

MC-202

Sobre a disciplina

Rafael C. S. Schouery
rafael@ic.unicamp.br

Universidade Estadual de Campinas

Atualizado em: 2024-09-18 10:58

Sobre o que é MC202?

MC102 - programação básica:

- o importante era aprender os conceitos básicos
 - variáveis
 - condicionais (if)
 - laços (for, while)
 - funções
 - etc
- resolver problemas computacionais simples

MC202 - programas rápidos:

- queremos lidar com uma grande quantidade de dados
- não basta resolver o problema, precisa que seja rápido

Iremos estruturar nossos dados usando ideias inteligentes

- usaremos algoritmos rápidos para manipular as estruturas
- sempre com uma garantia matemática 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 a base teórica da computação:

- Permite que façamos sistemas complexos
 - Ex: Redes Sociais, Netbanking, etc.
- Ou resolvamos problemas complexos
 - Sequenciamento de Genoma, Otimizações, etc.
- E rápidos mesmo para uma grande quantidade de dados

Muitos destes algoritmos utilizam estruturas de dados

- Ex: Google Maps e rota mais rápida

E muitos problemas envolvem armazenar dados

- Ex: praticamente todo site da internet

ED é um dos cursos principais para a 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)?

- Desmistificar 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

Aprendendo C

Inicialmente, teremos um curso rápido de C:

- Será suficiente para ver o conteúdo básico de C
- Mas vocês precisam estudar bastante agora no começo
 - Aproveitem as monitorias e aulas de laboratório
 - Leiam os materiais sugeridos
 - Façam as tarefas
 - Aprendam a usar o debugger e o valgrind
 - Instalem tudo o que vocês precisam no computador de vocês
 - E aprenda a usar direito
 - Vocês vão produzir códigos longos e precisam de um ambiente de desenvolvimento bom
- O sucesso na disciplina depende bastante desse esforço inicial

Site da Disciplina

<https://ic.unicamp.br/~rafael/mc202>

No site, você encontra, para cada aula:

- Slides
- Gravações
- Exercícios resolvidos
- Links interessantes
- Leituras recomendadas

Também tem alguns tutoriais úteis e listas de exercícios

Já tem muita coisa lá e será atualizado durante o semestre

- Os slides têm a data de atualização na primeira página

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

Mais referências bibliográficas no PDD

- E outros materiais no site da disciplina

Atendimento

Presencialmente na aula de laboratório

- segundas das 14h às 18h
- não importa se você é da turma A ou B
- na sala 300 do IC3

Presencialmente em horários específicos

- que serão divulgados em breve

Online via Discord

- <https://discord.gg/cR4wcwAmWJ>

Tarefas

Durante o semestre teremos várias tarefas:

- A entrega será via GitHub
- 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
- Pode ter descontos pela qualidade do programa apresentado
 - Desconto de no máximo 2 pontos
 - Ex: nomes de variáveis, funções e estruturas
 - Ex: funções muito longas, muito complexas ou com muitas responsabilidades
 - Ex: código muito repetitivo ou muito aninhado
 - Ex: soluções muito complicadas para problemas simples
 - Ex: código lento
 - Ex: vazamento de memória

Primeira e Segunda Chance

Toda tarefa tem uma primeira e uma segunda chance

Primeira Chance:

- Prazo de entrega normal da tarefa
- Normalmente nas “segundas-feiras”
 - Na verdade, terças-feiras às 06h
- 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
- Nem toda tarefa tem segunda chance

Não deixem as tarefas acumularem!

Nota das Tarefas

Notas:

- l_i^1 é a sua nota da primeira chance
 - É zero se você não entregou
- l_i^2 é a sua nota da segunda chance
 - É zero se você não entregou

Nota final da tarefa:

$$l_i = \begin{cases} \max(l_i^1, l_i^2), & \text{se } l_i^1 \geq 5, \\ \max(l_i^1, 0,8 \cdot l_i^2), & \text{caso contrário.} \end{cases}$$

Ou seja, se você foi minimamente bem na primeira chance, não tem desconto de nota na segunda chance!

- E pode consertar os erros da primeira chance

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 final

Cálculo da Média

- ML é a média das tarefas
- MT é a média dos testes

$$M = \begin{cases} \min(ML, MT), & \text{se } ML < 5 \text{ ou } MT < 5 & (1) \\ \frac{9ML + MT}{10}, & \text{caso contrário.} & (2) \end{cases}$$

Laboratórios têm um peso maior, mas não pode ir mal nos testes!

Exame

O exame final consiste em entregar tarefas e testes propostos durante o semestre até **10/12/2024**

Tarefas terão uma nova nota calculada usando a fórmula

$$l_i := \max(l_i, 0,5 \cdot l_i^e)$$

Testes também

$$t_i := \max\{t_i, t_i^e\}.$$

A média final será calculada usando a mesma fórmula de M

- Mas é no máximo **5** para alunos que ficaram de exame

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.
 - Mas toda discussão deve ser feita em canais abertos no Discord
 - e não é permitido compartilhar algoritmos para resolver a tarefa (em português, código ou pseudocódigo), nem ideias prontas de solução, etc.
- O material disponibilizado e os trabalhos submetidos nas avaliações são para uso exclusivo da disciplina.
 - Não compartilhem seus trabalhos mesmo após o término da disciplina.
 - Se desejar publicar algum trabalho que realizou, converse antes com o professor.

Fraudes

Caso o aluno realize uma fraude e se arrependa, ele deve

- entrar em contato com o professor explicando o que ocorreu
- e quem foram os envolvidos
 - Sem não contar, não será considerado arrependimento

A penalidade será obter nota zero nas atividades envolvidas

- Tal atitude só será válida se ocorrer antes do professor detectar e acusar a fraude
- O aluno não ficará imune a ser reprovado por outras fraudes existentes, apenas pela fraude declarada
- Outros participantes da fraude que não se manifestarem serão enquadrados pela regra da reprovação

Os alunos podem, a qualquer momento, contatar o professor, inclusive de maneira anônima, para esclarecer se determinado comportamento é considerado fraude ou não

Fraudes

É melhor não entregar do que ser pego por fraude!

Serei bem inflexível em relação a fraudes

- Laboratórios e testes são as únicas forma de avaliação
- Acusados de fraude poderão apresentar uma defesa ao professor
- E poderão pedir a formação de uma comissão para avaliar o caso

Sempre tem aluno que ia passar com nota alta...

- Mas passa a tarefa para o amigo
- E reprova por fraude...

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 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!

Bom Curso!