

Arrays (1/2)

- *O propósito de um array é permitir o armazenamento e manipulação de uma grande quantidade de dados de mesmo tipo*
- *Exemplos de dados armazenados através de arrays:*
 - *Notas de alunos*
 - *Nucleotídeos em uma cadeia de DNA*
 - *Frequencia de um sinal de audio*

Arrays (2/2)

- *Arrays são especialmente importantes quando é necessário o acesso direto aos elementos de uma representação de uma coleção de dados*
- *Arrays são relacionados ao conceito matemático de função discreta, que mapeia valores em um conjunto finito de índices consecutivos (por exemplo, um subconjunto de inteiros não negativos) em um conjunto qualquer de objetos de mesmo tipo.*
- *$F(x) \rightarrow S, x \in U$ tal que U é um conjunto finito de valores*

Arrays unidimensionais (1/2)

- *Os elementos de um array são identificados através de índices*
- *Arrays cujos elementos são indicados por um único índice são denominados arrays unidimensionais*

Arrays unidimensionais (2/2)

- *Um elemento em uma posição indicada por um índice i , em um array A , é acessado através do indentificador do array seguido do índice i (entre chaves ou parênteses, dependendo da linguagem)*

$a(n-1)$
$a(n-2)$
$a(4)$
$a(2)$
$a(1)$
$a(0)$

Um array com n elementos

Arrays unidimensionais em Java (1/3)

- *A criação de um array em Java requer 3 passos:*
 - *Declaração do nome do array e seu tipo*
 - *Criação do array*
 - *Inicialização de seus valores*
- *Exemplo: array de 10 elementos de tipo double*

```
double[ ] a;
```

```
a = new double[10];
```

```
for (int i = 0; i<10;i++)
```

```
    a[i] = 0.0;
```

Arrays unidimensionais em Java (2/3)

- *O número de elementos de um array em Java pode ser determinado através do nome do array seguido de `.length()`*
- *Exemplo: `a.length()`*
- *Arrays em Java são objetos (mais detalhes serão vistos posteriormente)*
- *Arrays em Java tem índice base igual a zero*

Arrays unidimensionais em Java (3/3)

- *Arrays em Java podem ser inicializados em tempo de compilação*
- *Exemplos:*
 - *`String[] naipe = {"copas", "ouros", "paus", "espadas"};`*
 - *`double[] temperaturas = {45.0, 32.0, 21.7, 28.2, 27.4};`*

Arrays multidimensionais em Java

- *Arrays multidimensionais representam agregados homogêneos cujos elementos são especificados por mais de um índice*
- *Em Java é muito simples especificar um array multidimensional*
- *Exemplo: array contendo as notas de 3 provas de 30 alunos*
 - *`int[][] notas = new int[30][3];`*

Exercícios: Conversão de cores (1/2)

- Diferentes sistemas são utilizados para representar cores
- Por exemplo, o sistema mais comum para representação de cores em display LCD, câmeras digitais e páginas web conhecido como sistema RGB, especifica os níveis de vermelho(R), verde(G) e azul(B) em uma escala de 0 a 255
- O sistema utilizado na publicação de livros e revistas, conhecido como CMYK, especifica os níveis de ciano, magenta, amarelo e preto em uma escala de 0.0 a 1.0

Exercícios: Conversão de cores (1/2)

- Escreva, um programa Java que receba três inteiros r,g e b representando um cor no sistema RGB e imprima os valores das componentes c,m,y,k correspondentes no sistema CMYK
- Se r=g=b=0 então c=m=y=0 e k = 1, caso contrário utilize a fórmula abaixo:

$$w = \max(r / 255, g / 255, b / 255);$$

$$c = (w - (r / 255)) / w$$

$$m = (w - (g / 255)) / w$$

$$y = (w - (b / 255)) / w$$

$$k = 1 - w$$

Exercícios: Padrão de divisores

- Escreva um programa Java que receba um inteiro N e imprima um tabela $N \times N$ com um asterístico na linha i e coluna j se ou i divide j ou j divide i .

Exercícios: Padrão de divisores

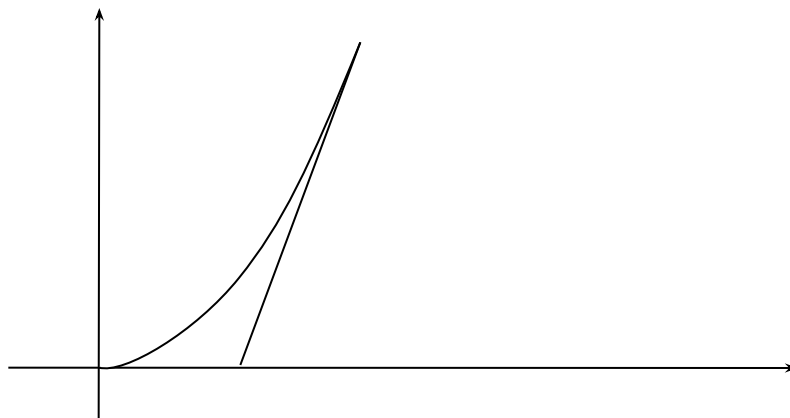
- Escreva um programa Java que receba um inteiro N e imprima um tabela $N \times N$ com um asterístico na linha i e coluna j se ou i divide j ou j divide i .

Exercícios: Fatoração de inteiros

- Escreva um programa Java que receba um inteiro N e imprima sua fatorização (sequência de inteiros primos que multiplicados iguala a N)

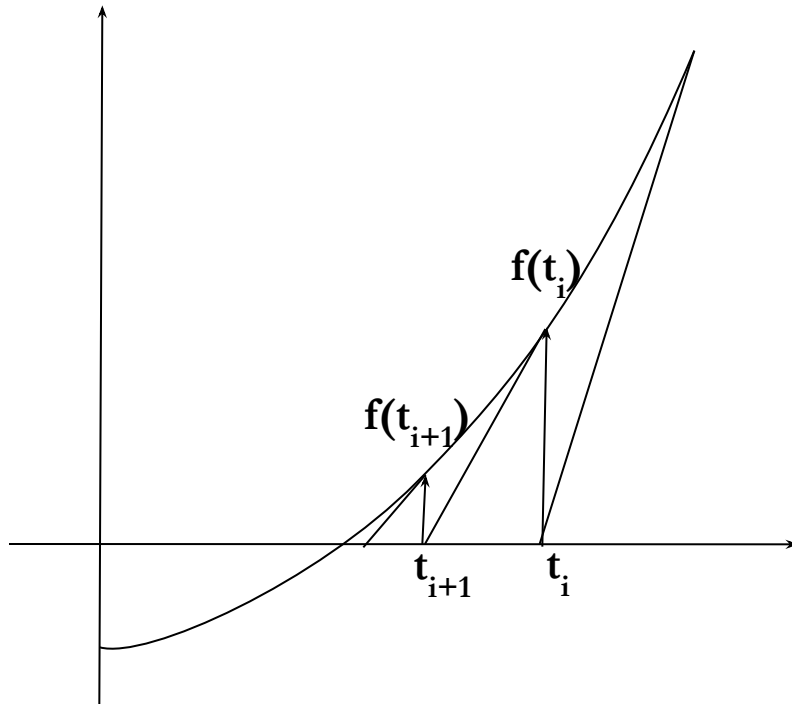
Exercícios: Cálculo de raiz quadrada

- Escreva um programa que ache a raiz quadrada de um número real c utilizando método de Newton-Raphson.
- Sob certas condições, dada uma função $f(x)$, o método de Newton-Raphson é capaz de encontrar as raízes de uma equação $f(x) \{x|f(x)=0\}$.
- O método inicia com uma estimativa da raiz t_0
- A partir de uma estimativa t_i , compute uma nova estimativa t_{i+1} onde t_i é a interseção da linha tangente a curva y no ponto $(t_i, f(t_i))$ com o eixo das abscissas.



Exercícios: Cálculo de raiz quadrada

- O método inicia com uma estimativa da raiz t_0
- A partir de uma estimativa t_i , compute uma nova estimativa t_{i+1} onde t_i é a interseção da linha tangente a curva y no ponto $(t_i, f(t_i))$ com o eixo das abscissas.



$$f'(t_i) = \frac{f(t_i) - 0}{t_i - t_{i+1}}$$

$$f'(t_i)(t_i - t_{i+1}) = f(t_i)$$

$$t_{i+1} = t_i + \frac{f(t_i)}{f'(t_i)}$$

Exercícios: Cálculo de raiz quadrada

- Computar a raiz de um número equivale a achar a raiz da função $f(x)=x^2-c$
- Considere uma estimativa inicial $t_0 = c$
- Se $t_i * t_i - c = \text{epsilon}$ então tome t_i como raiz de c

Exercícios: Fatorização

- Escreva um programa que receba um número inteiro N e gere todos os fatores primos de N
- Escreva uma versão mais eficiente do seu algoritmo

Exercícios: Sequencia de símbolos

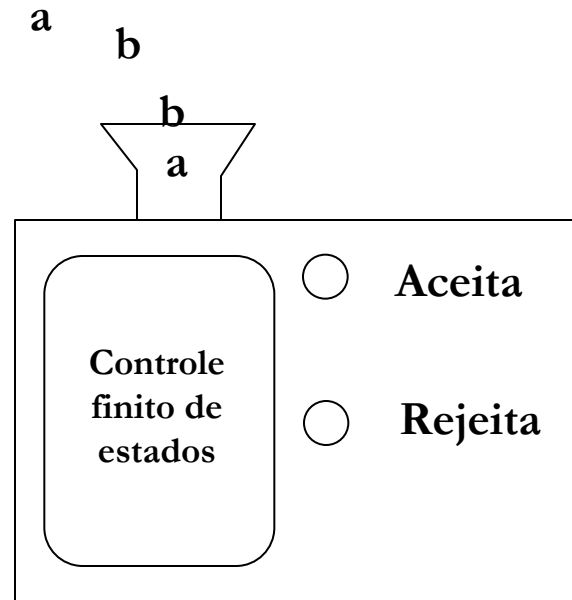
- Escreva um programa em Java que receba uma sequencia de letras da linha de comando formada por um conjunto de símbolos e gere as seguintes mensagens:
 - Para uma letra **a** “Soco”
 - Para uma letra **b** “Chute lateral”
 - Para duas letras **a’s** consecutivas “Chute circular”
 - Para uma letra **a** seguida de uma letra **b** imprime a mensagem (“dragon punch”)

Exercícios: Sequencia de símbolos

- O problema pode ser resolvido através de uma automato finito determinístico, que é um modelo para definição de linguagens regulares composto de cinco elementos: $\langle \Sigma, S, s_0, \delta, F \rangle$, onde:
 - Σ é o alfabeto sobre o qual a linguagem é definida;
 - S é um conjunto finito de estados não vazio;
 - s_0 é o estado inicial, $s_0 \in S$;
 - $\delta: S \times \Sigma \rightarrow S$ é a função de transição de estados;
 - F é o conjunto de estados finais $F \subseteq S$

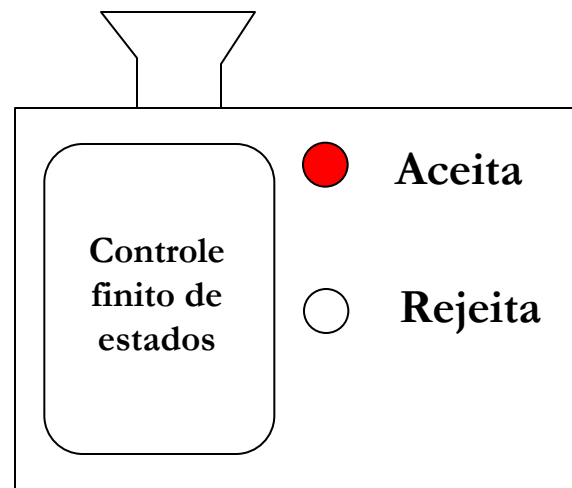
Exercícios: Sequencia de símbolos

- Um AFD é uma máquina reconhecedora de cadeias que pertencem a linguagem
- Ele recebe uma cadeia e diz se ela pertence ou não a linguagem modelada
- Ele possui um controle de estados S
- O CFE sempre coloca a máquina em um estado pertencente a S



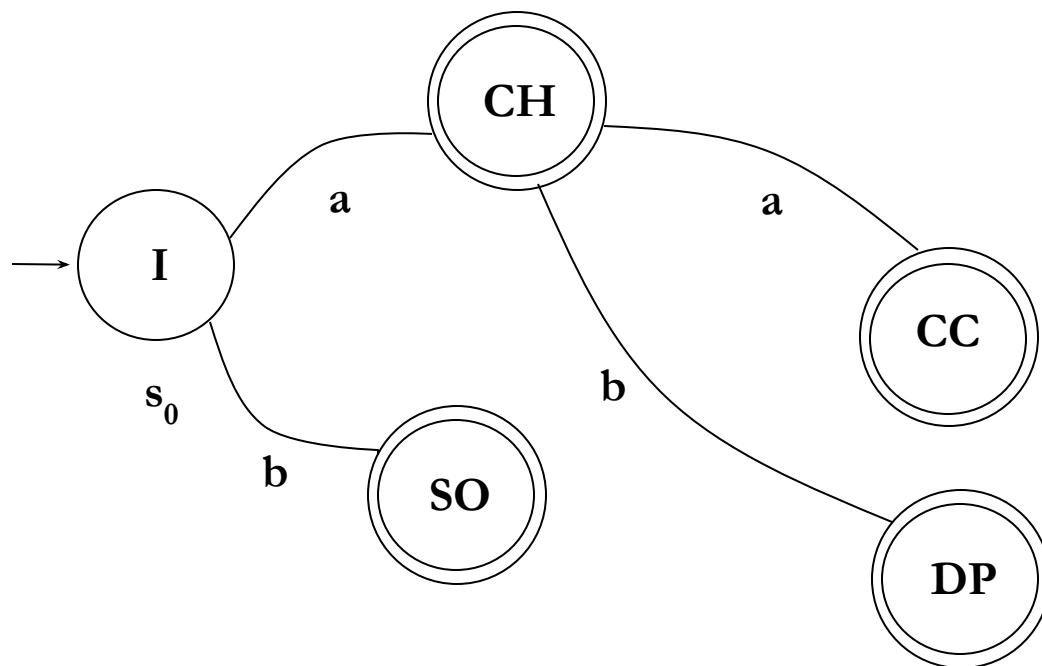
Exercícios: Sequencia de símbolos

- A função F diz como a máquina deve mudar de estado, à medida em que os símbolos da cadeia são analisados
- Após processar todos os símbolos e realizar as mudanças determinadas o AFD aceita ou não a cadeia
- Uma cadeia é rejeitada quando o autômato para em um estado que não é final
- Referência: Como construir um compilador utilizando ferramentas Java – Márcio Delamaro – novatec)



Exercícios: Sequencia de símbolos

- Autômato finito determinístico para o problema da sequência de símbolos



S	a	b	mensagem
I	C	B	
CH	CC	DP	Chute
SO			Soco
CC			Chute circular
DP			Dragon Punch

Exercícios: Caminhos aleatórios sem auto-interseção

- Suponha que você abandone seu cão no meio de uma grande cidade cujas ruas formam uma estrutura de reticulado
- Considera-se que existam N ruas na direção norte-sul e M na direção leste-oeste
- Com o objetivo de escapar da cidade, o cão faz uma escolha aleatória de qual direção ir em cada interseção, mas sabe através do faro como evitar visitar um lugar previamente visitado
- Apesar de tudo é possível que o cão fique perdido em um beco sem saída onde a próxima escolha obrigatoriamente leva a um lugar já percorrido

Exercícios: Caminhos aleatórios sem auto-interseção

- Escreva um programa Java que receba como parâmetros a largura e altura do reticulado e simule o caminho percorrido por um cão T vezes
- O programa deve determinar o número de vezes em que o cão fica sem saída