Capítulo 1

Aula 1

1.1 Algoritmos

Definição 1. Um algoritmo é uma sequência de instruções precisas para realizar uma computação ou resolver um problema.

Exemplo 2. Descreva uma algoritmo para exibir o valor máximo (maior) em uma sequência finita de inteiros.
Solução: Vejamos os passos:
Passo 1: Faça o máximo (temporariamente) igual ao primeiro inteiro da sequência.
Passo 2: Compare o próximo inteiro da sequência com o máximo temporário, e se este for maior faça dele o máximo temporário.
Passo 3: Repita o **Passo 2** para os demais inteiros da sequência.
Passo 4: Pare quando não restar mais inteiros na sequência. O máximo temporário (atual) é o maior inteiro da sequência.
Em linguagem simbólica, temos:

Procedimento: $max(a_1, a_2, \ldots, a_n : inteiros)$
1. $max = a_1$
2. para $i = 2$ **até** n
3. **se** $max < a_i$ **então** $max = a_i$
Retorne $max\{$ **é o maior elemento** $\}$

Um algoritmo deve gozar das seguintes propriedades:

1. ENTRADA: Um algoritmo possui um conjunto de entrada especificado.
2. SAÍDA: De cada conjunto de entrada, um algoritmo produz um conjunto de valores de saída específico.
3. DEFINITIVIDADE: Os passos de um algoritmo devem ser descritos com precisão.
4. EXATIDÃO: Um algoritmo deve produzir valores de saída corretos para cada conjunto de entrada.
5. FINITUDE: Deve ser possível executar cada etapa de um algoritmo em uma quantidade de tempo finita.
6. GENERALIDADE: O procedimento deve ser aplicável para todos os problemas da forma desejada, não apenas para um conjunto de valores particular de entrada.

Para Casa

Estude os seguintes problemas:
- The Linear Search (Pesquisa Linear): [Rosen, 194]
- The Binary Search (Pesquisa Binária): [Rosen, 194]
- Sorting (Classificação): [Rosen, 196]

1.2 O Problema da Parada

Vejamos um dos mais famosos teoremas da teoria computacional. Ele afirma que existe um problema que não pode ser resolvido usando qualquer procedimento. O problema que nós estudaremos é o problema da parada. Dada a entrada de um programa de computador e certa entrada, determinar se o programa para quando executado com essa entrada. Seria conveniente ter tal procedimento, se existisse. Em 1936 Alan Turing mostrou que tal procedimento não existe. Observe que não podemos rodar um programa e observar se este vai parar e quando vai parar. Vejamos a prova da não existência de tal programa.

Demonstração. Assuma que exista uma solução para o problema da parada cujo procedimento designaremos por $H(P, I)$, com duas entradas, o programa P e a entrada do programa I. $H(P, I)$ gera uma string 'para' se H determinar que P para quando submetido à entrada I. Do contrário $H(P, I)$ gera a string 'loop infinito' como saída. O que é uma contradição. □

1.3 Exercícios

Exercício 1. Construa um algoritmo para obter o máximo valor de uma lista de inteiros. Em seguida aplique-o à lista:

$$1,\ 8,\ 12,\ 9,\ 11,\ 2,\ 14,\ 5,\ 10,\ 4$$

Exercício 2. Construa um algoritmo que exiba a soma de todos os inteiros de uma lista. Em seguida aplique-o à lista:

$$5,\ 4,\ 7,\ 8,\ 3,\ 10$$

Exercício 3. Construa um algoritmo que tome como entrada uma lista de inteiros em ordem decrescente e produz uma lista de todos aqueles que aparecem mais de uma vez. Aplique-o à lista

$$12,\ 11,\ 11,\ 10,\ 6,\ 5,\ 5,\ 5,\ 2,\ 2,\ 1$$

Exercício 4. Construa um algoritmo que tome como entrada uma lista de n inteiros e exiba a localização dos inteiros pares nesta lista ou imprima 0 se não houver inteiros pares nesta lista. Aplique-o às sequências:

$$(1)\quad 3,\ 5,\ 4,\ 6,\ 7,\ 8$$

$$(2)\qquad 1,\ 7,\ 11,\ 9$$

Exercício 5. Construa um algoritmo que intercambeie os valores de duas variáveis x e y. Qual o mínimo de instruções é preciso para esta tarefa?

Exercício 6. Liste todos os passos para pesquisar o valor 9 na lista:

$$1,\ 3,\ 4,\ 5,\ 6,\ 8,\ 9,\ 11$$