Algoritmos e Estruturas de Dados I

Apresentação da Disciplina

Profa. Teoria: Mirtha Lina Fernández Venero, Sala 529-2,

mirtha.lina@ufabc.edu.br

http://professor.ufabc.edu.br/~mirtha.lina/aedi.html

Prof. Prática: Paulo Henrique Pisani, Sala 507-2

paulo.pisani@ufabc.edu.br

http://professor.ufabc.edu.br/~paulo.pisani/2019Q1/AEDI/index.html

14 de fevereiro de 2019

Universidade Federal do ABC

CMCC Centro de Matemática, Computação e Cognição

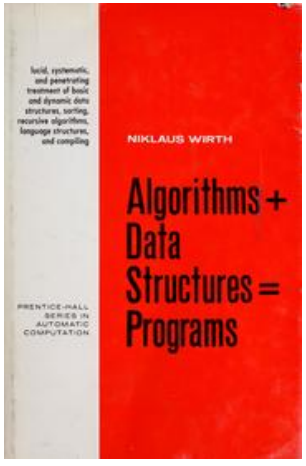
Agenda

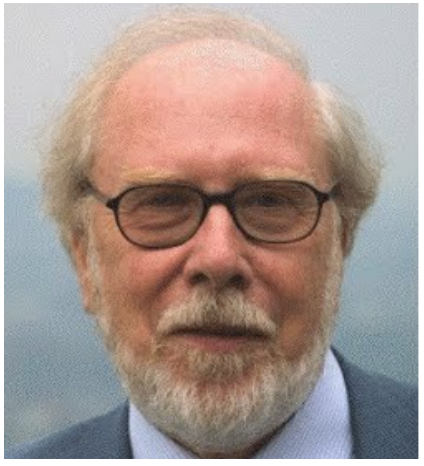
Universidade Federal do ABC

CMCC Centro de Matemática, Computação e Cognição

O que é um programa?

"Programs, after all, are concrete formulations of abstract algorithms based on particular representations and structures of data."

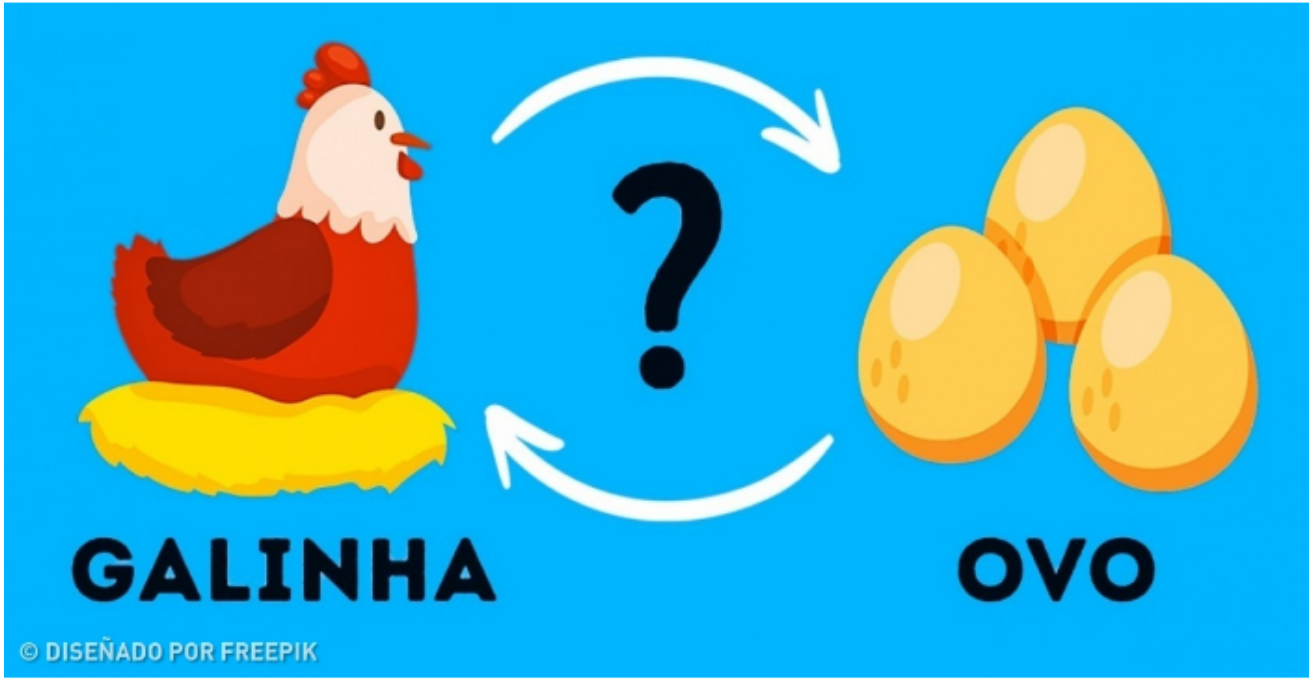




- Niklaus Wirth: Prêmio Turing 1984 pelo desenvolvimento das linguagens Euler, Algol W, Pascal, Modula, Modula-2, Oberon, Oberon-2, Oberon-07 e PL/0.
- Program Development by Stepwise Refinement (leitura recomendada pelo menos a Introdução e Conclusões)
- Top Niklaus Wirth Quotes (vídeo com 20)

"C++ is an insult to the human brain"

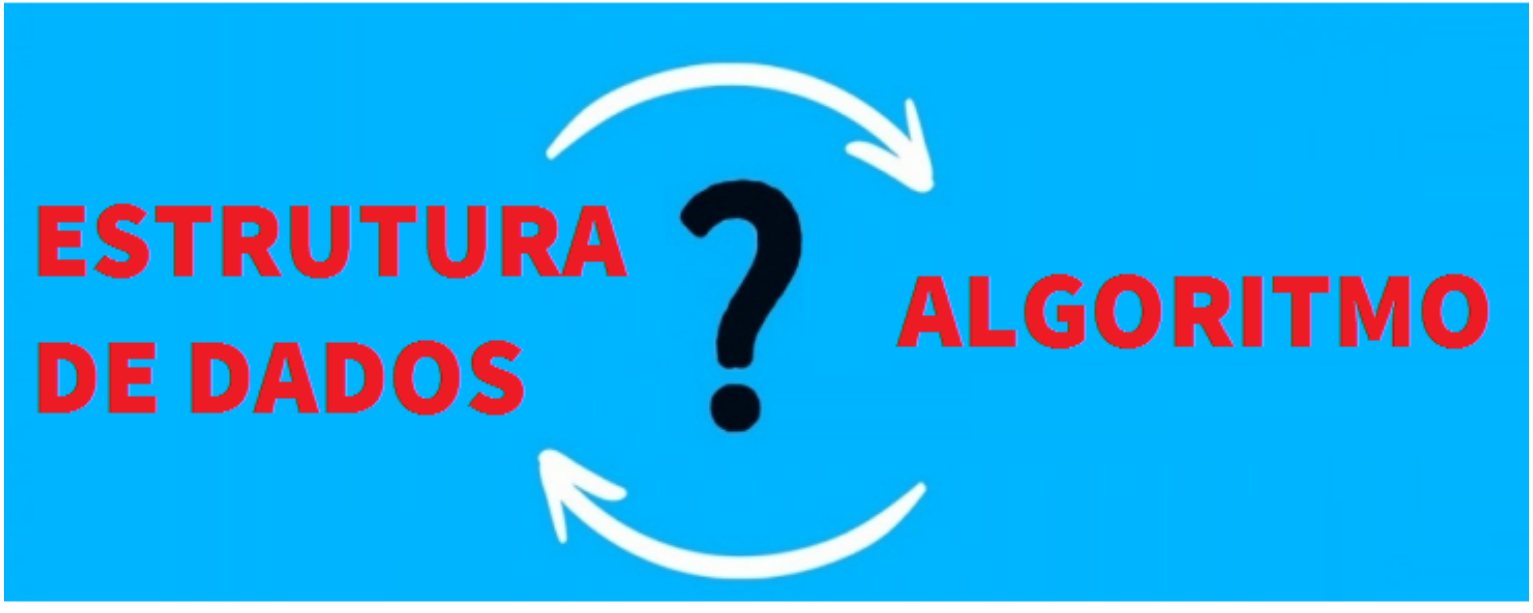
O que veio primeiro?



Universidade Federal do ABC

CMCC
Centro de Matemática, Computação e Cognição

O que vem primeiro?



Dados, Tipos e Estruturas de dados

Dados: valores que representam a abstração dum objeto ou fenômeno dum problema ou situação da vida real

- "Não é possível"ter dados sem operações nem operações sem dados

Tipo de Dados: conjunto de dados e suas operações

Exemplo:

Dados, Tipos e Estruturas de dados

Dados: valores que representam a abstração dum objeto ou fenômeno dum problema ou situação da vida real

- "Não é possível"ter dados sem operações nem operações sem dados

Tipo de Dados Abstrato (TDA): conjunto de dados e suas operações sem detalhes de implementação

Exemplo:

Dados, Tipos e Estruturas de dados

Dados: valores que representam a abstração dum objeto ou fenômeno dum problema ou situação da vida real

- "Não é possível"ter dados sem operações nem operações sem dados

Tipo de Dados Abstrato (TDA): conjunto de dados e suas operações

Exemplo: **Número** com operações de +, -, *, /. Exemplos de implementações: inteiro (`int`), real (`float`), racional (`RATIO`, LISP), complexo (`COMPLEX`, FORTRAN), irracional (?)

Dados, Tipos e Estruturas de dados

Dados: valores que representam a abstração dum objeto ou fenômeno dum problema ou situação da vida real

- "Não é possível"ter dados sem operações nem operações sem dados

Tipo de Dados Abstrato (TDA): conjunto de dados e suas operações

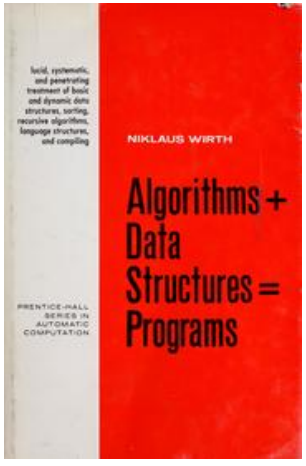
Exemplo: **Número** com operações de +, -, *, /. Exemplos de implementações: inteiro (`int`), real (`float`), racional (`RATIO`, LISP), complexo (`COMPLEX`, FORTRAN), irracional (?)

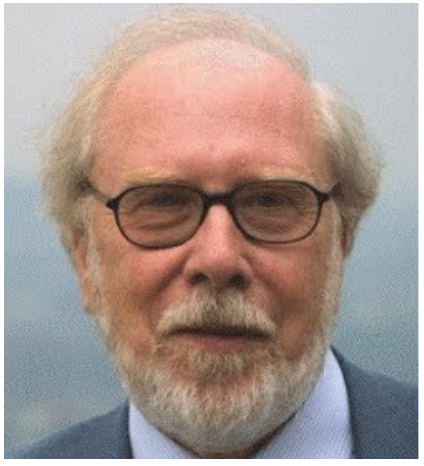
Estrutura de dados: representação concreta de um TDA definindo como organizar, armazenar, acessar, modificar e realizar as operações sobre os dados de forma eficiente

- a escolha da estrutura de dados pode mudar o algoritmo

O que é um programa?

"Programs, after all, are concrete formulations of abstract algorithms based on particular representations and structures of data."





$$Program = Algorithms \Leftrightarrow Data\ Strutures$$

Universidade Federal do ABC

CMCC Centro de Matemática, Computação e Cognição

Agenda

Universidade Federal do ABC

CMCC Centro de Matemática, Computação e Cognição

Objetivos da disciplina

- Fornecer uma abordagem científica e sistemática para construir programas que envolvem grandes volumes de dados
- Conhecer as vantagens e desvantagens das estruturas de dados básicas e seus algoritmos
- Reforçar a importância da análise e escolha da melhor solução pra um problema.
- Aprofundar as habilidades para a modelagem e solução computacional de problemas
- Melhorar as habilidades de codificação, depuração, manutenção de programas e trabalho em equipe
- Aumentar a confiança de que um programa funciona de forma correta e eficiente

$\Rightarrow$ Ajudar nas entrevistas de emprego e concursos da área

Universidade Federal do ABC

CMCC Centro de Matemática, Computação e Cognição

Ementa

- Breve introdução à linguagem C
- Noções básicas de análise de complexidade de tempo de algoritmos
- Estruturas lineares
- Busca e ordenação
- Árvores balanceadas

Prerrequisitos

- Habilidades básicas de resolução algorítmica de problemas e programação numa linguagem de alto nível, e.g. Python/...
- Capacidade de aprender as estruturas básicas da linguagem C

É recomendável, porém não indispensável, ter aprovado a disciplina de Programação Estruturada do BCC. Ver sites de professores que já ministraram a matéria (e.g. Fabrício, Jesús, Paulo)

Universidade Federal do ABC

CMCC Centro de Matemática, Computação e Cognição

Não é objetivo da disciplina

https://docs.oracle.com/javase/7/docs/api/java/util/package-tree.html

Overview Package Class Use Tree Deprecated Index Help — Java™ Platform Standard Ed. 7

Prev Next Frames No Frames All Classes

Hierarchy For Package java.util

Class Hierarchy

- java.lang.**Object**
  - java.util.**AbstractCollection**<E> (implements java.util.Collection<E>)
    - java.util.**AbstractList**<E> (implements java.util.List<E>)
      - java.util.**AbstractSequentialList**<E>
        - java.util.**LinkedList**<E> (implements java.lang.Cloneable, java.util.Deque<E>, java.util.List<E>, java.io.Serializable)
      - java.util.**ArrayList**<E> (implements java.lang.Cloneable, java.util.List<E>, java.util.RandomAccess, java.io.Serializable)
      - java.util.**Vector**<E> (implements java.lang.Cloneable, java.util.List<E>, java.util.RandomAccess, java.io.Serializable)
        - java.util.**Stack**<E>
    - java.util.**AbstractQueue**<E> (implements java.util.Queue<E>)
      - java.util.**PriorityQueue**<E> (implements java.io.Serializable)
    - java.util.**AbstractSet**<E> (implements java.util.Set<E>)
      - java.util.**EnumSet**<E> (implements java.lang.Cloneable, java.io.Serializable)
      - java.util.**HashSet**<E> (implements java.lang.Cloneable, java.io.Serializable, java.util.Set<E>)
        - java.util.**LinkedHashSet**<E> (implements java.lang.Cloneable, java.io.Serializable, java.util.Set<E>)
      - java.util.**TreeSet**<E> (implements java.lang.Cloneable, java.util.NavigableSet<E>, java.io.Serializable)
    - java.util.**ArrayDeque**<E> (implements java.lang.Cloneable, java.util.Deque<E>, java.io.Serializable)

Interface Hierarchy

- java.lang.**Iterable**<T>
  - java.util.**Collection**<E>
    - java.util.**List**<E>
    - java.util.**Queue**<E>
      - java.util.**Deque**<E>

Universidade Federal do ABC

CMCC Centro de Matemática, Computação e Cognição

Metodologia: T-P-I, 2-2-4

Na semana, 2h de **T**eoria e 2h de **P**rática em laboratório

- **Aulas Teóricas**: conceitos essenciais (não detalhes de implementação), exemplos, dicas, resolução de algum trecho de exercício

Niklaus Wirth Quotes

"My being a teacher had a decisive influence on making language and systems as simple as possible so that in my teaching, I could concentrate on the essential issues of programming rather than on details of language and notation."

"Clearly, programming courses should teach methods of design and construction, and the selected examples should be such that a gradual development can be nicely demonstrated."

Universidade Federal do ABC

CMCC Centro de Matemática, Computação e Cognição

Metodologia: T-P-I, 2-2-4

Na semana, 2h de **T**eoria e 2h de **P**rática em laboratório

- **Aulas Teóricas**: conceitos essenciais (não detalhes de implementação), exemplos, dicas, resolução de algum trecho de exercício
- **Aulas no laboratório**: Responsável Prof. Paulo
  1. rápida revisão de alguns conceitos da aula teórica
  2. implementação de algoritmos e estruturas de dados
  3. exercícios de forma individual ou duplas
- **Avaliação dos exercícios do Laboratório**: entrega até a sexta seguinte

$$\mathbf{NotaLab} = 10 * (\sum_i NotaLab_i) \ / \ \sum_i ValorLab_i$$

Mais detalhes a serem divulgados posteriormente

Universidade Federal do ABC

CMCC Centro de Matemática, Computação e Cognição

Metodologia: T-P-I, 2-2-4

- **Atividades e exercícios para resolver de forma Individual**
  Resolva a maior quantidade possível, no seu ritmo, como preparação para as provas
- **Monitoria** (?): datas e horários a serem definidos
- **Atendimento dos Professores**: quinta 13-15h (Mirtha) e quarta 17-18h (Paulo)

Importante: Quatro horas de teoria+prática é pouco $\Rightarrow$ Quatro horas de estudo independente não é suficiente. É necessário dedicar mais tempo à matéria. O que você **não deve fazer**:

- entrar em pânico, desistir
- procurar na Internet ou **copiar** a solução de um colega
- programar sem pensar nos algoritmos e estruturas de dados
- deixar pra estudar tudo na semana antes da prova

Universidade Federal do ABC

CMCC Centro de Matemática, Computação e Cognição

Metodologia: T-P-I, 2-2-4

- **Atividades e exercícios para resolver de forma Individual**
  Resolva a maior quantidade possível, no seu ritmo, como preparação para as provas
- **Monitoria** (?): datas e horários a serem definidos
- **Atendimento dos Professores**: quinta 13-15h (Mirtha) e quarta 17-18h (Paulo)

Importante: Quatro horas de teoria+prática é pouco $\Rightarrow$ Quatro horas de estudo independente não é suficiente. É necessário dedicar mais tempo à matéria. O que você **deve fazer**:

- faça anotações e escreva em papel seus programas
- estude toda semana resolvendo os exercícios propostos
- **discuta** suas soluções com colega/monitor/professor(a)
- programe aplicando os conceitos estudados nas aulas teóricas

Universidade Federal do ABC

CMCC Centro de Matemática, Computação e Cognição

Agenda

Agenda Preliminar (sujeita a alterações)

1	12/02	Introdução à linguagem C
	14/02	Apresentação da disciplina. Ponteiros e estruturas dinâmicas
2	19/02	Ponteiros e estruturas dinâmicas
	21/02	Estruturas lineares (listas ligadas)
3	26/02	Estruturas lineares (pilhas e filas)
	28/02	Custos dos algoritmos
4	05/03	Feriado
	07/03	Algoritmos de busca
5	12/03	Árvores de busca
	14/03	Revisão de exercícios
6	19/03	Prova
	21/03	Árvores de busca balanceadas

Agenda Preliminar (sujeita a alterações)

7	26/03	Árvores de busca balanceadas
	28/03	Árvores de busca balanceadas
8	02/04	Árvores de busca balanceadas
	04/04	Algoritmos simples de ordenação
9	09/04	Ordenação
	11/04	Algoritmos de ordenação eficientes
10	16/04	Ordenação
	18/04	Outros métodos de ordenação
11	23/04	Ordenação
	25/04	Revisão de exercícios
12	30/04	Prova
	02/05	Prova Substitutiva

Relação Nota - Conceito

Nota = 0.35 * Prova I + 0.35 * Prova II +
0.3 * Laboratório +
(Bônus = ParticipaçãoTeoria + Desafio)

- **A** $\geq 9 \Rightarrow$ excelente participação e compreensão da disciplina
- **B** $= [7.5, 9) \Rightarrow$ boa participação e compreensão da disciplina
- **C** $= [6, 7.5) \Rightarrow$ compreensão do conteúdo mais importante da disciplina e capacidade para seguir estudos mais avançados
- **D** $= [5, 6) \Rightarrow$ compreensão mínima do conteúdo da disciplina e deficiências para prosseguir estudos avançados
- **F** $= [0, 5) \Rightarrow$ insuficiente compreensão do conteúdo
- **O** $\Rightarrow$ Reprovado por faltas (mais de 6 aulas $= 25\%$)

Prova Substitutiva 02/05/2019

Ausência a **uma prova regular por motivo justificado**.
Obrigatória apresentação de documento que comprove o motivo **em até três dias úteis após a prova**

- Doença ou acidente incapacitante - atestado médico com CRM, período de repouso e CID
- Falecimento de familiar - atestado de óbito
- Boletim de Ocorrência Policial (B.O.) e/ou declaração de obrigações legais
- Greve nos transportes, alagamento, acidente de grandes proporções - notícia de jornal/site de conhecimento geral
- Certificado de participação em atividades acadêmicas oficiais e relevantes ou em Conselhos da Universidade

Resolução CONSEPE N° 181

Prova de recuperação 08/05/2019

- Cobre toda a matéria
- Aberta, qualquer estudante aprovado que quiser melhorar sua nota pode fazer a REC
- Estudante com mais de 25% de faltas **não** tem direito a REC
- **Atenção: a REC substitui a nota de P1+P2.**

Nota = 0.7 * REC + 0.3 * Laboratório + Bônus

Importante: as notas não são arredondadas

Resolução CONSEPE N° 182

Algumas Regras de Conduta

- Presença nas aulas será controlada com lista de presença
- Entrar e sair da sala frequentemente, comer, usar celular, bater papo, etc não serão condutas permitidas
- Chegar repetidamente atrasado será desencorajado. A presença não é obrigatória, por isso se vai chegar muito atrasado considere não chegar e aproveitar melhor o tempo.
- Emails para mirtha.lina@ufabc.edu.br com assunto relacionado a **AEDI**, **nome**, **RA** e **turma**. Arquivos adjuntos somente em **pdf**. Outros emails serão descartados; também, emails que podem ser respondidos sem necessidade de escrever aos professores.
- **Cola, Fraude ou Plágio** $\Rightarrow$ **Nota 0** para os envolvidos

Universidade Federal do ABC

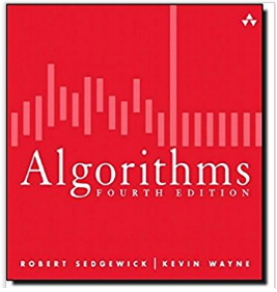
CMCC Centro de Matemática, Computação e Cognição

Agenda

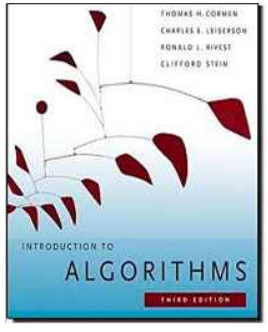
Bibliografia

1. Robert Sedgewick. **Algorithms in C/C++/Java, Parts 1-4 (Fundamental Algorithms, Data Structures, Sorting, Searching)**, Addison-Wesley Professional

   https://algs4.cs.princeton.edu/home/



2. Thomas H. Cormen; Charles Eric Leiserson; Ronald L. Rivest; Clifford Stein. **Introduction to Algorithms**, MIT Press



3. Donald E. Knuth. **The Art of Computer Programming**. vols. 1 e 3, Addison-Wesley, 1973
4. Nivio Ziviani. **Projeto de Algoritmos: com implementações em Pascal e C**, 2009
5. Slides em http://professor.ufabc.edu.br/~mirtha.lina/eadi.html