

ALGORITMOS PROBABILÍSTICOS

INTRODUÇÃO

ALGORITMOS

- A grande maioria dos algoritmos estudados no início do curso de Engenharia são algoritmos determinísticos.
- Um algoritmo, intuitivamente, é uma sequência finita de instruções ou operações básicas cuja execução, em tempo finito resolve um problema computacional qualquer que seja a sua instância.
- Para desenvolver um algoritmo pode ser feito usando-se técnicas top down que consiste em identificar partes ou etapas na estratégia de resolução do problema, elaborando inicialmente um esboço da resolução.

DESCRIÇÃO DE UM ALGORITMO

- A descrição de um algoritmo pode ser feita através de um pseudocódigo ou através de fluxogramas no fluxograma, cada operação básica entre parênteses instrução é representado por um desenho ponto na linguagem algorítmica cada operação básica é escrita em uma linguagem semelhante à linguagem natural com algumas regras comuns as linguagens de programação.
- Os fluxogramas são preferíveis em alguns casos por permitir uma visualização global do processo de resolução e de suas partes.
- A linguagem algorítmica, por sua vez, apresenta outras vantagens: é mais fácil escrever do que desenhar e a codificação em linguagem de programação acaba se tornando uma simples transcrição de palavras.

PROJETO DE PROGRAMA

- A partir de estratégias de resolução do problema desenvolvemos um algoritmo e definimos as estruturas de dados que serão utilizadas objetivando otimizar o binômio tempo espaço isso é visando obter um programa que apresente um tempo de execução mínimo com melhor aproveitamento de espaço de memória.
- A implementação de um algoritmo é a sua codificação em uma linguagem de programação.
- O algoritmo pode ser implementado em qualquer linguagem de programação e essa implementação pode ser trivial como uma simples transcrição de operações básicas ou trabalhosa dependendo principalmente das características da linguagem escolhida e dos tipos de dados nela definidos.

ESTRUTURA LÓGICA DE UM ALGORITMO

- Do ponto de vista lógico longo ritmo é constituído por 3 estruturas lógicas sequencial, repetitiva, e seletiva. Isto é, um algoritmo pode ser constituído por qualquer combinação destas 3 estruturas.
- Qual será a estrutura de um algoritmo que deve somar dois números?
 - O ser humano interpreta essa situação como sendo uma estrutura simples.
 - Leitura de dados,
 - Soma dos valores lidos,
 - Saida do resultado.

ALGORITMO DE ADIÇÃO

- Objetivo: dados dois números, calcular a sua soma.

- Entrada: Variável A, Variável B

- Saída: Soma das variáveis.

- Adição ();

```
leia(A,B);  
SOMA <- A+B;  
Imprima(SOMA)
```

- O nome identifica qual o algoritmo que está sendo escrito.

ALGORITMO PARA ENCONTRAR O ULTIMO ALGARISMO DE UM NUMERO

- Ao escrever um número inteiro no sistema de numeração decimal, o ultimo algarismo da direita representa a unidade. Lembre-se de que utilizamos a base 10 para nosso sistema matemático, e se o número tiver mais de um dígito com certeza o digito da direita é a nossa unidade.
- Sendo assim para encontrar a unidade de qualquer número, basta calcular o resto da divisão do número fornecido por 10. Este resto é exatamente o resultado desejado.

Ultimo ();

```
leia (A);  
Unidade <- A MOD 10;  
Imprima (Unidade)
```

ESTRUTURAS DE REPETIÇÃO

- Como devemos agir para obter a soma dos N primeiros números pares positivos.
 - Lembrando que N é um valor fornecido, sendo um inteiro positivo

$$N=6 \rightarrow 2+4+6+8+10+12 = 42$$

$$N=3 \rightarrow 2+4+6 = 12$$

$$S = \sum_{i=1}^n 2i$$

Para calcular essa soma podemos simplesmente guardar cada resultado parcial obtido, sem ter a necessidade de guardar todos os números utilizados. Basta guardar o último resultado acumulado para depois somá-lo ao número par seguinte, repetindo até somar o último número.

ESTRUTURAS DE REPETIÇÃO

Esboço do Algoritmo

Entrada de dados (valor de n)

Repetir n vezes

Calcular um número par
acumular numa soma

Saída do resultado

Soma Par ();

Leia (n);

S \leftarrow 0;

Para i de 1 até n repita

T \leftarrow 2 x i;

S \leftarrow S + t;

Imprima (S)

ESTRUTURAS DE REPETIÇÃO

Imprimir os n primeiros números ímpares em ordem oposta.

Geramos números ímpares desde o n-ésimo até o primeiro. À medida que os números são gerados, eles são impressos, para isso utilizaremos a instrução para-repita com acréscimo unitário negativo para descrever um loop em que a variável de controle assume valores na ordem decrescente.

ImprimaImparInvertido();

```
    Leia (n);  
    Para i de n até 1 passo -1 repita
```

```
        T <- 2 x i - 1;  
        Imprima(t)
```

ESTRUTURAS DE REPETIÇÃO

Imprimir os n primeiros termos da sequência matemática abaixo.

$$Y_{k+1} = Y_k + 2, k = 1, 2, 3, \dots \text{ com } Y_1 = 1$$

numerolmpar();

```
    Leia (n);  
    Y <- 1; imprima(y);  
    Para i de 1 até n-1 repita
```

```
        Y <- y+2;  
        Imprima (y)
```

EQUAÇÃO DA RAIZ QUADRADA

A sequência abaixo:

$$X_{n+1} = \frac{1}{2} \left(X_n + \frac{A}{X_n} \right), X_0 = 1, n \in N$$

Converge para a raiz quadrada de A, sendo $A > 0$, Calcule um valor aproximado da raiz quadrada de um número dado A, após cinco interações.