

Algoritmos e Programação I

Introdução a Computação

Prof. Fernando Maia da Mota
mota.fernandomaia@gmail.com
CPCX/UFMS
© Fernando Maia da Mota

Computadores são dispositivos que só sabem fazer um tipo de coisa: executar algoritmos para processar informação. Para cientistas da Computação, algoritmo é o conceito central da Computação.

Algoritmos

- ❖ Um algoritmo é uma sequência de instruções para resolver um problema, a qual possui as seguintes propriedades:
 - ❖ **Garantia de término:** o problema a ser resolvido possui condições específicas que, quando satisfeitas, a execução do algoritmo é encerrada e o problema é então tido como “resolvido”. Além disso, estas condições devem ser satisfeitas após uma quantidade finita de tempo, a ser contado a partir do início da execução do algoritmo.
 - ❖ **Exatidão:** a intenção de cada instrução no algoritmo deve ser suficientemente clara, de forma que não haja ambiguidade na interpretação da intenção.
 - ❖ **Efetividade:** cada instrução deve ser básica o suficiente para ser executada, pelo menos em princípio, por qualquer agente usando apenas lápis e papel.

Algoritmos

- ❖ Considere o seguinte algoritmo para encontrar o máximo divisor comum (MDC) de dois números naturais (N^*):
 1. Chame o maior número de a e o menor de b
 2. Divida a por b e chame o resto de r
 3. Se r é igual a zero então o MDC é igual a b e a execução das instruções encerra aqui. Caso contrário, siga para a próxima instrução.
 4. Atribua o valor de b a a e o valor de r a b
 5. Volte para a instrução 2.

Algoritmos

- ❖ Este algoritmo se seguido resolve qualquer ocorrência do problema.
- ❖ Se seguida, resolve qualquer ocorrência do problema. A execução da sequência sempre pára após uma quantidade finita de tempo. Isto é garantido pela instrução 3, que compara o valor de r a zero e termina a execução se r é igual a 0.
- ❖ Cada instrução da sequência é clara e possível de ser executada por qualquer pessoa que saiba, pelo menos, dividir dois números.

Algoritmos

- ❖ Como era de se esperar, nem toda sequência de instruções para resolver um determinado problema pode ser considerada um algoritmo.
- ❖ Por exemplo, uma instrução “Escreva todos os números ímpares”.
 - ❖ Neste caso, temos uma instrução que não pode ser executada porque a execução nunca terminará, apesar de sabermos exatamente como determinar os números ímpares.
 - ❖ Se modificássemos a instrução para “Escreva todos os números ímpares menores do que 100”, ela poderia, perfeitamente, fazer parte de um algoritmo.

Algoritmos

- ❖ Um problema para o qual existe uma solução na forma de algoritmo é dito um **problema algorítmico**.
 - ❖ O problema de encontrar o MDC de dois números naturais quaisquer é, portanto, um problema algorítmico.
- ❖ Um algoritmo é dito **correto** quando ele sempre termina e produz a resposta correta para todas as ocorrências de um dado problema.
- ❖ Algoritmo, então, pode ser imaginado como a especificação de um processo “mecânico” que, quando executado, leva-nos à solução de algum problema.

Algoritmos

- ❖ A execução de um algoritmo por um computador é denominada processamento de dados e consiste de três partes: **uma entrada, um processo e uma saída.**
- ❖ Todo algoritmo deve possuir um meio de obter os dados da entrada. Esta tarefa é conhecida como leitura da entrada.
- ❖ Todo algoritmo deve possuir um meio de revelar o resultado da computação. Isto é conhecido como escrita dos dados da saída.

Algoritmos

- ❖ A capacidade para resolver problemas pode ser vista como uma habilidade a ser adquirida. Esta habilidade, como qualquer outra, pode ser obtida pela combinação de duas partes:
 - ❖ **Conhecimento:** adquirido pelo estudo. Em termos de resolução de problemas, está relacionado a que táticas, estratégias e planos usar e quando usar;
 - ❖ **Destreza:** adquirida pela prática. A experiência no uso do conhecimento nos dá mais agilidade na resolução de problemas.

Algoritmos

- ❖ Independente do problema a ser resolvido, ao desenvolvermos um algoritmo devemos seguir os seguintes passos:
 - ❖ **Análise preliminar:** entender o problema com a maior precisão possível, identificando os dados e os resultados desejados;
 - ❖ **Solução:** desenvolver um algoritmo para o problema;
 - ❖ **Teste de qualidade:** executar o algoritmo desenvolvido com uma entrada para a qual o resultado seja conhecido;
 - ❖ **Alteração:** se o resultado do teste de qualidade não for satisfatório, altere o algoritmo e submeta-o a um novo teste de qualidade;
 - ❖ **Produto final:** algoritmo concluído e testado, pronto para ser aplicado.

Algoritmos e a Abstração

- ❖ “o processo de identificar certas propriedades ou características de uma entidade material e usá-las para especificar uma nova entidade que representa uma simplificação da entidade da qual ela foi derivada”. Esta “nova entidade” é o que chamamos de **abstração**.
- ❖ **Maria tinha cinco maçãs e João tinha três. Quantas maçãs eles tinham juntos?**

Provavelmente, um adulto resolveria este problema fazendo uma abstração das maçãs como se elas fossem os números 5 e 3 e faria a soma de tais números. Uma criança poderia imaginar as cinco maçãs de Maria como cinco palitinhos e as três de João como três palitinhos. Daí, faria uma contagem dos palitinhos para chegar à solução. Em ambos os casos, os elementos do problema foram substituídos por outros (números e palitinhos) e a solução foi encontrada através da manipulação dos novos elementos.

Algoritmos e a Abstração

- ❖ Algoritmos bem projetados são organizados em níveis de abstração, pois um mesmo algoritmo deve ser entendido por pessoas com diferentes graus de conhecimento.
- ❖ Para isso o uso de **módulos**, que agrupa instruções que realizam uma determinada tarefa do algoritmo se mostra como uma poderosa ferramenta.

Algoritmos e Computadores

- ❖ De acordo com o *Webster's New World Dictionary of the American Language* (segunda edição), um computador é “uma máquina eletrônica que, por meio de instruções e informações armazenadas, executa rápida e frequentemente cálculos complexos ou compila, correlaciona e seleciona dados”.
- ❖ Basicamente, um computador manipula **dados**.
- ❖ O que faz dos computadores uma máquina notável é a extrema rapidez e precisão com que eles podem armazenar, recuperar e manipular dados.

Algoritmos e Computadores

- ❖ Quando desejamos utilizar um computador para nos auxiliar na tarefa de processamento de dados, deparamo-nos com alguns problemas inerentes a este processo: “Como informaremos ao computador o algoritmo que deve ser executado para obtermos o resultado desejado?”, “Como forneceremos a entrada do algoritmo?” e “Como receberemos o resultado do algoritmo?”.
- ❖ O ato de instruir o computador para que ele resolva um determinado problema é conhecido como **programação**.
- ❖ A forma como descrevemos algoritmos computacionais, é através de uma **linguagem de programação**.
- ❖ O termo **programa** é comumente empregado para designar o algoritmo em uma linguagem de programação.

Algoritmos e Computadores

- ❖ Cada computador possui uma linguagem de programação própria, denominada **linguagem de máquina**, e, em geral, distinta das linguagens de máquina dos demais modelos de computador.
- ❖ Para evitar que para cada modelo de computador tivéssemos que aprender uma nova linguagem, linguagens independentes de modelo de computadores foram criadas.
- ❖ Se você aprende uma linguagem independente de máquina, estará apto, pelo menos em princípio, a programar qualquer computador.
- ❖ Estas linguagens não são compreendidas por computador, para isso é necessário a utilização de um outro programa denominado **compilador**.

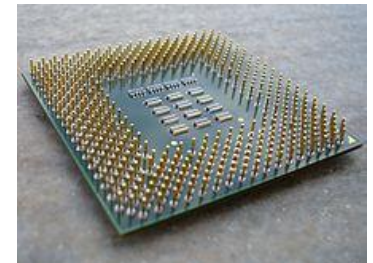
Algoritmos e Computadores

- ❖ Arquitetura básica de um computador
 - ❖ Dispositivos de entrada:
 - ❖ Teclado, mouse.
 - ❖ Dispositivos de saída:
 - ❖ Monitor, impressora.
 - ❖ CPU
 - ❖ UC.
 - ❖ ULA.
 - ❖ Memória:
 - ❖ Principal ou temporária;
 - ❖ Memória secundária.



Algoritmos e Computadores

- ❖ A CPU também possui seus próprios elementos de memória, denominados como **registradores**.
- ❖ Os registradores armazenam, em cada instante, os dados a serem imediatamente processados, isto é, os dados referenciados pela instrução processada no momento e que foram trazidos da memória principal.
- ❖ Os registradores possibilitam o aumento de velocidade na execução das instruções, pois os resultados intermediários da instrução não precisam ser armazenados na memória principal.
- ❖ Com o avanço da microeletrônica é possível construir toda uma CPU em uma única pastilha de silício. Essa pastilha, ou chip, denomina-se microprocessador.



Algoritmos e Computadores

- ❖ Os microprocessadores são classificados pelo tamanho da **palavra** ou comprimento, em bits, da unidade de informação que são capazes de processar de uma só vez.
- ❖ Os primeiros microprocessadores foram de 8 bits, seguidos pelos de 16 bits, depois pelos de 32 bits e, mais recentemente, pelos de 64 bits

Algoritmos e Computadores

- ❖ Tanto as **instruções** quanto os **dados** são localizados na memória através dos **endereços** das posições de memória que os contêm.
- ❖ O tamanho de cada posição de memória é dado em *bytes*.
 - ❖ Cada 1024*bytes* representam 1 *quilobyte* (1*K*).
 - ❖ Cada 1024*K* representam 1 *megabyte* (1*M*).
 - ❖ Cada 1024*M* representam 1 *gigabyte* (1*G*).
 - ❖ Cada 1024*G* representam 1 *terabyte* (1*T*).
- ❖ Se o tamanho de uma posição de memória de um computador mede 2 *bytes* e a memória possui 640 *Kbytes* de capacidade de armazenagem, o número de posições de memória é igual a 327.680.
- ❖ Se o tamanho de uma posição mede 4 *bytes* com uma memória de 128*Mbytes*, qual é o número de posições disponível?

Algoritmos e Computadores

- ❖ A memória principal perde todo o seu conteúdo no momento em que o computador é desligado.
- ❖ Ela possui o propósito de armazenar instruções e dados de um programa em execução.
- ❖ Sua principal característica é a rapidez com que as informações nela armazenadas são lidas e escritas pela CPU.
- ❖ Toda vez que um programa deve ser executado, ele é inserido antes na memória primária.

Linguagens de Programação

- ❖ A primeira geração de linguagens de programação remonta aos dias de codificação em linguagem de máquina.
- ❖ Esta linguagem é formada por instruções descritas como sequências de bytes.
- ❖ Um programa em linguagem de máquina poderia se parecer com a seguinte sequência de bytes:

01000011 00111010 00111011 01000001 00101011 01000100

- ❖ Para programar em linguagem de máquina, nós devemos conhecer a sequência de bits que determina cada instrução e também como codificar os dados em binário. Além disso, você deve conhecer os dispositivos internos do computador, pois as instruções de uma linguagem de máquina envolvem diretamente tais dispositivos.

Linguagens de Programação

- ❖ A programação em linguagem de máquina é, na maioria das vezes, inadequada, pois o desenvolvedor perde mais tempo com os detalhes da máquina do que com o próprio problema.
- ❖ O próximo passo na evolução das linguagens de programação foi a criação da linguagem montadora ou **assembly**.
- ❖ Instruções através de **mnemônicos**.

00000010 11001111 ← **ADD CL,BH**

- ❖ Para que o computador pudesse executar um programa escrito em linguagem montadora foi desenvolvido um compilador denominado **montador** ou **assembler**.

Linguagens de Programação

- ❖ O sucesso da linguagem montadora animou os pesquisadores a criarem linguagens em que a programação fosse realizada através de instruções na língua inglesa, deixando para o próprio computador a tarefa de traduzir o código escrito em tais linguagens para sua linguagem de máquina.
- ❖ 1957 – Fortran (primeira linguagem amplamente aceita)
- ❖ Anos 70 – Surgimento da linguagem C (destaca-se pela facilidade de uso).
- ❖ Anos 80 – Surgimento da linguagem C++ (uso de Orientação a Objetos).

Software

- ❖ O software pode ser classificado como sendo de dois tipos: **básico** ou **aplicativo**.
- ❖ Básicos: Sistema operacional, utilitários, compiladores, interpretadores, depuradores.
- ❖ Aplicativo: Um software aplicativo é aquele que realiza tarefas mais especializadas e que, apoiado nos softwares básicos, torna o computador uma ferramenta indispensável às organizações. Por exemplo, editores de texto, programas de desenho e pintura, programas de automação contábil, entre outros.

Referências

- ❖ **SIQUEIRA, Marcelo F. Algoritmos e Estrutura de Dados. Mato Grosso do Sul: CCET/CPCX - UFMS, 2007.**