

## Plano de ensino

**Curso:** CCI-BAC - Bacharelado em Ciência da Computação

**Turma:** CCI122-05U - CIÊNCIA DA COMPUTAÇÃO FASE 05U

**Disciplina:** CAL0001 - COMPLEXIDADE DE ALGORITMOS

**Período letivo:** 2018/1

**Carga horária:** 72

**Professor:** 211020727 - DIEGO BUCHINGER

### Ementa

1. Estudo de complexidade via métodos de desenvolvimento de algoritmos. Modelos de computação e ferramentas para notação para análise de algoritmos. Algoritmos iterativos e recursivos. Solubilidade de problemas. Intratabilidade de problemas. Análise da complexidade de algoritmos clássicos na área da computação.

### Objetivo geral

1. Analisar a complexidade de tempo e espaço de algoritmos. Identificar o melhor caso, o pior caso e o caso médio de execução de algoritmos. Identificar problemas tratáveis e intratáveis.

### Objetivo específico

1. Capacitar o aluno a analisar a complexidade de tempo e espaço de algoritmos e ser capaz de identificar problemas considerados intratáveis.

### Conteúdo programático

1. Apresentação do planejamento e da ementa da disciplina.  
Introdução à disciplina.
2. Conceitos Básicos de Complexidade:
  - Notação O grande (limite superior)
  - Ordens de complexidade
  - Análise de complexidade com uma variável
3. Conceitos Básicos de Complexidade:
  - Análise de complexidade de tempo de algoritmos recursivos
4. Conceitos Básicos de Complexidade:
  - Somatórios
5. Conceitos Básicos de Complexidade:
  - Complexidade de Espaço
  - Notações Assintóticas (limite inferior e limite firme)
  - Teorema Mestre
6. Análise de Algoritmos de Ordenação:
  - Merge Sort, Quick Sort e Heap Sort
7. Análise de Algoritmos de Ordenação:
  - Counting Sort e Bucket Sort
8. Análise de Complexidade envolvendo Estruturas de Dados Elementares
9. Análise de Complexidade com múltiplas variáveis:
  - Strings
  - Hash
  - Grafos
10. Análise de Complexidade de operações elementares com inteiros grandes
11. Complexidade em Teoria dos Números e Criptografia:
  - números primos
  - aritmética modular
12. Complexidade em Teoria dos Números e Criptografia:
  - algoritmo de criptografia RSA
13. Abordagens para Resolução de Problemas:
  - Indução matemática
  - Divisão e conquista
  - Algoritmos gulosos
  - Algoritmos de tentativa e erro
14. Abordagens para Resolução de Problemas:
  - Programação dinâmica
  - Algoritmos de aproximação

## Plano de ensino

- |  |
|--|
| 15. Vínculo entre Complexidade e Teoria da Computação: <ul style="list-style-type: none"><li>- Problemas tratáveis e intratáveis;</li><li>- Classes de problemas: P, NP, NP-Completo e NP-Difícil;</li><li>- Redução de problemas;</li></ul> |
| 16. Vínculo entre Complexidade e Teoria da Computação: <ul style="list-style-type: none"><li>- Problemas NP-Completo: SAT, 3-CNF-SAT, Clique, Cobertura de Vértices</li></ul>  |
| 17. Vínculo entre Complexidade e Teoria da Computação: <ul style="list-style-type: none"><li>- Problemas NP-Completo: Ciclo Hamiltoniano, Caixeiro Viajante, Subset-Sum</li><li>- Algoritmos pseudo-polinomiais</li></ul>                    |

### *Metodologia*

- |  |
|--|
| 1. Aulas expositivas acompanhadas de trabalhos práticos relacionados aos conteúdos apresentados na aula. Aulas práticas em laboratório que objetivam a implementação ou pesquisa dos conceitos apresentados nas aulas teóricas. Listas de exercícios para auxiliar na fixação do conteúdo apresentado. Seminários para proporcionar a busca de forma autônoma pelo conhecimento. Provas teóricas para avaliar o conteúdo conceitual aprendido. Trabalhos para avaliar a capacidade do uso dos conceitos aprendidos. Até 20% do conteúdo programático poderá ser ministrado na forma de ensino a distância. |
|--|

### *Sistema de avaliação*

- |   |
|---|
| 1. Os alunos serão avaliados com base no seu desempenho nas provas, trabalhos e listas de exercícios, sendo que ao final do semestre o aluno deverá ter comparecimento mínimo de 75% às aulas e desempenho mínimo de 70% nas avaliações. O grau de desempenho do aluno será avaliado com base no desenvolvimento das seguintes atividades e com os seguintes critérios: <ul style="list-style-type: none"><li>a) Provas: 60%<ul style="list-style-type: none"><li>a.1) Prova 1 (20%)</li><li>a.2) Prova 2 (20%)</li><li>a.3) Prova 3 (20%)</li></ul></li><li>b) Trabalhos: 30%<ul style="list-style-type: none"><li>b.1) Comparação entre algoritmos de ordenação (6,5%)</li><li>b.2) Trabalho comparativo entre busca sequencial, binária e hash (6,5%)</li><li>b.3) Pesquisa e apresentação de complexidade de algoritmo (7%)</li><li>b.4) Implementação e análise do algoritmo de criptografia RSA (10%)</li></ul></li><li>c) Exercícios: 10%<ul style="list-style-type: none"><li>c.1) Exercícios I (3%)</li><li>c.2) Exercícios II (3%)</li><li>c.3) Exercícios III (4%)</li></ul></li></ul> |
|---|

### *Bibliografia básica*

- |  |
|--|
| 1. CORMEN, Thomas H. Algoritmos: teoria e prática. 3. ed. Rio de Janeiro: Elsevier, 2012.<br>CORMEN, Thomas H. Desmistificando algoritmos. Rio de Janeiro: Elsevier, 2014.<br>DASGUPTA, Sanjoy; PAPADIMITRIOU, Christos H; VAZIRANI, Umesh Virkumar. Algoritmos. São Paulo: McGraw-Hill, 2009. |
|--|

### *Bibliografia complementar*

- |   |
|---|
| 1. AHO, Alfred V; HOPCROFT, John E.; ULLMAN, Jeffrey D.; Data structures and algorithms. Reading, MA: Addison Wesley, 1987.<br>KNUTH, Donald Ervin. The art of computer programming - Fundamental Algorithms. 3rd ed. Massachusetts: Addison Wesley, 2008.<br>LEVITIN, Anany. Introduction to the design & analysis of algorithms. 2nd ed. Pearson/Addison Wesley, 2007.<br>TOSCANI, Laura Vieira; VELOSO, Paulo A. S. Complexidade de algoritmos. 3.ed. Porto Alegre: Bookman, 2012.<br>ZIVIANI, Nivio. Projeto de algoritmos: com implementações em Pascal e C. 3. ed. rev. e ampl. São Paulo: Pioneira Thomson Learning, 2011. |
|---|