Algoritmos, Fundamentos Matemáticos

INF 332 - Projeto e Análise de Algoritmos

José Elias C. Arroyo

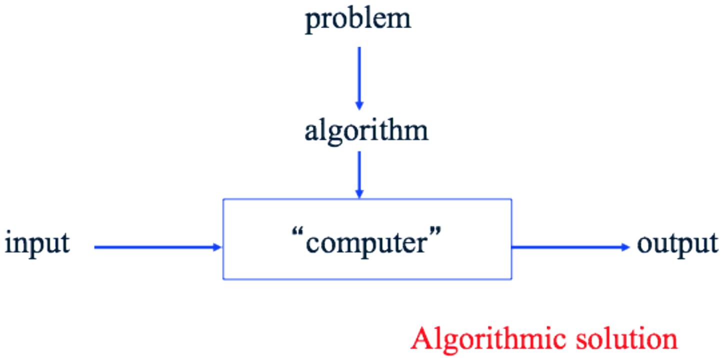
Departamento de Informática
Universidade Federal de Viçosa
jarroyo@ufv.br

INF 332 - 2022/2

Outline

Algoritmo

Um algoritmo é um **processo computacional bem definido** que recebe uma **entrada válida**, processa essa entrada e produz uma **saída correta**.



Algoritmo

Em geral, um algoritmo é uma **descrição passo a passo**, uma **receita**, um **procedimento**, **rotina**, **técnica**, ou **método**, para resolver um problema em um **tempo finito.**

Todo algoritmo possui os seguintes requisitos:

- A **entrada** deve ser **valida**;
- Deve **terminar** após um número finito de passos;
- Os *passos* devem ser **simples**, **bem definidos**, **não ambíguos** e executáveis computacionalmente;
- A **saída** produzida deve ser **correta.**

Algoritmo

- Um algoritmo pode ser **representado de várias formas** (diferentes **pseudocódigos** e liguagens de programação).
- Para resolver um problema podem existir **vários algoritmos** com diferentes **tempos de execução** e **consumo de memória**.

Máximo Divisor Comum (MDC)

Problema do MDC

Encontrar mdc(m, n), o **maior divisor comum** de m e n, sendo que m e n são não negativos e um deles é diferente de zero.

Exemplos

- mdc(60,24) = 12
- mdc(60,0) = 60
- mdc(0,0) = ?

Algoritmo de Euclides

Para calcular o mdc(m, n), o algoritmo de Euclides usa a seguinte fórmula recursiva:

mdc(m,n) = mdc(n, m mod n), para $n > 0$.

mdc(m,0) = m.

(m e n são não negativos e um deles é diferente de zero).

Algoritmo de Euclides

Algoritmo de Euclides

Para calcular o mdc(m, n), o algoritmo de Euclides usa a seguinte fórmula recursiva:

mdc(m,n) = mdc(n, m mod n), para $n > 0$.

mdc(m,0) = m.

(m e n são não negativos e um deles é diferente de zero).

Exemplo

mdc(60,24) = mdc(24, 60 mod 24) = mdc(24,12) =

= mdc(12, 24 mod 12) = mdc(12,0) = 12

Descrições do Algoritmo de Euclides

Algoritmo de Euclides

1. Se $n = 0$, **retorne** m; caso contrário, vá para passo 2
2. **Divida** m por n e determine o **resto** r (ou seja, $r \leftarrow m \bmod n$)
3. Substitua m por n e n por r (ou seja, $m \leftarrow n$ e $n \leftarrow r$)
   Vá para o passo 1.

Descrições do Algoritmo de Euclides

Algoritmo de Euclides

1. Se $n = 0$, **retorne** m; caso contrário, vá para passo 2
2. **Divida** m por n e determine o **resto** r (ou seja, $r \leftarrow m \bmod n$)
3. Substitua m por n e n por r (ou seja, $m \leftarrow n$ e $n \leftarrow r$)
   Vá para o passo 1.

Algoritmo de Euclides

Euclides (m, n)
    while $n \neq 0$ **do**
        $r \leftarrow m \bmod n$
        $m \leftarrow n$
        $n \leftarrow r$
    return m

Descrições do Algoritmo de Euclides

Algoritmo de Euclides

1. Se $n = 0$, **retorne** m; caso contrário, vá para passo 2
2. **Divida** m por n e determine o **resto** r (ou seja, $r \leftarrow m \bmod n$)
3. Substitua m por n e n por r (ou seja, $m \leftarrow n$ e $n \leftarrow r$)
   Vá para o passo 1.

Algoritmo de Euclides

Euclides (m, n)
    while $n \neq 0$ **do**
        $r \leftarrow m \bmod n$
        $m \leftarrow n$
        $n \leftarrow r$
    return m

O algoritmo de Euclides sempre termina?

Método Ingênuo para Calcular mdc(m,n)

Verificação Consecutiva de Inteiros

1. Determine $t = \min(m, n)$

   #Verificar se t é divisor de m e n:

2. Divida m por t. Se o resto for 0, vá para o passo 3; caso contrário, vá para o passo 4.
3. Divida n por t. Se o resto for 0, **retorne** t; caso contrário, vá para o passo 4.
4. Faça $t = t - 1$ e vá para o passo 2.

Método Ingênuo para Calcular mdc(m,n)

Verificação Consecutiva de Inteiros

1. Determine $t = \min(m, n)$

   #Verificar se t é divisor de m e n:

2. Divida m por t. Se o resto for 0, vá para o passo 3; caso contrário, vá para o passo 4.
3. Divida n por t. Se o resto for 0, **retorne** t; caso contrário, vá para o passo 4.
4. Faça $t = t - 1$ e vá para o passo 2.

Exemplo: calcular mdc(60, 24)

$t = min(60, 24) = 24$
$t = 24$ não é divisor de 60
$t = 23$ não é divisor de 60
....
$t = 12$ **é divisor** de 60 e 24. $\Rightarrow mdc(60, 24) = 12$.

Método Ingênuo para Calcular mdc(m,n)

Verificação Consecutiva de Inteiros

1. Determine $t = \min(m, n)$

   Verificar se t é divisor de m e n:
2. Divida m por t. Se o resto for 0, vá para o passo 3; caso contrário, vá para o passo 4.
3. Divida n por t. Se o resto for 0, **retorne** t; caso contrário, vá para o passo 4.
4. Faça $t = t - 1$ e vá para o passo 2.

O algoritmo está correto?

O algoritmo, como apresentado não funciona quando uma das entradas é **zero**.
Isto ilustra a importância de especificar explicitamente e cuidadosamente o **dominio das entradas** do algoritmo.

Método Aprendido na Escola para Calcular mdc(m,n)

Método da Fatorização

1. Encontre os fatores primos de m
2. Encontre os fatores primos de n
3. Encontre todos os fatores primos em comum
4. Calcule o **produto de todos os fatores primos em comum** e retorne como o resultado de mdc(m,n)

Método Aprendido na Escola para Calcular mdc(m,n)

Método da Fatorização

1. Encontre os fatores primos de *m*
2. Encontre os fatores primos de *n*
3. Encontre todos os fatores primos em comum
4. Calcule o **produto de todos os fatores primos em comum** e retorne como o resultado de mdc(m,n)

Exemplo

mdc(60, 24) = ?
$60 = 2 \cdot 2 \cdot 3 \cdot 5$
$24 = 2 \cdot 2 \cdot 2 \cdot 3$
Fatores primos comuns: $2 \cdot 2 \cdot 3 = 12 = mdc(60, 24)$

Método Aprendido na Escola para Calcular mdc(m,n)

Método da Fatorização

1. Encontre os fatores primos de m
2. Encontre os fatores primos de n
3. Encontre todos os fatores primos em comum
4. Calcule o **produto de todos os fatores primos em comum** e retorne o resultado como mdc(m,n)

Pergunta

Os passos do algoritmo (Método da Fatorização) estão bem definidos?

Método Aprendido na Escola para Calcular mdc(m,n)

Método da Fatorização

1. Encontre os fatores primos de m
2. Encontre os fatores primos de n
3. Encontre todos os fatores primos em comum
4. Calcule o **produto de todos os fatores primos em comum** e retorne o resultado como mdc(m,n)

Pergunta

Os passos do algoritmo (Método da Fatorização) estão bem definidos?

Os passos 1, 2 e 3 ***não*** *estão bem definidos.*

Por que Estudar Algoritmos?

Importância teórica

- Parte central da Ciência da Computação

Importância prática

- Algoritmos eficientes são usados para resolver diferentes problemas da vida real:
  - Projetos de genomas,
  - Rede mundial de computadores,
  - Sistemas de informação geográfica,
  - Comercio eletrônico,
  - Planejamento da produção nas industrias,
  - Logística de distribuição, Etc.

Dois problemas principais relacionados a algoritmos

- Como projetar algoritmos
- Como analisar a eficiência de algoritmos

Técnicas de Projeto de Algoritmos

- Força bruta
- Dividir para conquistar
- Técnica gulosa
- Programação dinâmica
- *Backtracking* e *Branch and bound*
- Algoritmos Aproximados

O algoritmo é bom?

- Eficiência de tempo
- Eficiência de espaço

Existe um algoritmo melhor para resolver o problema?

- limite inferior
- otimalidade

Fundamentos Matemáticos para Análise de Algoritmos

Somatórios - Definição

- Operador matemático que permite representar de forma sucinta somas arbitrariamente longas.
- Representado pela letra grega Σ (sigma), e definido como:

$$\sum_{i=m}^{n} x_i = x_m + x_{m+1} + x_{m+2} + \cdots + x_{n-1} + x_n$$

- onde
  - i: variável índice do somatório
  - m: valor inicial de i, chamado limite inferior
  - n: valor final de i, chamado limite superior ($n \geqslant m$)

Somatórios - exemplos

Exemplo #1

$$\sum_{i=1}^{10} i$$

Exemplo #2

$$\frac{\sum_{i=1}^{n} x_i}{n}$$

Exemplo #3

$$\sum_{i=1}^{n} (2i - 1)$$

Somatórios - exemplos

Exemplo #1 - Soma dos 10 primeiros números naturais

$$\sum_{i=1}^{10} i = 1 + 2 + 3 + \cdots + 9 + 10$$

Exemplo #2

$$\frac{\sum_{i=1}^{n} x_i}{n}$$

Exemplo #3

$$\sum_{i=1}^{n} (2i - 1)$$

Somatórios - exemplos

Exemplo #1 - Soma dos 10 primeiros números naturais

$$\sum_{i=1}^{10} i = 1 + 2 + 3 + \cdots + 9 + 10$$

Exemplo #2

$$\frac{\sum_{i=1}^{n} x_i}{n}$$

Exemplo #3

$$\sum_{i=1}^{n} (2i - 1)$$

Somatórios - exemplos

Exemplo #1 - Soma dos 10 primeiros números naturais

$$\sum_{i=1}^{10} i = 1 + 2 + 3 + \cdots + 9 + 10$$

Exemplo #2 - Média de *n* valores

$$\frac{\sum_{i=1}^{n} x_i}{n} = \frac{x_1 + x_2 + \cdots + x_n}{n}$$

Exemplo #3

$$\sum_{i=1}^{n} (2i - 1)$$

Somatórios - exemplos

Exemplo #1 - Soma dos 10 primeiros números naturais

$$\sum_{i=1}^{10} i = 1 + 2 + 3 + \cdots + 9 + 10$$

Exemplo #2 - Média de n valores

$$\frac{\sum_{i=1}^{n} x_i}{n} = \frac{x_1 + x_2 + \cdots + x_n}{n}$$

Exemplo #3

$$\sum_{i=1}^{n} (2i - 1)$$

Somatórios - exemplos

Exemplo #1 - Soma dos 10 primeiros números naturais

$$\sum_{i=1}^{10} i = 1 + 2 + 3 + \cdots + 9 + 10$$

Exemplo #2 - Média de *n* valores

$$\frac{\sum_{i=1}^{n} x_i}{n} = \frac{x_1 + x_2 + \cdots + x_n}{n}$$

Exemplo #3 - Soma dos *n* primeiros ímpares positivos

$$\sum_{i=1}^{n} (2i - 1) = 1 + 3 + 5 + 7 + \cdots + (2n - 1)$$

Somatórios importantes em Análise de Algoritmos

- $\sum_{i=1}^{n} 1$
- $\sum_{i=k}^{u} 1$
- $\sum_{i=1}^{n} i$
- $\sum_{i=1}^{n} i^2$
- $\sum_{i=1}^{n} i^k$

Somatórios importantes em Análise de Algoritmos

- $\sum_{i=1}^{n} 1 = 1 + 1 + \cdots + 1 = n$
- $\sum_{i=k}^{u} 1$
- $\sum_{i=1}^{n} i$
- $\sum_{i=1}^{n} i^2$
- $\sum_{i=1}^{n} i^k$

Somatórios importantes em Análise de Algoritmos

- $\displaystyle\sum_{i=1}^{n} 1 = 1 + 1 + \cdots + 1 = n$
- $\displaystyle\sum_{i=k}^{u} 1$
- $\displaystyle\sum_{i=1}^{n} i$
- $\displaystyle\sum_{i=1}^{n} i^2$
- $\displaystyle\sum_{i=1}^{n} i^k$

Somatórios importantes em Análise de Algoritmos

- $\sum_{i=1}^{n} 1 = 1 + 1 + \cdots + 1 = n$
- $\sum_{i=k}^{u} 1 = 1 + 1 + \cdots + 1 = u - k + 1$
- $\sum_{i=1}^{n} i$
- $\sum_{i=1}^{n} i^2$
- $\sum_{i=1}^{n} i^k$

Somatórios importantes em Análise de Algoritmos

- $\sum_{i=1}^{n} 1 = 1 + 1 + \cdots + 1 = n$
- $\sum_{i=k}^{u} 1 = 1 + 1 + \cdots + 1 = u - k + 1$
- $\sum_{i=1}^{n} i$
- $\sum_{i=1}^{n} i^2$
- $\sum_{i=1}^{n} i^k$

Somatórios importantes em Análise de Algoritmos

- $\sum_{i=1}^{n} 1 = 1 + 1 + \cdots + 1 = n$
- $\sum_{i=k}^{u} 1 = 1 + 1 + \cdots + 1 = u - k + 1$
- $\sum_{i=1}^{n} i = 1 + 2 + \cdots + n = \frac{n(n+1)}{2}$
- $\sum_{i=1}^{n} i^2$
- $\sum_{i=1}^{n} i^k$

Somatórios importantes em Análise de Algoritmos

- $\sum_{i=1}^{n} 1 = 1 + 1 + \cdots + 1 = n$
- $\sum_{i=k}^{u} 1 = 1 + 1 + \cdots + 1 = u - k + 1$
- $\sum_{i=1}^{n} i = 1 + 2 + \cdots + n = \frac{n(n+1)}{2}$
- $\sum_{i=1}^{n} i^2$
- $\sum_{i=1}^{n} i^k$

Somatórios importantes em Análise de Algoritmos

- $\sum_{i=1}^{n} 1 = 1 + 1 + \cdots + 1 = n$
- $\sum_{i=k}^{u} 1 = 1 + 1 + \cdots + 1 = u - k + 1$
- $\sum_{i=1}^{n} i = 1 + 2 + \cdots + n = \frac{n(n+1)}{2}$
- $\sum_{i=1}^{n} i^2 = 1^2 + 2^2 + \cdots + n^2 = \frac{n(n+1)(2n+1)}{6}$
- $\sum_{i=1}^{n} i^k$

Somatórios importantes em Análise de Algoritmos

- $\displaystyle\sum_{i=1}^{n} 1 = 1 + 1 + \cdots + 1 = n$
- $\displaystyle\sum_{i=k}^{u} 1 = 1 + 1 + \cdots + 1 = u - k + 1$
- $\displaystyle\sum_{i=1}^{n} i = 1 + 2 + \cdots + n = \frac{n(n+1)}{2}$
- $\displaystyle\sum_{i=1}^{n} i^2 = 1^2 + 2^2 + \cdots + n^2 = \frac{n(n+1)(2n+1)}{6}$
- $\displaystyle\sum_{i=1}^{n} i^k$

- $\displaystyle\sum_{i=1}^{n} 1 = 1 + 1 + \cdots + 1 = n$
- $\displaystyle\sum_{i=k}^{u} 1 = 1 + 1 + \cdots + 1 = u - k + 1$
- $\displaystyle\sum_{i=1}^{n} i = 1 + 2 + \cdots + n = \frac{n(n+1)}{2}$
- $\displaystyle\sum_{i=1}^{n} i^2 = 1^2 + 2^2 + \cdots + n^2 = \frac{n(n+1)(2n+1)}{6}$
- $\displaystyle\sum_{i=1}^{n} i^k = 1^k + 2^k + \cdots + n^k$

Somatórios importantes em Análise de Algoritmos

- $$\sum_{i=1}^{n} 1 = 1 + 1 + \cdots + 1 = n$$
- $$\sum_{i=k}^{u} 1 = 1 + 1 + \cdots + 1 = u - k + 1$$
- $$\sum_{i=1}^{n} i = 1 + 2 + \cdots + n = \frac{n(n+1)}{2}$$
- $$\sum_{i=1}^{n} i^2 = 1^2 + 2^2 + \cdots + n^2 = \frac{n(n+1)(2n+1)}{6}$$
- $$\sum_{i=1}^{n} i^k = 1^k + 2^k + \cdots + n^k \approx \frac{1}{k+1} n^{k+1}$$

Somatórios importantes em Análise de Algoritmos

- $\sum_{i=0}^{n} 2^i$
- $\sum_{i=0}^{n} a^i$
- $\sum_{i=1}^{n} i2^i$
- $\sum_{i=1}^{n} \frac{1}{i}$
- $\sum_{i=1}^{n} \log_2 i$

1

Somatórios importantes em Análise de Algoritmos

- $\sum_{i=0}^{n} 2^i = 1 + 2 + 4 + \cdots + 2^n = 2^{n+1} - 1$
- $\sum_{i=0}^{n} a^i$
- $\sum_{i=1}^{n} i2^i$
- $\sum_{i=1}^{n} \frac{1}{i}$
- $\sum_{i=1}^{n} \log_2 i$

1

Somatórios importantes em Análise de Algoritmos

- $\sum_{i=0}^{n} 2^i = 1 + 2 + 4 + \cdots + 2^n = 2^{n+1} - 1$
- $\sum_{i=0}^{n} a^i$
- $\sum_{i=1}^{n} i2^i$
- $\sum_{i=1}^{n} \frac{1}{i}$
- $\sum_{i=1}^{n} \log_2 i$

1

Somatórios importantes em Análise de Algoritmos

- $\sum_{i=0}^{n} 2^i = 1 + 2 + 4 + \cdots + 2^n = 2^{n+1} - 1$
- $\sum_{i=0}^{n} a^i = 1 + a + a^2 \cdots + a^n = \dfrac{a^{n+1} - 1}{a - 1}$
- $\sum_{i=1}^{n} i2^i$
- $\sum_{i=1}^{n} \dfrac{1}{i}$
- $\sum_{i=1}^{n} \log_2 i$

1

- $\sum_{i=0}^{n} 2^i = 1 + 2 + 4 + \cdots + 2^n = 2^{n+1} - 1$
- $\sum_{i=0}^{n} a^i = 1 + a + a^2 \cdots + a^n = \dfrac{a^{n+1} - 1}{a - 1}$, para $a \neq 1$ (PG)
- $\sum_{i=1}^{n} i2^i$
- $\sum_{i=1}^{n} \frac{1}{i}$
- $\sum_{i=1}^{n} \log_2 i$

1

Somatórios importantes em Análise de Algoritmos

- $\sum_{i=0}^{n} 2^i = 1 + 2 + 4 + \cdots + 2^n = 2^{n+1} - 1$
- $\sum_{i=0}^{n} a^i = 1 + a + a^2 \cdots + a^n = \dfrac{a^{n+1} - 1}{a - 1}$, para $a \neq 1$ (PG)
- $\sum_{i=1}^{n} i2^i$
- $\sum_{i=1}^{n} \frac{1}{i}$
- $\sum_{i=1}^{n} \log_2 i$

1

Somatórios importantes em Análise de Algoritmos

- $\displaystyle\sum_{i=0}^{n} 2^i = 1 + 2 + 4 + \cdots + 2^n = 2^{n+1} - 1$
- $\displaystyle\sum_{i=0}^{n} a^i = 1 + a + a^2 \cdots + a^n = \frac{a^{n+1} - 1}{a - 1}$, para $a \neq 1$ (PG)
- $\displaystyle\sum_{i=1}^{n} i2^i = 1 \times 2 + 2 \times 2^2 + \cdots + n \times 2^n = (n-1)2^{n+1} + 2$
- $\displaystyle\sum_{i=1}^{n} \frac{1}{i}$
- $\displaystyle\sum_{i=1}^{n} \log_2 i$

1

- $\sum_{i=0}^{n} 2^i = 1 + 2 + 4 + \cdots + 2^n = 2^{n+1} - 1$
- $\sum_{i=0}^{n} a^i = 1 + a + a^2 \cdots + a^n = \frac{a^{n+1} - 1}{a - 1}$, para $a \neq 1$ (PG)
- $\sum_{i=1}^{n} i2^i = 1 \times 2 + 2 \times 2^2 + \cdots + n \times 2^n = (n-1)2^{n+1} + 2$
- $\sum_{i=1}^{n} \frac{1}{i}$
- $\sum_{i=1}^{n} \log_2 i$

1

- $\sum_{i=0}^{n} 2^i = 1 + 2 + 4 + \cdots + 2^n = 2^{n+1} - 1$
- $\sum_{i=0}^{n} a^i = 1 + a + a^2 \cdots + a^n = \frac{a^{n+1} - 1}{a - 1}$, para $a \neq 1$ (PG)
- $\sum_{i=1}^{n} i2^i = 1 \times 2 + 2 \times 2^2 + \cdots + n \times 2^n = (n-1)2^{n+1} + 2$
- $\sum_{i=1}^{n} \frac{1}{i} = \frac{1}{1} + \frac{1}{2} + \cdots + \frac{1}{n}$
- $\sum_{i=1}^{n} \log_2 i$

1

- $\sum_{i=0}^{n} 2^i = 1 + 2 + 4 + \cdots + 2^n = 2^{n+1} - 1$
- $\sum_{i=0}^{n} a^i = 1 + a + a^2 \cdots + a^n = \frac{a^{n+1} - 1}{a - 1}$, para $a \neq 1$ (PG)
- $\sum_{i=1}^{n} i2^i = 1 \times 2 + 2 \times 2^2 + \cdots + n \times 2^n = (n-1)2^{n+1} + 2$
- $\sum_{i=1}^{n} \frac{1}{i} = \frac{1}{1} + \frac{1}{2} + \cdots + \frac{1}{n} \approx \ln n + \gamma$, onde $\gamma = 0,5772...$[1]
- $\sum_{i=1}^{n} \log_2 i$

[1] chamada série harmônica, e γ constante de Euler-Mascheroni

Somatórios importantes em Análise de Algoritmos

- $\sum_{i=0}^{n} 2^i = 1 + 2 + 4 + \cdots + 2^n = 2^{n+1} - 1$
- $\sum_{i=0}^{n} a^i = 1 + a + a^2 \cdots + a^n = \frac{a^{n+1} - 1}{a - 1}$, para $a \neq 1$ (PG)
- $\sum_{i=1}^{n} i2^i = 1 \times 2 + 2 \times 2^2 + \cdots + n \times 2^n = (n-1)2^{n+1} + 2$
- $\sum_{i=1}^{n} \frac{1}{i} = \frac{1}{1} + \frac{1}{2} + \cdots + \frac{1}{n} \approx \ln n + \gamma$, onde $\gamma = 0,5772...$[1]
- $\sum_{i=1}^{n} \log_2 i$

[1] chamada série harmônica, e γ constante de Euler-Mascheroni

Somatórios importantes em Análise de Algoritmos

- $\sum_{i=0}^{n} 2^i = 1 + 2 + 4 + \cdots + 2^n = 2^{n+1} - 1$
- $\sum_{i=0}^{n} a^i = 1 + a + a^2 \cdots + a^n = \frac{a^{n+1} - 1}{a - 1}$, para $a \neq 1$ (PG)
- $\sum_{i=1}^{n} i2^i = 1 \times 2 + 2 \times 2^2 + \cdots + n \times 2^n = (n-1)2^{n+1} + 2$
- $\sum_{i=1}^{n} \frac{1}{i} = \frac{1}{1} + \frac{1}{2} + \cdots + \frac{1}{n} \approx \ln n + \gamma$, onde $\gamma = 0,5772...$[1]
- $\sum_{i=1}^{n} \log_2 i = \log_2 1 + \log_2 2 + \cdots + \log_2 n$

[1] chamada série harmônica, e γ constante de Euler-Mascheroni

Somatórios importantes em Análise de Algoritmos

- $$\sum_{i=0}^{n} 2^i = 1 + 2 + 4 + \cdots + 2^n = 2^{n+1} - 1$$
- $$\sum_{i=0}^{n} a^i = 1 + a + a^2 \cdots + a^n = \frac{a^{n+1} - 1}{a - 1}$$, para $a \neq 1$ (PG)
- $$\sum_{i=1}^{n} i2^i = 1 \times 2 + 2 \times 2^2 + \cdots + n \times 2^n = (n-1)2^{n+1} + 2$$
- $$\sum_{i=1}^{n} \frac{1}{i} = \frac{1}{1} + \frac{1}{2} + \cdots + \frac{1}{n} \approx \ln n + \gamma$$, onde $\gamma = 0,5772...$[1]
- $$\sum_{i=1}^{n} \log_2 i = \log_2 1 + \log_2 2 + \cdots + \log_2 n \approx n \log_2 n$$

[1] chamada série harmônica, e γ constante de Euler-Mascheroni

Algumas regras para manipulação

- $\displaystyle\sum_{i=l}^{u} ca_i = c\sum_{i=l}^{u} a_i$
- $\displaystyle\sum_{i=l}^{u} (a_i \pm b_i) = \sum_{i=l}^{u} a_i \pm \sum_{i=l}^{u} b_i$
- $\displaystyle\sum_{i=l}^{u} a_i = \sum_{i=l}^{m} a_i + \sum_{i=m+1}^{u} a_i$, onde $l \leq m < u$

Algumas regras para manipulação

- $n! \approx \sqrt{2\pi n} \cdot \left(\frac{n}{e}\right)^n$ (fórmula de Stirling)
- $(\log_2 n)' = \dfrac{\log_2 e}{n}$ ou $(\log_2 n)' = \dfrac{1}{n \cdot \ln 2}$
- $\left(\sqrt{n}\right)' = \dfrac{1}{2\sqrt{n}}$
- $\lim\limits_{n\to\infty} \dfrac{f(n)}{g(n)} = \lim\limits_{n\to\infty} \dfrac{f(n)'}{g(n)'}$ (Regra de L'Hospital)
- $\lim\limits_{n\to\infty} a^n = 0$, onde $0 \leqslant a < 1$
- $\lim\limits_{n\to\infty} \dfrac{1}{n} = 0$
- $\lim\limits_{n\to\infty} n^a = \infty$, onde $a \geq 1$

Logaritmos

- Definição: $log_b n = x \Leftrightarrow b^x = n$
- $b^{log_b n} = n$
- $n = m \quad \Leftrightarrow \quad log_b n = log_b m$
- Logaritmo de um produto: $log_b(n.m) = log_b n + log_b m$
- Logaritmo de uma divisão: $log_b(\frac{n}{m}) = log_b n - log_b m$
- Logaritmo de uma potência: $log_b n^x = x.log_b n$
- Troca de base: $log_b n = \frac{log_a n}{log_a b}$

Logaritmos

- Definição: $log_b n = x \Leftrightarrow b^x = n$
- $b^{log_b n} = n$
- $n = m \quad \Leftrightarrow \quad log_b n = log_b m$
- Logaritmo de um produto: $log_b(n.m) = log_b n + log_b m$
- Logaritmo de uma divisão: $log_b(\frac{n}{m}) = log_b n - log_b m$
- Logaritmo de uma potência: $log_b n^x = x.log_b n$
- Troca de base: $log_b n = \frac{log_a n}{log_a b}$

Logaritmos

- Definição: $log_b n = x \Leftrightarrow b^x = n$
- $b^{log_b n} = n$
- $n = m \quad \Leftrightarrow \quad log_b n = log_b m$
- Logaritmo de um produto: $log_b(n.m) = log_b n + log_b m$
- Logaritmo de uma divisão: $log_b(\frac{n}{m}) = log_b n - log_b m$
- Logaritmo de uma potência: $log_b n^x = x.log_b n$
- Troca de base: $log_b n = \frac{log_a n}{log_a b}$

Logaritmos

- Definição: $log_b n = x \Leftrightarrow b^x = n$
- $b^{log_b n} = n$
- $n = m \quad \Leftrightarrow \quad log_b n = log_b m$
- Logaritmo de um produto: $log_b(n.m) = log_b n + log_b m$
- Logaritmo de uma divisão: $log_b(\frac{n}{m}) = log_b n - log_b m$
- Logaritmo de uma potência: $log_b n^x = x.log_b n$
- Troca de base: $log_b n = \frac{log_a n}{log_a b}$

Logaritmos

- Definição: $log_b n = x \Leftrightarrow b^x = n$
- $b^{log_b n} = n$
- $n = m \quad \Leftrightarrow \quad log_b n = log_b m$
- Logaritmo de um produto: $log_b(n.m) = log_b n + log_b m$
- Logaritmo de uma divisão: $log_b(\frac{n}{m}) = log_b n - log_b m$
- Logaritmo de uma potência: $log_b n^x = x.log_b n$
- Troca de base: $log_b n = \frac{log_a n}{log_a b}$

Logaritmos

- Definição: $log_b n = x \Leftrightarrow b^x = n$
- $b^{log_b n} = n$
- $n = m \quad \Leftrightarrow \quad log_b n = log_b m$
- Logaritmo de um produto: $log_b(n.m) = log_b n + log_b m$
- Logaritmo de uma divisão: $log_b(\frac{n}{m}) = log_b n - log_b m$
- Logaritmo de uma potência: $log_b n^x = x.log_b n$
- Troca de base: $log_b n = \frac{log_a n}{log_a b}$

Logaritmos

- Definição: $log_b n = x \Leftrightarrow b^x = n$
- $b^{log_b n} = n$
- $n = m \quad \Leftrightarrow \quad log_b n = log_b m$
- Logaritmo de um produto: $log_b(n.m) = log_b n + log_b m$
- Logaritmo de uma divisão: $log_b(\frac{n}{m}) = log_b n - log_b m$
- Logaritmo de uma potência: $log_b n^x = x.log_b n$
- Troca de base: $log_b n = \frac{log_a n}{log_a b}$

Funções piso e teto

- O **tempo** de execução de um algoritmo é geralmente expresso por meio de uma **quantidade inteira**, como o **número de operações** realizadas. Assim, na análise de algoritmos pode, às vezes, envolver o uso das chamadas funções "piso" e "teto", que são definidas respectivamente como:
- $\lfloor x \rfloor$ = o maior inteiro menor ou igual a x.
- $\lceil x \rceil$ = o menor inteiro maior ou igual a x.
- Exemplos:
  $\lfloor 5.7 \rfloor = 5$
  $\lceil 5.1 \rceil = 6$