

Estrutura de Dados

Matheus Werner

Sala 502

matheus.werner@fgv.br

Atendimento:

2as de 9h30 às 11h30

4as de 9h30 às 11h30

Informações do curso

- Material
<https://github.com/matwerner/fgv-ed/>
- Lista de Exercícios
<https://eclass.fgv.br/>

Motivação

- Até agora...
 - Em *Introdução à Computação*
 - *Conceitos de computação*
 - *Comandos básicos no Linux*
 - *Introdução a Python*
 - Em *Linguagens de Programação*
 - *Qualidade de Software*
 - *Orientação a Objetos*

Motivação

- Estrutura de Dados
 - Estrutura de dados
 - Algoritmos para manipular tais estruturas

Exemplo: Fila prioridade

- Pense no sistema de atendimento de um hospital, responsável por organizar a ordem de atendimento de diversas pessoas
- Cada pessoa é associada a um nível de severidade (prioridade) que pode ter como critérios: Horário de chegada, Idade, condição de saúde, etc
- Do ponto de vista computacional, como deveríamos implementar uma solução que diz qual é a próxima pessoa a ser atendida?



Conteúdo do curso

- Estrutura de dados
 - Estratégias básicas para organização de dados na construção de algoritmos
- Objetivo
 - Explorar e implementar estruturas de dados comuns, presentes em diversas aplicações e bibliotecas.
 - Entender como escolher a estrutura de dados mais adequada para uma aplicação.

Cronograma

- A1

- Linguagem C / C++
- Complexidade Computacional
- Tipo Abstrato de Dados (TAD)
 - Listas
 - Filas
 - Pilhas
- Ordenação
 - Insert
 - Select
 - Bubble

- A2

- Ordenação
 - Merge
- Árvores
 - Binárias de Busca
 - AVL
 - Rubro-negras ou B
- Heap
- Hash ← Talvez

Cronograma

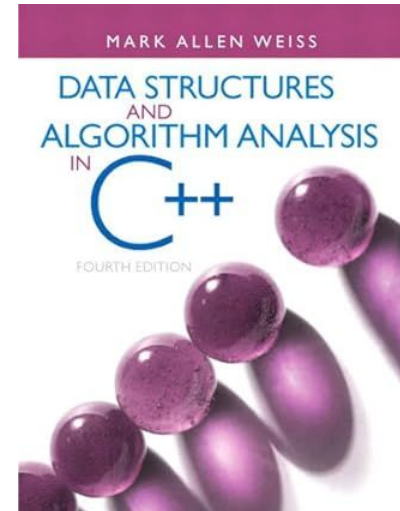
- Uma aula teórica e uma aula de implementação
(* Exceto: Linguagem C / C++)
 - Por favor, trazer laptops → Pair programming

Critério de Avaliação

- Nota final (NF) = $(A1 + A2) / 2$
 - Se $NF \geq 6.0$, sucesso.
 - Se não, AS
 - $NF = (\text{MAX}(A1, A2) + AS) / 2$
- Avaliação = 80% Prova + 20% Listas
 - **Observação**: Esperando a definição de se haverá monitor na disciplina

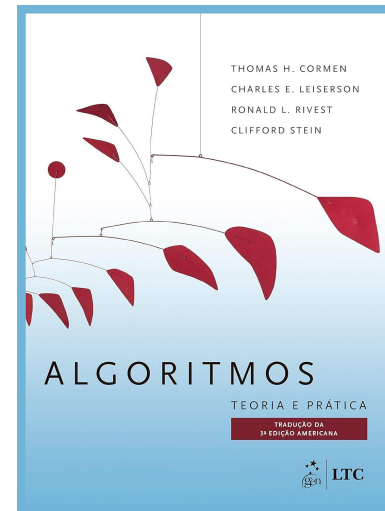
Bibliografia

- Mark A. Weiss. 2013. Data Structures and Algorithm Analysis in C++ (4rd Edition). Editora Pearson.
 - Cap 2: Complexidade Computacional
 - Cap 3: Listas, Filas, Pilhas
 - Cap 4: Árvores
 - Cap 5: Hash
 - Cap 6: Heap
 - Cap 7: Ordenação



Bibliografia

- Thomas H. Cormen. 2012. Algoritmos - Teoria e Prática (3ª edição). Editora GEN LTC.
 - Cap 2: Ordenação
 - Cap 3: Complexidade Computacional
 - Cap 6: Heap
 - Cap 10: Listas, Filas, Pilhas
 - Cap 11: Hash
 - Cap 12: Árvore Binária de Busca
 - Cap 13: Árvore Rubro-Negra e AVL
 - Cap 18: Árvore B



Outras referências

- Data structures and Algorithms in X minutes:
<https://www.youtube.com/@MichaelSambol>
- MIT 6.006 Introduction to Algorithms, Spring 2020
https://www.youtube.com/watch?v=ZA-tUyM_y7s&list=PLUI4u3cNGP63EdVPNLG3ToM6LaEUuStEY&index=1

Dúvidas?

Matheus Werner

Sala 502

matheus.werner@fgv.br

Atendimento:

2as de 9h30 às 11h30

4as de 9h30 às 11h30

Ok... por que C?

- Maior controle sobre a execução
 - Menos Abstração
 - Controle de Memória
 - Proximidade do Hardware
- Eficiência e Performance
- Base para outras Linguagens



Abhishek (e/acc) 
@Abhishekcure

Follow

- >Linux is C
- >Git is C
- >Python interpreter (CPython) is C
- >Ruby interpreter (MRI) is C
- >PostgreSQL is C
- >SQLite is C
- >Redis is C
- >MySQL is C++
- >MongoDB is C++
- >Unreal Engine is C++
- >Chrome is C++
- >Firefox is C++
- >Windows kernel is C
- >macOS kernel (XNU) is C
- >Photoshop is C++
- >VMware is C++
- >TensorFlow (Core) is C++

-still, you are not convinced to learn C and C++

13:46 · 23 Dec 24 · 712K Views

1,243 Reposts 192 Quotes 12.2K Likes