

## Programa

Este é o plano de desenvolvimento da disciplina e um guia de estudos. Leia-o com atenção e consulte este documento durante todo o semestre. Também, sempre acompanhe os avisos na página da disciplina: <http://www.ic.unicamp.br/~lehilton/mc458a/>.

### Objetivos

Ao final do curso, @ alun@ deverá ser capaz de:

- modelar problemas e subproblemas comuns: busca e ordenação;
- projetar e desenvolver algoritmos utilizando técnicas clássicas: divisão e conquista, algoritmos gulosos e programação dinâmica;
- analisar a correção e a complexidade de algoritmos de maneira formal.

### Pré-requisitos

Ao início do curso, @ alun@ deverá ser capaz de:

- escrever algoritmos recursivos;
- estimar informalmente o tempo de execução de um algoritmo;
- demonstrar afirmações usando indução;
- escrever programas de computador utilizando bibliotecas e estruturas de dados adequadas (exemplo para C++: algorithm, vector, priority queue, etc.).

No primeiro dia de aula, haverá uma avaliação diagnóstica que será realizada em duplas. A avaliação não terá nota atribuída e serve para verificar os requisitos. Se você não estiver presente na primeira aula, procure fazê-la e entregá-la ao professor posteriormente.

### Atividades

Questionário de leitura (avaliação): Antes do início de cada unidade, **em casa**, devem-se ler os capítulos ou seções correspondentes do livro-texto. Os assuntos serão divulgados na página da disciplina com antecedência. Haverá **até 10** questionários de leitura. **Antes da primeira aula de cada unidade**, @s alun@s preencherão um questionário **individualmente e sem consulta** com duração de 5 a 15 min (informado pelo professor cada vez). Para agilizar a aula, não cheguem atrasad@s. Os questionários conterão questões simples e de múltipla escolha acerca do conteúdo lido e serão corrigidos automaticamente. Os questionários têm por objetivo incentivar a leitura e o estudo contínuo da disciplina. A nota de cada questionário valerá de 0 a 10.

Aula expositiva: **Durante as aulas**, @s alun@s devem participar levantando dúvidas, sugestões, ou possivelmente resolvendo os problemas solicitados. As aulas correspondem a cerca de 20% do tempo total dedicado a disciplina; não desperdice esse tempo, participe!

Exercícios de fixação: **Após as aulas**, serão propostos diversos exercícios de fixação (retirados do livro-texto e de outras fontes) para serem feitos em casa. O conteúdo das listas é considerado parte integrante do curso e @s alun@s devem resolvê-los para se preparar para as avaliações. Recomenda-se tentar fazer as atividades individualmente e, depois, discutir em grupo. As principais dúvidas devem ser **anotadas** e levadas ao monitor, ou ao professor, durante as aulas.

Teste de unidade (avaliação): Serão realizados 5 testes de unidades **individuais ou em dupla e sem consulta** com duração de 15 a 40 min (informado pelo professor cada vez), no horário das aulas, em datas a serem divulgadas na página da disciplina. Alguns dos testes conterão questões selecionadas dentre os exercícios de fixação, com conteúdo igual ou com pequenas modificações. A nota de cada teste será de 0 a 10.

Exercícios (avaliação): Serão realizadas 10 aulas de exercícios, em datas a serem divulgadas na página da disciplina. Nessas aulas @s @lunos deverão resolver os exercícios propostos e deverão submeter a resolução, no prazo de 24 h. A nota do trabalho corresponderá a **um único exercício**, que será sorteado **somente depois** do prazo de entrega e valerá de 0 a 10. A dinâmica será:

1. @s alun@s serão separados em grupos definidos pelo professor; os grupos terão tamanho **um, dois ou três**, a depender da unidade, e serão divulgados na página da disciplina com antecedência; **identifiquem e/ou entrem em contato com seu grupo antes da aula**;
2. **durante a primeira parte da aula**, os grupos devem se reunir e resolver as questões; é permitida a consulta a livros, mas não será permitida discussão com pessoas fora do grupo; na parte final da aula, o professor e alun@s voluntári@s (com auxílio do professor) apresentarão para a turma uma sugestão de resolução dos problemas que se mostrarem mais difíceis;
3. cada grupo deverá escrever as resoluções de todas as questões (sem auxílio de outros grupos) e entregá-las até 24 h depois do início da aula, **manuscritas** e digitalizadas no formato PDF por meio do sistema SUSY (trabalhos copiados, desorganizados, incompletos, ilegíveis ou rasurados terão nota zerada); **ATENÇÃO**: os exercícios devem ser resolvidos em sala; o prazo de 24 h serve apenas para corrigir pequenos erros, passar a limpo e submeter; **é obrigatória a presença**: alun@s que faltarem terão nota da atividade zerada.

**Observação**: para otimizar o tempo e exercitar a escrita formal, os trabalhos entregues devem ser escritos à mão; utilize um escâner de mesa ou um aplicativo de celular com a função scan para criar um PDF; observe que o SUSY impõe um limite de 1,9MiB nos arquivos submetidos; verifique a legibilidade e o tamanho arquivo gerado com antecedência e teste sua senha; não serão aceitos trabalhos fora do prazo.

Exercícios de programação (avaliação): haverá 4 trabalhos de programação individuais, a serem submetidos no SUSY, com prazos de uma semana (incluindo eventuais falhas do sistema) divulgados na página, valendo de 0 a 10. Os critérios de correção serão especificados em cada tarefa.

## Avaliação

Média: Serão calculadas as médias aritméticas dos questionários de leitura (L), testes individuais (T), dos exercícios (E) e dos exercícios de programação (P). Depois será calculada a média do semestre  $M := (2L + 4T + 4E + 2P)/12$ . Será aprovado com nota de aproveitamento  $A := M @ alun@$  que satisfizer todos os critérios abaixo:

1. 75% de frequência;
2.  $L \geq 6$ ;  $T \geq 6$ ;  $E \geq 6$ ;  $P \geq 6$ ;
3.  $T4 + T5 + E9 + E10 \geq 20$ , ou seja, a média dos dois últimos testes e exercícios deve ser pelo menos 5.

Do contrário, a nota de aproveitamento do semestre será  $A := \text{mínimo} \{4, M\}$ .

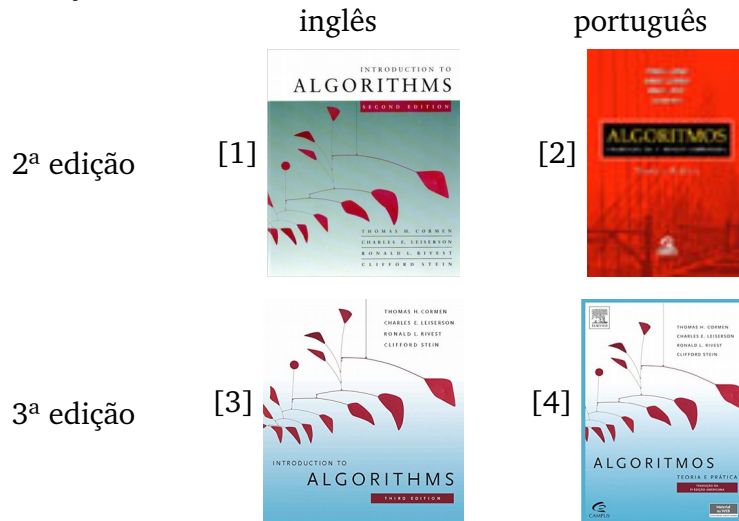
**Observação**: A frequência será controlada por listas de presença ou pela participação nas avaliações; não poderá assinar a lista @ alun@ que chegar com 20 min de atraso ou sair com 20 min de antecedência; em caso de atestado, envie uma cópia por e-mail ao professor no prazo de 15 dias e mantenha o documento original. Alun@s que, justificadamente, não puderem comparecer ao menos a 75% das aulas, devem conversar com o professor no início do semestre. **Importante**: Em qualquer momento, se tiver dúvidas ou tiver dificuldade em cumprir os critérios, procure o professor (o quanto antes!).

Exame: Poderá realizar exame, com nota E, entre 0 e 10, no dia 11/12/2017, @ alun@ com 75% de frequência e nota  $A \geq 2,5$ . Se @ alun@ fizer o exame, a nota final será  $\text{mínimo} \{5, (A+E)/2\}$ , senão a nota final será A.

Fraude: Em caso de fraude (plágio, atestado falso, assinar lista por colegas, abandonar aula após assinar, usar bibliotecas não permitidas, etc.), os envolvidos serão reprovados com nota 0.

## Bibliografia

Será adotado como livro-texto: “Algoritmos – Teoria e Prática”, de T. Cormen, C. Leiserson, R. Rivest, C. Stein (CLRS). Poderá ser usada a segunda ou a terceira edição, tanto na versão em inglês quanto na versão traduzida. Por causa das diferentes numerações entre as edições, os exercícios solicitados do livro poderão ser transcritos. Havendo discrepância de conteúdo, dê preferência à terceira edição.



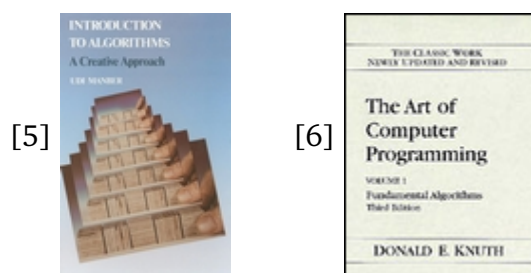
- [1] T. Cormen, C. Leiserson, R. Rivest, C. Stein. Introduction to Algorithms (2ª edição), 2001.  
[2] T. Cormen, C. Leiserson, R. Rivest, C. Stein. Algoritmos – Teoria e Prática (2ª edição), 2002.  
[3] T. Cormen, C. Leiserson, R. Rivest, C. Stein. Introduction to Algorithms (3ª edição), 2009.  
[4] T. Cormen, C. Leiserson, R. Rivest, C. Stein. Algoritmos – Teoria e Prática (3ª edição), 2012.

Além do livro de CLRS, em algumas unidades, também será utilizado o livro de Udi Manber, particularmente nas unidades referentes a indução e projeto de algoritmos baseado em indução.

- [5] U. Manber, Algorithms. A Creative Approach, 1989.

Além desses, poderão ser consultados outros livros, disponíveis na biblioteca, como o livro clássico de Donald Knuth, que é considerado o pai da análise de algoritmo.

- [6] D. E. Knuth. The Art of Computer Programming, 1974.



Há alguns livros pouco mais recentes, como o livro do Kleinberg e da Tardos,  
[7] J. Kleinberg, E. Tardos. Algorithm Design, 2005.

e o livro de Sanjoy Dasgupta, Christos Papadimitriou e Umesh Vazirani, que é um pouco mais informal e com um ritmo diferente do CLRS.

- [8] S. Dasgupta, C. Papadimitriou, U. Vazirani. Algorithms, 2006.

Livros de autores brasileiros importantes também estão disponíveis.

- [9] J. L. Szwarcfiter. Grafos e Algoritmos Computacionais, 1984.  
[10] N. Ziviani. Projeto de Algoritmos (2ª edição), 2004.

Não menos importante, na internet há diversos e excelentes recursos que podem servir para pesquisa. Em particular, veja o curso de análise de algoritmos do prof. Paulo Feofiloff.

- [11] P. Feofiloff. Análise de Algoritmos. [https://www.ime.usp.br/~pf/analise\\_de\\_algoritmos/](https://www.ime.usp.br/~pf/analise_de_algoritmos/).

Um bom conjunto de notas de aulas está livremente disponível.

[12] J. Erickson. Algorithms, Etc. <http://jeffe.cs.illinois.edu/teaching/algorithms/>.

E, lendo com cuidado, a Wikipédia é uma excelente fonte de consulta.

[13] Wikipédia. [https://en.wikipedia.org/wiki/Analysis\\_of\\_algorithms](https://en.wikipedia.org/wiki/Analysis_of_algorithms).

**Sugestão:** não tente ler várias referências de uma vez; na maior parte do conteúdo, atenha-se ao livro-texto ([1]-[4]) e consulte o livro [5] nos capítulos em que ele for usado. Consulte a bibliografia complementar sempre que solicitado, para se aprofundar, ou se tiver dificuldade com algum assunto do livro-texto e quiser uma apresentação diferente. Ler os exercícios de outras fontes (e tentar resolver os que achar mais interessantes) também é uma boa maneira de estudar. Finalmente, antes de começar a ler e fazer exercícios, certifique-se de que entendeu o espírito, a forma e o porquê de uma demonstração matemática. Se não, então o link [O que é uma prova matemática?](#) da referência [11] é um bom lugar para começar a estudar.

## Material didático

Por conveniência, as notas de aula utilizadas serão disponibilizadas no formato de slides. Note que os slides servem apenas como guia para o professor e **não** substituem a leitura do livro-texto. A maioria dos slides foram ou criados e gentilmente cedidos pelo prof. [Cid Carvalho de Souza](#) e pela profa. [Cândida Nunes da Silva](#) (particularmente com modificações do prof. [Orlando Lee](#)); ou criados e gentilmente cedidos pelo prof. [Flávio Keidi Miyazawa](#). A maioria dos exercícios sugeridos foi retirada da bibliografia. Eu estou adaptando o material disponibilizado e possivelmente introduzindo erros, que podem e devem ser reportados a mim.

## Rotina de estudo

**Importante:** Note que não há provas final e de meio de semestre. Portanto é fundamental criar uma rotina de estudos contínua. Planeje-se e separe algumas horas e alguns dias por semana para ler o livro e resolver as listas de exercícios. @s alun@s também são encorajados a se reunir e estudar em grupo (sempre depois de tentar resolver os exercícios individualmente). Cada um@ tem sua própria maneira de estudar. Não obstante, algumas sugestões são úteis para o bom desenvolvimento da disciplina:

1. Leia o capítulo ou seção do livro correspondente ao conteúdo antes da aula correspondente (veja a ordem dos conteúdos abaixo); utilize a aula principalmente para tirar dúvidas e confirmar o seu entendimento.
2. Faça ou tente fazer os exercícios correspondentes a uma aula no próximo horário que tiver reservado para estudar a disciplina; anote as principais dificuldades e discuta com colegas ou monitor e, persistindo, com o professor no início das próximas aulas.
3. Veja o resultado das avaliações! Elas servem para que você identifique os problemas (erro de lógica, incorreção, conceitos incorretos, dificuldade com formalismo, incompletude, erros de português e escrita matemática, etc.). Tente refazer as tarefas e, se não puder identificar o que está errado ou não conseguir corrigir o problema, anote a dúvida e leve-a ao monitor ou ao professor (mas evite pedir aumento de nota, ou comparar notas de colegas).

## Horário e local

As aulas serão ministradas das 8 às 10 h, na sala XXXX às segundas-feiras e na XXXX às quartas.

## Atendimento

@s alun@s contarão com atendimento de monitoria (Hugo) todas as XXXX, a partir do dia XXXX, na sala XXXX, das XXXX até as XXXX. Para haver atendimento, é necessário que pelo menos um@ alun@ confirme presença enviando e-mail ao monitor até as XXXX do dia anterior. @s alun@s são fortemente encorajados a frequentar a monitoria, tirando dúvidas sobre o conteúdo e exercícios.

Havendo demanda e necessidade, um@ alun@ ou grupo de alun@s poderá combinar um horário para atendimento com o professor via e-mail, desde que solicitado com, pelo menos, três dias de antecedência.

## Organização do curso

O curso tem 10 unidades. As datas abaixo são apenas uma tentativa e servem para que @ alun@ se planeje e estude para as aulas. Datas, ordem e conteúdo podem variar a depender do andamento. **Sempre consulte também os avisos na página da disciplina!**

### i - Introdução

- 7/8/2017: Parte 1 - Introdução a análise e projeto de algoritmos
- 9/8/2017: Parte 2 - Um tour pelo problema da ordenação
- 14/8/2017: **Aula de exercícios 1**

### ii - Complexidade de algoritmo

- 16/8/2017: Parte 3 - Notação assintótica e crescimento de funções
- 21/8/2017: Parte 4 - Recorrências
- 23/8/2017: **Aula de exercícios 2**

### iii - Correção de algoritmos

- 28/8/2017: Parte 5 - Revisão de técnicas de demonstração
- 30/8/2017: Parte 6 - Princípio da indução
- 4/9/2017: Parte 7 - Invariantes de laço e demonstração de correção
- 6/9/2017: **Aula de exercícios 3**

### iv - Projeto de algoritmos por indução e divisão e conquista

- 11/9/2017: Parte 8 - Projeto de algoritmos por indução
- 11/9/2017: Divulgação do exercício de programação 1: entrega até 18/9/2017
- 13/9/2017: Parte 9 - Divisão e conquista
- 13/9/2017: **Teste de unidades 1**
- 18/9/2017: **Aula de exercícios 4**

### v - Algoritmos de ordenação

- 20/9/2017: Parte 10 - Visão geral de algoritmos de ordenação
- 25/9/2017: Parte 11 - Heapsort
- 25/9/2017: **Teste de unidades 2**
- 27/9/2017: **Aula de exercícios 5**

### vi - Introdução a algoritmos aleatorizados e ordenação por particionamento

- 2/10/2017: Parte 12 - Revisão de conceitos básicos de probabilidade
- 2/10/2017: Divulgação do exercício de programação 2: entrega até 9/10/2017
- 4/10/2017: Parte 13 - Quicksort
- 9/10/2017: **Aula de exercícios 6**

### vii - Ordenação em tempo linear

- 11/10/2017: Parte 14 - Cotas inferiores
- 16/10/2017: Parte 15 - Algoritmos lineares
- 16/10/2017: **Teste de unidades 3**
- 18/10/2017: **Aula de exercícios 7**

### viii - Estatísticas de ordem

- 23/10/2017: Parte 16 - Algoritmos para estatísticas de ordem
- 23/10/2017: Divulgação do exercício de programação 3: entrega até 30/10/2017
- 25/10/2017: Parte 17 - Seleção em tempo linear
- 30/10/2017: **Aula de exercícios 8**

### ix - Programação dinâmica

- 1/11/2017: Parte 18 - Algoritmos de programação dinâmica
- 1/11/2017: **Teste de unidades 4**
- 6/11/2017: Parte 19 - Exemplos de programação dinâmica
- 6/11/2017: Divulgação do exercício de programação 4: entrega até 13/11/2017
- 8/11/2017: Parte 19 - Exemplos de programação dinâmica
- 13/11/2017: **Aula de exercícios 9**

### x - Algoritmos gulosos

- 22/11/2017: Parte 20 - Algoritmos gulosos
- 27/11/2017: Parte 21 - Exemplos de algoritmos gulosos
- 27/11/2017: **Teste de unidades 5**
- 29/11/2017: **Aula de exercícios 10**