
Estrutura de Dados

Revisão

Prof. Nilton Luiz Queiroz Junior

Algoritmos

- Algoritmos estão presente no dia a dia das pessoas;
 - Instruções para uso de medicamentos;
 - Passos necessários para fazer um sanduíche;
 - Como montar um aparelho qualquer;
 - etc...
- Mas como podemos definir o que é um Algoritmo?

Algoritmos

- Como definir o que é um algoritmo?
 - Uma descrição de um padrão de comportamento, expresso em termos de um conjunto finito de ações;
 - Sequência finita de instruções ou operações cuja execução, em tempo finito, resolve um problema computacional, qualquer que seja sua instância;
 - Sequência de passos que visa atingir um objetivo bem definido;

Algoritmos

- Para fins computacionais algoritmos manipulam objetos em função do tempo;
 - Os objetos são parte do algoritmo;
 - Os objetos tem funções específicas;
 - As ações de um algoritmo devem ser direcionadas à objetos específicos;
 - Descrevem o estado computacional ditado pelo algoritmo;
 - Os objetos são também conhecidos por variáveis;

Variáveis

- Toda variável tem um conjunto de características
 - Nome (identificador);
 - Célula de memória;
 - Valor associado;
 - Tipo;

Tipos de dados

- Dentre os tipos de uma variável temos:
 - Tipos primitivos:
 - Inteiros;
 - Reais;
 - Booleanos;
 - etc..
 - Tipos estruturados:
 - Agregados homogêneos;
 - Agregados heterogêneos;
- Onde cada tipo de dados tem suas operações e valores válidos;

Controle de fluxo

- No projeto de algoritmos muitas vezes é necessário o uso de estruturas que alteram o fluxo:
 - Estruturas de Seleção:
 - Se-então;
 - Se-então-senão;
 - "Case of";
 - Estruturas de repetição:
 - Por contagem:
 - Para-até
 - Por condição:
 - Enquanto-faça;
 - Repita-até

Subprogramação

- Muitas vezes por questões de abstração e reutilização de código os programas podem ser construídos a nível de módulos;
 - Subprogramação;
 - Função;
 - Procedimento;
- A modularização do programa aborda também outras questões como:
 - Escopo de objetos;
 - Local;
 - Global;
 - Passagens de Parâmetro:
 - Por valor;
 - Por referência;

Recursividade

- Alguns subprogramas podem chamar a si mesmo para solucionar problemas;
 - Em outras palavras, uma chamada da função que está sendo implementada aparece no corpo dela mesmo;
- Subprogramas que chamam a si próprios para solucionar o problema são ditos recursivos;

Ponteiros

- Além dos tipos de dados já apresentados anteriormente existe também um tipo de dados que será frequentemente usado na implementação das estruturas de dados abstratas abordadas na disciplina: os ponteiros;
 - São dados que apontam para um outro dado;
 - Em outras palavras, o valor que um ponteiro armazena é um endereço de memória;
- Deve-se conhecer o tipo de dado que o ponteiro aponta;

Exercícios

1. Considere qualquer número natural a seguinte operação sobre qualquer natural:

$$f(n) = \begin{cases} \frac{n}{2} & \text{se } n \text{ é par} \\ 3n + 1 & \text{se } n \text{ é ímpar} \end{cases}$$

A conjectura de Collatz diz que se tal operação for aplicada repetidamente sobre qualquer número natural diversas vezes, sempre será alcançado o valor 1.

- a. Faça um algoritmo **iterativo** que conte a quantidade de passos que foram necessários para chegar ao valor 1 partindo de um número natural N.
- b. Faça um algoritmo **recursivo** que conte a quantidade de passos que foram necessários para chegar ao valor 1 partindo de um número natural N.

Exercícios

2. Faça uma função que receba dois vetores, A e B, e gere um novo vetor C, onde C é a primeira metade do vetor A concatenado a segunda metade do vetor B. Em caso de vetores com tamanho ímpar, considere como primeira metade as posições entre a primeira e a divisão do tamanho por 2 arredondada para baixo e segunda metade o restante dos elementos.

A

1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---

B

10	20	30	40	50
----	----	----	----	----

C

1	2	3	4	30	40	50
---	---	---	---	----	----	----

Exercícios

3. Um número natural, com n dígitos, é dito narcisista quando soma de todos seus dígitos elevado a n -ésima potência resulta no próprio número, por exemplo:

$$153 = 1^3 + 5^3 + 3^3$$

$$8208 = 8^4 + 2^4 + 0^4 + 8^4$$

Faça um algoritmo que receba um número e informe se ele é um número narcisista ou não.

Dicas:

Crie uma função para contar a quantidade de dígitos de um número;

Crie uma função para somar todos dígitos de um número elevado a uma determinada potência;

Use a função `pow` da biblioteca `math.h`

Exercícios

4. Declare um agregado heterogêneo chamado ponto contendo as coordenadas x, y e z de um ponto no espaço e em seguida:
 - a. Faça uma função que receba 3 valores e retorne um ponteiro para um ponto do espaço com suas coordenadas sendo os valores passados para a função. Obs: o ponto deve ser alocado dinamicamente.
 - b. Faça uma função para o cálculo de distância entre dois pontos.
 - c. Use as funções implementadas em (a) e (b) para criar 2 pontos e calcular a distância euclidiana entre eles.
5. Faça uma função **recursiva** que some todos elementos de um vetor.
Dica: Use um dos parâmetros da função como o índice do elemento a ser somado.