



ESTRUTURA DE DADOS

Aula 01 - Apresentação

Apresentação

- **Objetivos:** Desenvolver algoritmos elementares para solução de problemas de ciência de dados, por meio da compreensão e aplicação das estruturas de dados básicas da computação e das técnicas para sua representação em linguagem de programação e métodos de operação sobre as mesmas. Analisar e avaliar as estruturas adequadas para o armazenamento de dados em memória.

Apresentação - Ementa

Listas simples. Listas ordenadas. Listas encadeadas. Listas circulares. Pilhas. Filas.

Árvores. Árvores binárias. Algoritmos de Pesquisa e Ordenação. Hash.

Noções de balanceamento. Algoritmos recursivos em listas e árvores.

Noções de complexidade computacional.

Grafos. Algoritmos básicos em grafos: busca primeiro na extensão, busca primeiro na largura, ordenação topológica, componentes fortemente conectados.

Bibliografia Básica:

- CORMEN, T. H. LEISERSON, Charles E. RIVEST, Ronald. Algoritmos: Teoria e Prática. 3. ed. São Paulo, SP: Elsevier, 2012.
- FORBELLONE, A. L. Lógica de Programação. 3ª Ed. Editora Prentice Hall Brasil. 2015.
- WAZLAWICK, R. S. Introdução a Algoritmos e Programação com Python. 1. ed. São Paulo, SP: Elsevier, 2017.

Bibliografia Complementar:

- RAMALHO, L. Python Fluente: programação clara, concisa e eficaz. 1. ed. São Paulo, SP: Novatec, 2015.
- MENEZES, N. N. Coutinho. Introdução à Programação com Python: algoritmos e lógica de programação para iniciantes. 3. ed. rev. e atual. São Paulo, SP: Novatec, 2019.

Apresentação

Avaliação

- A avaliação será contínua para evidenciar os fatores de participação, presença e envolvimento no contexto geral na realização das atividades individualmente ou em grupo, para isso serão usadas avaliações formativas e somativas.
 - **Avaliação Formativa:** Exercícios para prática. Análise e Resolução de Problemas acompanhado de rubrica de avaliação.
 - **Avaliação Somativa:** Provas. Projetos. Avaliação em pares e Trabalhos Interdisciplinares.

Avaliação

- **P1**

Avaliação Formativa: atividades (03 pontos)

Avaliação Somativa: avaliação P1 (07 pontos)

- **P2**

Avaliação Formativa: atividades (03 pontos)

Avaliação Somativa: avaliação P1 (07 pontos)

- **Média: $(P1 + P2) / 2$**

- Alunos que não atingirem a média 6,0, deverão realizar a avaliação **P3**, que será uma avaliação escrita individual, sem consulta e que englobará todo o conteúdo do semestre além da necessária entrega da parte prática.

Avaliação

- **Estratégias de Recuperação da Aprendizagem:**

Recuperação da Aprendizagem da parte teórica.

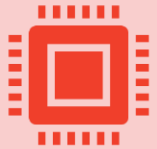
- a) Diagnóstico: análise das questões com maior índice de erros.
- b) Na vista de prova, usar o segundo horário da aula para fazer uma revisão com os alunos das questões com maior índice de erros.
- c) Na aula seguinte, reservar um espaço da aula para que os alunos possam responder às questões que tiveram maior dificuldades como meio de checagem se houve a recuperação da aprendizagem.

Recuperação da Aprendizagem da parte prática

- a) Como a parte prática consiste em elaborar aplicar ferramentas para a análise do ambiente interno e externo, o mesmo deverá ser realizado, caso não tenha feito com êxito.
- b) Para os trabalhos com fragilidades, os alunos terão após serem orientados quanto às fragilidades, até a aula seguinte para apresentar uma errata, evidenciando os pontos frágeis e as possíveis medidas corretivas.



Linguagem de programação: Python



Ambientes de desenvolvimento: IDLE, PyCharm, Jupyter, VS Code, Atom, Wing Python IDE, Thonny, Pydev, Elpy, Spyder Python IDE, Pydroid 3, etc.



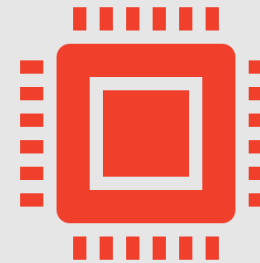
Aulas de Estrutura de Dados: IDLE, PyCharm.

Ferramentas

Apresentação



Professor:
César Torres Fernandes



Formação:
Mestrado Profissional em Tecnologia
Especialização em Redes de Computadores
Bacharel em Ciência da Computação
Licenciatura em Matemática

05/09/2022

INTRODUÇÃO

Estrutura de Dados

Ao projetarmos um algoritmo, uma etapa fundamental é a especificação e o projeto de seus dados de entrada. Esses dados são projetados pensando-se na aplicação e apresentarão tipos distintos.

Dentre os tipos primitivos de dados, citamos: inteiro, real, caractere e booleano.

Para cada uma dessas categorias, o espaço ocupado na memória será diferente e dependerá da arquitetura da máquina, do sistema operacional e da linguagem de programação, bem como da forma de manipulação desses dados e do seu tratamento pelo programa.

Estrutura de Dados

Estrutura de Dados

Os tipos de dados manipulados por um algoritmo podem ser classificados em duas categorias distintas: os dados primitivos (ou atômicos), que são dados indivisíveis, como inteiro, real, caractere ou lógico; dados complexos (ou compostos), que podem ser divisíveis em mais partes menores.

Todo o **dado atômico** é aquele no qual o conjunto de dados manipulados é indivisível, ou seja, você trata-os como sendo um único valor.

Dados complexos são aqueles cujos elementos do conjunto de valores podem ser decompostos em partes mais simples. Se um dado pode ser dividido, isso significa que ele apresenta algum tipo de organização estruturada e, portanto, é chamado de dado estruturado, o qual faz parte de uma estrutura de dados.

Estruturas de dados
homogêneas



```
graph TD; A[Estruturas de dados homogêneas] --> B[Estruturas de dados homogêneas são aquelas que manipulam um só tipo de dado.]; B --> C[Exemplos: vetor e matrizes];
```

Estruturas de dados
homogêneas são aquelas
que manipulam um só tipo
de dado.

Exemplos: vetor e matrizes

05/09/2022

Estrutura de
Dados

Estruturas de dados
heterogêneas



Existem estruturas
capazes de manipular
mais do que um tipo,
por exemplo,
empregando dados
inteiros e caracteres,
simultaneamente em
uma só estrutura. São
as chamadas
estruturas de dados
heterogêneas.

Estrutura de Dados

ANÁLISE DE COMPLEXIDADE DE ALGORITMOS

Quando desenvolvemos algoritmos, as possibilidades de solução do problema tendem a ser bastante grandes e distintas.

Iremos aprender diversos algoritmos distintos para solucionar problemas envolvendo manipulação de estruturas de dados. Desse modo, como poderemos saber qual deles é o mais eficiente para solucionar um determinado problema? Que parâmetros de desempenho devemos analisar?

ANÁLISE DE COMPLEXIDADE DE ALGORITMOS

Ao analisarmos o desempenho de um algoritmo, existem dois parâmetros que precisam ser observados:

Tempo de execução - quando tempo um código levou para ser executado;

Uso de memória volátil - a quantidade de espaço ocupado na memória principal do computador.

05/09/2022

FIM

cesar.fernandes3@fatec.sp.gov.br