# APRESENTAÇÃO DA DISCIPLINA 2020/1

GSI512 - Estrutura de Dados II

Prof. Dr. Rafael D. Araújo rafael.araujo@ufu.br

http://www.facom.ufu.br/~rafaelaraujo



#### Objetivo da disciplina

- Fornecer uma visão geral da análise de algoritmos afim de fornecer subsídios para a construção e/ou utilização de algoritmos e estrutura de dados adequados à solução eficiente de problemas.
- Carga horária total: 60h (30h teóricas e 30h práticas)

#### Conteúdo Programático

#### 1<sup>a</sup> parte

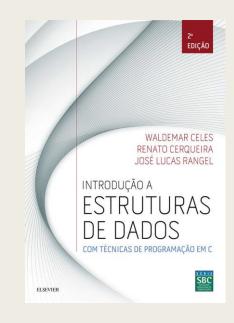
- Revisão estruturas de dados lineares
- Tipo Abstrato de Dados
- Princípios de Análise de Algoritmos
- Algoritmos de Busca
- Hashing

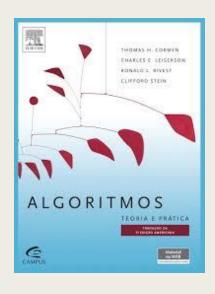
#### 2ª parte

- Algoritmos recursivos
- Programação Dinâmica
- Heap
- Algoritmo de Huffman
- Grafos

#### Bibliografia Básica

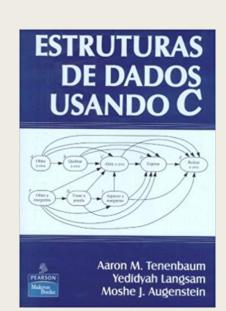
■ CELES, W. et al. Introdução a Estruturas de Dados. Rio de Janeiro: Campus. (1ª e 2ª edição).





■ CORMEN, T.H. et al. Algoritmos: teoria e prática. 2. ed. Rio de Janeiro: Campus, 2002.

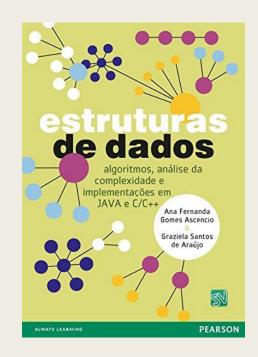
■ TENENBAUM, A. M. et al. Estrutura de Dados usando C. São Paulo: Pearson, 1995.



#### Bibliografia Complementar

■ ASCENCIO, A. F. G.; ARAÚJO, G. S. Estruturas de Dados: algoritmos, análise de implementações em JAVA e C/C++. São Paulo: Pearson, 2011.

■ ZIVIANI, N. Projeto de Algoritmos: com implementações em PASCAL e C. 3.ed. São Paulo: Cengage Leaming, 2011.





#### Material de apoio online

- FEOFILOFF, P. Minicurso de Análise de Algoritmos. In: CARVALHO, A. P. L. F.; KOWALTOWSKI, T. (org.) Jornadas de Atualização em Informática 2009. Edição atualizada. Bento Gonçalves, RS: SBC, 2009. Disponível em: https://www.ime.usp.br/~pf/livrinho-AA/AA-BOOKLET.pdf
- LINTZMAYER, C. N.; MOTA, G. O. Análise de Algoritmos e Estruturas de Dados. CMCC Universidade Federal do ABC, 2020. Disponível em: http://professor.ufabc.edu.br/~carla.negri/cursos/materiais/Livro-Analise.de.Algoritmos.pdf

#### Metodologia



Encontro
síncrono para
discussão do
conteúdo da
semana



Estudo
 assíncrono dos
 materiais da
 semana



Encontro
 síncrono para
 dúvidas e
 resolução de
 exercícios

Testes



 Resolução de exercícios da semana (individual/ em equipe)

Icons made by Freepik.

#### Avaliação A + E + P + Q + O = 100pts



• A = Apresentação de trabalho em vídeo: 10pts



- E = Exercícios de implementação (individual/equipe): 25pts
- P = Avaliação por pares: 10pts



- Q = 3 testes: **15pts** (5pts cada)
  - 22/03/2021, 10/05/2021, 09/06/2021



- 0 = 2 avaliações individuais: **40pts** (20pts cada)
  - 07/04/2021 e 26/05/2021

Câmeras ligadas

Icons made by Freepik.

#### Avaliação A + E + P + Q + O = 100pts

NÃO haverá avaliação substitutiva!



• A = Apresentação de trabalho em vídeo: 10pts



- E = Exercícios de implementação (individual/equipe): 25pts
- P = Avaliação por pares: 10pts



- Q = 3 testes: **15pts** (5pts cada)
  - 22/03/2021, 10/05/2021, 09/06/2021



- 0 = 2 avaliações individuais: **40pts** (20pts cada)
  - 07/04/2021 e 26/05/2021

Icons made by Freepik.

#### Assiduidade

Presenças nas atividades síncronas + porcentagem de atividades entregues

■ É necessário um mínimo de 75% de frequência para aprovação na disciplina

#### Atendimento

- Mensagens diretas ao professor dentro da plataforma Microsoft Teams. Tempo de resposta: até 48h.
- Agendamento síncrono (via mensagem direta no Microsoft Teams), preferencialmente nos horários:
  - Segunda-feira 16h00-17h00
  - Quarta-feira 16h00-17h00
- Agendar os atendimentos via chat do Microsoft Teams
- Equipe Microsoft Teams: "[2020/1] ED2 BSIMC"
  - ATENÇÃO: os slides são <u>apenas</u> notas de aula e não devem ser utilizados como fonte única de referência!!!

## Visão geral da disciplina

## Como os computadores resolvem problemas?

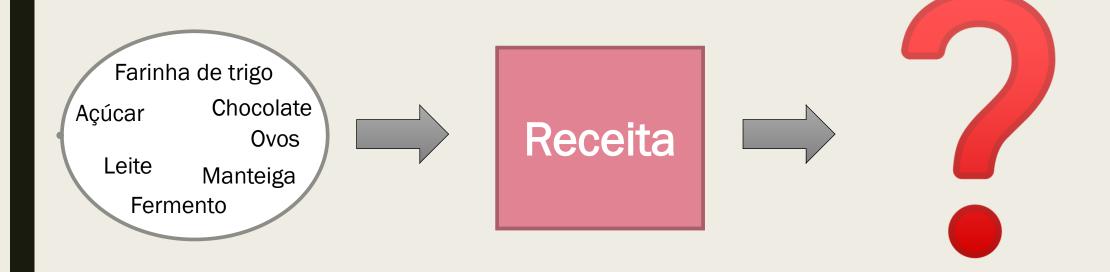
- Como duas compras efetuadas no mesmo momento pela internet chega nos endereços corretamente?
  - Como o número do seu cartão de crédito é protegido contra alguém que o intercepta?
- Como você consegue jogar um jogo com outras pessoas pela Internet em tempo real?
- Como o GPS consegue encontrar, entre uma infinidade de possíveis rotas, o caminho mais rápido para o seu destino e fazer isso em poucos segundos?

### O que é um algoritmo?

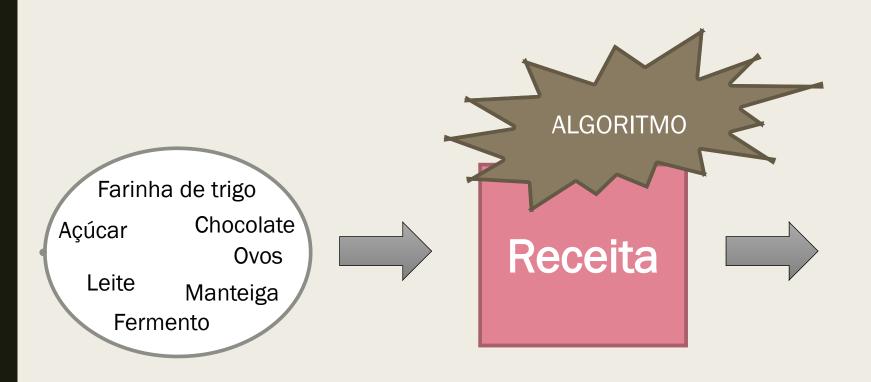
- Um algoritmo é um conjunto finito de passos que devem ser seguidos com um conjunto de dados de entrada para se chegar à solução de um problema.
  - Um problema, geralmente, pode ser resolvido por muitos algoritmos diferentes.

- A existência de um algoritmo não implica necessariamente que este problema possa realmente ser resolvido na prática.
  - Há restrições de tempo e de espaço de memória.

#### Exemplo: como fazer um bolo?

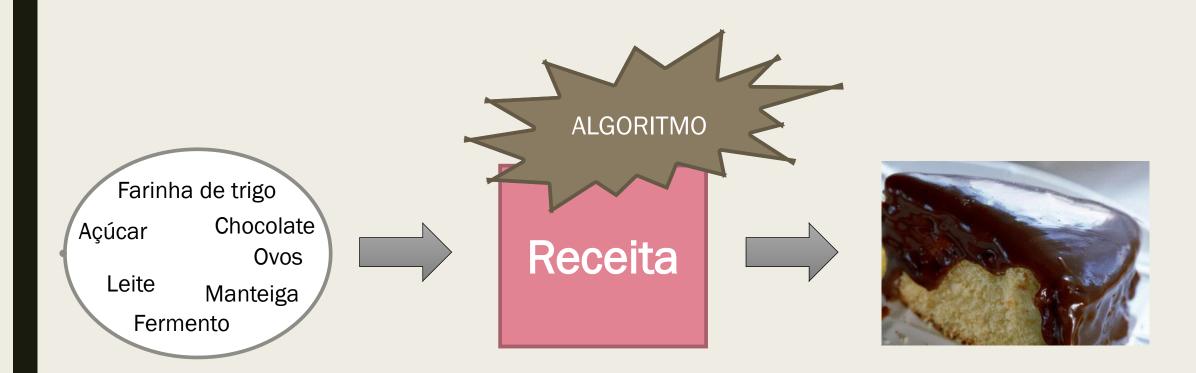


#### Exemplo: como fazer um bolo?





#### Exemplo: como fazer um bolo?



## O que distingue um algoritmo executado em um computador de um algoritmo que as pessoas executam?

- No exemplo anterior, o açúcar acabou, o que fazer?
- Outro exemplo:
  - Se você vai de carro para o trabalho, o seu algoritmo poderia dizer "se o tráfego estiver ruim, pegue uma rota alternativa".
  - O que significa "tráfego ruim"?

### Algoritmo vs Programa

#### Algoritmo vs Programa

- Um algoritmo é uma ideia abstrata de como resolver um determinado problema.
  - A princípio, é independente do computador que o executará e de suas características.
- Um programa é uma implementação de um algoritmo em uma linguagem particular.
  - Executado em um computador particular.
- Um programa está sujeito às limitações físicas da máquina onde será executado
  - memória, processador, periféricos etc.

#### Problema vs Instância

#### Problema vs Instância

- Um problema computacional é um problema formulado em termos da descrição das possíveis entradas e suas respectivas saídas.
- Uma instância é uma materialização do problema.
- Exemplo: problema de ordenação crescente de números

#### Problema de ordenação crescente de números:

Entrada: uma sequência de n números  $a_1$ ,  $a_2$ ,  $a_3$ , . . . ,  $a_n$  Saída: uma sequência de n números  $a_1$ ,  $a_2$ ,  $a_3$ , . . . ,  $a_n$ , tal que  $a_i$  <  $a_i$  se i < j



#### Instância:

Entrada: 3, 1, 2, 4

Saída: 1, 2, 3, 4

#### Estratégias de projeto de algoritmos

 Existem diferentes estratégias para se projetar um algoritmo, ou seja, diferentes formas de pensamento (paradigmas)

- Força bruta
- Divisão e conquista
- Programação dinâmica
- Estratégia gulosa
- Algoritmos aproximados

Cada estratégia pode utilizar diferentes estruturas de dados!

**–** ...

#### Eficiência de algoritmos

- Qual é o custo de usar um dado algoritmo para resolver um problema específico?
  - análise do número de vezes que cada parte do algoritmo deve ser executada
  - estudo da quantidade de memória necessária
  - consumo de energia, ...
- Como saber se um algoritmo é eficiente?
- É possível fazer um algoritmo mais eficiente para resolver o mesmo problema?

#### Corretude de algoritmos

- Um algoritmo **resolve** um problema (ou seja, é correto) quando, para qualquer entrada, produz uma resposta correta, se forem concedidos tempo e memória suficientes para sua execução.
  - Dada qualquer entrada, o algoritmo termina sua execução retornando saídas corretas
- Note que um algoritmo estar correto não significa que a entrada de dados está!
  - No exemplo do GPS, imagine que o trânsito esteja congestionado na rota de menor distância. O resultado não deixa de estar correto.

#### Algoritmos incorretos

- Não para quando um dado caso de entrada é introduzido
- Para com uma saída incorreta
- O fato de um algoritmo resolver um problema na teoria,
   não significa que seja aceitável na prática
  - Pode ser que o gasto de memória e tempo seja inviável
- Em alguns casos, uma solução aproximada pode ser suficiente para o objetivo almejado

#### Problemas que não podem ser resolvidos

- Por que alguns problemas parecem ser (computacionalmente) mais difíceis do que outros?
- Exemplo:
  - Entrada: um programa de computador e uma entrada qualquer (arbitrária) para este programa
  - Problema: este programa irá parar quando alimentado com a entrada dada?

## Áreas da Ciência da Computação

- Análise de Algoritmos: estuda a eficiência de algoritmos (para saber se são viáveis)
  - Quantidade de recursos computacionais necessários
  - Diferentes estratégias para construção de algoritmos

- Teoria da Computação: estuda a solubilidade computacional (computabilidade) de problemas
  - Capacidades e limitações dos computadores
  - Modelos formais

#### Nessa disciplina...

■ Análise de Algoritmos x Estruturas de Dados

- O que preciso fazer para ser aprovado(a) na disciplina?
  - Comprometimento
  - Interação
  - Fazer as atividades