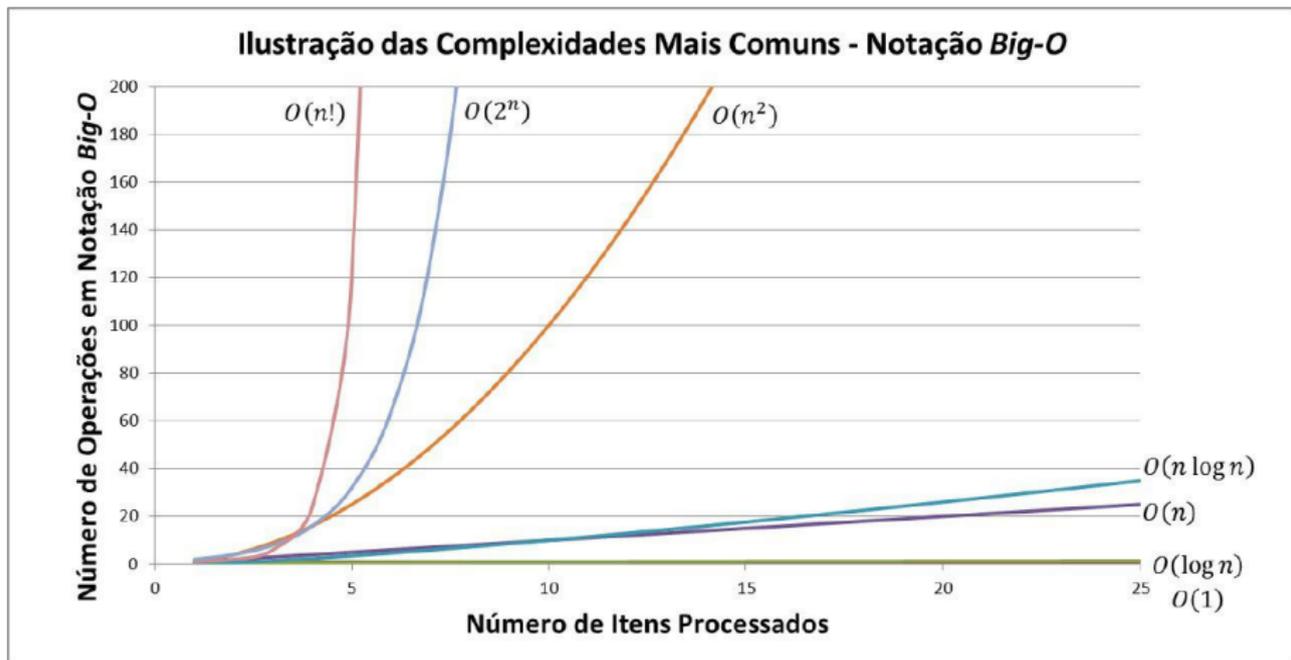
Notação O: Valores Comuns

Complexidade Assintótica

- $f(n) = O(2^n)$:
 - ▶ Complexidade exponencial
 - ▶ Resultantes de problemas resolvidos por força bruta
- $f(n) = O(n!)$:
 - ▶ Complexidade fatorial
 - ▶ Resultantes de problemas resolvidos por força bruta

Notação O: Valores Comuns

Complexidade Assintótica



Notação O: Valores Comuns

Complexidade Assintótica

- Ressalta-se que algoritmos de complexidade polinomial e os de complexidade exponencial tem significativa distinção quando o tamanho de n cresce
 - ▶ Um problema é considerado b em resolvido quando existe um algoritmo polinomial para resolvê-lo
- Para exemplificar algumas complexidades, pode-se citar a classe de problema de ordenação, que possui algoritmos e complexidades no caso médio como:
 - ▶ Selection, Insertion e Bubble: $O(n^2)$
 - ▶ Merge, Heap e Quick: $O(n \log n)$

Sumário

- 1 A Análise...
 - Análise de Complexidade de Algoritmos
- 2 Complexidade Assintótica
 - Notações
 - Notação O
- 3 E para o futuro?

E para o futuro?

- Existem outras notações?
 - ▶ Sim! Há a notação Ω (omega), 'o', ω (omegazinho), e θ (theta) que serão vistas em aulas futuras.
 - ▶ O que diferem entre elas? A referência de comparação das funções $f(n)$ e $g(n)$: “cresce no máximo tão rápido quanto”, “cresce no mínimo tão lento quanto”, “cresce mais lentamente que”, “cresce mais rapidamente que”, “cresce tão rápido quanto”.
- Para Algoritmos 2:
 - ▶ Técnicas de análise de complexidade, mais exemplos e exercícios
- Desafio - faça hoje: faça um algoritmo que resolva um problema proposto e calcule a sua complexidade:
 - ▶ Problema "Número Proibido" no spoj.
 - ▶ <https://br.spoj.com/problems/PROIBIDO/>

E para o futuro?: Descrição do Problema

Os números proibidos são números que possuem alguma representação problemática, por exemplo, número do azar, de algo ruim, e até números que são senhas do governo.

O número proibido mais conhecido é um número primo¹ que foi descoberto em 2001 e representa o arquivo binário da versão compactada do código C que implementa o algoritmo DeCSS, que pode ser utilizado para lograr o sistema de proteção do DVD.

Luan, um rapaz que tem muito receio de ser procurado por agências espãs internacionais coletou um conjunto de números ilegais e está filtrando esses números de todos os seus arquivos no computador.

Infelizmente, Luan ainda não sabe programar muito bem e pediu a sua ajuda para implementar um programa que receba um conjunto de números ilegais e responda se um outro conjunto de números fazem parte dos números ilegais.

continua...

¹http://en.wikipedia.org/wiki/Illegal_prime

E para o futuro?: Descrição do Problema

Entrada

A entrada é composta por um único caso teste que possui diversas linhas. A primeira linha possui um número N ($1 \leq N \leq 140000$) que representa a quantidade de números proibidos existentes. A segunda linha do caso de teste possui N números P_i ($0 \leq P_i \leq 231$) representando os números proibidos.

Depois existirão diversas linhas contendo um único número que se quer saber se é proibido ou não.

A entrada termina em EOF.

Saída

Para cada número da consulta deve-se imprimir uma única linha contendo a palavra sim se o número for proibido, ou não caso o número não seja proibido.

Detalhes e restrições:

<https://br.spoj.com/problems/PROIBIDO/>

Submissão:

<https://moj.naquadah.com.br/cgi-bin/index.sh>

Complexidade Assintótica

Bruno César Ribas

Universidade Tecnológica Federal do Paraná - Campus Pato Branco

21 de setembro de 2018