

Introdução

Nesta aula:

- Primeiro contato com o conceito de algoritmos
- Quais são as partes fundamentais de um algoritmo
- Como criar um algoritmo
- Como que um algoritmo nos faz:
 - Pensar de forma astúcia em problemas
 - Encontrar a melhor solução para eles
- Fluxogramas (forma de representar um algoritmo)

Quais são os objetivos da aula?

- Explicar o conceito de um algoritmo
 - Analisar e resolver um problema do mundo real com um algoritmo
 - Usar o conceito de abstração para criar algoritmos
 - Compreender fluxogramas e criá-los a partir de algoritmos
-

O Que São Algoritmos?

Algoritmos:

- Na computação:

Lista encadeada de comandos (instruções)

- Não se tem uma definição foral única
- Cormen define como:

Qualquer procedimento computacional bem definido que toma um valor ou conjunto de valores como entrada e produz um conjunto de valores como saída. Portanto é uma sequência de passos computacionais que transformam a entrada em saída.

- Essa definição trata da parte técnica, voltado diretamente para a computação
- Podemos também considerar muitas das atividades que realizamos no nosso dia a dia como algoritmos:

Algoritmo é uma sequência de passos ordenados e finitos, que visam resolver um problema

Passos:

1. Ordenados

- Ordem entre os passos
 - Mudança na ordem influência na execução do algoritmo
2. Finitos
 - O algoritmo não pode ter uma quantidade infinita de passos
 3. Precisos
 - Sem ambiguidade

Características de um Algoritmo:

1. Tem um nome
2. Tem início e fim
3. Resolve um problema
4. Tem entrada de dados
5. Produz saída de dados
6. Passos e tempo de execução finitos
7. Passos precisos sem ambiguidade

Algoritmos são como uma receita, quais passos devemos executar para resolver um problema / alcançar um objetivo

Outras características

1. Algoritmos não são exclusivos para computadores
2. Na computação são a base para criar programas

Função de um algoritmo

- O computador não é dedutivo
- Precisamos dar a ele instruções explícitas e claras
- Os algoritmos nos dão a base do que devemos escrever em linguagem de programação
- Sequência de traduções:
 1. Criação de um algoritmo
 2. Tradução do algoritmo para linguagem de programação
 3. Tradução da linguagem de programação
- O algoritmo afetará muito mais a qualidade do programa no fim do que o conhecimento sobre linguagens
- O que realmente altera a fundo a eficiência de um programa é bom desenvolvimento de um algoritmo, e não a pro-eficiência em alguma linguagem de programação por si só:
 - Acho que da para comparar com a lingua falada, o que faz um texto genial é as ideias e os conceitos que o texto está transmitindo e não a lingua em si em que foi escrito

- É muito menos custoso consertar erros nos planos de projetos do que após o projeto já ter sido implementado (em qualquer área da vida)
- Da mesma forma é menos custoso consertar erros ao montar um algoritmo do que depois que o código fonte já está pronto

Compilação x Interpretação:

Compilação e interpretação são formas de tradução de um código fonte para linguagem de máquina, eles variam apenas no aspecto de quando fazem a tradução

Compilação	Interpretação
Previamente	Na hora da execução

Construção de Algoritmos

- Quais são seus elementos fundamentais?
- Como organizar os passos da construção de um?
- Compreender bem o objetivo de um algoritmo é fundamental:
 - Nos permite identificar corretamente os elementos-chave:
 - * Entradas
 - * Saídas
 - E abstrair tudo o que for desnecessário
- Processo criativo:
 - Análise da tarefa que será realizada
 - Identificação do que é necessário para realizá-la
 - Criação de passos que executa a tarefa com sucesso
 - Produção do resultado esperado

Não existe uma regra fixa para criação de algoritmos, isso é só um guia

Conceito de memória

- Basicamente: ***variáveis***
- Computadores precisam de instruções *explicitas* para usar sua memória (declarar e armazenar valores em memória)
- Quanto mais complexo um algoritmo ele tende a precisar de mais memória

Qualidade de Algoritmos

- Geralmente existem várias maneiras de resolver um problema
- Devemos considerar as diversas opções e escolher o que for mais conveniente para cada caso:
 - Maior eficiência
 - Menor tempo de execução
 - Maior simplicidade
 - ...

Cuidados especiais com algoritmos

- Uma vez que um algoritmo está pronto deve ser testado exaustivamente para garantir que:
 - Os passos serão executados na ordem certa
 - O resultado final / a saída será o que esperamos
 - Adicionar condições que previnam erros encontrados nos testes

Testes de Algoritmo

- Algoritmos podem conter muitas falhas
- Para encontrar falhas devemos **testar os algoritmos**
- Testar:
 - Validar se um algoritmo realmente faz o que promete
 - Verificar se ele produz os resultados esperados para todas as situações (valores de entrada) às quais pode ser submetido
- Como testar:
 1. Crie um conjunto de dados de entrada
 2. Execute o algoritmo
 3. Verifique se a saída está de acordo com o resultado esperado:
 - Se não for, o algoritmo possui erros
- Classes de entrada:
 - Devemos buscar todos os tipos de classes de entrada para algum algoritmo
 - Cada classe representa um tipo de entrada
 - Criar um conjunto de valores para cada classe
 - Cada conjunto é chamado de **caso de teste**:
 - Quantos mais casos de testes melhor:
 - * Priorizar buscar classes diferentes para cada caso de teste

A Importância dos Algoritmos

- Porque não escrever direto em vez de criar o algoritmo primeiro?
 - Isso resulta em programas com falhas, desempenhos ruins, ou nem sequer funcionais

- O processo de escrever o algoritmo primeiro antes do código ajuda a identificar falhas mais cedo, e perceber detalhes menos aparentes de um problema
- Ajustes no algoritmo são mais fáceis do que no código fonte

Introdução a Fluxogramas

Objetivo:

- Qual é a função de um fluxograma
- Qual é a estrutura básica dele

Fluxograma

- Outra forma de representar um algoritmo
- São diagramas indicados pra representar eventos dinâmicos (que seguem um fluxo)
- Forma mais simples de um algoritmo:
 - Circulo: início ou fim do algoritmo
 - Retângulo: Um passo do algoritmo (uma instrução)
 - Seta: Indica o caminho do fluxo entre os elementos do fluxograma

Função do Fluxograma

- Fluxogramas trazem uma representação visual para