

MC-202
Sobre a disciplina

Lehilton Pedrosa
lehilton@ic.unicamp.br

Universidade Estadual de Campinas

Segundo semestre de 2024

Sobre o que é MC202?

MC102: problemas computacionais simples

- temos poucos dados ou conjuntos de dados muito simples
- estruturas de controle (condicionais, laços, funções, etc.)
- estruturas de dados primitivas (variáveis, listas, matrizes)

MC202: problemas computacionais mais difíceis

- queremos lidar com uma grande quantidade de dados
- ou dados com estruturas muito mais complicadas
- não basta resolver o problema, mas resolver rapidamente
- não basta guardar dados, mas acessá-los eficientemente

Iremos organizar nossos dados usando ideias inteligentes

- usaremos algoritmos rápidos para manipular as estruturas
- sempre com alguma garantia de qualidade

Ementa

Estruturas básicas para representação de informações: listas, árvores, grafos e suas generalizações. Algoritmos para construção, consulta e manipulação de tais estruturas. Desenvolvimento, implementação e testes de programas usando tais estruturas em aplicações específicas.

Programa

- 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

Importância de Estruturas de Dados e Algoritmos

Algoritmos são fundamentos da Computação

- Manipulam sistemas complexos: redes sociais, bancos, etc.
- Resolvem problemas complexos: genoma, roteamento, etc.
- E devem ser rápidos mesmo para grande quantidade de dados

E onde estão as estruturas de dados?

- Em todo lugar que armazenarmos dados no computador
- E em todo algoritmo que manipular esses dados
- Mas algumas estruturas de dados são bem mais elaboradas:
 - Ex: no CRUD de praticamente todo site da internet
- E outras estruturas são extremamente sofisticadas:
 - Ex: na rota mais rápida do Google Maps

ED é um dos principais cursos da Computação!

C vs. Python

Por que MC102 em Python (ao invés de C)?

- Python é (talvez) mais simples de aprender do que C
 - É uma linguagem de nível mais alto
 - O aluno precisa se preocupar com menos detalhes
 - strings
 - listas e dicionários
 - entrada e saída
 - Aprender lógica de programação
 - Sem ter tantas barreiras de entrada
- Python tem muitas bibliotecas disponíveis
 - Permite fazer várias coisas diferentes facilmente
 - jogos
 - processamento de informações
 - plotar gráficos
 - desenvolvimento web
 - Muito útil para alunos de todas as áreas

C vs. Python

Por que MC202 em C (ao invés de Python)?

- Dismistificar o Python
 - Como funciona um dicionário (dictionary)?
 - Como representar um conjunto (set)?
 - Como representar uma string?
 - Como funciona recursão?
- Ter um entendimento mais profundo de programação
 - Entender como a memória funciona
 - Ser capaz de manipular a memória
 - Entender melhor outras linguagens
- Aprender uma segunda linguagem de programação
 - C é muito usado para diversas tarefas
 - Muito mais rápido do que Python

Bibliografia

As duas principais referências para o curso são:

1. “**Algorithms in C - Third Edition**” de R. Sedgewick
2. “**Introduction to Algorithms**” de T. Cormen, C. E. Leiserson, R. L. Rivest e C. Stein
 - Pegue a versão em inglês (segunda ou terceira edição)
 - Ou a terceira edição em português

Outras referências estão no PDD.

Material didático

- Todo material está disponível no site da disciplina:
 - <https://www.ic.unicamp.br/~lehilton/mc202>
 - leia o plano de desenvolvimento da disciplina com atenção!

No site, você encontra, para cada aula:

- Slides
- Listas de exercícios de fixação
- Exercícios resolvidos
- Links interessantes
- Leituras recomendadas

Também tem alguns tutoriais úteis e listas de exercícios

Horários das aulas

As aulas teóricas

- Quartas e quintas às 19h na CB13

As aulas de laboratório

- Terças 21h no Lab 300
- Com os monitores das disciplinas
- Usem o tempo para fazer exercícios e tarefas

Cronograma de aulas:

- 15/10 (não haverá aula, participem da avaliação de curso)
- 23/10 (não haverá aula)
- 20/11 (não haverá aula)

Atendimento

Professor:

- Durante a segunda parte das aulas
- Se necessário, em horário agendado
- Ou sempre que minha porta estiver aberta

Monitores:

- Atendimento assíncrono via Discord
 - <https://discord.gg/cR4wAmWJ>
- Atendimento individual
 - presencial e online
 - veja escala de horários no Discord

Exercícios de fixação

Para cada unidade, haverá uma lista de exercícios de fixação

- Exercícios simples e mais complicados
- Vocês podem discutir e compartilhar código dos exercícios de fixação livremente
 - Mas isso deve ser indicado no código
 - Servem para praticar e colaborar com seus colegas
- Devem ser colocados no repositório pessoal
 - Não é obrigatório fazer e não serão corrigidos
 - Mas quem fizer será bonificado com **nota de participação**

Testes

Durante o semestre teremos vários testes

- No Google Classroom
- Rápidos e curtos
- Podem ser refeitos quantas vezes vocês quiserem
 - Para corrigir os erros
 - Até o prazo de entrega

Tarefas

Durante o semestre teremos várias tarefas:

- A entrega será via GitLab
- Não aceitaremos entregas por outras formas
- A tarefa zero terá as instruções
- Mas não é ruim começar a estudar o básico de `git`

Notas das tarefas:

- Depende do sucesso em vários casos de teste
- Pode ser zerado por não satisfazer critérios do enunciado
- Desconto de até 2 pontos por problemas de qualidade:
 - nomes de variáveis, funções e estruturas
 - funções muito longas, muito complexas
 - funções com muitas responsabilidades
 - código muito repetitivo ou muito aninhado
 - soluções muito complicadas para problemas simples
 - código lento
 - vazamento de memória

Prazos das tarefas

Você tem até duas chances para entregar uma tarefa

Primeira Chance:

- Prazo de entrega normal da tarefa
- Normalmente nas “segundas-feiras”
 - Na verdade, terças-feiras às 6h
- Logo depois do conteúdo ser ensinado

Segunda Chance:

- Algum tempo após a entrega da nota da primeira chance
- Permite aumentar a nota da tarefa entregue
- Permite entregar uma tarefa não entregue
 - nesse caso, tem desconto de 20%
 - mas se tiver entregue com nota ≥ 5 , não desconta
- Nem toda tarefa tem segunda chance

Não deixem as tarefas acumularem!

Média do semestre

- M é a média ponderada das tarefas
- T é a média ponderada dos testes
- F é a frequência

Nota do semestre é

$$A = \begin{cases} 0,9M + 0,1T & \text{se } M \geq 6, T \geq 6 \text{ e } F \geq 75\% \\ \min\{M, 4\} & \text{caso contrário.} \end{cases}$$

Maior parte da nota vem das tarefas, mas deve fazer os testes!

Exame

O exame consistirá em completar um subconjunto das tarefas:

- só pode fazer exame se $2,5 \leq A < 5$ e $F \geq 75\%$
- deverá entregar até 10/12/2024
- precisará **apresentar** oralmente ao professor
- serão combinados horário e data com antecedência
- o professor atribuirá nota E
- a nota final será $\min\{5, (E + A)/2\}$

Fraudes

Qualquer tentativa de fraude, os envolvidos serão reprovados com nota 0 e será registrada a ocorrência no histórico escolar.

Exemplos de fraude:

- plágio
- atestado falso
- assinar lista de presença por colegas
- usar bibliotecas não permitidas
- copiar quaisquer trechos da internet sem autorização expressa
- mostrar ou distribuir tarefa de programação individual
- cola independentemente de origem
- consulta a material proibido
- tentar burlar testes automáticos

Fraudes

Fique atento:

- Cada pessoa é responsável por manter seguros os arquivos de seu repositório.
 - Não compartilhe sua senha nem deixe cópias de arquivos em computadores compartilhados.
- É proibido utilizar ajuda de terceiros para realizar as tarefas sem autorização expressa do professor da disciplina.
 - Ex: colegas, amigos, professores particulares
- É permitido tirar dúvidas e discutir a respeito de tarefas de programação com colegas.
 - toda discussão deve ocorrer em canais abertos no Discord
 - não é permitido compartilhar algoritmos para tarefa (português, código ou pseudocódigo), nem ideias prontas
- O material disponibilizado e os trabalhos submetidos nas avaliações são para uso exclusivo da disciplina.
 - Não compartilhem os trabalhos mesmo após o semestre
 - Se quiser publicar algo, converse antes com o professor.

E o tal do ChatGPT?

Pode usar?

- Não para resolver as tarefas
- Mas pode ser uma ferramenta de aprendizado!

O mesmo vale para outras ferramentas de IA, sites da internet, livros, etc.

Como deixar a disciplina mais fácil?

Tire dúvidas e peça ajuda (e peça ajuda logo)

- Perguntas em sala de aula
- Aulas de laboratório
- Monitorias
- Discord

Use um bom ambiente computacional

- Considere usar GNU/Linux ao invés de Windows
- Ou explore as alternativas que o Windows dá
- Use um bom editor de texto (ex: VSCode)
- Use um debugger
- Use o valgrind

E invista tempo em aprender a usar bem essas ferramentas

Estude C agora no começo do curso

- Principalmente ponteiros e recursão...

E não deixe as tarefas acumularem!