

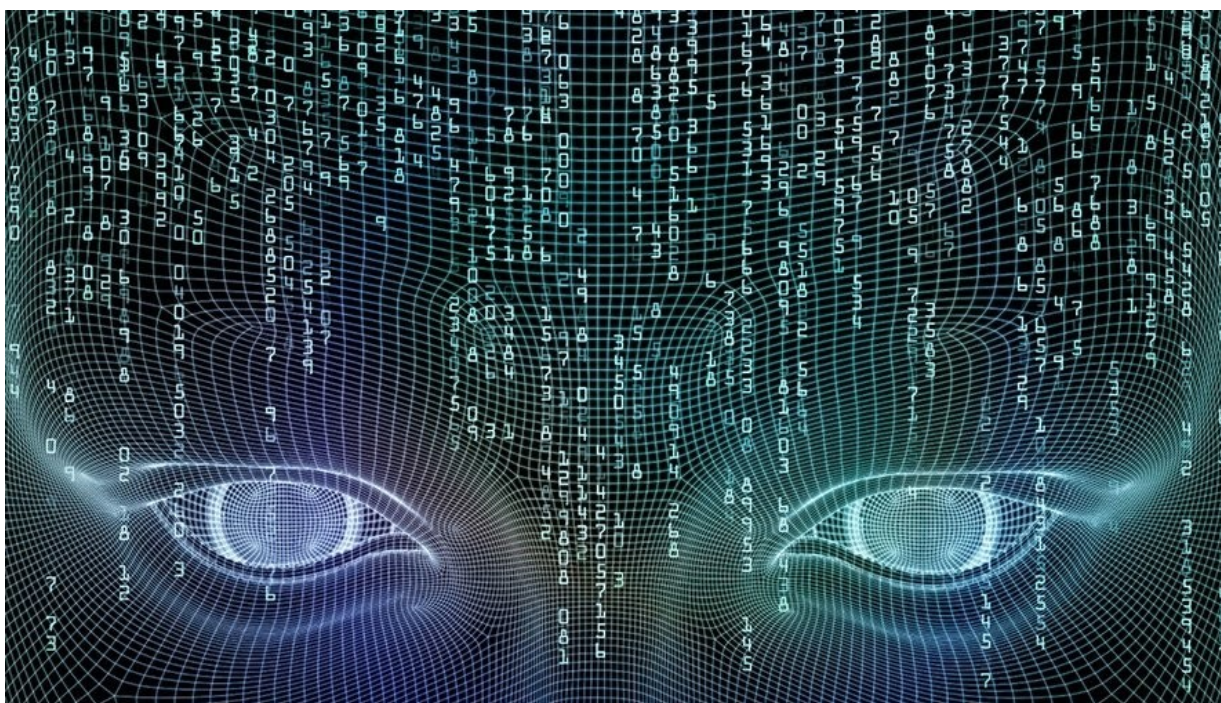
Algoritmos

SSC 301 – IC para Engenharia Ambiental

Prof. Marcio Delamaro

Algoritmo

- O que é?



Algoritmo

- O que é?
 - Substantivo masculino
Matemática Sequência de raciocínios ou operações que oferece a solução de certos problemas.
(dicio.com.br)
 - Algoritmo é uma sequência finita de instruções bem definidas e não ambíguas, cada uma das quais devendo ser executada mecânica ou eletronicamente em um intervalo de tempo finito e com uma quantidade de esforço finita. (wikipedia)

Wikipedia

- Um algoritmo não representa, necessariamente, um programa de computador, e sim os passos necessários para realizar uma tarefa. Sua implementação pode ser feita por um computador, por outro tipo de autômato ou mesmo por um ser humano.
- O conceito de algoritmo é frequentemente ilustrado pelo exemplo de uma receita culinária, embora muitos algoritmos sejam mais complexos.

Receita culinária

- Problema: preparar peixe com molho de maracujá



Receita culinária

- Problema: preparar peixe com molho de maracujá
- Algoritmo (solução)
 - Entre na cozinha
 - Prepare o peixe e o molho
 - Sirva em prato com o molho de maracujá por cima, acompanhado de arroz branco



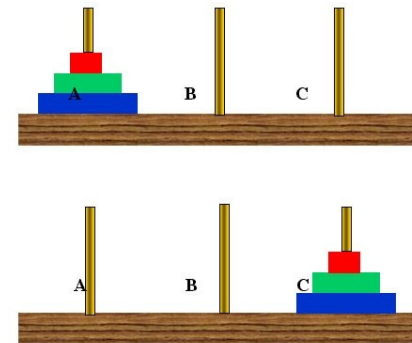
Receita culinária

- Problema: preparar peixe com molho de maracujá
- Algoritmo (solução)
 - Derreta uma colher de sopa de manteiga
 - Acrescente uma colher de sopa de açúcar
 - Acrescente a polpa do maracujá
 - Ferva por alguns minutos até reduzir na consistência de molho, reserve
 - Tempere o peixe com sal e pimenta-do-reino
 - Passe um dos lados na farinha de trigo e retire o excesso
 - Frite em uma colher de manteiga, primeiro o lado com a farinha
 - Após 3 minutos vire e frite o outro lado por mais 3 minutos
 - Sirva em prato com o molho de maracujá por cima, acompanhado de arroz branco



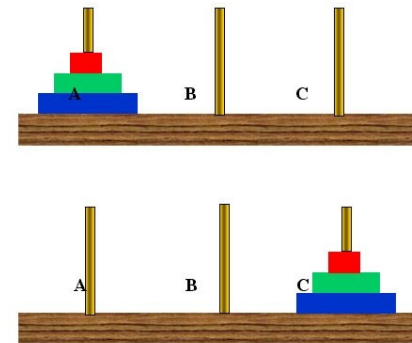
Torres de Hanoi

- Tem-se três hastes. Uma das hastes serve de suporte para os discos. Deseja-se mover todos os discos para outra haste, porém deve-se movimentar um disco de cada vez e um disco maior nunca pode ser colocado sobre um disco de menor tamanho.



Torres de Hanoi

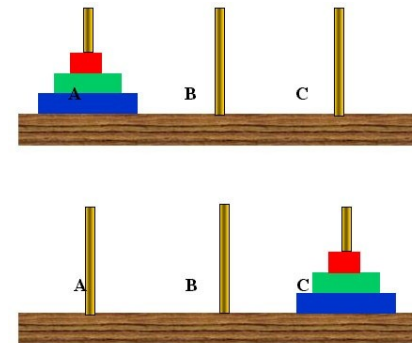
- Tem-se três hastes. Uma das hastes serve de suporte para os discos. Deseja-se mover todos os discos para outra haste, porém deve-se movimentar um disco de cada vez e um disco maior nunca pode ser colocado sobre um disco de menor tamanho.



Torres de Hanoi

- Tem-se três hastes. Uma das hastes serve de suporte para os discos. Deseja-se mover todos os discos para outra haste, porém deve-se movimentar um disco de cada vez e um disco maior nunca pode ser colocado sobre um disco de menor tamanho.

1. Move de A para C	5. Move de B para A
2. Move de A para B	6. Move de B para C
3. Move de C para B	7. Move de A para C
4. Move de A para C	



Existem algoritmos ruins?



Existem algoritmos ruins?

- Algoritmos errados
- Algoritmos certos mais ruins
- Problemas difíceis

Errados

- Não resolvem o problema ou não resolvem em todos os casos
- Contar as pessoas na sala
 - *inicie com valor 0*
para cada par de pessoas:
aumente o valor em 2 unidades

Certos mas ruins

- Estão corretos, ou seja, resolvem o problema mas “na prática” não pode ser usados
- Buscar um livro na biblioteca



k6855664 fotosearch ©

Problemas difíceis

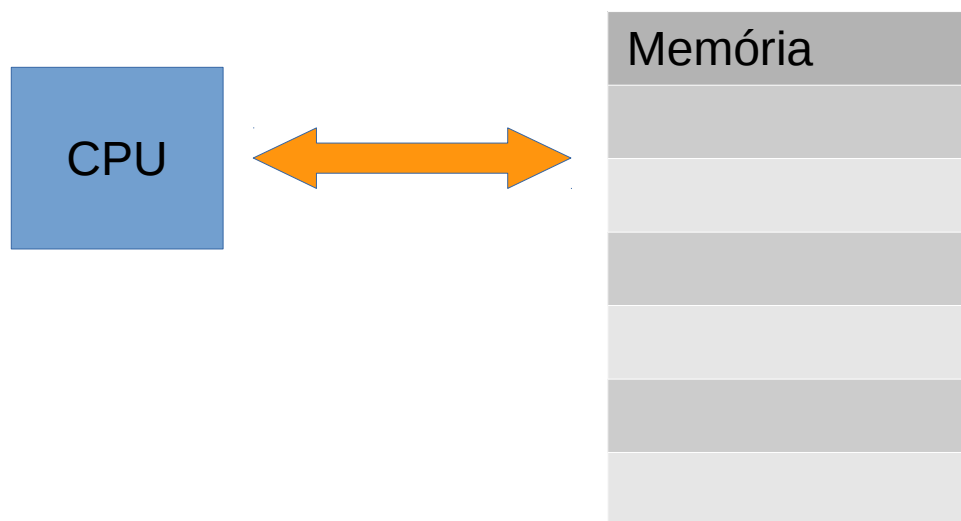
- O problema em si, dependendo do seu “tamanho” não admite uma solução adequada
- Torres de Hanoi
 - Para solucionar com 4 discos, são necessários 15 movimentos
 - Para solucionar com 7 discos, 127 movimentos
 - Para solucionar com 15 discos, 32.767 movimentos
 - Para solucionar com 64 discos, como diz a lenda, 18.446.744.073.709.551.615 movimentos. (580 séculos)

Em resumo

- Para resolver alguns problemas, nós precisamos de um algoritmo que nos instrua
- Precisamos também de uma forma de expressar o algoritmo
 - Depende de quais são os “passos” que sabemos realizar

Para utilizar o computador

- Ainda precisamos de um algoritmo
- As instruções que o computador executa (os passos) são muito primitivas



Hanoi Hanoi

```
push %rbp
mov %rsp,%rbp
sub $0x10,%rsp
mov %fs:0x28,%rax
mov %rax,-0x8(%rbp)
xor %eax,%eax
mov $0x400774,%edi
mov $0x0,%eax
callq 4004c0 <printf@plt>
lea -0xc(%rbp),%rax
mov %rax,%rsi
mov $0x400787,%edi
mov $0x0,%eax
callq 4004e0 <__isoc99_scanf@plt>
```

```
mov -0xc(%rbp),%eax
mov %eax,%ecx
mov $0x43,%edx
mov $0x42,%esi
mov $0x41,%edi
callq 400666 <hanoi>
mov $0x0,%eax
mov -0x8(%rbp),%rdx
xor %fs:0x28,%rdx
je 400664 <main+0x6e>
callq 4004b0 <__stack_chk_fail@plt>
leaveq
retq
```

Linguagem de alto nível

- Permite que expressemos os algoritmos de uma forma mais simples
- Tem um conjunto de instruções que são mais fáceis de usar

Linguagem de alto nível

- Permite que expressemos os algoritmos de uma forma mais simples
- Tem um conjunto de instruções que são mais fáceis de usar

```
import copy

# Utiliza Laplace para computar o determinante
def determinante(M):
    # fim da recursão. Qdo tamanho da matriz é 1
    if len(M) == 1:
        return M[0][0]

    det = 0.0
    for i in range(len(M)):
        #faz uma copia da matriz sem linha 0 e sem coluna i
        M2 = copia_matriz(M,0,i)

        # como a linha é sempre 0, olha a coluna para determinar
        # o sinal do cofator
        if i % 2 == 0:
            sig = 1
        else:
            sig = -1

        # calcula o determinante da matriz reduzida
        d = determinante(M2)
```

Compilador



Interpretador

```
push %rbp
mov %rsp,%rbp
sub $0x10,%rsp
mov %fs:0x28,%rax
mov %rax,-0x8(%rbp)
xor %eax,%eax
mov $0x400774,%edi
mov $0x0,%eax
callq 4004c0 <printf@plt>
lea -0xc(%rbp),%rax
mov %rax,%rs
```


Python

- Linguagem de programação baseada em um interpretador
- Homenagem ao grupo Monty Python
- Crescido muito nos últimos anos
- Flexível e “fácil” de utilizar

Bhaskara

- Para começarmos a conhecer o interpretador Python, vamos usar esse algoritmo
- Tente descrever os passos desse algoritmo
- Ver material do livro

Prática

- Seguir o material do livro sobre o interpretador Python
- Anotar dúvidas, problemas e sugestões para melhorar o texto