



UNIVERSIDADE  
FEDERAL DO CEARÁ

Aula Inaugural

---

TÓPICOS ESPECIAIS EM COMPUTAÇÃO II e ESTUDOS ESPECIAIS

UFC – Engenharia da Computação – 2023-2

Prof. Fischer Jônatas Ferreira

# Agenda

➡ Discussões iniciais sobre algoritmos

➡ Organização da disciplina



# Professor



Fischer Ferreira

## Formação:

- Graduação: FURG
- Especialização: UCP
- Mestrado: PUC-Rio
- Doutorado: UFMG

## Atividade como docente anteriores:

- UEMG (2018 -2021)
- UIT (2014-2021)

## Desenvolvedor:

- Marinha do Brasil
- Ministério Público MG

## Pesquisador:

- LabSoft UFMG

[Lattes](#)



# Organização da disciplina



# Desafios da disciplina

Graduação: **TÓPICOS ESPECIAIS EM COMPUTAÇÃO**

Mestrado: **ESTUDOS ESPECIAIS**

- Modalidade nova de disciplina a graduação juntamente com o mestrado
- Primeira vez ministrada na UFC (prof.Fischer)
- Disciplina com finalidade de ser prática
- Diferentes níveis de conhecimento dos discentes sobre os assuntos tratados
- Exige muita maturidade dos discente para a disciplina poder funcionar

# Comunicação

➡ Ambiente Google Sala de Aula - Código da turma: **uqyrswu**

- Compartilhamento de arquivos
- Entrega de atividades
- Comunicação geral com a turma
- Retiradas de dúvidas

E-mail: [fischer.ferreira@sobral.ufc.br](mailto:fischer.ferreira@sobral.ufc.br) - Gabinete 17



# Monitor

## Lucas Gabriel

- Aluno de mestrado
- Estágio em docência
- Contatos
  - [lucasggs.com@gmail.com](mailto:lucasggs.com@gmail.com)
  - 88 9359-0602 (WhatsApp)



# Entrega das atividades no classroom

Entregas não consideradas:

- 1) Atividades submetidas mas não enviada
  - a) Submeta e envie sua tarefa
- 2) Atividades respondidas na parte do comentário
  - a) Submeta sua resposta na atividade
- 3) Código postado como zip
  - a) Código deve ser postado no GitHub - link do código deve ser postado em resposta a atividade
- 4) Links quebrados
  - a) Tome cuidado com a organização do repositório no GitHub
- 5) Atividades enviadas com atraso





# Encontros e frequência nas aulas

➡ Terça-feira: 08h00 às 10h00

Quinta-feira: 08h00 às 10h00

Frequência computada em cada aula



# Implementação de algoritmos

Foco da disciplina é na experimentação dos algoritmos, desta forma:

- As implementações dos algoritmos em si podem ser buscada na literatura relacionada
- A linguagem utilizada pode ser aquela que o aluno tiver mais familiaridade
- A não faz parte da ementa da disciplina discussão sobre estatística, mas é aconselhado o uso nos relatórios
- Caprichar na exibição dos dados nos relatórios

# Processo de avaliação realizado

## Mestrado:

- Atividades práticas de aula
- Três trabalhos práticos
- **Apresentação de artigo** - Seminário de apresentação de artigo
- **Escrita de um artigo** do trabalho final

## Graduação:

- Atividades práticas de aula
- Três trabalhos práticos
- **Escrita de relatório** - Seminários de apresentação de artigo
- **Participação** na equipe de escrita de um artigo trabalho final



# Agenda

## Unidade 1:

- 1.1) Análise de Algoritmos: Complexidade de Tempo e Notação Assintótica;
- 1.2) Algoritmos Fundamentais para Ordenação
- 1.3) Engenharia de Software experimental



# Agenda

## Unidade 2:

2.1) Algoritmos de Divisão e Conquista

2.2) Técnicas de Projeto de Algoritmos

Trabalho Prático 2



# Agenda

➡ Unidade 3:

## 3.1) Algoritmos para Problemas em Grafos

Trabalho Prático 3



# Agenda

➡ Unidade 4:

4.1) Técnicas para tratar problemas NP-Completo

Trabalho Prático 4



# Objetivo da disciplina

Trabalhar com algoritmos clássicos da Ciência da Computação para resolução de problemas práticos.





# Objetivos específicos

- Conhecer as especificações dos principais algoritmos da Ciência da Computação;
- Saber identificar problemas onde os algoritmos são aplicados;
- Saber identificar a complexidade de cada algoritmo;
- Familiarizar-se com a implementação dos principais algoritmos;
- Comparação dos algoritmos estudados por meio de experimentos;



# Algoritmos: definição

- Sequência de ações executáveis para a obtenção de uma solução para um determinado tipo de problema [Loureiro].
- 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 como saída [Cormen]



# Algoritmo - definição

- Problema:

- é simplesmente uma tarefa a ser executada;
- é uma função ou associação de entradas com saídas.

- Algoritmo:

- é um método ou um processo usado para resolver um problema.

- Programa:

- é uma instanciação de um algoritmo em uma linguagem de programação computacional.

# Algoritmos

- 1) Qual é a importância do estudo sobre algoritmos?
- 1) Algoritmo é um assunto específico dos cursos de computação?
- 1) Qual a importância do estudo de algoritmos para sua formação?
- 1) Qual é a importância do conhecimento sobre algoritmo para o profissional de computação (engenheiro, cientista, analista e etc...)



# Aplicação de algoritmos

- Presentes em todas as áreas da computação na resolução dos mais diversos tipos de problemas;
- Permitem que problemas do mundo real possam ser trabalhados de forma estruturada e conseqüentemente possam ser resolvidos por um computador;



# O papel de algoritmos em computação

- Definição: um algoritmo é um conjunto finito de instruções precisas para executar uma computação.
  - Um algoritmo pode ser visto como uma ferramenta para resolver um problema computacional bem especificado.
- Um algoritmo pode receber como entrada um conjunto de valores e pode produzir como saída um outro conjunto de valores.
  - Um algoritmo descreve uma seqüência de passos computacionais que transforma a entrada numa saída, ou seja, uma relação entrada/saída.

# Algumas perguntas importantes

- Um programa pode ser visto como um algoritmo codificado em uma linguagem de programação que pode ser executado por um computador.
- Qualquer computador pode executar qualquer programa?
- Todos os problemas ligados às ciências exatas possuem algoritmos?
- Todos os problemas computacionais têm a mesma dificuldade de resolução? Como algoritmos diferentes para um mesmo problema podem ser comparados/avaliados?

# Algumas perguntas importantes

- Precisamos apenas fornecer um algoritmo para um dado problema?
  - Basta apenas resolver o problema?
  - Basta escolher uma biblioteca específica e o pronto?
- 
- E o tempo computacional gasto?
  - E o consumo de memória?
  - E o espaço em disco utilizado?
  - E a técnica utilizada para projeto desse algoritmo?
  - E a dificuldade de implementação?
  - E o custo com a manutenção desse algoritmo?
  - Esse algoritmo tem uma implementação escalável?



# Algumas perguntas importantes

- Estamos mais preocupados em quais frameworks ou ferramentas utilizar?
- Ainda sabemos projetar e implementar algoritmos eficientes?
- Precisamos projetar algoritmos?

# Algoritmos do futuro

- Como serão os algoritmos do futuro?
- Como nos preparamos para ser um desenvolvedor do futuro?
- Quais são seus diferenciais para lidar com sua concorrência?
- Existe ou existirá computação sem algoritmos?

# Porquê o estudo da Complexidade?

A performance é extremamente importante na ciência da computação, pois existe uma necessidade constante de melhorar os algoritmos.

Apesar de parecer contraditório, com o aumento da velocidade dos computadores, torna-se cada vez mais importante desenvolver algoritmos mais eficientes, devido ao aumento constante do "tamanho" dos problemas a serem resolvidos.

# Análise de algoritmos:

Critérios:

- Correção
- Otimalidade
- Eficiência temporal
- Eficiência espacial

# Análise de algoritmos

## Abordagem experimental

- Implementar o algoritmo e executar o programa para um conjunto de dados de teste
- Observar seu comportamento
- Comparar algoritmos com o mesmo propósito em ambiente controlado (pseudo controlado)

# Análise de algoritmos: Limitações

- Não podemos testar todas as possíveis entradas
- Podemos esquecer algum caso em que o algoritmo falha ou caso em que o desempenho do algoritmo é particularmente bom ou ruim
- Os resultados dependem de aspectos de implementação

# Análise de algoritmos: empírica

Um plano geral para a análise empírica da eficiência de uma algoritmo pode seguir os seguintes passos:

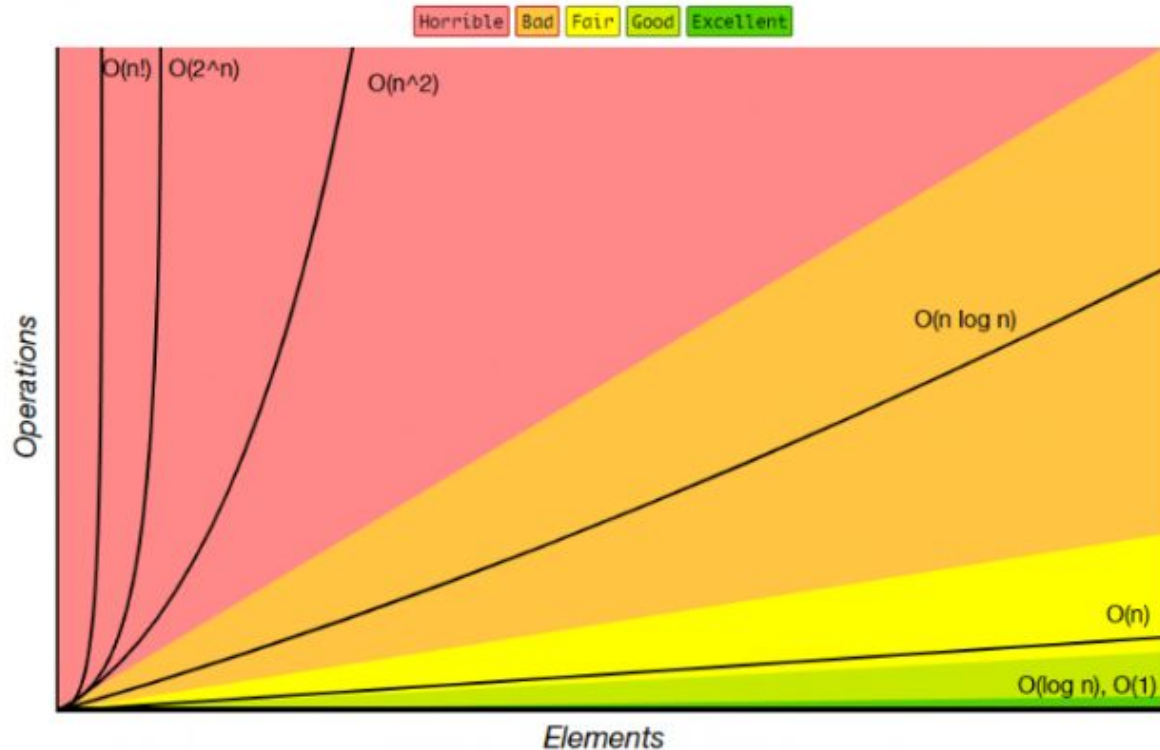
1. Entender o propósito do experimento
2. Escolher uma métrica de eficiência  $M$  a ser medida e a unidade de medida (unidade de tempo ou número de operações básicas)
3. Escolher as características da amostra de entrada
  - a. (repetição de dados, tamanho etc...)

# Análise de algoritmos: empírica

4. Escrever um programa implementando o algoritmo para a experimentação
5. Escolher ou gerar amostra de entradas
6. Executar o algoritmo sobre as amostras e guardar os dados obtidos
7. Analisar os dados obtidos



# Análise de algoritmos: empírica



# Análise de algoritmos: empírica

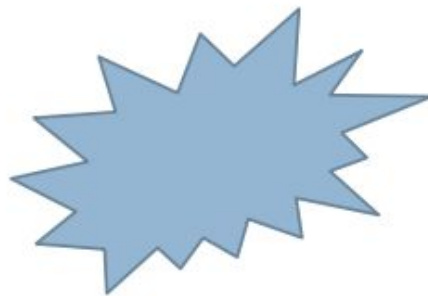
|    |                 |             |
|----|-----------------|-------------|
| P  | $O(1)$          | Constante   |
|    | $O(\log n)$     | Logarítmica |
|    | $O(n)$          | Lineal      |
|    | $O(n \log n)$   |             |
|    | $O(n^{1.x})$    |             |
|    | $O(n^2)$        | Cuadrática  |
|    | $O(n^2 \log n)$ |             |
|    | $O(n^3)$        | Cúbica      |
|    | $O(n^K)$        |             |
| NP | $O(1, x^n)$     | Exponencial |
|    | $O(2^n)$        |             |
|    | $O(n!)$         | Factorial   |
|    | $O(n^n)$        |             |

# Complexidade assintótica

Qual é a complexidade do algoritmo abaixo?

```
int n=;  
int soma=0;  
  
for (int i = 0; i < n; i++) {  
    soma=soma+1;  
}
```

- a)  $n=100$
- b)  $n=1000$
- c)  $n=1.000.000.000.000.000.000$



# Complexidade assintótica

Qual é a complexidade do algoritmo abaixo?

```
int n=;  
int soma=0;  
  
for (int i = 0; i < n; i++) {  
    soma=soma+1;  
}
```

- a)  $n=100$
- b)  $n=1000$
- c)  $n=1.000.000.000.000.000.000$

$O(n)$

# Complexidade assintótica

Qual é a complexidade do algoritmo abaixo?


```
int n = ;  
int m = ;  
int soma = 0;  
  
for (int i = 0; i < n; i++) {  
    for (int j = 0; j < m; j++) {  
        soma = soma + 1;  
    }  
}
```



# Complexidade assintótica

Qual é a complexidade do algoritmo abaixo?

```
int n = ;  
int m = ;  
int soma = 0;  
  
for (int i = 0; i < n; i++) {  
    for (int j = 0; j < m; j++) {  
        soma = soma + 1;  
    }  
}
```



$O(n^2)$

# Complexidade assintótica

Qual é a complexidade do algoritmo abaixo?

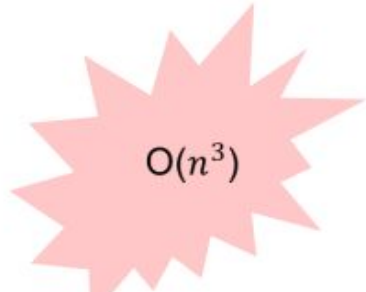
```
int n = ;  
int m = ;  
int p = ;  
int soma = 0;  
  
for (int i = 0; i < n; i++) {  
    for (int j = 0; j < m; j++) {  
        for (int l = 0; l < p; l++) {  
            soma = soma + 1;  
        }  
    }  
}
```



# Complexidade assintótica

Qual é a complexidade do algoritmo abaixo?

```
int n = ;  
int m = ;  
int p = ;  
int soma = 0;  
  
for (int i = 0; i < n; i++) {  
    for (int j = 0; j < m; j++) {  
        for (int l = 0; l < p; l++) {  
            soma = soma + 1;  
        }  
    }  
}
```



$O(n^3)$



# Complexidade assintótica

Qual desses algoritmos tem a maior complexidade?

a)

```
int n=100;
int soma=0;

for (int i = 0; i < n; i++) {
    soma=soma+1;
}
```

b)

```
int n = 100;
int soma = 0;

for (int i = 0; i < n; i++) {
    soma = soma + 1;
    soma = soma + 2;
    soma = soma + 3;
    soma = soma + 4;
    soma = soma + 5;
}
```

# Complexidade assintótica

Qual desses algoritmos tem a maior complexidade?

a)

```
int n=100;  
int soma=0;  
  
for (int i = 0; i < n; i++) {  
    soma=soma+1;  
}
```

$O(n)$

b)

```
int n = 100;  
int soma = 0;  
  
for (int i = 0; i < n; i++) {  
    soma = soma + 1;  
    soma = soma + 2;  
    soma = soma + 3;  
    soma = soma + 4;  
    soma = soma + 5;  
}
```

$O(n)$

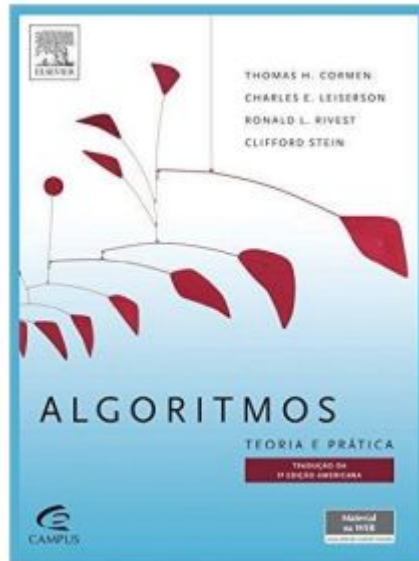
# Atividade 1: Apresentação pessoal

Postar um vídeo (máximo 5 minutos, com a câmera ligada) no YouTube como não listado e colocar o link no Classroom.

- Nome
- Matrícula
- Período
- Cidade de origem
- Vertente pretendida
- Conte um pouco se você gosta de desenvolvimento de software
- Conte um pouco da sua experiência em desenvolvimento de software e se participou de algum projeto ou desenvolveu alguma aplicação
- Informe as linguagens que você já teve contato
- Informe o que você espera para o restante do curso de Engenharia de Computação
- Informe o que você espera da disciplina de TÓPICOS ESPECIAIS EM COMPUTAÇÃO II e ESTUDOS ESPECIAIS

# Bibliografia principal

Cormen, Thomas H., et al. "Algoritmos: teoria e prática." Editora Campus 2 (2002).



# Bibliografia alternativa

- Dasgupta, Sanjoy, Christos Papadimitriou, and Umesh Vazirani. Algoritmos. AMGH Editora, 2009.
- Ziviani, Nivio. Projeto de algoritmos: com implementações em Pascal e C. Vol. 2. Thomson, 2004.
- Lintzmayer, Carla. Análise de Algoritmos e de Estruturas de Dados <http://professor.ufabc.edu.br/~carla.negri/cursos/materiais/Livro-Analise.de.Algoritmos.pdf>