

Google Sala de Aula Para coordenar as atividades, o professor utilizará o Google Sala de Aula (<https://classroom.google.com>). Os alunos são inscritos automaticamente com o email @dac.unicamp.br pela DAC. Caso o aluno não seja inscrito automaticamente, ele deve entrar em contato com o professor com urgência.

Aulas A parte teórica do curso será fornecida de maneira assíncrona. O professor disponibilizará (no Google Sala de Aula) aulas teóricas gravadas em semestres anteriores e os slides (em formato PDF). Os alunos deverão assistir as aulas teóricas para aprender o conteúdo. No horário previsto para as aulas teóricas (terças e quintas-feiras, às 10h), o professor dará atendimento coletivo para os alunos.

Atendimento O professor fará atendimento no horário usual da aula através do Google Meet. Já os monitores farão o atendimento por texto, áudio e vídeo através do Discord (<https://discord.com/>). O professor fornecerá o convite para a plataforma no começo do semestre. Os atendimentos ocorrerão no horário previsto para a aula de laboratório (segundas às 14h e 16h) e em outros horários a serem divulgados no começo do semestre.

Programa da Disciplina • Estruturas ligadas: nó, apontador, variável apontadora, alocação dinâmica de memória • Listas ligadas simples: operações básicas • Comparação de listas ligadas com vetores • Algoritmos gerais para listas simples: enumeração, inversão, cópia, concatenação • Pilhas, filas, e aplicações • Intercalação (merge) de listas e mergesort; análise informal • Variações: listas circulares, duplamente ligadas, com cabeça. Lista livre • Algoritmos de ordenação • Árvores binárias: representação e percurso (recursivo) • Aplicação: árvores de busca (com inserção e remoção) • Árvores binárias de busca balanceadas • Fila de prioridade (heap) implementação com vetor e heapsort • Árvores B e generalizações • Introdução ao espalhamento (hashing): conceito, implementação com listas ligadas. • Grafos: conceito, representação por matrizes e listas ligadas • Percurso de grafos em largura e profundidade

Linguagens de Programação A linguagem de programação *C* será utilizada.

Tarefas Teremos diversas tarefas (ou “laboratórios”) a serem entregues durante o semestre. Todas as tarefas terão um prazo total de 7 dias ou mais para a primeira entrega e deverão ser feitos **individualmente**.

Para a correção dos tarefas, será utilizado um sistema criado pelo Prof. Leilton Pedrosa. Cada programa desenvolvido pelo aluno para uma tarefa específica será automaticamente avaliado por este sistema em vários testes. As informações de como usar o sistema serão dadas no começo do semestre.

A nota da tarefa dependerá de testes automáticos realizados. De qualquer forma, a nota pode sofrer descontos de acordo com a qualidade do programa apresentado ou caso

o programa submetido não satisfaça os critérios estabelecidos no enunciado da tarefa. Assim, mesmo que o código seja capaz de resolver todos os casos de teste, a nota final ainda pode ser menor do que 10.

Cada tarefa terá uma data para a primeira entrega da solução. Porém, até o final do semestre, os alunos poderão submeter tarefas que não tenham sido submetidas dentro do prazo da primeira entrega ou resubmeter tarefas já entregues com o objetivo de aumentar a nota obtida. Caso o aluno falhe em aumentar a nota, ele continuará com a nota original, isto é, a nota não será diminuída da primeira entrega para a entrega final. Cada tarefa será corrigido uma vez após o prazo inicial e apenas mais uma vez até o final do semestre (16/12/2021).

Testes Durante o semestre, vários testes, corrigidos automaticamente, serão propostos no Google Sala de Aula. O aluno poderá submeter o teste mais de uma vez para corrigir as respostas dadas e, assim, aprender com os erros cometidos. Os testes deverão ser entregue até o final do semestre (16/12/2021).

Avaliação

- Cada tarefa i terá um de acordo com a dificuldade.
- A média ML das tarefas será calculada como a média ponderada das tarefas.
- Cada teste i terá um peso de acordo com o número de questões e dificuldade.
- A média MT dos testes será calculada como a média ponderada dos testes.
- A média M , antes do exame, será a média harmônica ponderada entre ML (com peso 8) e MT (com peso 2), isto é,

$$M = \begin{cases} 0, & \text{se } ML = 0 \text{ ou } MT = 0, \\ \frac{10}{\frac{2}{MT} + \frac{8}{ML}}, & \text{caso contrário.} \end{cases} \quad (1)$$

- Ao invés da existência de um exame escrito, todos os alunos podem, opcionalmente, entregar tarefas e testes propostos durante o semestre até 16/12/2021 com o objetivo de elevar a média M . O aluno estará aprovado caso sua média M seja maior ou igual a 5,0 e estará reprovado caso contrário.

Observações

- Qualquer tentativa de fraude nas tarefas ou testes implicará em nota final $F = 0$ (zero) para todos os envolvidos, sem prejuízo de outras sanções. Exemplos de fraudes são:
 - Compartilhar trechos de código de qualquer forma.
 - Utilizar trechos de códigos da internet ou de outras fontes.
 - Copiar ou comprar um tarefa.
 - Disponibilizar soluções de laboratórios online antes do término completo do semestre letivo (16/12/2021).

Referências O professor não seguirá um livro texto específico, entretanto, os livros abaixo cobrem o que será visto em aula. Em particular, as principais referências são os livros 1 e 2 da seguinte lista.

1. R. Sedgewick, Algorithms in C. Addison-Wesley, 1990.
2. T. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein. Algoritmos — Teoria e Prática. Campus, 2002.
3. A. V. Aho, J. E. Hopcroft, J. Ullmann. Data Structures and Algorithms. Addison-Wesley, 1983.
4. W. Celes, R. Cerqueira, J. L. Rangel. Introdução a Estruturas de Dados. Campus, 2004.
5. M. J. Folk e B. Zoellick. File Structures. Addison-Wesley, 1992.
6. F. Lorenzi, P. N. de Mattos, T. P. de Carvalho. Estruturas de Dados. Thomson, 2007.
7. S. L. Pereira. Estruturas de Dados Fundamentais. Érica, 1996.
8. E. M. Reingold e W. J. Hanson, Data Structures. Little-Brown (1983).
9. J. L. Szwarcfiter e L. Markenzon. Estruturas de Dados e Seus Algoritmos. Editora LTC (1994).
10. D. E. Knuth, The Art of Computer Programming, Vol I: Fundamental Algorithms. Addison-Wesley (1978).
11. N. Wirth, Algorithms + Data Structures = Programs. Prentice-Hall (1976).
12. A. M. Tenenbaum. Estruturas de Dados Usando C. Makron Books, 1995.
13. N. Ziviani. Projeto de Algoritmos. Thomson, 2004.