

Técnicas de Programação

Introdução

O que é um algoritmo?

Algoritmo: sequência de passos (finita) para resolver um problema

Algoritmos

- é garantido que acaba
- dá a resposta correta 100% das vezes
- cada passo é bem definido
- dada uma entrada, devolve sempre a mesma resposta
- independe da linguagem de programação usada

Algoritmos não são

1. misteriosos
2. imprevisíveis
3. inexplicáveis
4. maliciosos

Desenvolver algoritmos é resolver problemas

Que tipo de problemas?

1. Problemas de busca/decisão

- Essa coleção de dados contém X?
- Dadas essas restrições, a situação Y é possível?

2. Problemas de Ordenação

- Ranqueamento
- Comparações, Igualdades

3. Problemas de otimização

- menor X, dado um conjunto de restrições
- maior Y, dado um conjunto de restrições

Como descrever a solução de um problema?

Como descrever a solução de um problema?

1. Mudaremos a linguagem usada (Java)
2. Escreveremos algoritmos em *Pseudo Código*

descrição de um algoritmo usando estruturas de controle simplificadas com o objetivo de ser legível para pessoas. Pode incluir trechos em linguagem natural e fórmulas matemáticas se isso ajudar a compreensão.

Como descrever a solução de um problema?

Escrever solução de maneira que uma pessoa sem conhecimento específico do algoritmo em questão consiga simular sua execução

Estratégias de solução de problemas computacionais

- estruturas de dados
- divisão e conquista
- buscas por largura e profundidade
- backtracking

Adaptar algoritmos clássicos que usem essas estratégias para resolver novos problemas

Burocracias

Organização do curso

- 7 módulos
 - motivação com problema "real"
 - exercícios de sala (pseudo código)
 - APS para entregar (código Java)
- Nota final

$$NF = 35\%APS + 25\%PI + 40\%PF$$

0. Algoritmos

1. Tipos Abstratos de Dados

2. Busca em Arrays

3. Ordenação

PI

4. Algoritmos em Strings

5. Caminhos entre pontos

6. Otimização com restrições

PF

Horários

- **QUA:** 13:30
- **SEX:** 13:30
- Atendimento **QUA** 15:45

Objetivos de Aprendizagem (formal)

1. Implementar em Java um algoritmo descrito em alto nível
2. Empregar Backtracking para resolver problemas computacionais
3. Empregar a técnica Divisão e Conquista para resolver problemas computacionais
4. Empregar estruturas de dados lineares (listas, matrizes, pilhas, filas, mapeamentos, conjuntos) para resolver problemas computacionais de maneira eficiente
5. Identificar como estratégias computacionais clássicas (busca, ordenação, otimização) podem ser adaptadas para resolver novos problemas computacionais
6. Estimar a complexidade computacional de um algoritmo usando uma argumentação informal baseada na contagem de vezes que uma linha executa

Atividades: Módulo 00-Algoritmos

(mostrar site e projeto configurado do VSCode)