

MCCC003-23 -- Algoritmos em Grafos -- 2024.3

Sep 25, 2024

Atualizado em 26/09/24

Expediente

- Professor: [Aritanan Gruber](#), sala S-539.2
- Aulas: Ter. 08–10h e Qui. 10–12h na S-108.0
- Atendimento: Ter. 10–12h (na S-539.2)
- Monitoria: Guilherme Afonso Gigeck, Zzz. XX–YYh na sala de monitoria (5o andar, torre 2)
- Moodle: [AG \(Q24.3-DA1\)](#)
andamento do curso, links úteis, material, avaliações, notas, mensagens, etc.

Ementa

Revisão da terminologia básica de Teoria dos Grafos. Noções de análise de algoritmos. Estruturas de dados para representação de grafos. Buscas em largura e profundidade e suas aplicações: caminhos mínimos sem pesos, componentes conexas. Grafos ponderados. Árvores geradoras mínimas: algoritmos de Prim e Kruskal. Grafos Eulerianos e Hamiltonianos. O Problema do Caixeiro Viajante. Digrafos: definições básicas, componentes fortemente conexas e ordenação topológica. Caminhos mínimos em digrafos ponderados: algoritmos de Dijkstra, Bellman-Ford e Floyd–Warshall. Fluxo máximo: algoritmo de Ford-Fulkerson e suas aplicações.

Objetivos

Mostrar como representar um grafo computacionalmente (Matriz de Adjacências e Lista de Adjacências), discutir as vantagens e desvantagens de cada abordagem e discutir como adaptar tais estruturas para representar grafos nos quais os vértices e/ou arestas possuam atributos (por exemplo, grafos orientados, grafos ponderados, grafos com os vértices coloridos). Apresentar algoritmos eficientes para problemas clássicos em grafos e discutir, sem a necessidade de muito rigor formal, os tempos de execução dos algoritmos estudados. Discutir o impacto das estruturas de dados escolhidas para a implementação dos algoritmos sobre o tempo de execução desses. Espera-se que ao final do curso o aluno seja capaz de modelar problemas em grafos. Espera-se que este conheça os principais problemas em grafos e os algoritmos eficientes que os resolvem. Espera-se também que o aluno tenha demonstrado capacidade de implementar tais algoritmos em uma linguagem de alto nível. Por fim, espera-se que o aluno tenha noções da complexidade de tempo de execução dos algoritmos cobertos ao longo do curso. Este curso tem como objetivo desenvolver tanto habilidades teóricas (algoritmos) como práticas (implementações eficientes de tais algoritmos).

Recomendação

Natureza da Informação, Funções de Uma Variável, Algoritmos e Estruturas de Dados I e II, Programação Estruturada, Matemática Discreta I e II.

Avaliações e critérios

- duas listas de exercícios L_1 e L_2 ; datas de entrega a serem definidas ao longo do quadrimestre;
- dois exercícios programados E_1 e E_2 ; datas de entrega a serem definidas ao longo do quadrimestre;
- duas provas regulares P_1 e P_2 e uma prova substitutiva P_3 aberta;
 - P1 : Qui. 07/11 @ 10h
 - P2 : Ter. 17/12 @ 08h
 - P3 : Qui. 19/12 @ 10h

Nota nominal (N): com

$$P = \frac{1}{2} \max \left\{ \sum_{j \in S} P_j : S \in \binom{[3]}{2} \right\} = \frac{1}{2} \max \{P_1 + P_2, P_1 + P_3, P_2 + P_3\}, \quad P_j \in [0, 10]$$

$$L = \frac{1}{2}(L_1 + L_2) \quad \text{e} \quad E = \frac{1}{2}(E_1 + E_2),$$

tem-se que

$$N = \frac{3}{5}P + \frac{1}{5}L + \frac{1}{5}E.$$

Conceito nominal (C_N): reflete o seu desempenho frente ao material apresentado e às avaliações realizadas; obtido pelo encaixe de N em um dos intervalos:

$$-\infty < \mathbf{F} < 5.0 \leq \mathbf{D} \leq 6.0 < \mathbf{C} \leq 7.0 < \mathbf{B} \leq 8.5 < \mathbf{A} < \infty.$$

Recuperação

Caso seu conceito C_N seja **D** ou **F**, você tem direito a uma prova de recuperação P_R . Esta será única e contemplará toda a matéria do quadrimestre. Uma nova nota nominal $\bar{N} = (N + P_R)/2$ será utilizada para gerar um novo conceito (nominal) *final pós-recuperação* \bar{C}_N . Note que seu conceito final pós-recuperação pode ser menor que o pré-recuperação: uma vez feita, a recuperação é parte integrante da sua avaliação.

Bibliografia

Primária

- [CLRS] T. Cormen, C. Leiserson, R. Rivest, C. Stein, [Introduction to Algorithms](#), 3rd ed., MIT University Press (2009) [é possível se virar com a 2a ed.]
Versão em Português: *Algoritmos: Teoria e Prática*, 3a ed., Elsevier (2012)
[Não confunda com a tradução da 2a edição pela Campus (2001), que tem baixa qualidade.]
- [Fe] P. Feofiloff, [Algoritmos para Grafos via Sedgewick](#), self-published (2020)
- [KT] J. Kleinberg, E. Tardos, [Algorithm Design](#), Pearson Education, Inc. (2006)

Secundária

- [AMO] R.K. Ahuja, T.L. Magnanti, J.B. Orlin, *Network Flows: Theory, Algorithms, and Applications*, Prentice Hall (1993)
- [BM] J.A. Bondy, U.S.R. Murty, *Graph Theory*, Springer-Verlag (2008)
- [DPV] S. Dasgupta, C. Papadimitriou, U. Vazirani, [Algorithms](#), McGraw-Hill (2006)
Versão em Português: *Algoritmos*, AMGH (2009)
- [Er] J. Erickson, [Algorithms](#), self-published (2019)
- [SW] R. Sedgewick, K. Wayne, [Algorithms](#), 4th ed., Addison-Wesley (2011)

Lista de tópicos por semana (tentativa)

Detalhes de cada tópico (coberto nas aulas) serão atualizados no Moodle ao longo do quadrimestre.

| Aulas | Datas | Tópicos |
|-------|-------|---|
| A01 | 01/10 | Grafos e dígrafos: conceitos elementares, exemplos e representações |
| A02 | 03/10 | " |
| A03 | 08/10 | Busca em largura (BFS), propriedades e aplicações (distâncias) |
| A04 | 10/10 | " |
| A05 | 15/10 | Busca em profundidade (DFS), propriedade e aplicações (ord. topológica) |
| A06 | 17/10 | " |
| A07 | 22/10 | Componentes conexas e aresta-biconexas |
| A08 | 24/10 | Componentes fortemente conexas (Tarjan, Kosaraju) |
| A09 | 29/10 | " |
| A10 | 31/10 | Trilhas e ciclos eulerianos |
| A11 | 05/11 | Grafos bipartidos, circuitos ímpares e bi-coloração |
| A12 | 07/11 | Prova P1 |
| A13 | 12/11 | Árvores geradoras de custo mínimo (Prim, Kruskal, Boruvka) |
| A14 | 14/11 | " |
| A15 | 19/11 | Arborescências de custo mínimo (Edmonds) |
| A16 | 21/11 | Caminhos de custo mínimo: origem simples (Dijkstra, Bellman-Ford) |
| A17 | 26/11 | Caminhos de custo mínimo: todos os pares (Floyd-Warshall, Johnson) |
| A18 | 28/11 | Fluxos-máximos / cortes-mínimos: teoria |
| A19 | 03/12 | Fluxos-máximos / cortes-mínimos: algoritmos (Edmonds-Karp, Dinics) |
| A20 | 05/12 | Fluxos-máximos / cortes-mínimos: aplicações |
| A21 | 10/12 | Emparelhamentos em grafos bipartidos: algoritmo húngaro |
| A22 | 12/12 | Circuitos hamiltonianos e TSP: condições suficientes, NP-hardness, prog.din. (Held-Karp), aprox. (Christofides) |
| A23 | 17/12 | Prova P2 |
| A24 | 19/12 | Prova P3 (substitutiva) |

Estudando para esta disciplina

A natureza do tópico, o posicionamento do curso na grade, e a lista de pré-requisitos indicam que esta é uma disciplina de nível intermediário; e será tratada como tal. Você deve assistir às aulas, estudar a bibliografia indicada, e dedicar-se às listas de exercícios.

Caso seus pré-requisitos não estejam tão sólidos quanto desejável (falta de familiaridade com formalismo matemático e raciocínio algorítmico, atitude passiva com relação ao aprendizado, tempo dedicado insuficiente, etc.), será possível fazer confusões e sentir-se perdido no início.

Alguns procedimentos que costumam funcionar em cursos introdutórios para mitigar os motivos relacionados também costumam funcionar por aqui:

- Refaça os exemplos e re-prove os resultados fornecidos em sala de aula.
- Preste atenção aos processos de solução (aprenda-os!) e não foque somente nos resultados finais.
- Assista ativamente às aulas; resolva os exercícios nelas propostos e os contidos nas listas.
- Estude a bibliografia indicada, monte grupos de estudo, e faça um bom uso dos horários de atendimento.
- Tenha sempre em mente que aprendizado é uma tarefa *ativa*; não fique somente assistindo. “Ouvir” às aulas e “ler” os livros tem pouco ou nenhum efeito neste curso – e em qualquer disciplina matemática/algorítmica que o valha.
- Se ainda assim, sentir-se perdido, repita os passos acima. Mais cedo ou mais tarde, eles convergirão à compreensão.

Note que você não será convidado a regurgitar respostas fornecidas em aula ou presente nos livros. As questões em listas e provas testarão sua capacidade de entender os problemas e apresentar uma solução para eles; às vezes, serão uma adaptação simples ou uma extensão direta do que foi visto, outras, será necessário relacionar dois ou mais métodos ou conceitos apresentados, e outras ainda, irão requerer análise e raciocínio mais profundo (o que leva tempo, então não deixe nada para a última hora!).

Tenha em mente: além do escrito acima, para aproveitar bem este curso, você deve ler os slides e familiarizar-se com o material na leitura sugerida correspondentes antes da aula, e estudá-los com afinco depois.

Integridade acadêmica e transgressões

O Artigo 25 do [Código de Ética da UFABC](#) estabelece, à página 23: “Quanto aos trabalhos acadêmicos, é eticamente inaceitável que os discentes:

- I - fraudem avaliações;
- II - fabriquem ou falsifiquem dados;
- III - plajeiem ou não creditem devidamente autoria;
- IV - aceitem autoria de material acadêmico sem participação na produção;
- V - vendam ou cedam autoria de material acadêmico próprio a pessoas que não participaram da produção.”

Trabalhos (listas, provas, programas) suspeitos de cópia ou de outra representação fraudulenta acarretarão aos envolvidos conceitos **F** (falha) no curso. A atividade será reportada à [Comissão Disciplinar Discente](#) da universidade para que sejam tomadas todas as providências disciplinares cabíveis.

LLMs, GPTs e similares

Representam um grande avanço da inteligência artificial generativa sendo assim, por este ponto de vista, resultados magníficos. De qualquer forma, soluções transcritas e/ou entregues que tenham sido produzidas por eles enquadram-se no Artigo 25 acima.

Para pensar ao longo do curso: *Do que adianta as máquinas aprenderem e os alunos não?*



Aritanan Gruber

Assistant Professor

“See, if y’all haven’t the same feeling for *this*, I really don’t give a damn. If you ain’t feeling it, then dammit *this* ain’t for you!”
(desconheço a autoria; agradeço a indicação)

[✉](#) [G](#) [R](#) [ID](#) [in](#) [T](#) [CV](#)

