

Lição 1



Conceitos Básicos e Notações

Objetivos

Ao final desta lição, o estudante será capaz de:

- Explicar os processos de resolução de problemas
- Definir tipos de dados (*data type*), tipos de dados abstratos (*abstract data type*) e estrutura de dados (*data structure*)
- Identificar as propriedades de um algoritmo
- Diferenciar os dois métodos de endereçamento – endereçamento computado e endereçamento por *link*
- Utilizar as funções matemáticas básicas para analisar algoritmos
- Mensurar a complexidade dos algoritmos expressando a eficiência em termos de complexidade de tempo e notação *Big-O*



Processo de Resolução de Problemas

- Programação – um processo de resolução de problemas pode ser visto em termos de domínio
- Domínio de problema
 - entrada, ou dado bruto, em um processo
 - Saída, ou dado processado
- Domínio de máquina
 - meios de armazenamento
 - unidades processadas
- Domínio de solução



Processo de Resolução de Problemas: Domínio de Problema

Dados Brutos

Dados pessoais
Formulário 5S
Planilhas
Etc.

Dados Processados

Dados acadêmicos
Indic. de performance
Transcrições
Etc.

Processo de Resolução de Problemas: Domínio de Máquina

Meio de Armazenamento

Bits
Bytes
Words
Etc.

Unidades de Processamento

Adicionar, Subtrair
Multiplicar
Dividir
Etc.

Processo de Resolução de Problemas: Domínio de Solução

Representações de Dados

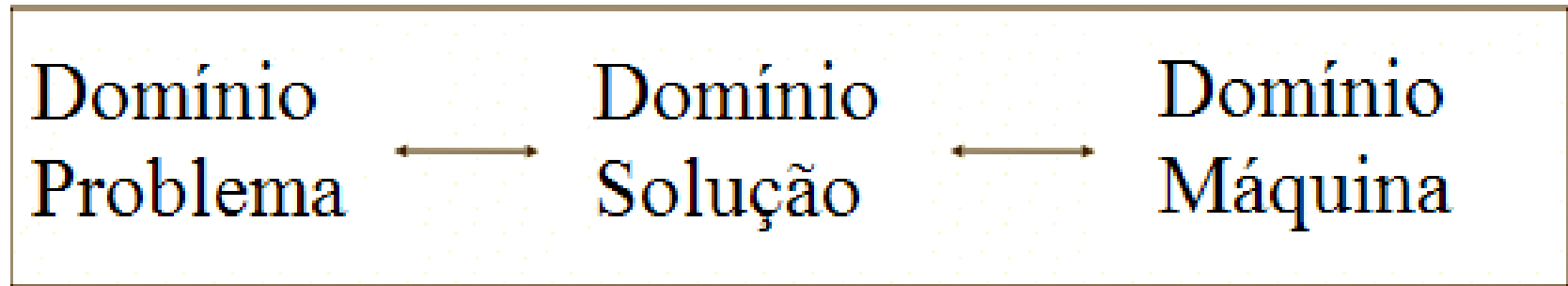
Listas
Arquivos
Tabelas
Árvores

Algoritmos

Pesquisando
Ordenando
Gerenc. de Memória
Transversal



Processo de Resolução de Problemas



- Duas tarefas relacionadas no domínio da solução
 - Estruturação da representação de dados de alto nível
 - Síntese do algoritmo
- Estrutura de dados e algoritmos são os blocos de construção dos programas de computador

Tipo de Dado, Tipo de Dado Abstrato e Estrutura de Dados

- Tipo de dado
- Tipo de Dado Abstrato (*Abstract Data Type* – ADT)
- Estrutura de Dados



Algoritmo

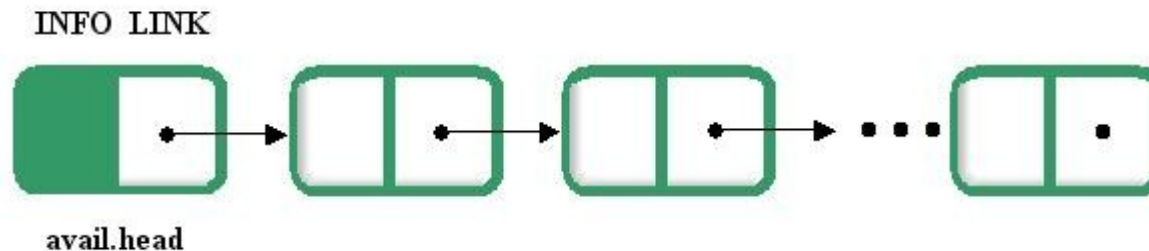
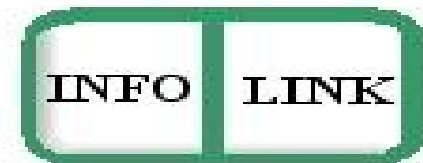
- Finito
- Definido
- Entrada
- Saída
- Efetivo



Métodos de Endereçamento

- Método de Endereçamento Computado
- Método de Endereçamento por Link
- Pool de Memória ou lista disponível

FIELD



Métodos de Endereçamento

- Passaremos agora para o NetBeans



Funções Matemáticas

- *Floor* de x ($\lfloor x \rfloor$)
- *Ceil* de x ($\lceil x \rceil$)
- **Módulo** – dados quaisquer dois números reais x e y ,
 - $x \bmod y = x$ se $y = 0$
 - $= x - y * \lfloor x / y \rfloor$ se $y \neq 0$



Funções Matemáticas

- Identidades

- $\lceil x \rceil = \lfloor x \rfloor$ se e somente se x é inteiro
- $\lceil x \rceil > \lfloor x \rfloor$ se e somente se x não é inteiro
- $\lfloor -x \rfloor = -\lceil x \rceil$
- $\lfloor x \rfloor + \lfloor y \rfloor \leq \lfloor x + y \rfloor$
- $x = \lfloor x \rfloor + x \bmod 1$
- $z (x \bmod y) = zx \bmod zy$



Complexidade de Algoritmos

- **Utilização de Espaço**
- **Eficiência relacionada ao Tempo**
- **Tempo de execução** estão incluídos:
 - Tamanho dos dados de entrada
 - Tipo da instrução
 - Velocidade da máquina
 - Qualidade do código-fonte do algoritmo implementado
 - Qualidade do código de máquina gerado a partir do código-fonte pelo compilador



Complexidade de Algoritmos

- Notação **Big-O** (ou simplesmente O)

<i>Big-Oh</i>	<i>Descrição</i>	<i>Algoritmo</i>
$O(1)$	Constante	
$O(\log_2 n)$	Logarítmico	Busca Binária
$O(n)$	Linear	Busca Sequencial
$O(n \log_2 n)$		Heapsort
$O(n^2)$	Quadrático	Insertion Sort
$O(n^3)$	Cúbico	Algoritmo de Floyd
$O(2^n)$	Exponencial	

<i>F(n)</i>	<i>Tempo de Execução</i>
$\log_2 n$	19.93 microsegundos
n	1.00 segundos
$n \log_2 n$	19.93 segundos
n^2	11.57 dias
n^3	317.10 séculos
2^n	Eternidade



Complexidade de Algoritmos

- Notação **Big-O** (ou simplesmente O)
- Regra para Adição
- Regra para Multiplicação



Complexidade de Algoritmos: Exemplos

- Considere o algoritmo abaixo:

```
for (i=1; i <= n, i++)  
    for (j=1; j <= n, j++)  
        // passos quando O(1)
```

- Já que os passos no laço interior irão executar $n + n-1 + n-2 + \dots + 2 + 1$ vezes, então o tempo de execução é:

$$n(n+1)/2 = n^2/2 + n/2 = O(n^2)$$



Complexidade de Algoritmos: Exemplos

- Passaremos agora para o NetBeans



Complexidade de Algoritmos

- Laços FOR
- Laços FOR aninhados
- Declarações consecutivas
- Condicional IF/ELSE

Sumário

- Processo de resolução de problemas
- Tipo de dado, tipo de dado abstrato e estrutura de dados
- Algoritmo
- Métodos de endereçamento
 - ♦ Método de endereçamento computado
 - ♦ Método de endereçamento por link
 - ♦ Dois procedimentos básicos
- Funções matemáticas
 - ♦ Identidade
- Complexidade de algoritmos



Parceiros

- Os seguintes parceiros tornaram JEDITM possível em Língua Portuguesa:

