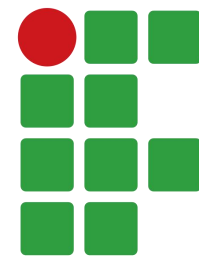


ALGORITMOS E ESTRUTURA DE DADOS

Curso de Engenharia de Software

Lucas Sampaio Leite



**INSTITUTO
FEDERAL**
Pernambuco

Alunos

Quem são vocês?

Qual é o seu background em programação?

Qual a sua experiência com C/C++?



Professor

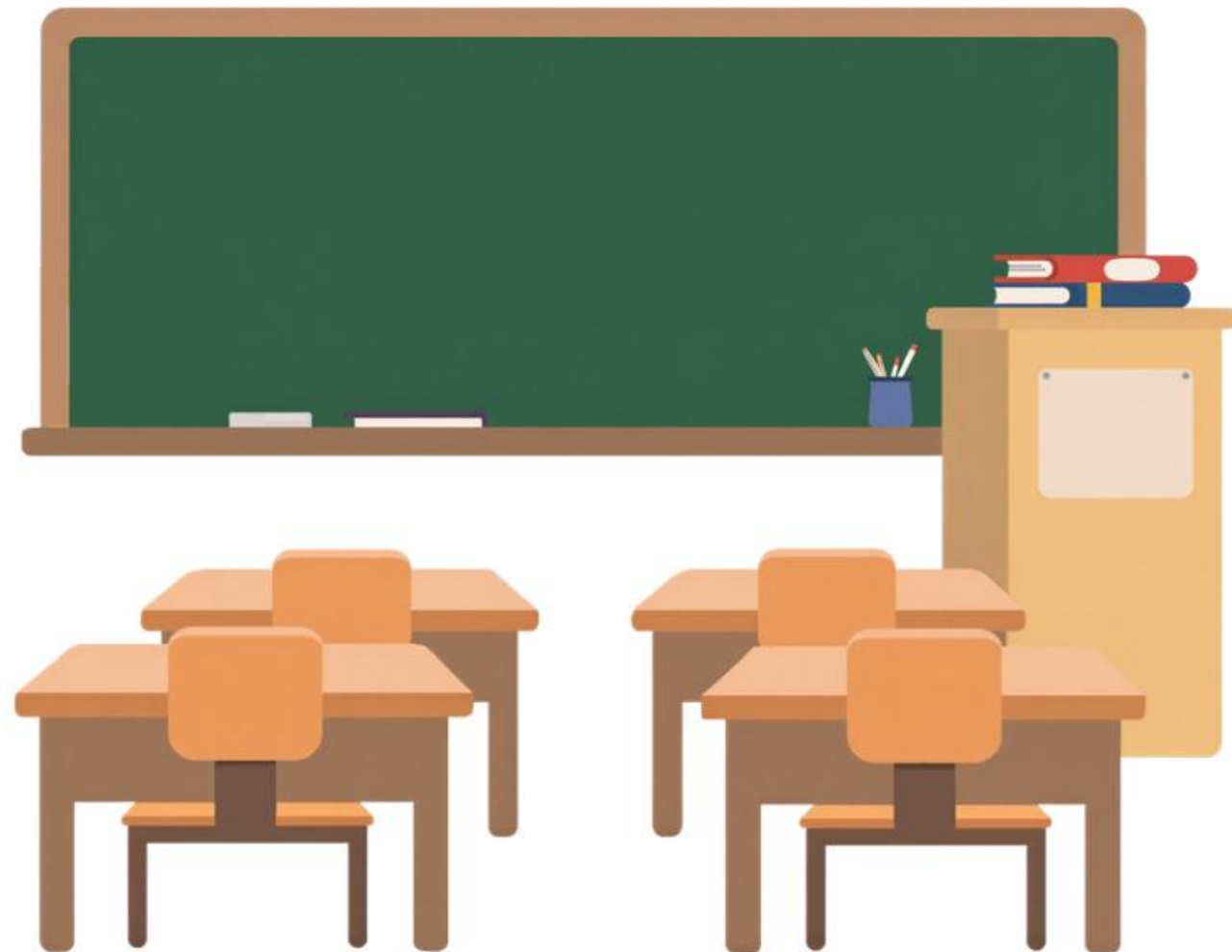
Lucas Sampaio Leite

lucas.leite@belojardim.ifpe.edu.br



A disciplina

- Quando serão as aulas?
 - Quintas-feiras das 16h40 às 18h10
 - Sextas-feiras das 13h30 às 15h00
15h10 às 16h40
- Carga Horária: 120h



Ementa

- Tipos de dados.
- Manipulação de estrutura de dados.
- Algoritmos de busca.
- Árvores no balanceamento de dados.
- Recursividade em programas modulares.

Conteúdo Programático

1. Introdução a tipo de dados e às Estruturas de Dados
2. **Alguma revisão???**
3. Listas Lineares: Conceitos Gerais e implementação a partir de vetores
4. Listas Encadeadas simples, duplas e circulares
5. Filas (Queue)
6. Pilhas (Stack)
7. Busca sequencial vs busca binária
8. Algoritmos de ordenação
9. Árvores: Estrutura e Conceitos Iniciais

Conteúdo Programático

10. Árvores Binárias

11. Travessias em Árvores Binárias

12. Árvores Binárias de Busca (BST)

13. Árvores Balanceadas

14. Árvores AVL

Ferramentas e ambientes de desenvolvimento

- Exemplos em Pseudocódigo, C/C++ e/ou Java
- IDEs: VS Code, Dev-C++, Codeblocks, etc - (livre)
- Implementações: C/C++
- Árbitro Virtual Beecrowd (listas de exercícios)
- Classroom?



Metodologia

- Aulas síncronas e presenciais
 - Aulas expositivas e dialogadas
 - Abordagem prática
 - Aulas de resolução de exercícios
- **Com chamada**

Avaliação

- Avaliação 1: Listas de exercício 30% + Prova 70%
- Avaliação 2: Listas de exercício 30% + Prova 70%
- Prova final



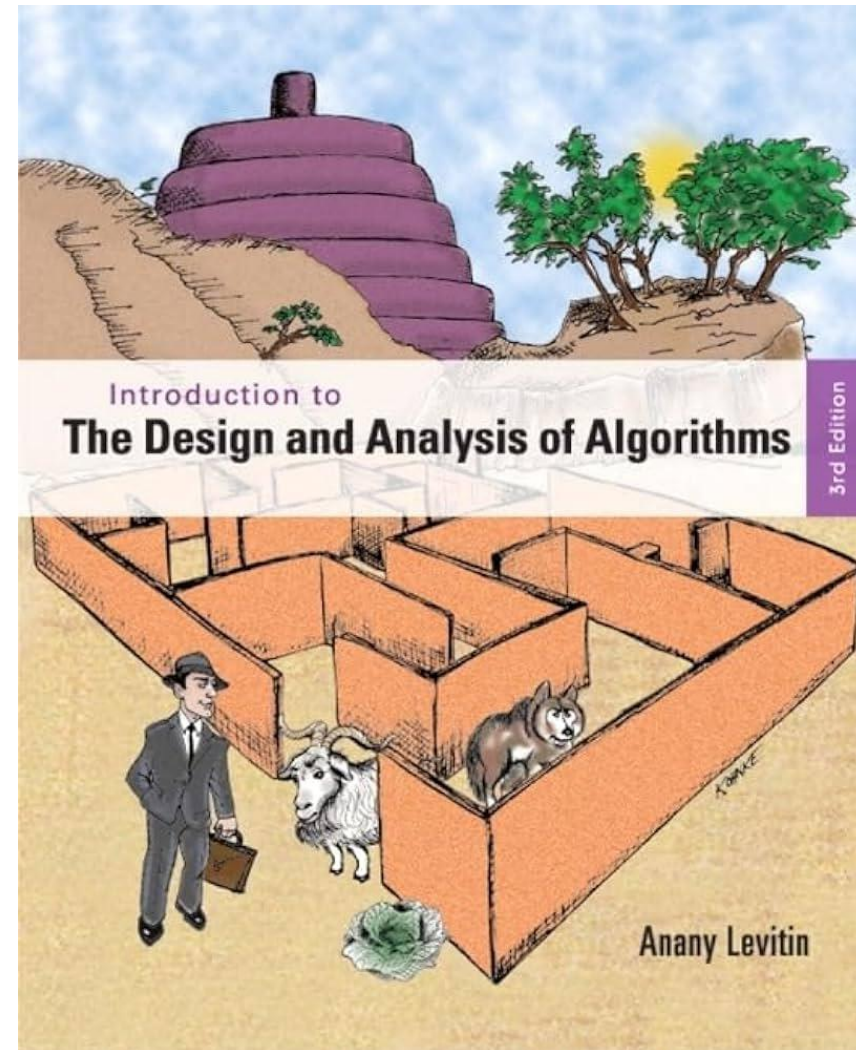
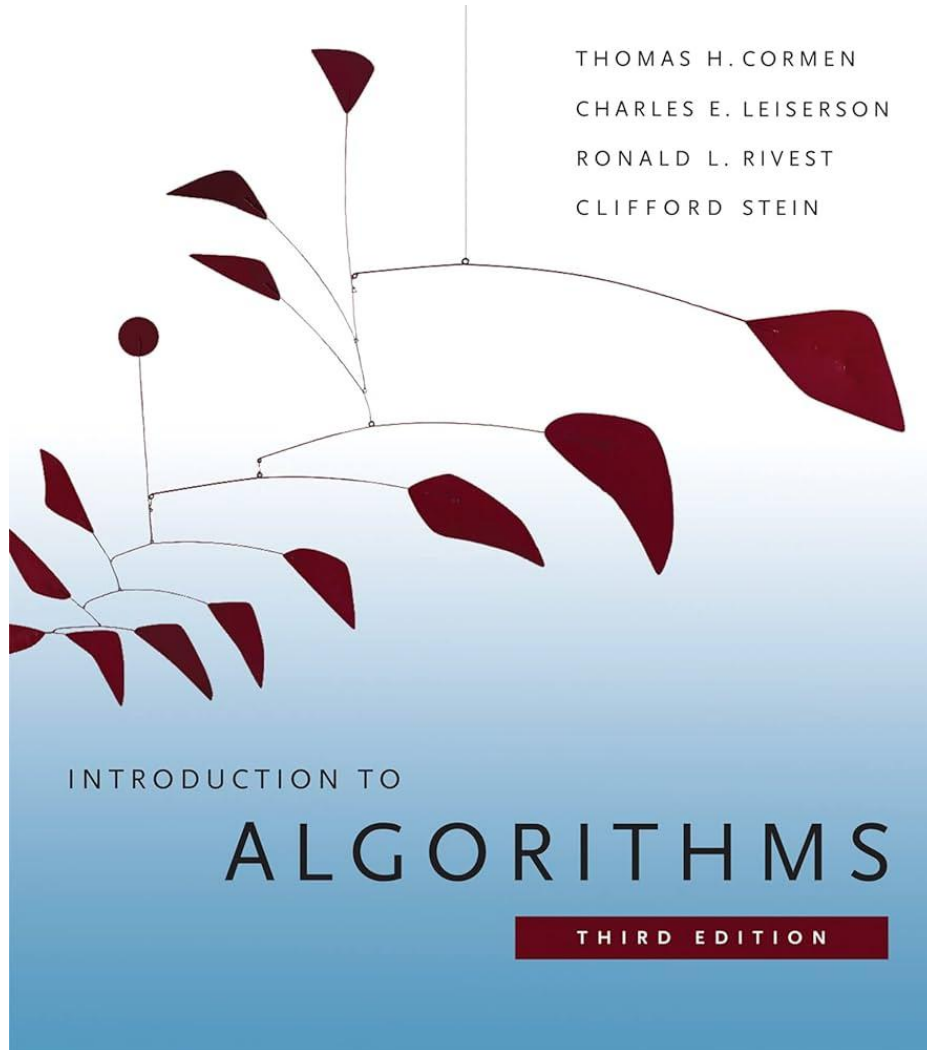
Bibliografia Básica

- CELES, W.; CERQUEIRA, R.; RANGEL, J. L. Introdução a estrutura de dados: com técnicas de programação em C. Rio de Janeiro: Elsevier, 2004.
- BANNEDUZZI, Humberto Martins; METZ, João Ariberto. Lógica e linguagem de programação: introdução ao desenvolvimento de software. Curitiba: Livro Técnico, 2010.
- FARRER, Harry; (et. al). Algoritmos estruturados. 3. ed. Rio de Janeiro: LTC, 1999.


Bibliografia Complementar

- GOODRICH, michael. Estruturas de dados e algoritmos em Java. 4 ed. Porto Alegre: BOOKMAN, 2007.
- PREISS, Bruno R.. Estruturas de dados e algoritmos: Padrões de projetos orientados a objeto com Java. Rio de Janeiro: Elsevier, 2000.
- DROZDEK, Adam. Estrutura de dados e algoritmos em C++. São Paulo: Cengage Learning, 2008.
- GUIMARAES, Angelo de Moura; LAGES, Newton Alberto de Castilho. Algoritmos e estruturas de dados. Rio de Janeiro: LTC, 2008.
- SZWARCFITER, J. L.; MARKENZON, L. Estruturas de Dados e seus Algoritmos. Rio de Janeiro: LTC, 1994.

Outras Referências



Material de aula e informações sobre a disciplina



Prof. Lucas Sampaio Leite
Instituto Federal Baiano - Campus Senhor do Bonfim

[Disciplinas](#) [Projetos](#) [Atendimento](#) [Contato](#)

Bem-vindo ao meu portfólio! Aqui você encontrará materiais atualizados e recursos das disciplinas que ministro no Instituto Federal Baiano, preparados para apoiar sua aprendizagem de forma clara e acessível.

Disciplinas

Curso Técnico Integrado em Informática

Lógica e Linguagem de Programação
Conteúdo e materiais atualizados.

[Acessar](#)

Programação Web II
Conteúdo e materiais atualizados.

[Acessar](#)

Curso Técnico Subsequente em Informática

Programação I
Conteúdo e materiais atualizados.

[Acessar](#)

Projeto Integrador II
Conteúdo e materiais atualizados.

[Acessar](#)

Lógica e Linguagem de Programação
Conteúdos e materiais da turma 2025.1

[Acessar](#)

Projeto Integrador I
Conteúdos e materiais da turma 2025.1

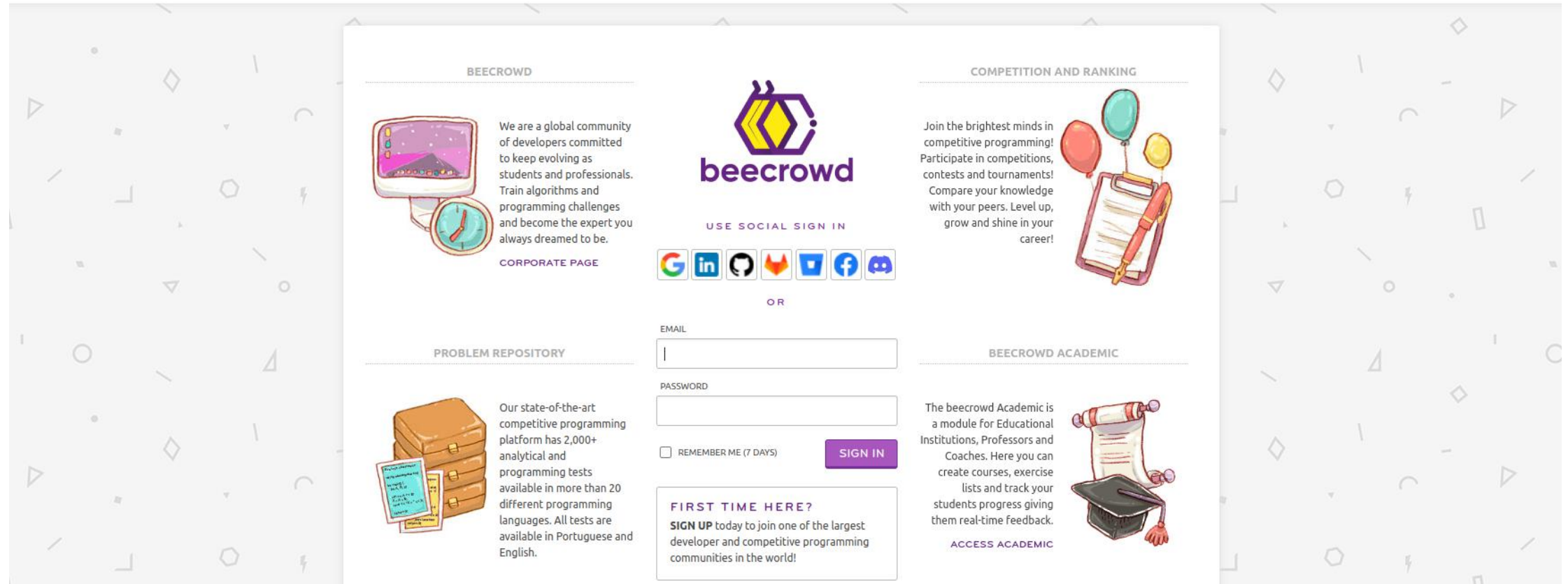
[Acessar](#)

Licenciatura em Ciências da Computação

Estrutura de Dados
Conteúdo e materiais atualizados.

[Acessar](#)

Referência de exercícios (com árbitro digital)



BEECROWD

We are a global community of developers committed to keep evolving as students and professionals. Train algorithms and programming challenges and become the expert you always dreamed to be.

[CORPORATE PAGE](#)

PROBLEM REPOSITORY

Our state-of-the-art competitive programming platform has 2,000+ analytical and programming tests available in more than 20 different programming languages. All tests are available in Portuguese and English.

COMPETITION AND RANKING

Join the brightest minds in competitive programming! Participate in competitions, contests and tournaments! Compare your knowledge with your peers. Level up, grow and shine in your career!

BEECROWD ACADEMIC

The beecrowd Academic is a module for Educational Institutions, Professors and Coaches. Here you can create courses, exercise lists and track your students progress giving them real-time feedback.

[ACCESS ACADEMIC](#)

USE SOCIAL SIGN IN

[G](#) [in](#) [O](#) [F](#) [T](#) [F](#) [D](#)

OR

EMAIL

PASSWORD

REMEMBER ME (7 DAYS) [SIGN IN](#)

FIRST TIME HERE?

SIGN UP today to join one of the largest developer and competitive programming communities in the world!

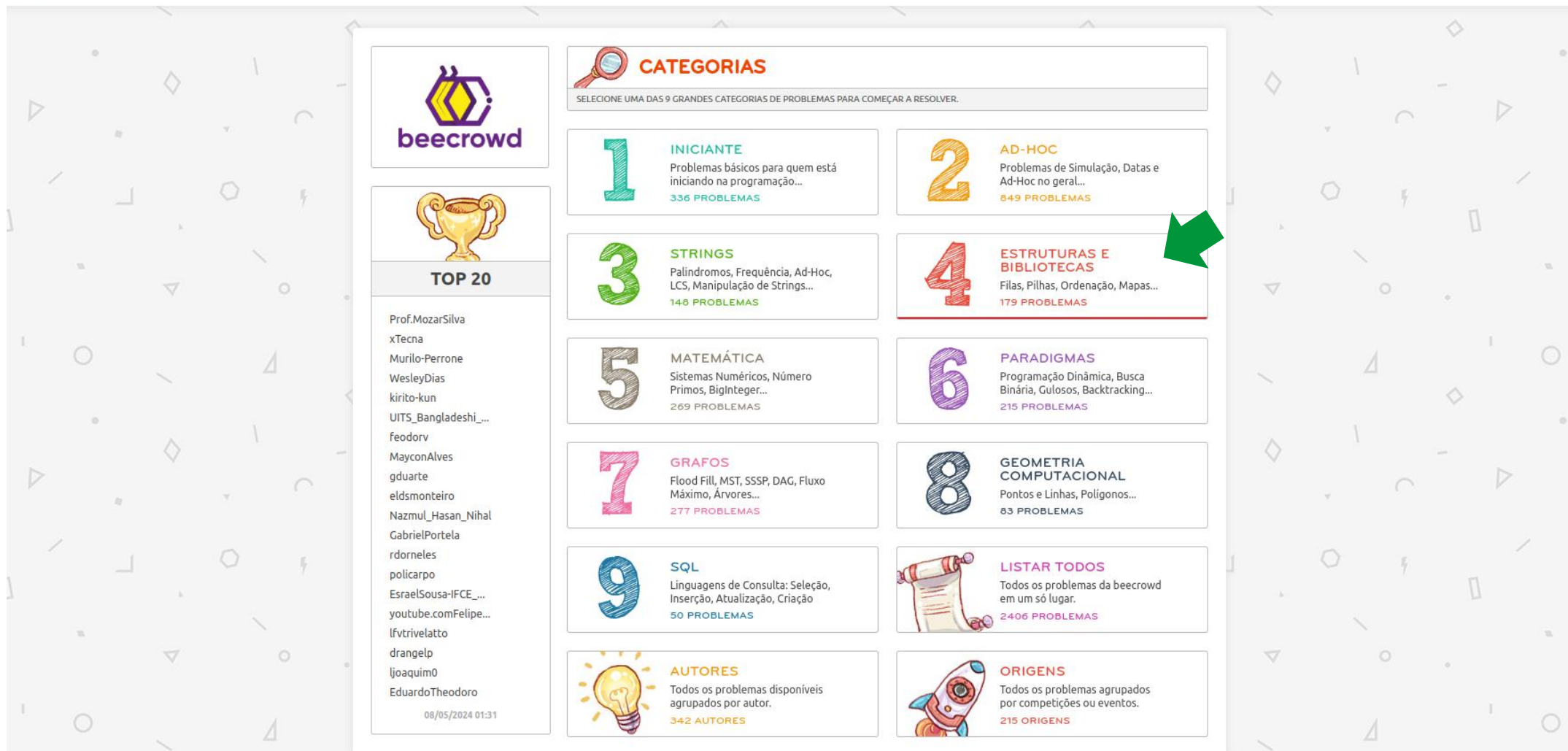
<https://judge.beecrowd.com/>

Referência de exercícios (com árbitro digital)

Hi, lucas.sampaio.leite
lucas.sampaio.leite@gmail.com

HOME PERFIL NEWS OPORTUNIDADES ACADEMIC CONTESTS PROBLEMAS SUBMISSÕES RANKS SAIR

beecrowd



beecrowd

TOP 20

- Prof.MozarSilva
- xTecna
- Murilo-Perrone
- WesleyDias
- kirito-kun
- UITS_Bangladeshi_...
- feodorv
- MayconAlves
- gduarte
- eldsmonteiro
- Nazmul_Hasan_Nihal
- GabrielPortela
- rdorneles
- policarpo
- EsraelSousa-IFCE_...
- youtube.comFelipe...
- lfvtrivelatto
- drangelp
- ljoaquim0
- EduardoTheodoro

08/05/2024 01:31

CATEGORIAS

SELECIONE UMA DAS 9 GRANDES CATEGORIAS DE PROBLEMAS PARA COMEÇAR A RESOLVER.

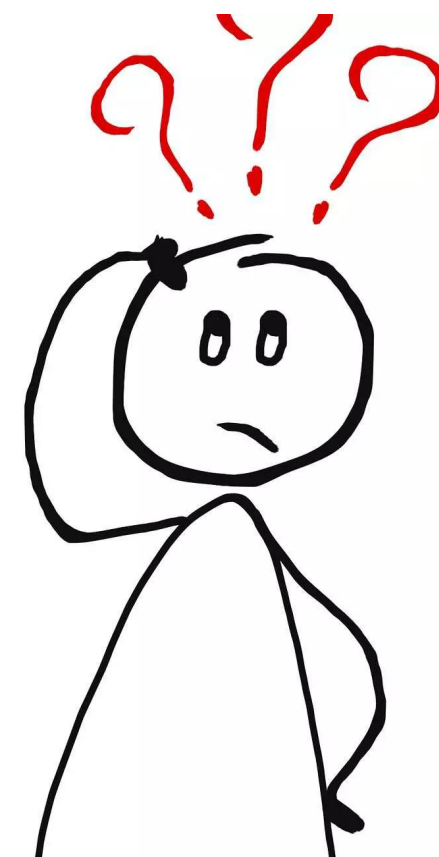
1 INICIANTE Problemas básicos para quem está iniciando na programação... 336 PROBLEMAS	2 AD-HOC Problemas de Simulação, Datas e Ad-Hoc no geral... 849 PROBLEMAS
3 STRINGS Palindromos, Frequência, Ad-Hoc, LCS, Manipulação de Strings... 148 PROBLEMAS	4 ESTRUTURAS E BIBLIOTECAS Filas, Pilhas, Ordenação, Mapas... 179 PROBLEMAS
5 MATEMÁTICA Sistemas Numéricos, Número Primos, BigInteger... 269 PROBLEMAS	6 PARADIGMAS Programação Dinâmica, Busca Binária, Gulosos, Backtracking... 215 PROBLEMAS
7 GRAFOS Flood Fill, MST, SSSP, DAG, Fluxo Máximo, Árvores... 277 PROBLEMAS	8 GEOMETRIA COMPUTACIONAL Pontos e Linhas, Polígonos... 83 PROBLEMAS
9 SQL Linguagens de Consulta: Seleção, Inserção, Atualização, Criação 50 PROBLEMAS	LISTAR TODOS Todos os problemas da beecrowd em um só lugar. 2406 PROBLEMAS
AUTORES Todos os problemas disponíveis agrupados por autor. 342 AUTORES	ORIGENS Todos os problemas agrupados por competições ou eventos. 215 ORIGENS

<https://judge.beecrowd.com/>

Boas práticas

- Para a dinâmica de aprendizagem da disciplina funcionar, é importante realizar as práticas e exercícios passados em sala (não deixem acumular atividades).
- Quem deixa acumular as atividades, tende a ter um desempenho inferior.
- Não deixem as listas de exercício para última hora.
- Organizem seu tempo entre as disciplinas.
- Silêncio e/ou participação.

O que é um Algoritmo?

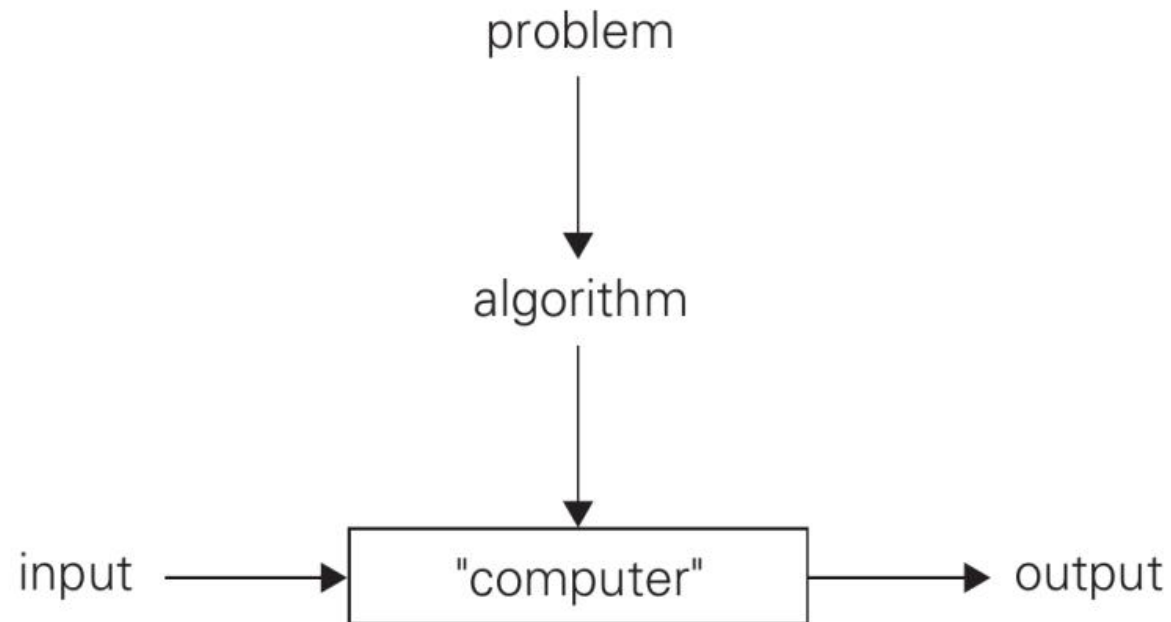


Definição informal (Cormen et al.)

- Informalmente, um algoritmo é qualquer procedimento computacional bem definido que toma algum valor ou conjunto de valores como entrada e produz algum valor ou conjunto de valores com saída.
- Portanto, um algoritmo é uma sequência de etapas computacionais que transformam a entrada em saída.

Definição (Levitin)

- Um algoritmo é uma sequência de instruções inequívocas para resolver um problema, ou seja, para obter uma saída necessária para qualquer entrada legítima em um período de tempo finito.



Algumas ponderações...

- O requisito de não ambiguidade não pode ser comprometido.
- O intervalo de entradas válidas deve ser especificado cuidadosamente.
- O mesmo algoritmo pode ser representado de diversas maneiras diferentes.
 - Podem existir vários algoritmos para resolver o mesmo problema.
 - Algoritmos podem resolver o mesmo problema com velocidades muito diferentes.

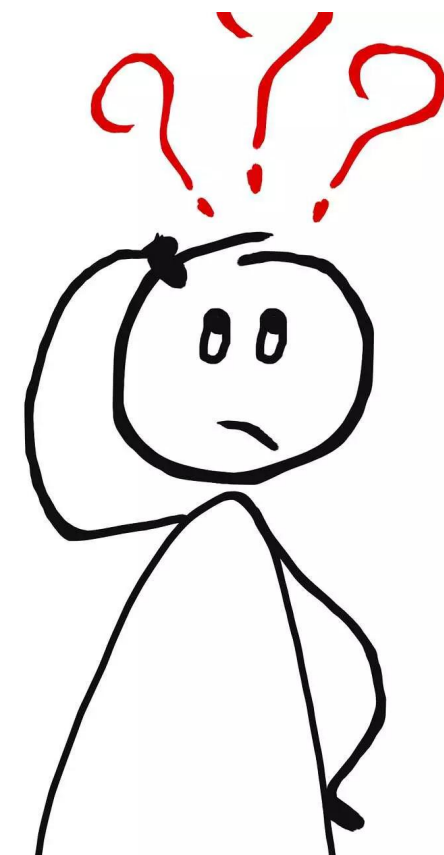
Exemplo (problema da ordenação)

- Entrada: Uma sequência de números inteiros $\langle a_1, a_2, \dots, a_n \rangle$
- Saída: Uma permutação (reordenação) $\langle a_1', a_2', \dots, a_n' \rangle$. da sequência de entrada, tal que $\langle a_1' \leq a_2' \leq \dots \leq a_n' \rangle$
- Exemplo:
 - Sequência de entrada (instância): $\langle 31, 41, 59, 26, 41, 58 \rangle$
 - Saída esperada: $\langle 26, 31, 41, 41, 48, 49 \rangle$

Exemplo (problema da ordenação)

- Em geral, uma **instância** de um problema consiste na entrada (que satisfaz quaisquer restrições impostas no enunciado do problema) necessária para calcular uma solução para o problema.
- Um algoritmo é **correto** se, para toda instância de entrada ele parar com a saída correta.
- Dizemos que um algoritmo correto **resolve** o problema computacional dado.

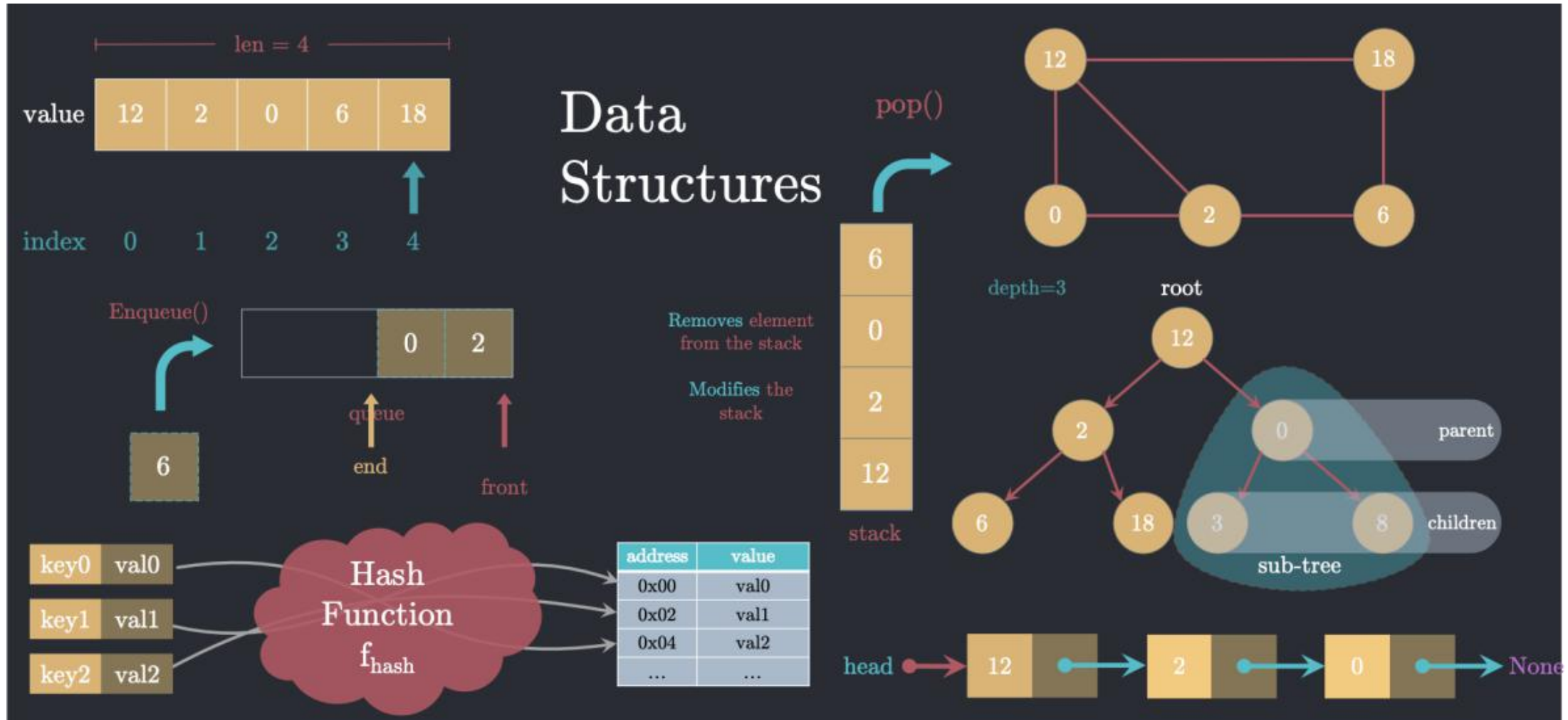
O que são estruturas de dados?



Definição informal (Cormen et al.)

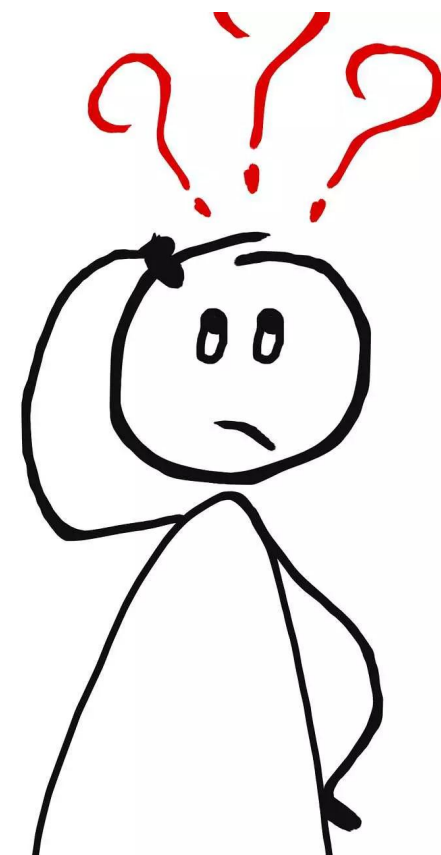
- Uma estrutura de dados é um modo de armazenar e organizar dados com o objetivo de facilitar acesso e modificações.
- Nenhuma estrutura de dados única funciona bem para todas as finalidades e, por isso, é importante conhecer os pontos fortes e limitações de várias delas.

Definição informal (Cormen et al.)



Fonte imagem: <https://brasap.com.br/8-estruturas-de-dados-que-todo-o-programador-precisa-conhecer/>

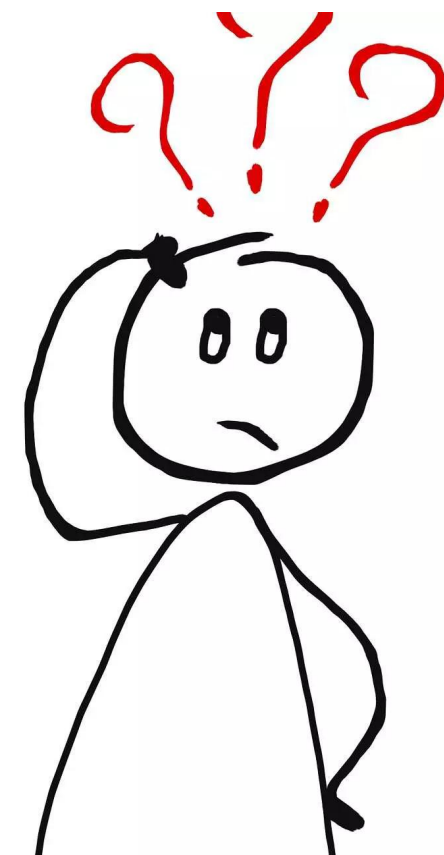
Qual a relação entre algoritmos e estruturas de dados?



Qual a relação entre algoritmos e estruturas de dados?

- Um algoritmo pode ser otimizado ou ineficiente dependendo da estrutura de dados utilizada para armazenar e manipular os dados envolvidos no problema.
- Exemplo:
 - Buscar um elemento em uma lista não ordenada.
 - Buscar um elemento em uma árvore de busca (BST) aplicando a propriedade de busca binária.

O que são tipos de dados primitivos e derivados em C?



Tipos primitivos e derivados em C

- Na linguagem C, os dados manipulados por um programa são armazenados na memória. Esses dados podem ser classificados em dois grandes grupos:
 - Tipos primitivos (básicos)
 - Tipos compostos ou derivados
- A distinção entre eles está principalmente em como os valores são armazenados e acessados na memória.

Tipos primitivos

- São os tipos básicos fornecidos pela linguagem. Armazenam valores diretamente na variável.
- Principais tipos primitivos em C:

Tipo	Descrição	Exemplo
int	Números inteiros	10, -5
float	Números reais (ponto flutuante)	3.14
double	Maior precisão que float	3.1415926535
char	Um único caractere	'A', 'b'

Tipos primitivos

```
#include <stdio.h>

int main() {
    int idade = 20;
    float altura = 1.75;
    char inicial = 'L';

    printf("Idade: %d\n", idade);
    printf("Altura: %.2f\n", altura);
    printf("Inicial: %c\n", inicial);

    return 0;
}
```

Tipos primitivos

```
#include <stdio.h>

int main() {
    int idade = 20;
    float altura = 1.75;
    char inicial = 'L';

    printf("Idade: %d\n", idade);
    printf("Altura: %.2f\n", altura);
    printf("Inicial: %c\n", inicial);

    return 0;
}
```

Características:

- Armazenam valores diretamente
- Têm tamanho fixo
- Acesso simples e direto
- Usados para dados básicos

Tipos derivados

- Diferente dos primitivos, esses tipos trabalham com endereços de memória ou estruturas mais complexas.
- Principais tipos derivados em C:
 - Ponteiros
 - Arrays (vetores)
 - Structs
 - Strings (arrays de char)

Tipos derivados

```
#include <stdio.h>

int main() {
    int x = 10;
    int *p;

    p = &x;

    printf("Valor de x: %d\n", x);
    printf("Endereço de x: %p\n", &x);
    printf("Valor de p: %p\n", p);
    printf("Valor apontado por p: %d\n", *p);

    return 0;
}
```

Tipos derivados

```
#include <stdio.h>

int main() {
    int x = 10;
    int *p;

    p = &x;

    printf("Valor de x: %d\n", x);
    printf("Endereço de x: %p\n", &x);
    printf("Valor de p: %p\n", p);
    printf("Valor apontado por p: %d\n", *p);

    return 0;
}
```

Um ponteiro armazena o endereço de memória de outra variável.

Tipos derivados

```
#include <stdio.h>

int main() {
    int numeros[3] = {10, 20, 30};

    printf("%d\n", numeros[0]);

    return 0;
}
```

Tipos derivados

```
#include <stdio.h>

int main() {
    int numeros[3] = {10, 20, 30};
    printf("%d\n", numeros[0]);

    return 0;
}
```

O nome do array funciona como um ponteiro para o primeiro elemento.

Tipos derivados

```
#include <stdio.h>

struct Aluno {
    int idade;
    float nota;
};

int main() {
    struct Aluno a1;

    a1.idade = 18;
    a1.nota = 9.5;

    printf("Idade: %d\n", a1.idade);
    printf("Nota: %.1f\n", a1.nota);

    return 0;
}
```

Tipos derivados

```
#include <stdio.h>

struct Aluno {
    int idade;
    float nota;
};

int main() {
    struct Aluno a1;

    a1.idade = 18;
    a1.nota = 9.5;

    printf("Idade: %d\n", a1.idade);
    printf("Nota: %.1f\n", a1.nota);

    return 0;
}
```

Structs permitem agrupar diferentes tipos de dados em uma única estrutura.

Tipos derivados

```
#include <stdio.h>

int main() {
    char nome[] = "Estruturas de Dados";

    printf("%s\n", nome);
}
```

Tipos derivados

```
#include <stdio.h>

int main() {
    char nome[] = "Estruturas de Dados";

    printf("%s\n", nome);
}
```

Em C, strings são arrays de caracteres terminados por '\0'

Tipos primitivos vs derivados

Característica	Primitivos	Referenciados
Armazenamento	Valor direto	Endereço ou estrutura
Complexidade	Simple	Mais complexos
Uso de memória	Menor	Pode ser maior
Flexibilidade	Baixa	Alta
Exemplos	int, float, char	ponteiros, arrays, structs

Exercícios rápidos

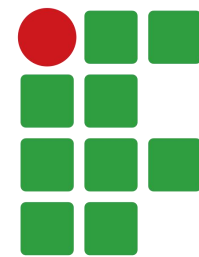
1. Criar uma variável int e imprimir seu valor e endereço;
2. Criar uma struct simples (ex: Produto).
3. Crie um programa para calcular a área de um quadrado de lado 10.
4. Crie um programa que solicite o valor de duas notas em C e calcule a média.
5. Crie um vetor de 10 posições em C e atribua valores a ele. Em seguida percorra e imprima todos os valores. Por fim, imprima apenas nos valores pares.

Dúvidas



ALGORITMOS E ESTRUTURA DE DADOS

Curso de Engenharia de Software
Lucas Sampaio Leite



**INSTITUTO
FEDERAL**
Pernambuco