

Algoritmos e ED 1

AULA 2

Neilor A. Tonin

Fatores a considerar na construção de um algoritmo

1. Complexidade:

- Percebeu-se, na medida em que colocávamos situações novas no problema a ser resolvido, que ia aumentando a complexidade do algoritmo.
- A complexidade tende a aumentar quando vamos incluindo situações novas em um problema, situações estas que devem ser tratadas pela solução. Embora a complexidade seja um mal necessário, devemos tentar diminuí-la ao máximo.

2. Legibilidade:

- Mede a capacidade de compreensão de um algoritmo por qualquer observador que não seja aquele que o construiu; a clareza com que a lógica está exposta.
 - Quanto mais legível, menos complexo...

Fatores a considerar na construção de um algoritmo

3. Portabilidade:

- A linguagem C++ será utilizada neste livro para implementação dos algoritmos. Juntamente com suas variantes (C#, objective C, etc) é uma das mais aceitas e mais utilizadas em todo o mundo.
- Se bem desenvolvido, um programa em linguagem C possui grande portabilidade. O mesmo código pode rodar sem nenhuma ou o mínimo de modificação em diferentes sistemas operacionais.

4. Método Cartesiano:

- A famosa frase de Descartes “Dividir para conquistar” é muito importante dentro da programação.
- É um método que ataca um problema grande, de difícil solução, dividindo-o em problemas menores, de solução mais fácil.
- Se necessário, divide-se novamente as partes não compreendidas. Método muito importante dentro da programação e é implementado através da criação de rotinas, que além de deixar o código mais limpo e claro, facilita a posterior reutilização.

Fatores a considerar na construção de um algoritmo

5. Planejamento reverso:

- Consiste em, a partir do resultado final, determinar quais são os componentes básicos. Ou seja, a partir da saída desejada, devemos poder determinar, de forma reversa, quais são os componentes da entrada de dados necessários.
- Por exemplo, se o desejado é um algoritmo para calcular as duas raízes de báskhara (denominadas aqui de $r1$ e $r2$). Quais são os valores de entrada que deverão ser fornecidos?

Metodologia para construir um algoritmo

Metodologia dos 5 passos

1. Ler atentamente o enunciado

- ▶ Deve-se reler o enunciado de um exercício várias vezes, até compreendê-lo completamente. A maior parte da resolução de um exercício consiste na compreensão completa do enunciado.

2. Retirar a relação das entradas de dados

- ▶ Através do enunciado, descobrimos quais são os dados que devem ser fornecidos ao programa.
- ▶ Na maioria das vezes os problemas informam de forma explícita qual deverá ser a entrada. Existem alguns casos raros de algoritmos que não possuem entrada.

Metodologia para construir um algoritmo

Metodologia dos 5 passos

3. Retirar a relação das SAÍDAS de dados

- ▶ Através do enunciado podemos descobrir quais são as informações que devem ser mostradas para compor o resultado final, objetivo do algoritmo.

4. Qual o processamento necessário?

- ▶ Aqui temos o algoritmo propriamente dito: que sequência de passos ou ações transformars nossos dados de entrada em informações de resultado?
- ▶ Para isso, utilizamos os fatores descritos anteriormente, tais como legibilidade, portabilidade, método cartesiano e planejamento reverso, e finalmente podemos construir o algoritmo.

Metodologia para construir um algoritmo

Metodologia dos 5 passos

5. TESTE é a palavra chave

- ▶ Do que adianta construir um algoritmo se a resposta que ele fornece não está correta?
 - Neste ponto deve-se analisar o formato que tem a saída, conferindo se está exatamente como pede o exercício ou problema. Isso diz respeito ao espaçamento, texto, formato dos valores, número de casas decimais, etc.
 - Acima de tudo, **nunca se baseie apenas pelos casos de teste fornecidos pelo programa**. Na maioria das vezes eles não cobrem todas as situações possíveis.

“Se um programa já foi testado e ao ser submetido ao portal URI Online Judge para correção retorna “Wrong Answer”, é necessário criar casos de testes adicionais com valores diferentes daqueles fornecidos (mas dentro dos limites especificados na descrição do problema), para verificar em que situação o programa não está apresentando a saída correta.”

Metodologia para construir um algoritmo

Complemento

- ▶ O estudante deve “parar para pensar” naquilo que está resolvendo.
- ▶ As novas gerações tendem a fazer várias coisas ao mesmo tempo sem focar realmente em algo específico. Isso pode acabar resultando em uma perda de detalhes que são extremamente importantes na construção de um algoritmo.
- ▶ Ao estruturar a forma inicial do algoritmo para calcular baskhara, por exemplo as seguintes perguntas podem ser feitas:
 - Foram utilizadas variáveis adequadas (inteiros, valores com ponto decimal, etc)?
 - Em que situação não é possível calcular as raízes da equação? O algoritmo prevê isso?
 - Quantos valores devem ser apresentados? Em que formato? Com quantas casas decimais?

Fórmula de Bhaskara

Adaptado por Neilor Tonin, URI  Brasil

Timelimit: 1

1036



Descrição

Tela Cheia

Enviar

Blocks **BETA**

Ranking

Fórum

uDebug

INICIANTE

SUGESTÃO

Sugerir a categoria, nível e tópicos deste problema.

ENVIA FEEDBACK

RESOLVIDO



Rank: 5°

Run: 3301

Tempo: 0.000

MINHA SOLUÇÃO

Leia 3 valores de ponto flutuante e efetue o cálculo das raízes da equação de Bhaskara. Se não for possível calcular as raízes, mostre a mensagem correspondente *"Impossível calcular"*, caso haja uma divisão por 0 ou raiz de numero negativo.

Entrada

Leia três valores de ponto flutuante (double) A, B e C.

Saída

Se não houver possibilidade de calcular as raízes, apresente a mensagem "Impossível calcular". Caso contrário, imprima o resultado das raízes com 5 dígitos após o ponto, com uma mensagem correspondente conforme exemplo abaixo. Imprima sempre o final de linha após cada mensagem.

Exemplos de Entrada	Exemplos de Saída
10.0 20.1 5.1	R1 = -0.29788 R2 = -1.71212
0.0 20.0 5.0	Impossível calcular
10.3 203.0 5.0	R1 = -0.02466 R2 = -19.68408
10.0 3.0 5.0	Impossível calcular

Técnicas para construir algoritmos

a) Divisão do problema em Entrada, Processamento e Saída

Esta técnica consiste em dividir todo o problema em três partes, relacionar o que é entrada, o que será saída e quais os cálculos necessários para se chegar ao resultado. É aplicável à maioria dos problemas e é uma técnica bastante simples.

Relacionar as entradas e seus tipos:

a, b, c : todos valores reais.

**Determinar o
processamento ou
Cálculos necessários:**

$$\Delta = b^2 - 4.a.c$$

$$R1 = \frac{-b + \sqrt{\Delta}}{2.a} \quad R2 = \frac{-b - \sqrt{\Delta}}{2.a}$$

Relacionar as saídas e tipos:

$R1, R2$: ambos os valores reais.

Técnicas para construir algoritmos

b) Sequência de Passos.

Semelhante à metodologia anterior e pode ser aplicada após a mesma e é um pouco mais detalhada. Define-se uma sequência de passos e cuida-se para não esquecer nenhum deles pois caso contrário, o resultado obtido pode ser bem diferente do resultado esperado. Este é um método muito utilizado. Para o problema em questão, a listagem dos passos seria:

- Definir e criar variáveis: `a, b, c, R1, R2, delta: todos valores reais`
- Ler a entrada fornecida armazenando nas respectivas variáveis: `ler a, b, c`
- Fazer cada um dos cálculos necessários, inclusive de variáveis auxiliares se for o caso:

$$\Delta = b^2 - 4.a.c$$

$$R1 = \frac{-b + \sqrt{\Delta}}{2.a} \quad R2 = \frac{-b - \sqrt{\Delta}}{2.a}$$

$$\text{Delta} = b * b - 4 * a * c$$

$$R1 = -b + \dots$$

$$R2 = -b - \dots$$

- Apresentar cada uma das variáveis que foi utilizada para armazenar os resultados dos cálculos

Apresentar R1 e R2 no formato especificado pelo problema.

Técnicas para construir algoritmos

c) Metodologia de Fluxogramas

Outra técnica muito explorada em muitos livros de algoritmos é a utilização de Fluxogramas. O Fluxograma é uma técnica gráfica que pode ser visualmente interessante quando os algoritmos se tornam um pouco mais complexos. Através do fluxograma fica fácil ver aonde existe um desvio condicional e quais são as ações realizadas de acordo com cada tomada de decisão.

No exemplo trabalhado aqui, o fluxograma é extremamente simples, **pois não estão sendo consideradas ainda as condições nas quais não é possível calcular as raízes de báskhara**, ou seja, o nosso algoritmo é apenas uma sequência de passos.

