

Algoritmos e linguagens

Marco A L Barbosa

malbarbo.pro.br

Departamento de Informática

Universidade Estadual de Maringá



Este trabalho está licenciado com uma Licença Creative Commons - Atribuição-Compartilhual 4.0 Internacional.

<http://github.com/malbarbo/na-programacao>

Fundamentos de Algoritmos

- Resolver problemas projetando programas que sejam bem escritos e funcionem corretamente.

O que são algoritmos e qual é a relação deles com resolução de problemas e programas de computadores?

O que é um algoritmo?

Dicionário Silveira Bueno

Conjunto predeterminado e definido de regras e processos destinados à solução de um problema, com um número finito de etapas.

Dicionário Merriam-Webster

Um procedimento passo a passo para resolver um problema ou atingir um objetivo.

Forbellone e Eberspächer

Uma sequência de passos que visam atingir um objetivo bem definido.

Thomas Cormen

Uma sequência bem definida de passos que toma um conjunto de valores como entrada e produz um conjunto de valores como saída com o objetivo de resolver um problema.

Donald Knuth

Um conjunto finito de regras que fornece uma sequência de operações para resolver um tipo específico de problema.

O que estas definições têm em comum?

- Sequência de etapas/passos/operações
- Resolução de um problema

Então, qual é a relação entre algoritmos, resolução de problemas e programas de computadores?

- Processos de resolução de problemas podem ser descritos com algoritmos.

E a relação com programas de computadores?

- Algoritmos podem ser implementados como programas de computadores (embora algoritmos possam ser usados em diversos contextos, estamos interessados apenas naqueles que podem ser executados por computadores)

Como expressar um algoritmo?

- Descrição textual
- Fluxograma
- Pseudocódigo
- Funções matemáticas
- Etc

Para cada algoritmo a seguir, vamos

- Ler o algoritmo
- Tentar executar o algoritmo
- Tentar descobrir o propósito do algoritmo

Considere a tabela ao lado.

Marque a linha 1 da tabela como linha de referência.

Olhando para a linha de referência e para as linhas que estão após ela, procure pela linha com a coluna **B** com maior valor e troque de lugar o conteúdo dessa linha com o conteúdo da linha de referência.

Se existe uma linha após a linha de referência, considere essa próxima linha como referência e repita o processo, senão pare.

	A	B
1	Paula	8
2	Jorge	10
3	Maria	7
4	José	6
5	Ana	9
6	Mário	8

Qual é o propósito do algoritmo?

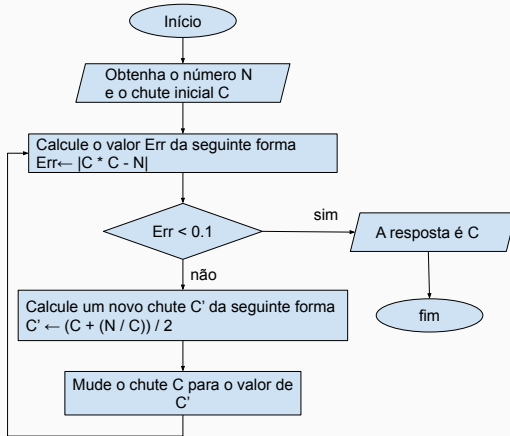
Organizar os nomes em ordem decrescente de pontos.

Características

- Linguagem natural
- Fácil entendimento
- Falta de precisão

Algoritmo 2 - Fluxograma

Usando $N = 4$ e $C = 1$, faça



Qual é o propósito do algoritmo?

Calcular a raiz quadrada (método de Newton).

Características

- Representação gráfica com símbolos padronizados
- Intuitivo e fácil de entender o fluxo
- Restrito a algoritmos curtos

Passo 0:

- Comece com $A = 52$ e $B = 36$

Passo 1:

- Se $A < B$, troque o valor de A por B e vice-versa

Passo 2:

- Se $B = 0$, então pare e dê como saída o valor de A
- Senão
 - $C \leftarrow A - B$
 - $A \leftarrow B$
 - $B \leftarrow C$
 - Repita o passo 1

Qual é o propósito do algoritmo?

Encontrar o MDC de dois números (algoritmo de Euclides).

Características

- Linguagem natural mais estruturada
- Mais preciso que descrição textual
- Não é muito padronizado

Calcule o valor de $f([2, 1, 4, 1, 2])$

$$f(x) = \begin{cases} 0 & \text{se } num_itens(x) = 0 \\ primeiro(x) + f(sem_primeiro(x)) & \text{caso contrário} \end{cases}$$

Qual é o propósito do algoritmo?

Somar os valores de uma lista.

Características

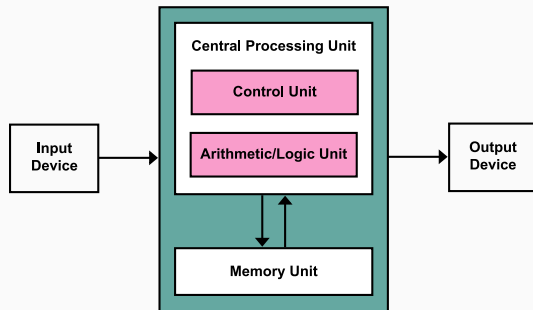
- Baseado em formalismo matemático
- Precisa e concisa
- Difícil para iniciantes

O que foi necessário para entender/executar esses algoritmos?

- Entender a notação
- Entender a semântica (significado)

O que esses algoritmos têm em comum?

- Entrada e saída
- Ações (processamento)
 - Cálculo de valores
 - Escolha entre ações
 - Repetição de ações
- Uso de memória



O que é possível fazer com algoritmos?

ATENÇÃO: sequência de perguntas a seguir!

Todos os problemas podem ser resolvidos com algoritmos?

O que é um problema?

Vamos considerar apenas problemas bem definidos, que especificam com precisão quais são as possíveis entradas e como as saídas estão relacionadas com as entradas.

Todos os problemas bem definidos podem ser resolvidos com algoritmos?

O que significa dizer que um algoritmo resolve um problema?

Que para todas as entradas possíveis, o algoritmo produz a saída correta.

E então, todos os problemas bem definidos podem ser resolvidos com algoritmos?

Não...

Os problemas que podem ser resolvidos com algoritmos são chamados de problemas computáveis.

Pode existir mais que um algoritmo que resolve o mesmo problema? Sim.

O que considerar na escolha ou no projeto de algoritmos?

- Simplicidade
- Uso de recursos (tempo de processamento, memória, energia, etc)
- Corretude e/ou precisão

Qual representação utilizar para escrever algoritmos que possam ser executados por um computador?

Nenhuma das que vimos, precisamos de uma linguagem.

Mas o que é uma linguagem e qual a sua função?

Um sistema estruturado para comunicação.

Podemos utilizar o português para programar um computador?

Não, porque o computador não entende o português! Ainda que entendesse, o português não seria adequado para programação porque não é preciso o bastante.

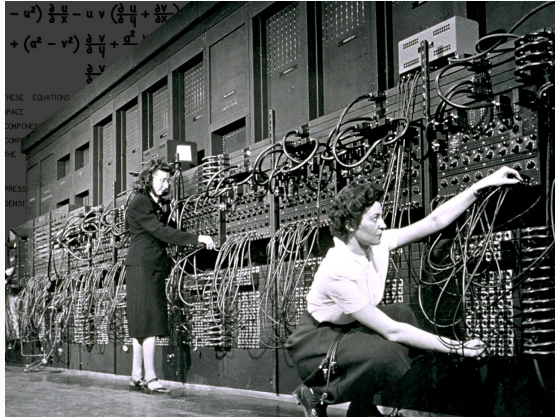
E que linguagem o computador entende?

Cada tipo de computador entende uma linguagem específica, a sua linguagem de máquina.

Então, precisamos programar um computador usando sua linguagem de máquina?

Hoje em dia não, mas os primeiros computadores eram programados usando linguagem de máquina!

O ENIAC (1945), um dos primeiros computadores, era programado reconectando cabos e ajustando interruptores.



Se não queremos utilizar linguagem de máquina, como podemos programar os computadores?

Como duas entidades que usam linguagens diferentes podem se comunicar?

Usando um tradutor.

Se a linguagem fonte e a linguagem alvo forem bem definidas, podemos escrever um programa que faça a tradução de forma automática.

A linguagem alvo (de máquina) é bem definida, precisamos de uma linguagem fonte que também seja bem definida.

Uma **linguagem de programação** é uma linguagem formal (bem definida) adequada para programar computadores.

Uma linguagem de programação é caracterizada por diversos aspectos, entre eles a sintaxe, a semântica e o modo de execução.

A **sintaxe** de uma linguagem é o conjunto de regras que define como os símbolos podem ser agrupados para criar as sentenças na linguagem.

Por exemplo, a instrução para soma de dois números é expressa:

- Na maioria das linguagens: `20 + 17`
- Em Lisp: `(+ 20 17)`
- Em Forth: `20 17 +`

A **semântica** de uma linguagem define o significado das construções da linguagem.

A expressão `12 + "3"` é válida sintaticamente em muitas linguagens, mas o significado é diferente entre elas.

- Lua: o texto `"3"` deve ser convertido para um número e depois somado com `12`, resultando em `15`.
- Javascript: o número `12` deve ser transformado para um texto e depois juntado com `"3"`, resultando em `"123"`.
- Na maioria das linguagens: é um erro semântico.

Os dois principais **modos de execução** das linguagens de programação são a compilação e a interpretação.

Em uma linguagem compilada o código do programa é primeiro traduzido (compilado) para código de máquina e posteriormente executado.

Em uma linguagem interpretada o código do programa é lido, traduzido e executado diretamente pelo interpretador.

Exemplo em C++ - Raiz quadrada

Parte do arquivo `raiz.cpp`

```
// Entrada
double n = stof(argv[1]);
double c = stof(argv[2]);

// Processamento
while (abs(c * c - n) >= 0.1) {
    double c_ = (c + n / c) / 2;
    c = c_;
}

// Saída
cout << c << endl;
```

Windows

Compilação

```
g++ -o raiz.exe raiz.cpp
```

Execução

```
> raiz.exe 4 1
2.00061
```

Linux

Compilação

```
g++ -o raiz raiz.cpp
```

Execução

```
> ./raiz 4 1
2.00061
```

Exemplo em Python - MDC

Parte do arquivo `mdc.py`

```
# Entrada
a = int(sys.argv[1])
b = int(sys.argv[2])
# Processamento
if a < b:
    a, b = b, a
while b != 0:
    c = a - b
    a = b
    b = c
    if a < b:
        a, b = b, a
# Saída
print(a)
```

Interpretação

```
> python3 mdc.py 52 36
4
```

Existem centenas de linguagens de programação, cada uma com suas próprias características e usos, então, qual delas aprender?

Nenhuma em particular.

O mais importante não são as linguagens em si, mas os princípios que são utilizados para criar os programas nas linguagens.

Dominando os fundamentos de algoritmos e paradigmas de programação, aprender uma nova linguagem não é uma tarefa difícil!

Mas precisamos de uma linguagem para começar.

Nessa disciplina vamos utilizar a linguagem Python!

Python é uma linguagem bastante utilizada e versátil, mas também é uma linguagem extensa!

Mas não se preocupe, vamos utilizar apenas construções básicas!

Vamos começar?

O que é um algoritmo?

- Uma sequência de passos que visa resolver um problema bem definido.

De que formas podemos expressar um algoritmo?

- Descrição textual, fluxograma, pseudocódigo, funções matemáticas, entre outras.

Todo problema bem definido pode ser resolvido com um algoritmo?

- Não. Os problemas que podem ser resolvidos com algoritmos são chamados de problemas computáveis.

O que é uma linguagem de programação?

- Uma linguagem formal (bem definida) adequada para programar computadores.

O que são sintaxe e semântica?

- A sintaxe define como os símbolos podem ser agrupados para formar sentenças. A semântica define o significado dessas construções.

Qual é a diferença entre compilação e interpretação?

- Na compilação, o código é primeiro traduzido para linguagem de máquina e depois executado. Na interpretação, o código é lido, traduzido e executado diretamente pelo interpretador.