TEG

Gilmário B. Santos

gilmario.santos@udesc.br

http://www.joinville.udesc.br/portal/pagina/gilmario

TEG

Ementa:

Conceitos e definições de grafos. Representação computacional. Conexividade, Isomorfismo, Planaridade e Coloração.

Ordenação topológica. Grafos Hamiltonianos, Eulerianos e Árvores. Buscas em Grafos. Caminho Mínimo. Árvore geradora.

Fluxos em Redes. Introdução ao estudo de estruturas combinatórias.

TEG

Avaliação: mínimo 3 atividades na forma de trabalho prático envolvendo a implementação autoral com a entrega do código fonte desenvolvido e relatório e/ou nota obtida em uma prova escrita.

Implementações práticas:

- Em “C” (e talvez em Python)

Moodle: repositório do material e atividades do curso.

TEG

Para as atividades na forma de trabalhos práticos:

- Relatório em formato PDF;
- Codificação em linguagem C (exceto se for especificado em contrário);
- Implementação em código compilável no Linux;
- Código, relatório etc são entregues exclusivamente via Moodle, compactados e dentro dos prazos;
- Nomeie o arquivo compactado como *T_#_iniciaisDosAutores*
  - Onde "#" é o número da tarefa. Exemplo: T_1_JS_PF.zip para a submissão da solução da tarefa n. 1, cujos autores são José Silva e Pedro Francisco.
- Se for o caso, haverá 0,25 de desconto para cada 6 horas de atraso do *upload*;

TEG

Para as atividades na forma de trabalhos práticos:

- Codificação em linguagem e compilável no Linux:
  - Utilize um C padrão sem bibliotecas específicas para SO Windows ou indique as bibliotecas equivalentes (às específicas) que devem ser usadas na compilação no SO Linux;
  - Compile usando "*gcc -Wall -o ./saida programa.c*" e elimine cada warning que ocorrer.

- Relatório: sempre em PDF, a função do relatório é:
  - Identificar a equipe de execução e os papéis de cada membro (se todo mundo fez tudo, então qualquer membro tem que ter a capacidade de responder a qualquer pergunta);
  - Apresentar um resumo descritivo de atividades realizadas bem como informar os dados e resultados coletados com elas. Sua estrutura apresenta: título, introdução, referências, desenvolvimento, conclusão e, em alguns casos, sugestões.
  - Maiores detalhes: https://mundoeducacao.uol.com.br/redacao/o-relatorio.htm

TEG

Bibliografia

Básica

LUCCHESI, C. L. et alli. Aspectos Teóricos da Computação, Parte C: Teoria dos Grafos, projeto Euclides, 1979.

SANTOS, J. P. O. et alli. Introdução à Análise Combinatória. UNICAMP; 1995.

SZWARCFITER, J. L. Grafos e Algoritmos Computacionais. Campus, 1986.

GERSTING, Judith L. Fundamentos Matemáticos para a Ciência da Computação. Rio de Janeiro. 3a Ed. Editora.

Complementar:

1.) CORMEN, T. Introduction to Algorithms, third edition, MIT press, 2009

2.) ROSEN, K. Discrete Mathematics and its applications, seventh edition, McGraw Hill, 2011.

3.) WEST, Douglas, B. Introduction to Graph Theory, second edition, Pearson, 2001.

4.) BONDY, J.A., MURTY, U.S.R., Graph Theory with applications , Springer, 1984.

5.) SEDGEWICK, R. Algorithms in C - part 5 - Graph Algorithms, third edition, 2002, Addison-Wesley.

6.) GOLDBARG, M., GOLDBARG E., Grafos: Conceitos, algoritmos e aplicações. Editora Elsevier, 2012.

7.) BONDY, J.A., MURTY, U.S.R., Graph Theory with applications , Springer, 1984

8.) FEOFILOFF, P., KOHAYAKAWA, Y., WAKABAYASHI, Y., uma introdução sucinta à teoria dos grafos. 2011. (www.ime.usp.br/~pf/teoriadosgrafos)

9.) DIESTEL, R. Graph Theory, second edition, springer, 2000

10.) FURTADO, A. L. Teoria de grafos. Rio de janeiro. Editora LTC. 1973.

11.) WILSON, R.J. Introduction to Graph Theory. John Wiley & Sons Inc., 1985

12.) BOAVENTURA NETTO , P. O. Grafos: Teoria, Modelos, Algoritmos. Edgard Blucher, SP, quinta edição

Tutoriais, artigos, notas de aula...

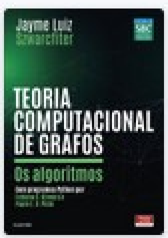
Vários livros podem ser acessados np formato eletrônico (e-book) via

https://www.udesc.br/bu/acervos/ebook

Exemplos:

Fundamentos Matemáticos para a Ciência da Computação

Judith L. Gersting



Teoria Computacional de Grafos - Os Algoritmos

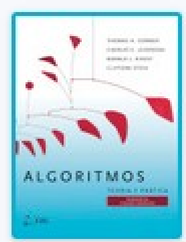
Jayme Luiz Szwarcfiter

Fundamentos Matemáticos para a Ciência da Computação

Judith L. Gersting

Grafos

Marco Goldbarg



Algoritmos - Teoria e Prática

Thomas Cormen

Aulas e interação

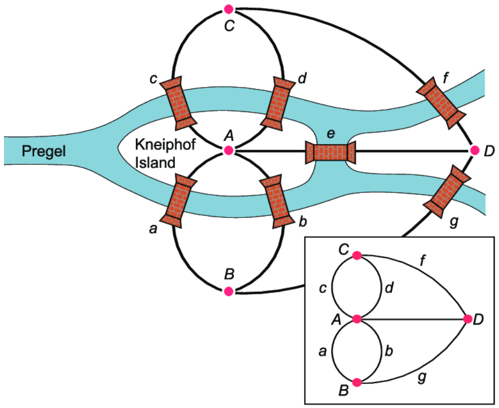
- O material do curso estará centrado no MOODLE

- Se necessário poderemos utilizar recurso de aula online via BBB:

BigBlueButton

Um dos primeiros registros históricos da utilização de grafos surgiu do problema das Pontes de Königsberg. A cidade de Königsberg é a antiga capital da Prússia Oriental, conhecida atualmente pelo nome de Kaliningrado.

A cidade é dividida em 4 zonas criadas pelo percurso do rio Pregel, no séc. XVII essas zonas estavam ligadas por sete pontes conforme a figura.



Por muito tempo os habitantes da cidade questionavam se era possível cruzar as sete pontes numa caminhada contínua sem passar duas vezes por qualquer uma delas!

Em 1736, o matemático Euler demonstrou que não existe tal trajeto, ao utilizar um modelo em grafos para uma generalização deste problema. Através desse modelo ele verificou que existe o desejado trajeto quando e somente quando em cada região concorrer um número par de pontes.

Na verdade o problema consiste na determinação de um caminho Euleriano, ou seja, um trajeto que usa cada ponte exatamente uma vez...

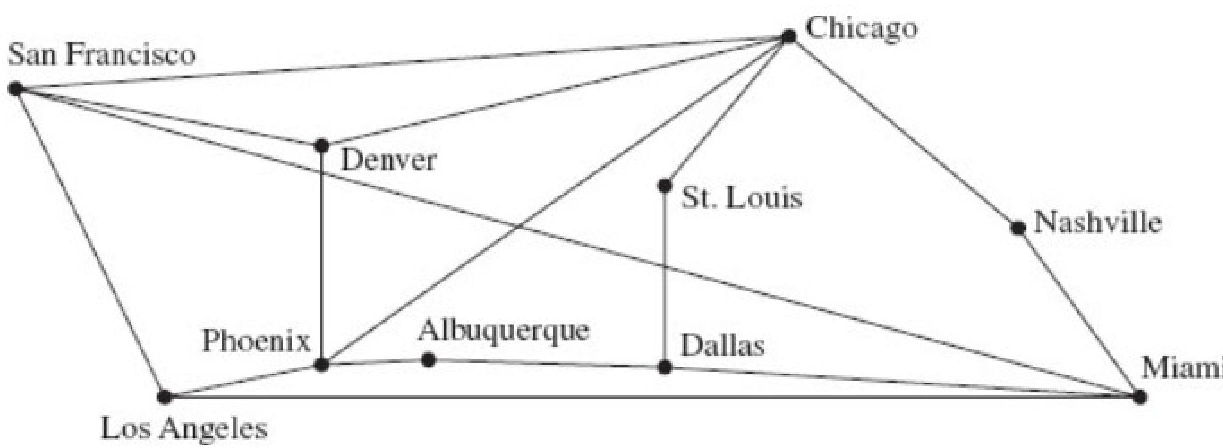
Onde você já viu um grafo?

Onde você já viu um grafo?

Um grafo (= *graph*) é um animal formado por dois conjuntos: um conjunto de coisas chamadas vértices e um conjunto de coisas chamadas arcos ou arestas;

Cada arco/aresta está associado a dois vértices: o primeiro é a ponta inicial do arco e o segundo é a ponta final.

Você pode imaginar que um grafo é um mapa rodoviário idealizado: os vértices são cidades e os arcos são estradas de mão única.



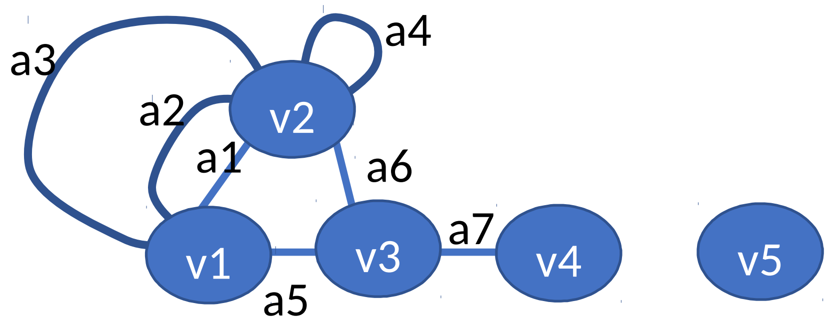
Formalmente:

Os grafos representam um tipo particular de gráfico, e corresponde a um conjunto finito e não vazio ‘N’ de nós/vértices e uma família finita ‘A’ de arcos/arestas constituída de pares não ordenados de elementos de ‘N’, ou seja, cada arco/aresta conecta dois nós/vértices.

Um grafo ‘G’ é representado por uma tripla ordenada (N, A, g) [Gersting, 2004], onde:

N = um conjunto não vazio de nós (vértices)
A = um conjunto de arcos (arestas), que correspondem a pares não ordenados.
g = uma função que associa a cada arco um par não ordenado x-y de nós, chamados de extremidades de a.
|N| = numero de vértices
|A| = número de arestas

12



N={v1,v2,v3,v4,v5}
A={a1,a2,a3,a4,a5,a6}
Onde: a1=(v1,v2), a2=(v1,v2), a3=(v2,v1), a4=(v2,v2) , a5=(1,3), a6=(2,3), a7=(3,4)
Ou seja, g(a1) estabelece a aresta entre v1 e v2 e assim sucessivamente…

Note que:
a4 é um laço;

a1, a2 e a3 são uma repetição da mesma aresta entre vértice 1 e vértice 2. Nesse caso se diz que são arestas múltiplas. Isso é permitido porque família de arestas pode repetir elementos e os pares de vértices ligados não são ordenados

v5 é um vértice isolado,

Outro exemplo:

V = { V1,V2,V3,V4,V5} E = {e1,e2,e3,e4,e5}

E1 → {V1,V2}

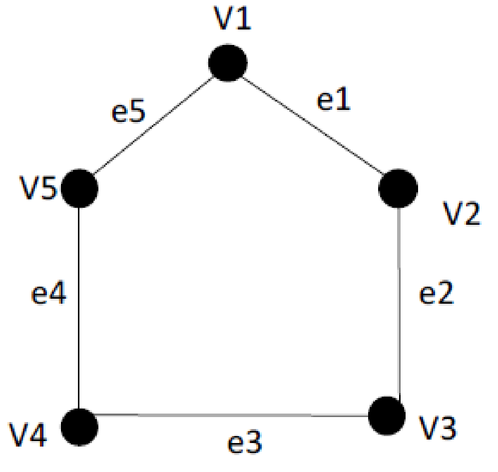
E2 → {V2,V3}

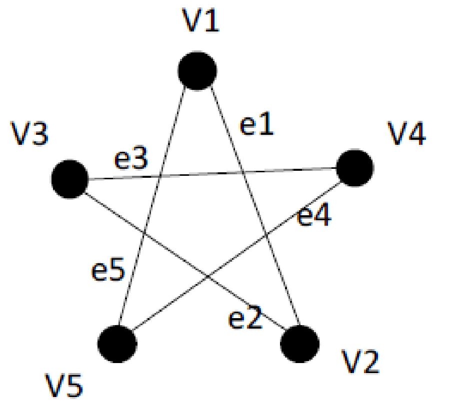
E3 → {V3,V4}

E4 → {V4,V5}

E5 → {V5,V1}

Planaridade, K2 é planar.





Digrafo:

Podemos querer que os arcos de um grafo comecem em um nó e terminem em outro, caso em que teríamos um *grafo direcionado* ou Digrafo [Gersting, 2004]. O dígrafo também é uma tripla ordenada (N, A, g), onde:

N = um conjunto não vazio de nós (vértices)

A = um conjunto de arcos (arestas), que correspondem a pares não ordenados.

|N| = numero de vértices e |A| = número de arestas

Porém:

g = uma função que associa a cada arco um par não ordenado x-y de nós, em que *x* é o **ponto inicial (extremidade inicial)** e *y* é o **ponto final (extremidade final)** de *a*.

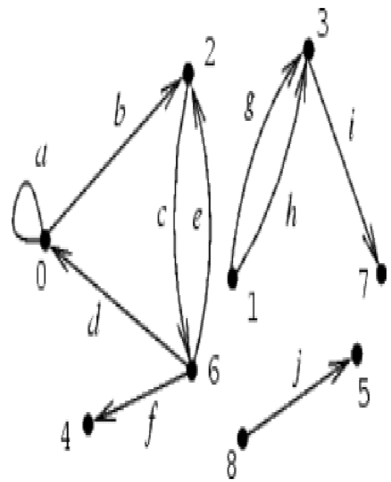
Nesse caso, a família de arestas é ordenada: a aresta (Vi,Vj) é diferente da aresta (Vj,Vi), por exemplo

Boa parte da nomenclatura e dos conceitos é análoga nos grafos e digrafos.

Digrafo:

EXEMPLO 1: Digamos que os vértices de nosso grafo são $0, 1, 2, 3, 4, 5, 6, 7, 8$ e os arcos são $a, b, c, d, e, f, g, h, i, j$. Então a seguinte tabela define um grafo:

ponta inicial	0	0	2	6	6	6	1	1	3	8
arco	a	b	c	d	e	f	g	h	i	j
ponta final	0	2	6	0	2	4	3	3	7	5



Se um arco a tem ponta inicial v e ponta final w, dizemos que a vai de v a w. Dizemos também que a sai de v e entra em w. Às vezes, um arco com ponta inicial v e ponta final w será denotado por (v,w) ou por v—w ou ainda por vw.

https://www.ime.usp.br/~pf/algoritmos_em_grafos/aulas/grafos.html

Digrafo versus grafo:

Iremos manter a nomenclatura “grafos” para fazer referência a grafos não direcionados/ dirigidos (digrafos).

Para cada grafo dirigido, existe um grafo simples (não dirigido) que é obtido removendo as direções das arestas, e os laços. Abaixo: no lado esquerdo um digrafo e no lado direito seu grafo simples: