Teoria de Grafos - Material Suplementar Exercícios

CCET-UCS

28 de fevereiro de 2023

Exercícios da Área 1

1) Qual dos grafos a seguir não é isomorfo dos outros? Por que?

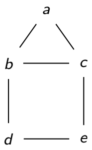


Figura: Grafo A

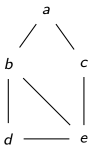


Figura: Grafo B

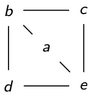


Figura: Grafo C

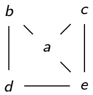


Figura: Grafo D

2) Desenhe 10 grafos não isomorfos de 4 vértices (existem 11 :-)).

Voltar para Isomorfismo

3)Os grafos G e H descritos a seguir são isomorfos? Justifique sua resposta.

$V_G = \{a, b, c, d, e, f, g\}$

$E_G = \{ab, bc, cd, cf, fe, gf, ga, gb\}$

$V_H = \{h, i, j, k, l, m, n\}$

$E_H = \{hn, nj, jk, lk, lm, li, ij, in\}$ Voltar para Isomorfismo

4) Quantas arestas possui K_{20}?
5) É possível haver um grupo de 7 pessoas no qual cada uma conhece exatamente 3 outras pessoas? Justifique sua resposta utilizando teoria de grafos. ???
6) Qual é o maior número possível de vértices em um grafo com 19 arestas e todos os vértices tem pelo menos grau 3? Explique sua resposta.
7) É possível que um grafo bipartido com um número ímpar de vertices seja hamiltoniano (isso é, contenha um ciclo hamiltoniano)? E euleriano? Justifique sua resposta.

8)Suponha um conjunto de n números inteiros positivos. Diga uma condição necessária para que esses números possam representar os graus dos n vértices de uma árvore. Essa condição é suficiente?

9) Descreva um possível algoritmo para verificar se um dado grafo é bipartido ou não.

10) Uma *caterpillar* é um grafo conexo, acíclico e que possui um caminho ao qual todo vértice do grafo pertence ou é adjacente a um vértice deste caminho. Este caminho é chamado a *espinha* da *caterpillar* e pode não ser único. Descreva um algoritmo para identificar se um grafo é uma *caterpillar* ou não.

11) Qual o número máximo de arestas de um grafo bipartido com 30 vértices?

12) Quais dentre os grafos a seguir são bipartidos?

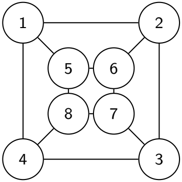


Figura: Grafo A

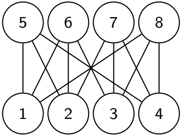


Figura: Grafo B

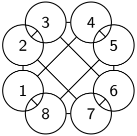
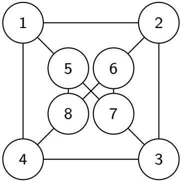
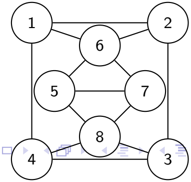
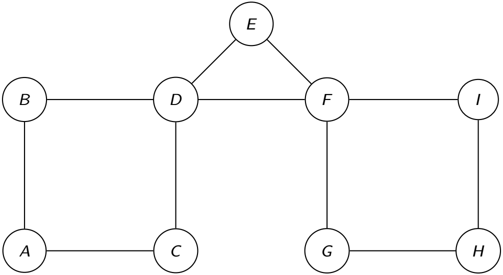


Figura: Grafo C



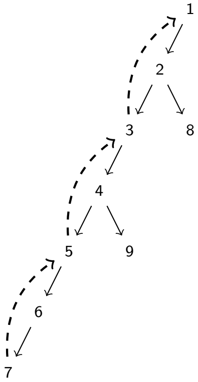


13) Decomponha o grafo a seguir em seus componentes biconexos, a partir do vértice e, utilizando o algoritmo dado. Mostre todos os passos do algoritmo (identificação das articulações, mostrando as arestas de retorno, identificação dos demarcadores e dos componentes a cada passo).

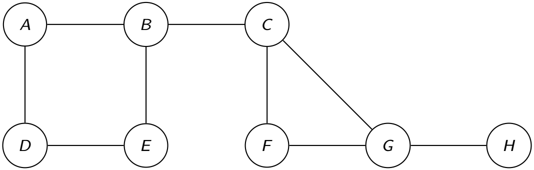


Voltar para Componentes Biconexos

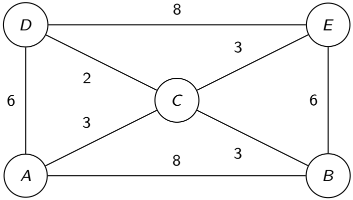
14) A figura a seguir representa a árvore resultante da busca em profundidade e as arestas de retorno de um grafo G. Identifique o lowpt de cada vértice, as articulações, demarcadores e componentes biconexas de G.



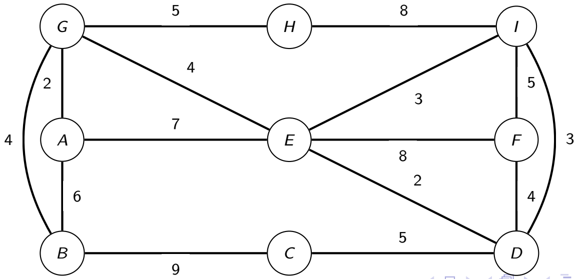
15) Decomponha o grafo a seguir em seus componentes biconexos, a partir do vértice e, utilizando o algoritmo visto em aula. Mostre todos os passos do algoritmo (árvore em profundidade mostrando as arestas de retorno, identificação das articulações, identificação dos demarcadores e dos componentes a cada passo).



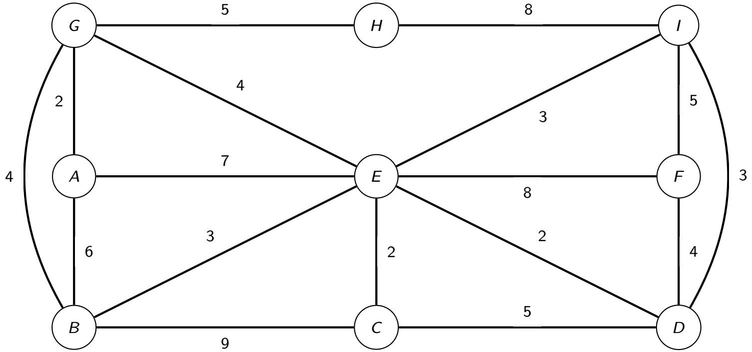
16) O problema do carteiro chinês consiste em, dado um grafo valorado (não necessariamente euleriano), encontrar o ciclo de menor custo total que percorra todas as arestas ao menos uma vez e volte para a aresta inicial. Descreva, em termos de teoria de grafos, como o problema pode ser resolvido. Mostre a solução para o grafo a seguir:



17) O grafo a seguir representa as esquinas e ruas de um bairro onde uma firma deve entregar iogurte a domicílio para um grande número de clientes. Sabendo que o depósito está situado no ponto (a) e que o grafo está valorado pela distância entre duas esquinas (em dezenas de metros), determinar o percurso mais curto que pode ser seguido pelo entregador, de modo que todas as entregas sejam feitas e que ele volte ao depósito.



18) O grafo a seguir representa as esquinas e ruas de um bairro onde uma firma deve entregar iogurte a domicílio para um grande número de clientes. Sabendo que o depósito está situado no vértice A e que o grafo está valorado pela distância entre duas esquinas (em dezenas de metros), determinar o percurso mais curto que pode ser seguido pelo entregador, de modo que todas as entregas sejam feitas e que ele volte ao depósito (parece o problema anterior mas o grafo é diferente). Considere que há iogurtes a serem entregues em todas as quadras.


G
H
I
A
E
F
B
C
D
5
8
4
2
3
5
4
7
8
3
3
6
2
2
4
9
5

19) Seja um jogo de dominó que contém 10 peças, cujas configurações são as seguintes: (1,2),(1,3),(1,4),(1,6),(2,4),(2,5),(2,6),(3,4),(3,5),(3,6),(4,5),(5,6). É possível colocar as peças em uma fila circular (isso é, em que a primeira peça toca a última), de tal maneira que o número de uma peça sempre toca um número igual na outra peça? E se a fila não for circular? E se for removida a peça (1,2)?

20) Considere todos os movimentos possíveis para o cavalo em um tabuleiro de xadrez de 8x8. Existe um percurso que o cavalo possa seguir onde ele execute todos os movimentos possíveis, executando cada movimento exatamente uma vez? Justifique sua resposta utilizando teoria de grafos.

21) Tertuliano Gonçalves havia prometido casamento a Josefina das Graças; o evento deveria se realizar, segundo ele, assim que acabasse o contrato de trabalho que acabava de assinar com uma empresa encarregada de pavimentar a rede de estradas que ligava Santana do Caixa Prego - cidade onde morava Josefina - às cidades vizinhas. O trabalho iria começar em Santana e prosseguir em continuidade, estrada após estrada, sem passar mais de uma vez na mesma estrada, terminando, segundo explicou Tertuliano - na própria Santana. A rede poderia ser representada pela matriz abaixo; Santana é a cidade número 1. Você que leu esta história, acha que Tertuliano está sendo sincero com Josefina? E se o itinerário 1-5-9-10 estivesse a cargo de outra empresa, estaria ele sendo sincero? Se sim, encontre uma ordem possível para a pavimentação das estradas.

	1	2	3	4	5	6	7	8	9	19
1		1	1		1					
2	1		1	1	1					
3	1	1			1	1				
4		1			1		1			1
5	1	1	1	1		1	1	1	1	
6			1		1			1		1
7				1	1			1		1
8					1	1	1		1	1
9					1			1		1
10				1		1	1	1	1	

- 22) A linha de produção de uma fábrica de tambores de aço está equipada para a produção de tambores de 100 e de 200 litros, ambos podendo ter tampa fixa com bujão ou tampa inteiramente removível.
- A mudança de tipo de tambor a ser produzido exige modificações na linha; em algumas máquinas isso só é necessário quando muda o tamanho do tambor, enquanto em outras a mudança do tipo de tampa implica também em modificações.
- A linha é formada de prensas para tampas e para fundos, soldadora, fresadora e recravadeira. As 4 opções de produção estão abreviadas abaixo da seguinte forma:
  - 1.tampa fixa, 100 litros
  - 2. tampa fixa, 200 litros
  - 3.tampa removível, 100 litros
  - 4. tampa removível, 200 litros

- Os valores, em homens-hora, do trabalho exigido para a passagem de uma máquina da opção i para a opção j são os seguintes, para a prensa de tampas e para a recravadeira:

	1	**2**	**3**	**4**
1	-	5	2	6
2	4.5	-	6.5	2.5
3	1.5	5.5	-	5.5
4	5	1.5	6	-

Tabela: Prensa de Tampa

	1	**2**	**3**	**4**
1	-	5	2	6
2	4.5	-	6.5	2.5
3	1.5	5.5	-	5.5
4	5	1.5	6	-

Tabela: Recravadeira

- Em relação às outras máquinas temos:

	Prensa fundos	Soldadora	Fresadora
100 para 200	4	6	0.5
200 para 100	3.5	6.5	1

- A fábrica dispõe de encomendas dos 4 tipos de tambores e deseja colocar a linha em funcionamento. Se nenhuma encomenda é mais urgente que as restantes, determine a seqüência de produção dos 4 tipos, mais econômica em relação ao custo do trabalho exigido pelas modificações:

  a)Se a situação das encomendas é repetitiva;

  b)Se não se pode considerá-la como repetitiva, isto é, se a fabricação dos 4 tipos ocorrerá apenas uma vez.

- 23) Uma fábrica faz oito variedades de palhetas de guitarras usando apenas uma máquina, a Presto Pick Puncher. Esta máquina precisa de diferentes ajustes a fim de fazer diferentes estilos de palhetas. A companhia deseja gastar 1 hora de cada dia de trabalho em cada estilo de palheta. No fim de cada hora os ajustes na máquina precisam ser convertidos para aqueles necessários para o próximo estilo de palheta. No fim do dia, a máquina deve ser convertida de volta para o estilo produzido no início do dia. O tempo necessário para conversão depende de cada estilo para outro, os quais são fornecidos a seguir. Desde que os tempos de conversões não são o mesmo, faz diferença em qual ordem os estilos são feitos. Você pode arranjar uma ordem, começando e finalizando com o estilo A, tal que nenhuma conversão leve mais do que 3 min? Se não, você pode fazê-la tal que nenhuma tarefa demore mais do que 4 min?

	A	B	C	D	E	F	G	H
A		5	2	6	6	5	2	5
B	5		3	4	5	6	1	6
C	2	3		1	6	5	5	6
D	6	4	1		2	4	5	5
E	6	5	6	2		3	6	2
F	5	6	5	4	3		3	6
G	2	1	5	5	6	3		1
H	5	6	6	5	2	6	1	

Tabela: Custo de conversão de máquinas para fabricação de cada tipo de palheta

24) A turma de Teoria de Grafos planeja secretamente uma insurreição. Os sete líderes chaves estão listados na tabela abaixo. É decidido que durante o teste, os detalhes escritos serão passados entre os sete começando com Abe. O processo é complicado por dois fatores. Primeiro, dilemas ideológicos entre alguns deles impedem que eles troquem mensagens entre si. Segundo, no interesse de evitar a interceptação, é necessário passar o pedaço de papel, começando e finalizando com Abe, entre teodos os sete sem ninguém receber duas vezes (exceto Abe). A tabela abaixo indica quais pares de estudantes têm relações amigáveis (A) e podem fazer contato para passar uma mensagem e quais pares não são amigos (NA) e não podem ter a mensagem passada diretamente entre eles. Pode a mensagem ser passada da forma definida? Em caso positivo, em que ordem?

	Abe	Tom	Dave	John	Lee	Gerry	Ren
Abe	-	A	NA	NA	A	NA	NA
Tom	A	-	A	A	NA	NA	A
Dave	NA	A	-	NA	A	A	A
John	NA	A	NA	-	A	A	NA
Lee	A	NA	A	A	-	A	A
Gerry	NA	NA	A	A	A	-	NA
Ren	NA	A	A	NA	A	NA	-

25) O professor descobriu a rebelião descrita no exercício anterior e deseja punir os sete líderes com um problema de teoria de grafos extra para fazer em casa. Seu plano é complicado pelo fato que estudantes amigos irão colaborar (assumindo que os inimigos não vão) se eles receberem o mesmo problema. Portanto, ele gostaria de ter certeza que estudante amigos receberão problemas diferentes. Ele gostaria de minimizar o número de problemas que ele atribuirá. Qual é o número mínimo?

26) Em um zoológico onde os animais vivem soltos existem 10 espécies, as quais estão listadas na tabela abaixo. As coisas ficam complicadas pelo fato que não é aconselhável permitir espécies que são inimigas naturais viver em um mesmo cercado. A tabela indica tais incompatibilidades pela letra X. Por razões de economia é desejado ter poucos cercados possíveis. Qual é o número mínimo de cercados? ????

	1	2	3	4	5	6	7	8	9	10
1			X							
2				X						
3	X			X						
4		X	X							
5						X				X
6					X		X			
7						X		X		X
8							X		X	
9								X		X
10					X		X		X	

27) Um chefe de escoteiros está planejando levar oito membros de sua tropa em carros para um congresso de escoteiros. Para evitar problemas, somente os escoteiros que são amigos (A) vão no mesmo carro. Qual é o menor número de carros que são necessários para transportar os escoteiros? Se os carros pudessem carregar somente dois escoteiros, haveria uma resposta diferente?

????

	1	2	3	4	5	6	7	8
1	-	A	NA	NA	NA	A	NA	A
2	A	-	A	A	NA	NA	NA	A
3	NA	A	-	NA	NA	A	NA	NA
4	NA	A	NA	-	A	A	NA	NA
5	NA	NA	NA	A		A	A	NA
6	A	NA	A	A	A	-	A	A
7	NA	NA	NA	NA	A	A	-	NA
8	A	A	NA	NA	NA	A	NA	-

28)O Departamento de Informática pretende oferecer 7 disciplinas para o nono semestre do Curso de Computação no próximo semestre.

(C) Compiladores
(G) Grafos
(L) Lógica
(N) Análise de Algoritmos

(P) Probabilidade
(S) Engenharia de Software
(T) Arquitetura de Computadores

Com base na pré-matrícula, verificou-se que cada estudante planeja cursar as disciplinas indicadas abaixo:

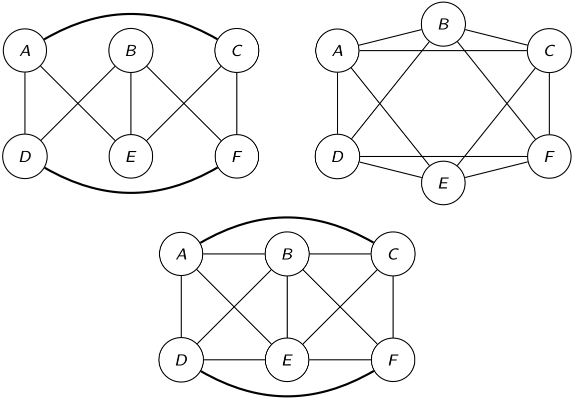
- Aldo C,L,T
- Batista C,G,S
- Carlos C,L
- Daniel G,N
- Eduardo L,N
- Frederico C,G
- Giovana N,P
- Homero G,L
- Iracema C,T
- Janete C,S,T
- Kevin P,S
- Luis P,T

Considerando que cada disciplina é dada num período, como se poderia determinar qual o menor número de períodos necessários para que os sete cursos possam ser oferecidos? Para os dados fornecidos, qual é esse número?

29)O grafo K_7, após a exclusão das arestas $(1,2)$,$(1,3)$,$(1,4)$,$(2,3)$ e $(2,4)$ é planar?

30) O grafo resultante da remoção das arestas (3,1), (3,2), (6,4), (6,5) e (5,4) de K_6 é planar? Justifique sua resposta.

31)Verificar se os grafos a seguir são planares. Se forem, mostrar uma representação planar. Se não forem, mostrar um homeomorfismo de $K_{3,3}$ ou K_5.



a. É planar. Basta inverter os vértices B e E.
b. É planar. Basta "puxar para fora" as arestas AC, CE e EA.
c. Não é planar. {A,B,F} e {D,E,C} formam $K_{3,3}$. Além disso, ao remover as arestas BF e EF resulta um homeomorfismo de K_5.

32) Idem (verifique a planaridade)

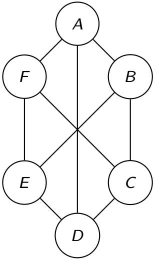


Figura: Grafo A

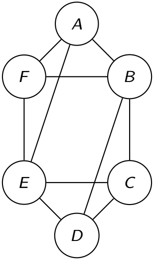


Figura: Grafo B

a. É o próprio $K_{3,3}$.
b. Planar. É so puxar para fora as arestas horizontais.

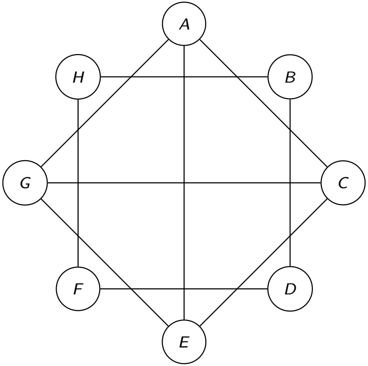


Figura: Grafo C

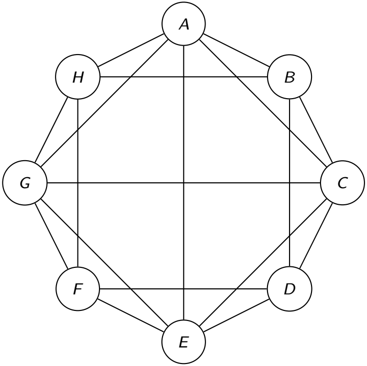


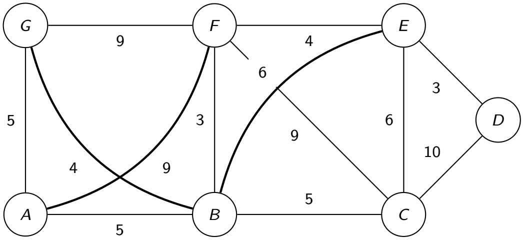
Figura: Grafo D

c. Planar. Há dois subgrafos disjuntos.
d. Não planar. Removendo as arestas AG, EF, FG, EG, EC resulta K_5. E removendo as arestas EF e FG resulta $K_{3,3}$ ({G,A,D} e {H,C,E}).

33) O grafo a seguir corresponde ao projeto de uma rede de computadores. Os vértices correspondem às máquinas e as arestas correspondem à possibilidade de conexão entre duas máquinas, juntamente com o correspondente custo de instalação. Essa rede deverá ser construída de forma que cada computador possa se comunicar (direta ou indiretamente) com cada um dos outros computadores, sempre a um custo mínimo, não importando o custo total da rede. Como a rede é bastante segura, não há a necessidade de haver mais de uma rota de comunicação entre dois computadores pois sempre será utilizada a de menor custo. Sendo assim, algumas conexões nunca serão utilizadas, de forma que não precisam ser implantadas.

a)O problema descrito pode ser resolvido através da geração da árvore geradora mínima?

b)Em caso contrário, como o problema pode ser resolvido?



34)Considere uma árvore composta de q vértices de grau 4 e p vértices de grau 1. Qual a relação matemática entre p e q? (essa família de grafos representa hidrocarbonetos em que o vértice de grau 4 representa átomos de carbono e os vértices de grau 1 representam átomos de hidrogênio)

35)É possível existir um grafo com um número ímpar de vértices e um número par de arestas que contenha um ciclo euleriano? Justifique sua resposta.

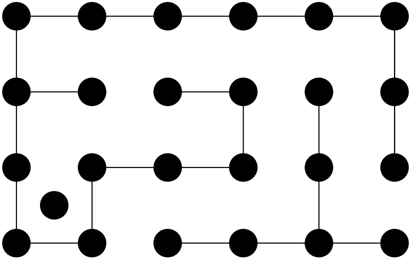
36)Descreva duas formas possíveis de obter um ciclo qualquer a partir de um grafo sabidamente cíclico.

37) Considere o grafo a seguir $G(V, A)$:
$V = \{A, B, C, D, E, F\}$
$A =$
$\{(A, D),(A, E),(B, D),(B, C),(C, D),(C, E),(D, E),(F, D),(F, E)\}$
a) Descreva os percursos em profundidade (DFS) e em amplitude (BFS) para o grafo G, iniciando pelo vértice A e seguindo sempre pelo vértice de menor ordem alfabética quando houver mais de um vértice adjacente.
b) Considere o grafo G como um dígrafo (onde o arco sempre parte do vértice com letra de menor ordem alfabética para o maior). Defina a árvore geradora obtida a partir de um percurso em profundidade, iniciando pelo vértice A e seguindo sempre pelo vértice de menor ordem alfabética quando houver mais de um vértice adjacente.

38)Mostre uma árvore geradora mínima do grafo representado pela matriz de adjacências a seguir:

	A	**B**	**C**	**D**	**E**	**F**
A		3	2		5	4
B	3		3	4		4
C	2	3		5	2	4
D		4	5		4	6
E	5		2	4		2
F	4	4	4	6	2	

39)Considere um labirinto perfeito qualquer, como o mostrado abaixo, onde há uma saída do labirinto a partir de qualquer uma das casas. Deseja-se descobrir, dentro do labirinto, qual das posições dentro do labirinto é a mais afastada da saída mais próxima. Por exemplo, no labirinto mostrado, a casa mais afastada das duas saídas é a casa marcada com um círculo. Diga como esse problema pode ser resolvido com o uso da teoria de grafos, ou seja, como modelar o problema com um grafo e que algoritmo utilizar para chegar à solução do problema.



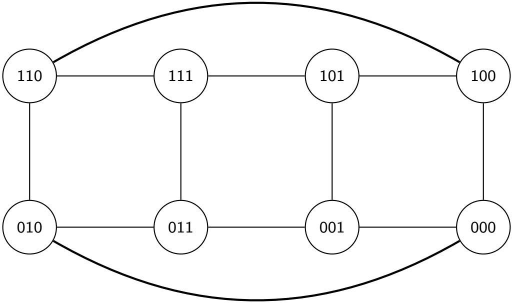
40) (busca em espaço de estados) Problema dos canibais e dos missionários - Três canibais e três missionários estão viajando juntos e chegam a um rio. Eles desejam atravessar o rio, sendo que o único meio de transporte disponível é um barco que comporta no máximo duas pessoas. Há uma outra dificuldade: em nenhum momento o número de canibais pode ser superior ao número de missionários, pois dessa forma os missionários estariam em grande perigo de vida. Como efetuar a travessia do rio?

Situação inicial: cccmmm -
- Vão dois canibais (cmmm - cc), volta um canibal (ccmmm - c)
- Vão dois canibais (mmm - ccc), volta um canibal (cmmm - cc)
- Vão dois missionários (cm - mmcc), volta um canibal e um missionário (ccmm - cm)
- Vão dois missionários (cc - cmmm), volta o canibal (ccc - mmm)
- Vão dois canibais (c - ccmmm), volta um canibal (cc - cmmm)
- Vão dois canibais (- cccmmm)

41) Problema dos três maridos ciumentos - Três esposas e seus respectivos marido desejam ir ao centro da cidade em uma moto, a qual comporta apenas duas pessoas. Como eles poderiam deslocar-se até o centro considerando que nenhuma esposa deveria estar com um ou ambos os outros maridos, a menos que seu marido também esteja presente?

42) Problema dos potes de vinho - Considere que temos 3 potes com capacidades de 8, 5 e 3 litros, respectivamente, os quais não possuem qualquer marcação. O maior deles está completamente cheio enquanto que os outros dois estão vazios. Estamos interessados em dividir o vinho em duas porções iguais de 4 litros, tarefa esta que pode ser resolvida por transvasos sucessivos de um vaso no outro. Qual o menor número de transvasos necessários para completar a divisão?

43) Considere a sequência de números binários de 0 a 7. É possível colocar os 8 números em uma sequência de modo que, de um número para o outro, mude apenas um bit? O grafo da possibilidades de transição é planar?


110
111
101
100
010
011
001
000

44) Um grafo conexo é dito euleriano se possui um ciclo euleriano. Desenho 3 grafos eulerianos não isomorfos com exatamente seis vértices (há 8)

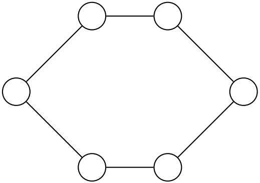


Figura: 222222

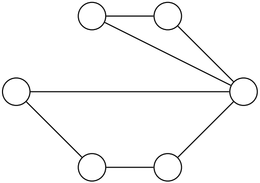


Figura: 222224

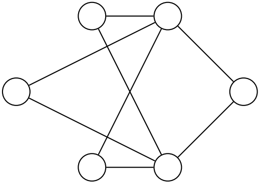


Figura: 222244

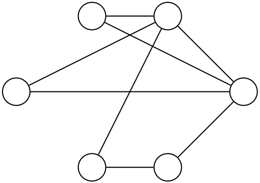


Figura: 222244

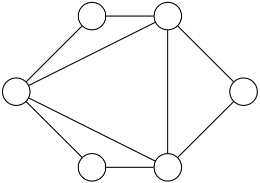


Figura: 222444

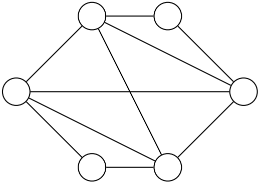


Figura: 224444

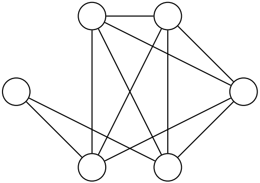


Figura: 244444

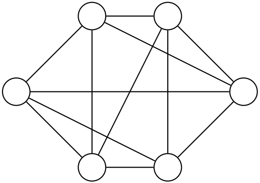
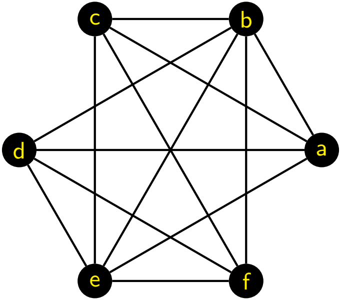


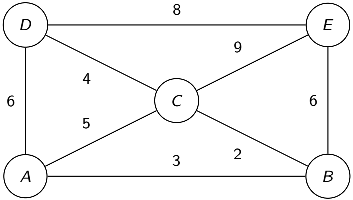
Figura: 444444

45) No grafo do exercício anterior, aplique o algoritmo de Tremaux, a partir do vértice A, mostrando como ficam ao final as marcações das passagens. Use como critério de escolha entre dois vértices, sempre o de menor ordem alfabética.

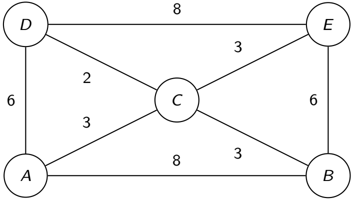
46) Aplique o teste de planaridade de Democron ao grafo a seguir para determinar se é planar ou não. Mostre o ciclo inicial e suas pontes. Mostre o desenho do grafo à medida que vão sendo acrescentados os caminhos a ele e indique as faces em que cada ponte restante pode ser inserida (para cada iteração do algoritmo).



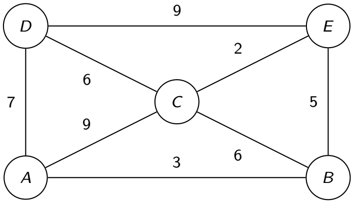
47) Aplique o algoritmo de Floyd-Warshall no grafo a seguir, mostrando a matriz de distâncias a cada iteração.



48) Aplique o algoritmo de Dijkstra ao grafo a seguir, para calcular a distância entre os vértices A e E. Mostre o vetor de distâncias ao final de cada iteração.



49) Aplique o algoritmo de Dijkstra ao grafo a seguir, para calcular a distância entre os vértices A e E. Mostre o vetor de distâncias ao final de cada iteração.



50) Para qualquer inteiro positivo k, um cubo de dimensão k (ou k-cubo) é o grafo definido da seguinte maneira: os vértices do grafo são todas as seqüências $b_1 b_2 ... b_k$ de bits; dois vértices são adjacentes se e somente se diferem em exatamente uma posição. Por exemplo, os vértices do cubo de dimensão 3 são 000, 001, 010, 011, 100, 101, 110, 111; o vértice 000 é adjacente aos vértices 001, 010, 100 e a nenhum outro; e assim por diante.

Faça uma figura do 1-cubo, do 2-cubo e do 3-cubo.

Mostre a matriz de adjacências de um 3-cubo.

Quantos vértices tem o k-cubo?

Quantas arestas tem o k-cubo?

O 3-cubo é planar?

51) Descreva sucintamente qual a função de cada um dos 8 algoritmos a seguir, deixando bem claras as diferenças entre eles, caso haja algoritmos com funções semelhantes:

Floyd-Warshall	Tremaux
Prim	Moore
Kruskal	Dijkstra
Demoucron	Ford

52) Indique os grafos bipartidos completos que são:

1. eulerianos;
2. semi-eulerianos.

53) Um grafo conexo é dito *euleriano* se possui um ciclo euleriano. Desenhe 3 grafos eulerianos não isomorfos com exatamente seis vértices (há 8).

54) É possível que um grafo bipartido com um número ímpar de vértices seja hamiltoniano (isso é, contenha um ciclo hamiltoniano)? E euleriano? Justifique sua resposta.

55) Determine todos os grafos eulerianos (não isomorfos) com menos de seis vértices (há exatamente 7 grafos).

56) Sejam G_1 e G_2 dois grafos conexos, sem vértices em comum. Seja x um vértice de G_1 e y um vértice de G_2. Seja G o grafo que se obtém de $G_1 \cup G_2$ acrescentando a aresta (x, y). É possível que o grafo G seja euleriano? E semi-euleriano (contem um caminho euleriano mas não contem um ciclo euleriano)? Em que condições?

57) Determine os valores de n para os quais:

1. K_n é euleriano;
2. K_n é semi-euleriano.

58)Seja G um grafo com 14 vértices e 25 arestas. Se todo vértice de G tem grau 3 ou 5, quantos vértices de grau 3 o grafo G possui?

59)Um grafo é dito r-regular se todos os vértices tem grau r. Dê um exemplo de um grafo 3-regular que não seja completo.

60)Uma floresta é um grafo desconexo em que todas as componentes conexas são árvores. Determine o número de arestas de uma floresta com n vértices e k componentes conexos, em função de n e k.

61) Quantos subgrafos induzidos com 2 vértices possui um grafo de n vértices e m arestas?

62) Desenhe um grafo que contenha 4 pontos centrais.

Exercícios da Segunda Área

1) Encontre um fluxo máximo para a rede abaixo. O número na aresta representa sua capacidade.

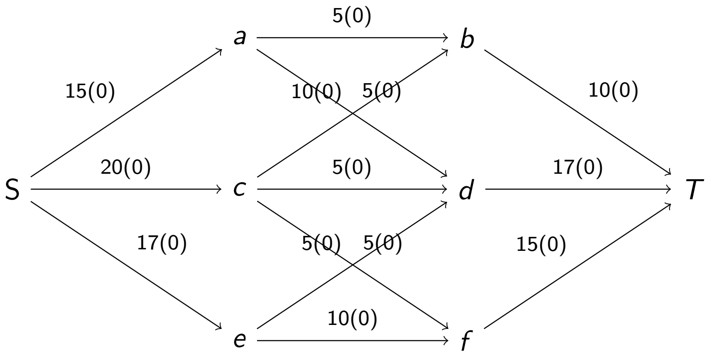


Figura: Encontre o fluxo máximo

2) Na rede a seguir, x_1, x_2 e x_3 são fontes do mesmo produto. A quantidade disponível em x_1 é 5, em x_2 é 10 e em x_3 é 5. Os vértices y_1, y_2 e y_3 são consumidores. A quantidade de produto necessária em y_1 é 5, em y_2 é 10 e em y_3 é 5. Descubra se é possível atender a estas condições (Dica: Uma forma de resolver este tipo de problema é introduzir uma fonte auxiliar s e um sumidouro t; conectar s a x_i por uma aresta de capacidade igual à quantidade disponível em x_i; conectar cada y_i a t através de uma aresta de capacidade igual à demanda de y_i; encontrar um fluxo máximo na rede resultante e observar se todas as demandas são atendidas. Produtor x Consumidor

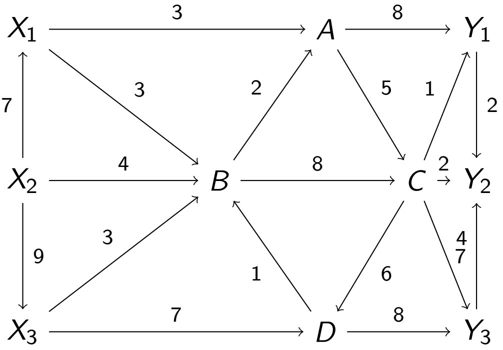
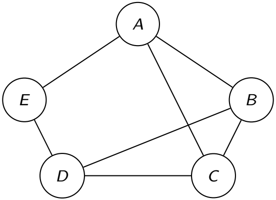


Figura: Problema produtor x consumidor

3) Dados os bom resultados obtidos no final do ano passado, a empresa de turismo Vemque Talimpo resolveu distribuir um lote de passagens (presente das companhias aéreas) entre seus funcionários. Com as passagens destinam-se a lugares específicos, o diretor da empresa viu-se diante de um dilema, pois não gostaria de presentear um funcionário com uma passagem para um lugar que não fosse de seu agrado. Sendo assim, ele resolveu fazer uma consulta aos funcionários solicitando que eles indicassem locais de sua preferência, dentre aqueles para os quais havia passagens. Como ele poderia, agora, obter uma distribuição de passagens que agradasse ao maior número possível de funcionários todos os funcionários? Para ilustrar sua solução, crie um exemplo com 5 passagens e 5 funcionários, cada um escolhendo 2 lugares. prox

Associação Máxima

4) Mostre como seria utilizado o método de alteração estrutural para obtenção do número cromático no grafo a seguir, mostrando todos os grafos gerados em cada passo da aplicação do método:



5) Considere o grafo K_6 de vértices a,b,c,d,e,f. Mostre os grafos gerados na utilização do método da alteração estrutural para obtenção do número cromático se forem retiradas as arestas (b,c) e (a,d). Qual o número cromático obtido? E se, a partir do grafo original, forem retiradas as arestas (b,c) e (b,d)?

9) Considere que você é o Chefe do Departamento de Informática e você deve fazer a alocação das disciplinas aos professores para o próximo semestre. Você dispõe, para cada professor, da lista de disciplinas que ele é capaz de ministrar e do número máximo de disciplinas que o professor pode ministrar. Além disso você dispõe da lista de disciplinas que deverá ser oferecida no próximo semestre, e o número de turmas que deve ser oferecido para cada disciplina. Descreva como você pode resolver o problema, ou seja, chegar a uma atribuição de disciplinas aos professores que respeite suas competências e sua carga horária máxima de forma todas as turmas de todas as disciplinas sejam atendidas. Se for necessário, invente uma situação hipotética para ilustrar sua resposta.

Associação Máxima

10) Suponha que existem três geradores de energia a serem construídos em três das sete cidades mais populosas de certo país. As distâncias entre cada par de cidades é dada na tabela abaixo. O objetivo é construir os geradores tal que cada cidade está próxima dentro de 50 milhas da cada um dos geradores. Como este problema pode ser resolvido? Qual é a solução?

	B	C	D	E	F	G
A	80	110	15	60	100	80
B		40	45	55	70	90
C			65	20	50	80
D				35	55	80
E					25	60
F						70

11) Em uma nação emergente, cada uma das seis principais cidades receberá uma estação de televisão. Diferentes estações podem usar a mesma faixa de frequência, contanto que estejam distanciadas de mais de 80 quilômetros, para evitar problemas de interferência. Suponha que as cidades tenham os seguintes nomes: A, B, C, D, E e F, e as distâncias entre elas estão descritas na tabela a seguir:

	B	C	D	E	F
A	150	135	96	76	84
B		96	147	90	102
C			72	76	51
D				69	48
E					24

Qual o número mínimo de faixas de frequência necessárias, e como pode ser feita a distribuição entre as cidades? Como isso pode ser resolvido através de Teoria de Grafos?

PERT1) Construa um diagrama PERT da tabela de tarefas a seguir. Calcule também o tempo mínimo para completá-lo, os nós do caminho crítico e, para cada evento, a data mais cedo, a data mais tarde e a folga.

Tarefa	Predecessoras	Duração
a	Nenhum	10
b	Nenhum	5
c	a	10
d	a,b	5
e	c,d	15
f	d	10
g	e,f	12
h	d	10
i	g,h	7

PERT2) Construa um diagrama PERT da tabela de tarefas a seguir. Calcule também o tempo mínimo para completá-lo, os nós do caminho crítico e, para cada evento, a data mais cedo, a data mais tarde e a folga.

Tarefa	Predecessoras	Duração
a	e	3
b	c,d	5
c	a	2
d	a	6
e	nenhum	2
f	a,g	4
g	e	4
h	b,f	1

PERT3) Construa um diagrama PERT da tabela de tarefas a seguir. Calcule também o tempo mínimo para completá-lo, os nós do caminho crítico e, para cada evento, a data mais cedo, a data mais tarde e a folga.

Tarefa	Descrição da Tarefa	Pred.	Tempo (dias)
b	escavar e concretar sapatas	-	4
c	concretar a fundação	b	2
d	levantar estrutura de madeira, inclusive a base do telhado	c	4
e	assentar a alvenaria	d	6
f	instalar esgoto e tubulação do subsolo	c	1
g	concretar o piso do subsolo	f	2
h	instalar a canalisação principal	f	3
i	instalar a rede elétrica geral	d	2
j	instalar aquecimento e ventilação	d,g	4
k	pregar as tábuas de revestimento e emboçar	i,j,h	10
l	assentar piso	k	3
m	instalar equipamento da cozinha	l	1
n	instalar a tubulação adicional	l	2
o	terminar carpintaria	l	3
p	terminar o telhado	e	2
q	instalar calhas e canais	p	1
r	colocar calhas para água da chuva	c	1
s	raspar e encerar o piso	o,t	2
t	pintar	m,n	3
u	terminar a instalação elétrica	t	1
v	terminar nivelamento do terreno	q,r	2
w	concretar as calçadas e completar o ajardinamento	v	5

PERT4) Construa um diagrama PERT da tabela de tarefas a seguir. Calcule também o tempo mínimo para completá-lo, as ATIVIDADES do caminho crítico e, para cada evento, a data mais cedo, a data mais tarde e a folga.

Tarefa	Predecessoras	Duração
a	Nenhum	10
b	Nenhum	5
c	a	5
d	a,b	10
e	b	5
f	c,d	10
g	d	7
h	d,e	16

PERT5) Construa um diagrama PERT da tabela de tarefas na página a seguir. Calcule também o tempo mínimo para completá-lo, os nós do caminho crítico e, para cada evento, a data mais cedo, a data mais tarde e a folga.

Tarefa	Predecessoras	Duração
a	Nenhum	10
b	Nenhum	20
c	Nenhum	30
d	Nenhum	40
e	a,b	15
f	b,c	10
g	c,d	12

PERT6) Suponhamos que se deseja levar a cabo um projeto de construção de locais comerciais para alugar, e para isto foram identificadas as seguintes atividades.

Descrição	Predecessoras	Dur.(semanas)
1 - Preparar desenho arquitetônico	Nenhuma	5
2 - Identificar novos arrendatários potenciais	Nenhuma	6
3 - Desenvolver prospectos de contrato para os arrendatários	1	4
4 - Selecionar construtor	1	3
5 - Preparar as licenças de construção	1	1
6 - Obter a aprovação das licenças de construção	5	4
7 - Levar a cabo a construção	4,6	14
8 - Formalizar os contratos com os arrendatários	2,3	12
9 - Entrada do arrendatários	7,8	2

Desenhe o diagrama PERT, e calcule a data mais cedo e a data mais tarde para cada evento e identifique as atividades que fazem parte do caminho crítico.

PERT7) Suponha que uma empreiteira ganhou uma concorrência para construir uma planta industrial e para isto foram identificadas as seguintes atividades.

Atividade	Descrição	Predecessoras	Dur.(semanas)
A	Escavação	Nenhuma	2
B	Fundação	A	4
C	Paredes	B	10
D	Telhado	C	6
E	Encanamento Exterior	C	4
F	Encanamento Interior	E	5
G	Muros	D	7
H	Pintura Exterior	E,G	9
I	Instalação Elétrica	C	7
J	Divisórias	F,I	8
K	Piso	J	4
L	Pintura Interior	J	5
M	Acabamento Exterior	H	2
N	Acabamento Interior	K,L	6

Desenhe o diagrama PERT, e calcule a data mais cedo e a data mais tarde para cada evento e identifique as atividades que fazem parte do caminho crítico.

3) Mostre todos os passos da utilização do algoritmo de programação dinâmica para encontrar o ciclo hamiltoniano de custo mínimo (problema do caixeiro viajante) para o grafo a seguir:

	1	2	3	4
1	0	3	∞	5
2	6	0	4	7
3	4	∞	0	2
4	2	8	3	0

15) Mostre todos os passos da utilização do algoritmo de programação dinâmica para encontrar o ciclo hamiltoniano de custo mínimo (problema do caixeiro viajante) para o grafo a seguir:

	1	2	3	4
1	0	6	2	5
2	6	0	4	∞
3	2	4	0	1
4	5	2	1	0

16)**Problema das 5 damas** - A dama é a peça mais poderosa do xadrez. Numa jogada ela pode mover-se tantas casas quantas quiser em qualquer direção (horizontal, vertical ou diagonal), desde que não haja nenhuma outra peça que obstrua sua passagem. O desenho que se segue mostra uma posição particular da dama na qual ela tem 27 possibilidades de movimento. Essas 27 casas (além daquela onde a dama está) estão sob o domínio da dama. Qualquer outra peça que estivesse numa destas casas estaria sob ataque da dama em questão. O problema consiste em procurar as posições a serem ocupadas por damas em um tabuleiro de xadrez de modo a que se possa controlar todas as casas do tabuleiro com o menor número de damas. Como esse problema pode ser modelado e resolvido com teoria de grafos?

			o				o
o			o			o	
	o		o		o		
		o	o	o			
o	o	o	D	o	o	o	o
		o	o	o			
	o		o		o		
o			o			o	

17) **Problema da Distribuição de Pontos de Observação** - Em um parque ecológico, deseja-se colocar pontos de observação de modo que se tenha uma visão de todo o parque. Para isso o parque foi dividido em várias áreas, sendo que para cada área foram identificados os pontos que mais propiciam sua observação. Conhecida esta relação de vigilância, observou-se que em vários casos um mesmo ponto de observação permite a vigilância de mais de uma área. Sendo assim, a direção do parque optou por procurar instalar o menor número de pontos de observação de forma a interferir o mínimo possível com a vida no parque. Como se poderia identificar esse conjunto mínimo de pontos de observação?

Questões do POSCOMP

(39 - Fundamentos - 2002) - O menor número possível de arestas em um grafo conexo com n vértices é:

1. 1
2. $n/2$
3. $n-1$
4. n
5. n^2

Voltar para o índice

(39 - Fundamentos - 2002) - O menor número possível de arestas em um grafo conexo com n vértices é:

1. 1
2. $n/2$
3. $n-1$
4. n
5. n^2

(40 - Fundamentos - 2002) - Considere um grafo G satisfazendo as seguintes propriedades:

- G é conexo
- Se removermos qualquer aresta de G, o grafo obtido é desconexo.

Então é correto afirmar que o grafo G é:

1. Um circuito
2. Não bipartido
3. Uma árvore
4. Hamiltoniano
5. Euleriano

(40 - Fundamentos - 2002) - Considere um grafo G satisfazendo as seguintes propriedades:

- G é conexo
- Se removermos qualquer aresta de G, o grafo obtido é desconexo.

Então é correto afirmar que o grafo G é:

1. Um circuito
2. Não bipartido
3. Uma árvore
4. Hamiltoniano
5. Euleriano

(33 - Fundamentos - 2004) - Considere as seguintes afirmações sobre um grafo G com $n > 0$ vértices:

I. Se G é conexo o número de arestas é maior que n;
II. G será acíclico somente se o número de arestas for menor que n;
III. Se G não tem triângulos então G é planar;
IV. G é Euleriano se, e somente se, todo grau é par.

As afirmativas verdadeiras são:

1. I e II
2. I e III
3. II e III
4. II e IV
5. II, III e IV

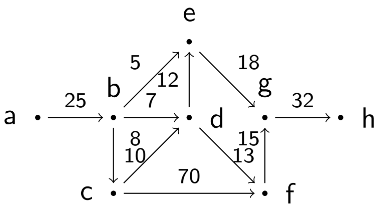
(33 - Fundamentos - 2004) - Considere as seguintes afirmações sobre um grafo G com $n > 0$ vértices:

I. Se G é conexo o número de arestas é maior que n;
II. G será acíclico somente se o número de arestas for menor que n;
III. Se G não tem triângulos então G é planar;
IV. G é Euleriano se, e somente se, todo grau é par.

As afirmativas verdadeiras são:

1. I e II
2. I e III
3. II e III
4. II e IV
5. II, III e IV

(39 - Fundamentos - 2005) - Os grafos $G = (V_G, E_G)$ e $H = (V_H, E_H)$ são isomorfos. Assinale a alternativa que justifica essa afirmação.



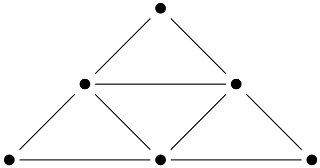


Figura: H

1. As seqüências dos graus dos vértices de G e H são iguais.
2. Os grafos têm o mesmo número de vértices e o mesmo número de arestas.
3. Existe uma bijeção de V_G em V_H que preserva as adjacências.
4. Cada vértice de G e de H pertence a exatamente quatro triângulos distintos.
5. Ambos os grafos admitem um circuito que passa por cada aresta exatamente uma vez.

(40 - Fundamentos - 2005) - Dadas as seguintes afirmações

I. Qualquer grafo conexo com n vértices deve ter pelo menos $n-1$ arestas.

II. O grafo bipartido completo $K_{m,n}$ é Euleriano desde que m e n sejam ímpares.

III. Em um grafo o número de vértices de grau ímpar é sempre par.

São verdadeiras:

1. Somente a afirmaçao (I).
2. Somente as afirmações (I) e (III).
3. Somente as afirmações (II) e (III).
4. Somente as afirmações (I) e (II).
5. Todas as afirmações.

(26 - Fundamentos - 2006) - A respeito da representação de um grafo de n vértices e m arestas é correto dizer que:

1. a representação sob a forma de matriz de adjacência exige espaço $\Omega(m^2)$.
2. a representação sob a forma de listas de adjacência permite verificar a existência de uma aresta ligando dois vértices dados em tempo $O(1)$.
3. a representação sob a forma de matriz de adjacência não permite verificar a existência de uma aresta ligando dois vértices dados em tempo $O(1)$.
4. a representação sob a forma de listas de adjacência exige espaço $\Omega(n + m)$.
5. todas as alternativas estão corretas.

(36 - Fundamentos - 2006) - Seja $G = (V, E)$ um grafo simples conexo não-euleriano. Queremos construir um grafo H que seja euleriano e que contenha G como subgrafo. Considere os seguintes possíveis processos de construção:

I. Acrescenta-se um novo vértice, ligando-o a cada vértice de G por uma aresta.
II. Acrescenta-se um novo vértice, ligando-o a cada vértice de grau ímpar de G por uma aresta.
III. Cria-se uma nova cópia G' do grafo G e acrescenta-se uma aresta ligando cada par de vértices correspondentes.
IV. Escolhe-se um vértice arbitrário de G e acrescentam-se arestas ligando este vértice a todo vértice de grau ímpar de G.
V. Duplicam-se todas as arestas de G.

Quais dos processos acima sempre constroem corretamente o grafo H?

1. Somente (II) e (IV)
2. Somente (II), (IV) e (V)
3. Somente (III), (V) e (VI)
4. Somente (II), (IV), (V) e (VI)
5. Somente (I), (III), (IV) e (V)

(31 - Fundamentos - 2007) - Considere o *problema do caixeiro viajante*, definido como se segue.

Sejam S um conjunto de n cidades, $n \geq 0$, e $d_{ij} > 0$ a distância entre as cidades i e j, $i, j \in S$, $i \neq j$. Define-se um *percurso fechado* como sendo um percurso que parte de uma cidade $i \in S$, passa exatamente uma vez por cada cidade de $S - \{i\}$, e retorna à cidade de origem. A distância de um percurso fechado é definida como sendo a soma das distâncias entre cidades consecutivas do percurso. Deseja-se encontrar um percurso fechado de distância mínima. Suponha um algoritmo guloso que, partindo da cidade 1, move-se para a cidade mais próxima ainda não visitada e que repita esse processo até passar por todas as cidades, retornando à cidade 1.

Considere as seguintes afirmativas.

I Todo percurso fechado obtido com esse algoritmo tem distância mínima.

II O problema do caixeiro viajante pode ser resolvido com um algoritmo de complexidade linear no número de cidades.

III Dado que todo percurso fechado corresponde a uma permutação das cidades, existe um algoritmo de complexidade exponencial no número de cidades para o problema do caixeiro viajante.

Em relação a essas afirmativas, pode-se afirmar que

(a) I é falsa e III é correta.

(b) I, II e III são corretas.

(c) apenas I e II são corretas.

(d) apenas I e III são falsas.

(e) I, II e III são falsas.

(39 - Fundamentos - 2007) - Seja $G = (V, E)$ um grafo simples e finito, onde $|V| = n$ e $|E| = m$.

Nesse caso, analise as seguintes afirmativas.

I. Se G é hamiltoniano, então G é 2-conexo em vértices.
II. Se G é completo, então G é hamiltoniano.
III. Se G é 4-regular e conexo, então G é euleriano.
IV. Se G é bipartite com partições A e B, então G é hamiltoniano se, e somente se, $|A| = |B|$.
V. Se G é euleriano, então G é 2-conexo.

A análise permite concluir que são **FALSOS**

1. apenas os itens I e II.
2. apenas os itens I e V.
3. apenas os itens II e III.
4. apenas os itens III e IV.
5. apenas os itens IV e V.

40 - Fundamentos - 2007
Tem figurinhas. Mostrar no arquivo da prova.

(25 - Fundamentos - 2008) - Um grafo $G(V, E)$ é uma árvore se G é conexo e acíclico.
Assinale a **definição** que **NÃO** pode ser usada para definir árvores.

1. G é conexo e o número de arestas é mínimo.
2. G é conexo e o número de vértices excede o número de arestas por uma unidade.
3. G é acíclico e o número de vértices excede o número de arestas por uma unidade.
4. G é acíclico e, para todo par de vértices v, w, que não são adjacentes em G, a adição da aresta (v, w) produz um grafo contendo exatamente um ciclo.
5. G é acíclico, e o número de arestas é mínimo.

(26 - Fundamentos - 2008) - Em um grafo $G(V, E)$, o grau de um vértice v é o número de vértices adjacentes a v.
A esse respeito, assinale a afirmativa **CORRETA**.

1. Num grafo, o número de vértices com grau ímpar é sempre par.
2. Num grafo, o número de vértices com grau par é sempre ímpar.
3. Num grafo, sempre existe algum vértice com grau par.
4. Num grafo, sempre existe algum vértice com grau ímpar.
5. Num grafo, o número de vértices com grau ímpar é sempre igual ao número de vértices com grau par.

(28 - Fundamentos - 2008) - Seja $G(V, E)$ um grafo tal que $|V| = n$ e $|E| = m$.

Analise as seguintes sentenças:

I. Se G é acíclico com no máximo $n-1$ arestas, então G é uma árvore.

II. Se G é um ciclo, então G tem n árvores geradores distintas.

III. Se G é conexo com no máximo $n-1$ arestas, então G é uma árvore.

IV. Se G é conexo e tem um ciclo, então para toda árvore geradora T de G, $E(G) - E(T) \neq \emptyset$

(24 - Fundamentos - 2009) - Assinalar a afirmativa correta, em relação a um grafo completo G com $n > 2$ vértices.

1. O grau de cada vértice é n.
2. O número cromático de G é igual a $n - 1$.
3. G não pode ser um grafo bipartido.
4. G não possui caminho hamiltoniano.
5. G possui ciclo euleriano.

(30 - Fundamentos - 2009) - Considere o algoritmo de busca em largura em grafos. Dado o grafo $V = A, B, C, D, E, F$ e $E = (A, C), (A, B), (B, D), (C, E), (C, D), (D, E), (D, F), (E, F)$ e o vértice A como ponto de partida, a ordem em que os vértices são descobertos é dada por:

1. $A\ B\ C\ D\ E\ F$
2. $A\ B\ D\ C\ E\ F$
3. $A\ C\ D\ B\ F\ E$
4. $A\ B\ C\ E\ D\ F$
5. $A\ B\ D\ F\ E\ C$

(43 - Fundamentos - 2010) Dados dois grafos não orientados $G_1(V_1, E_1)$ e $G_2(V_2, E_2)$:

- $G_1 : V_1 = \{a, b, c\}$ $E_1 = \{(a, b), (b, c), (a, c)\}$
- $G_2 : V_2 = \{d, e\}$ $E_2 = \{(d, e)\}$

Qual alternativa apresenta corretamente o grafo Gr(V,E) resultante da soma dos grafos G_1 e G_2?

a) Gr: $V = \{a, b, c, d, e\}$ $E = \{(a, b), (b, c), (a, c), (d, e)\}$
b) Gr: $V = \{a, b, c, d, e\}$ $E = \{(a, d), (a, e), (b, d), (b, e), (c, d), (c, e), (d, e)\}$
c) Gr: $V = \{a, b, c, d, e\}$ $E = \{(a, b), (b, c), (a, c), (a, d), (a, e), (b, d), (b, e), (c, d), (c, e)\}$
d) Gr: $V = \{a, b, c, d, e\}$ $E = \{(a, b), (b, c), (a, c), (a, d), (a, e), (b, d), (b, e), (c, d), (c, e), (d, e)\}$
e) Gr: $V = \{a, b, c, d, e\}$ $E = \{(a, b), (b, c), (c, d), (d, e), (e, a)\}$

(2010 - Fundamentos - 46) Qual é o número cromático do grafo $K_{3,2}$?

a) 2

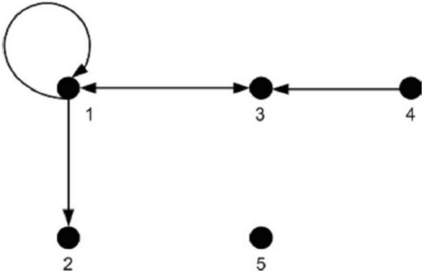
b) 3

c) 4

d) 5

e) 6

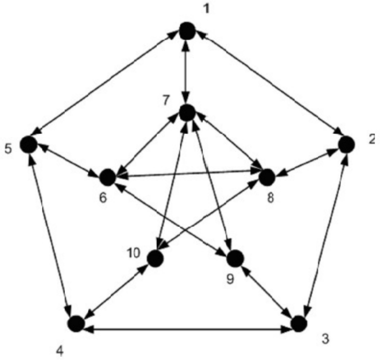
(2011 - Matemática - 13) Considere o grafo a seguir.



O grafo representa a relação:

a) $R = \{(1,1),(1,2),(1,3),(3,1),(4,3)\}$
b) $R = \{(1,1),(1,2),(1,3),(3,1),(3,4)\}$
c) $R = \{(1,1),(1,3),(2,1),(3,1),(3,4)\}$
d) $R = \{(1,1),(1,2),(1,3),(3,4),(4,3)\}$
e) $R = \{(1,1),(1,3),(2,1),(3,1),(4,3)\}$

(2011 - Matemática - 18) Sejam 10 cidades conectadas por rodovias, conforme o grafo a seguir.



Um vendedor sai de uma das cidades com o intuito de visitar cada uma das outras cidades uma única vez e retornar ao seu ponto de partida. Com base no grafo e nessa informação, considere as afirmativas a seguir.

I. O vendedor cumprirá seu propósito com êxito se sair de uma cidade par.

II. O vendedor cumprirá seu propósito com êxito se sair de uma cidade ímpar.

III. O vendedor não cumprirá seu propósito com êxito se sair de uma cidade par.

IV. O vendedor não cumprirá seu propósito com êxito se sair de uma cidade ímpar.

Assinale a alternativa correta.

a) Somente as afirmativas I e II são corretas.

b) Somente as afirmativas I e IV são corretas.

c) Somente as afirmativas III e IV são corretas.

d) Somente as afirmativas I, II e III são corretas.

e) Somente as afirmativas II, III e IV são corretas.

Um vendedor sai de uma das cidades com o intuito de visitar cada uma das outras cidades uma única vez e retornar ao seu ponto de partida. Com base no grafo e nessa informação, considere as afirmativas a seguir.

I. O vendedor cumprirá seu propósito com êxito se sair de uma cidade par.

II. O vendedor cumprirá seu propósito com êxito se sair de uma cidade ímpar.

III. O vendedor não cumprirá seu propósito com êxito se sair de uma cidade par.

IV. O vendedor não cumprirá seu propósito com êxito se sair de uma cidade ímpar.

Assinale a alternativa correta.

a) Somente as afirmativas I e II são corretas.

b) Somente as afirmativas I e IV são corretas.

c) Somente as afirmativas III e IV são corretas.

d) Somente as afirmativas I, II e III são corretas.

e) Somente as afirmativas II, III e IV são corretas.

(2011 - Fundamentos - 44) Qual a quantidade mínima de arestas que se deve remover do grafo completo com 6 vértices, K6, para se obter um grafo planar?

a) 1

b) 2

c) 3

d) 4

e) 5

(2011 - Fundamentos - 47) Seja G um grafo conexo. Considere a notação a seguir.

* c_v é o número cromático em vértices de G.
* c_e é o número cromático em arestas de G.
* g_{min} é o grau mínimo de G.
* g_{max} é o grau máximo de G.
* w é a quantidade de vértices do maior subgrafo completo de G.

Assinale a alternativa correta.

a) $c_v \leq c_e$

b) $c_v \leq w$

c) $c_e \leq g_{max}$

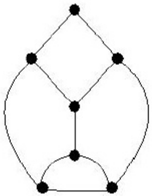
d) $c_v \leq g_{max} + 1$

e) $c_v \geq g_{min}$

(2011 - Fundamentos - 50) Seja G um grafo conexo com n vértices. Considere duas rotulações dos vértices de G obtidas por duas buscas em G, uma em largura, $l()$, e outra em profundidade, $p()$, ambas iniciadas no vértice v. Em cada rotulação, os vértices receberam um número de 1 a n, o qual representa a ordem em que foram alcançados na busca em questão. Assim, $l(v) = p(v) = 1$; enquanto $l(x) > 1$ e $p(x) > 1$ para todo vértice x diferente de v. Considere dois vértices u e w de G e denote por $d(u, w)$ a distância em G de u até w. Com base nesses dados, assinale a alternativa correta.

a) Se $l(u) < l(w)$ e $p(u) < p(w)$, então $d(v, u) < d(v, w)$.
b) Se $l(u) < l(w)$ e $p(u) > p(w)$, então $d(v, u) = d(v, w)$.
c) Se $l(u) > l(w)$ e $p(u) < p(w)$, então $d(v, u) \leq d(v, w)$.
d) Se $l(u) > l(w)$ e $p(u) > p(w)$, então $d(v, u) < d(v, w)$.
e) Se $l(u) < l(w)$ e $p(u) > p(w)$, então $d(v, u) \leq d(v, w)$.

(2013) Qual o número cromático do grafo a seguir?



(2014) Considerando que um grafo possui n vértices e m arestas, assinale a alternativa que apresenta, corretamente, um grafo planar.

a) $n = 5$, $m = 10$

b) $n = 6$, $m = 15$

c) $n = 7$, $m = 21$

d) $n = 8$, $m = 12$

e) $n = 9$, $m = 22$

(2016 - Fundamentos - 32) A matriz de um grafo G = (V,A) contendo n vértices é uma matriz n x n de bits, em que A[i,j] é 1 (ou verdadeiro, no caso de booleanos) se e somente se existir um arco do vértice i para o vértice j. Essa definição é uma:

(A) Matriz de adjacência para grafos não ponderados.

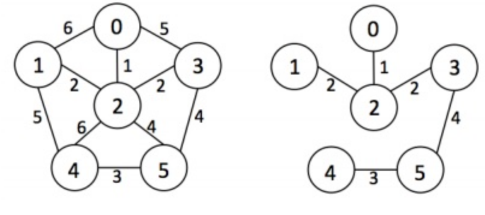
(B) Matriz de recorrência para grafos não ponderados.

(C) Matriz de incidência para grafos não ponderados.

(D) Matriz de adjacência para grafos ponderados.

(E) Matriz de incidência para grafos ponderados.

(2016 - Fundamentos - 36) A Figura (a) abaixo mostra o exemplo de um grafo não direcionado G com os pesos mostrados ao lado de cada aresta. Sobre a árvore T representada na Figura (b), é correto afirmar que:



(a) (b)

(A) T representa a árvore geradora mínima do grafo da Figura (a) cujo peso total é 12. T não é única, pois a substituição da aresta (3,5) pela aresta (2,5) produz outra árvore geradora de custo 12.

(B) T representa a árvore de caminhos mais curtos entre todos os pares de vértices do grafo da Figura (a). T não é única, pois a substituição da aresta (3,5) pela aresta (2,5) produz caminhos mais curtos entre os mesmos pares de vértices do grafo.

(C) T representa a árvore geradora mínima do grafo da Figura (a) cujo peso total é 12. A substituição da aresta (3,5) pela aresta (2,4) produz uma árvore geradora máxima cujo peso total é 14.

(D) T representa a ordenação topológica do grafo da Figura (a). O peso da aresta (0,2) indica que ela deve ser executada antes da aresta (2,3) e o peso da aresta (2,3) indica que ela deve ser executada antes da aresta (4,5) e assim sucessivamente.

(E) T representa a árvore de caminhos mais curtos do grafo da Figura (a) com origem única no vértice 2. T não é única, pois a substituição da aresta (3,5) pela aresta (2,4) produz caminhos mais curtos entre todos os pares de vértices do grafo.

(2016 - Fundamentos - 37) Em relação a Teoria dos Grafos, relacione a Coluna 1 à Coluna 2.

Coluna 1

1. Grafo Completo.
2. Hipergrafo.
3. Árvore Livre.
4. Grafo Planar.
5. Grafo não direcionado antirregular.

Coluna 2

( ) Grafo não direcionado, no qual todos os pares de vértices são adjacentes entre si.

( ) Grafo não direcionado em que cada aresta conecta um número arbitrário de vértices, ao invés de conectar dois vértices apenas.

( ) Grafo não direcionado acíclico e dirigido.

( ) Grafo em que seu esquema pode ser traçado em um plano, de modo que duas arestas quaisquer se toquem, no máximo, em alguma extremidade.

( ) Grafo que possui o maior número possível de graus diferentes em sua sequência.

A ordem correta de preenchimento dos parênteses, de cima para baixo, é:

(A) 1 - 2 - 3 - 4 - 5.

(B) 2 - 3 - 4 - 5 - 1.

(C) 3 - 4 - 5 - 1 - 2.

(D) 4 - 5 - 1 - 2 - 3.

(E) 5 - 1 - 2 - 3 - 4.

Questões do ENADE

- (Ciência da Computação 2014 - Questão 22) Considere o processo de fabricação de um produto siderúrgico que necessita passar por n tratamentos térmicos e químicos para ficar pronto. Cada uma das n etapas de tratamento é realizada uma única vez na mesma caldeira. Além do custo próprio de cada etapa do tratamento, existe o custo de se passar de uma etapa para outra, uma vez que, dependendo da sequência escolhida, pode ser necessário alterar a temperatura da caldeira e limpá-la para evitar a reação entre os produtos químicos utilizados. Assuma que o processo de fabricação inicia e termina com a caldeira limpa. Deseja-se projetar um algoritmo para indicar a sequência de tratamentos que possibilite fabricar o produto com o menor custo total. Nessa situação, conclui-se que:

A A solução do problema é obtida em tempo de ordem O(nLogn), utilizando-se um algoritmo ótimo de ordenação.

B Uma heurística para a solução do problema de coloração de grafos solucionará o problema em tempo polinomial.

C O problema se reduz a encontrar a árvore geradora mínima para o conjunto de etapas do processo, requerendo tempo de ordem polinomial para ser solucionado.

D A utilização do algoritmo de Dijkstra para se determinar o caminho de custo minimo entre o estado inicial e o final soluciona o problema em tempo polinomial.

E Qualquer algoritmo conhecido para a solução do problema descrito possui ordem de complexidade de tempo nao-polinomial, uma vez que o problema do caixeiro viajante se reduz a ele.

(Ciência da Computação 2014 - Questão 30) Uma fazenda possui um único poço artesiano que deve abastecer n bebedouros para o gado. Deseja-se determinar um projeto de ligação entre esses n+1 pontos através de encanamentos com a menor extensão total. Um algoritmo proposto para a solução do problema executa os seguintes passos:

1. Crie n+1 conjuntos unitérios, cada um contendo um dos pontos a serem ligados entre si e insira esses conjuntos em um conjunto C.
2. Crie um conjunto D contendo um registro para cada combinação possível de dois pontos distintos a serem ligados. Cada registro deve conter os campos c_i , $c_j j$ e d, em que c_i e c_j são os dois pontos a serem ligados e d é a distância entre eles.
3. Enquanto D não estiver vazio faça:

    1) Remova o registro de D com o menor valor de distância d.
    2) Se os valores de c_f e c_l. do registro removido pertencerem a conjuntos distintos de C, então
        a) Substitua estes dois conjuntos pela união entre eles.
        b) Guarde o registro removido em um conjunto-solução.

Com base na descrição do problema e do algoritmo proposto, conclui-se que

A O problema exemplifica a obtenção de uma árvore geradora mínima, portanto está no conjunto P.

B O algoritmo é uma heurística para o Problema do Caixeiro Viajante, logo apresenta complexidade polinomial.

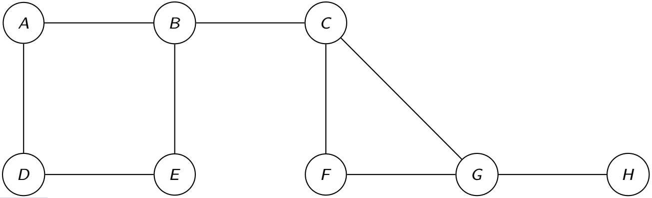
C O problema descrito é de otimização, logo pertence ao conjunto NP-difícil, mas não ao conjunto NP-completo.

D Uma alternativa para a solução do problema é usar o algoritmo de Dijkstra para obtenção do caminho mínimo entre dois pontos.

E O passo de maior custo do algoritmo é a criação do conjunto D com as combinações de pontos, apresentando complexidade computacional $\Theta(n!)$.

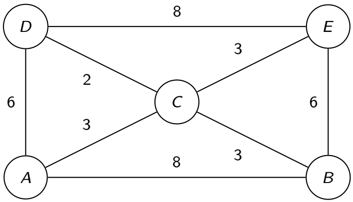
1) (2 pontos) Considere a sequência de números binários de 0 a 7. É possível colocar os 8 números em uma sequência de modo que, de um número para o outro, mude apenas um bit? Como isso pode ser resolvido através de teoria de grafos? Se necessário, desenhe o grafo associado ao problema.

2) (2 pontos) Decomponha o grafo a seguir em seus componentes biconexos, a partir do vértice *e*, utilizando o algoritmo visto em aula. Mostre todos os passos do algoritmo (árvore em profundidade mostrando as arestas de retorno, identificação das articulações, identificação dos demarcadores e dos componentes a cada passo).



3) (2 pontos) No grafo do exercício anterior, aplique o algoritmo de Tremaux, a partir do vértice A, mostrando como ficam ao final as marcações das passagens. Use como critério de escolha entre dois vértices, sempre o de menor ordem alfabética.

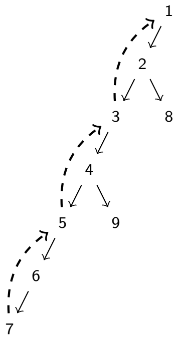
4) (2 pontos) Aplique o algoritmo de Dijkstra ao grafo a seguir, para calcular a distância entre os vértices A e E. Mostre o vetor com os λ a cada vez que passar pelo passo 3.



5)(1 ponto) Um grafo conexo é dito euleriano se possui um ciclo euleriano. E semi-euleriano se possui um caminho euleriano mas não possui um ciclo euleriano. Determine os valores de n para os quais K_n é euleriano e semi-euleriano.
6)(1 ponto) É possível que um grafo bipartido com um número ímpar de vertices seja hamiltoniano (isso é, contenha um ciclo hamiltoniano)? E euleriano? Justifique sua resposta.

Primeira Prova - 6/10/10

1) (2.0 pontos) A figura a seguir representa a árvore resultante da busca em profundidade e as arestas de retorno de um grafo G. Mostre o lowpt de cada vértice. Identifique as articulações, demarcadores e componentes biconexas de G.



2) (2.0 pontos) Para qualquer inteiro positivo k, um cubo de dimensão k (ou k-cubo) é o grafo definido da seguinte maneira: os vértices do grafo são todas as seqüências $b_1 b_2 \ldots b_k$ de bits; dois vértices são adjacentes se e somente se diferem em exatamente uma posição. Por exemplo, os vértices do cubo de dimensão 3 são 000, 001, 010, 011, 100, 101, 110, 111; o vértice 000 é adjacente aos vértices 001, 010, 100 e a nenhum outro; e assim por diante.

- (0.4 pontos) Faça uma figura do 1-cubo, do 2-cubo e do 3-cubo.
- (0.4 pontos) Mostre a matriz de adjacências de um 3-cubo.
- (0.4 pontos) Quantos vértices tem o k-cubo?
- (0.4 pontos) Quantas arestas tem o k-cubo?
- (0.4 pontos) O 3-cubo é planar?

3) (2.0 pontos) Descreva sucintamente qual a função de cada um dos 8 algoritmos a seguir, deixando bem claras as diferenças entre eles, caso haja algoritmos com funções semelhantes (.25 cada um):

Floyd-Warshall	Tremaux
Prim	Moore
Kruskal	Dijkstra
Demoucron	Ford

4) (2.0 pontos)O Departamento de Informática pretende oferecer 7 disciplinas para o nono semestre do Curso de Computação no próximo semestre.

(C) Compiladores
(G) Grafos
(L) Lógica
(N) Análise de Algoritmos
(P) Probabilidade
(S) Engenharia de Software
(T) Arquitetura de Computadores

Com base na pré-matrícula, verificou-se que cada estudante planeja cursar as disciplinas indicadas abaixo:

Aldo C,L,T	Eduardo L,N	Iracema C,T
Batista C,G,S	Frederico C,G	Janete C,S,T
Carlos C,L	Giovana N,P	Kevin P,S
Daniel G,N	Homero G,L	Luis P,T

Considerando que cada disciplina é dada num período, como se poderia determinar qual o menor número de períodos necessários para que os sete cursos possam ser oferecidos e todos os alunos possam cursar as disciplinas planejadas? Para os dados fornecidos, qual é esse número?

5)(2 pontos) Considere o problema 544 da ACM, descrito a seguir. Como esse problema poderia ser modelado e resolvido com teoria de grafos? Descreva em detalhes o grafo que modelaria o problema, o que se buscaria nesse grafo e qual o algoritmo utilizado. Caso seja necessário modificar o algoritmo, descreva o que deveria ser modificado no mesmo.

Problema 544 da ACM - Heavy Load (Carga Pesada)

O Problema

Big Johnsson Trucks Inc. é uma empresa especializada na fabricação de caminhões grandes. Seu último modelo, o V12 Godzilla, é tão grande que a quantidade de carga que pode transportar com ela nunca é limitado pelo próprio caminhão. É limitado apenas pelas restrições de peso que se aplicam às estradas ao longo do caminho que pretende conduzir.

Dadas a cidade de origem e de destino, seu trabalho é determinar a carga máxima do V12 Godzilla de forma que exista um caminho entre as duas cidades especificadas.

Entrada

O arquivo de entrada irá conter um ou mais casos de teste. A primeira linha de cada caso de teste contém dois números inteiros: o número N de cidades $(2 \leq n \leq 200)$ e o número r e segmentos de estrada $(1 \leq r \leq 19.900)$ que compõem a rede da rua. Então r linhas seguirão, cada uma descrevendo um segmento rodoviário, contendo as duas cidades ligadas pelo segmento e o limite de peso de caminhões que usam esse segmento. Os nomes não tem mais de 30 caracteres e não contêm caracteres de espaço em branco. Os limites de peso são números inteiros no intervalo 0-10000. Estradas sempre podem ser percorrido em ambos os sentidos. A última linha do caso de teste contém os nomes da cidade de origem e da cidade de destino. A entrada será encerrado por dois valores de 0 para n e r.

Saída

Para cada caso de teste, imprimir três linhas:

* Uma linha dizendo "Cenário x" onde x é o número do caso de teste

* Uma linha dizendo "y toneladas" onde y é a carga máxima possível

* Uma linha em branco

Exemplo de entrada

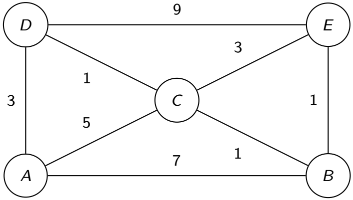
4 3
Karlsruhe Stuttgart 100
Stuttgart Ulm 80
Ulm Muenchen 120
Karlsruhe Muenchen
5 5
Karlsruhe Stuttgart 100
Stuttgart Ulm 80
Ulm Muenchen 120
Karlsruhe Hamburg 220
Hamburg Muenchen 170
Muenchen Karlsruhe
0 0

Exemplo de Saída
Cenário 1
80 toneladas
Cenário 2
170 toneladas

Primeira Prova - 09/05/11

1)(2.0 pontos) Considere a árvore obtida a partir de uma busca em profundidade em um grafo G, com raiz no vértice 3 e definida pelos seguintes arcos:
Arcos diretos: $\{(3,4),(4,5),(4,2),(5,6),(5,1),(6,9),(1,7),(1,8)\}$
Arcos de retorno: $\{(5,3),(9,5),(7,5)\}$
Mostre o lowpt de cada vértice. Identifique as articulações, demarcadores e componentes biconexas de G.
4)(1.0 ponto)Um grafo conexo é dito euleriano se possui um ciclo euleriano. Desenho 6 grafos eulerianos não isomorfos com exatamente seis vértices (há 8 grafos assim).

5)(2 pontos) Aplique o algoritmo de Dijkstra ao grafo a seguir, para calcular a distância entre os vértices A e E. Mostre o vetor com os λ a cada vez que passar pelo passo 3.



6)(1.0 ponto) Suponha um conjunto de n números inteiros positivos. Diga uma condição necessária para que esses números possam representar os graus dos n vértices de uma árvore. Essa condição é suficiente?

Primeira Prova - 3/10/11

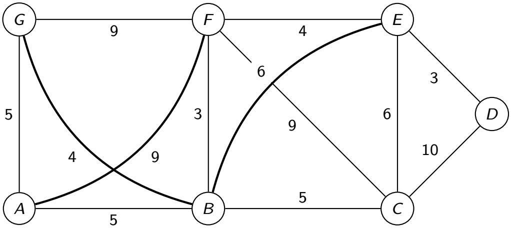
1) (1.0 ponto) O grafo resultante da remoção das arestas (3,1), (3,2), (6,4), (6,5) e (5,4) de K_6 é planar? Justifique sua resposta.

3) (1 ponto) Seja G um grafo com 14 vértices e 25 arestas. Se todo vértice de G tem grau 3 ou 5, quantos vértices de grau 3 o grafo G possui?

4) (1 ponto) Um grafo é dito r-regular se todos os vértices tem grau r. Dê um exemplo de um grafo 3-regular que não seja completo.

5) (1 ponto) Uma floresta é um grafo desconexo em que todas as componentes conexas são árvores. Determine o número de arestas de uma floresta com n vértices e k componentes conexos, em função de n e k.

8) (1 ponto) O grafo a seguir corresponde ao projeto de uma rede de computadores. Os vértices correspondem às máquinas e as arestas correspondem à possibilidade de conexão entre duas máquinas, juntamente com o correspondente custo de instalação. Essa rede deverá ser construída de forma que cada computador possa se comunicar (direta ou indiretamente) com cada um dos outros computadores, sempre a um custo mínimo, não importando o custo total da rede. Como a rede é bastante segura, não há a necessidade de haver mais de uma rota de comunicação entre dois computadores pois sempre será utilizada a de menor custo. Sendo assim, algumas conexões nunca serão utilizadas, de forma que não precisam ser implantadas. Descreva, do ponto de vista de Teoria de Grafos, como o problema pode ser resolvido.



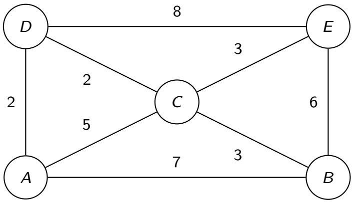
Primeira Prova - 18/10/11

1) (1.5 pontos) Um cachorro, um gato e um rato estão na margem de um rio. Um barqueiro deseja levá-los até a outra margem. Uma vez que o barco é pequeno, ele apenas pode levar um deles por vez. Por razões óbvias, nem o cachorro e o gato, nem o gato e o rato podem ficar sozinhos. Como o barqueiro irá fazer para transportá-los? Descreva como esse problema pode ser resolvido utilizando teoria de grafos e desenhe (pelo menos parcialmente, mínimo de 7 estados) o grafo que modela o problema.
2)(1 ponto) Quantos grafos diferentes de N vértices existem? Encontre uma expressão matemática do número de grafos em função de N.

3)(1.5 pontos) Como se pode determinar quantos caminhos diferentes existem entre um dado par de vértices em um grafo?
4)(1 ponto) Um grafo conexo é dito euleriano se possui um ciclo euleriano. E semi-euleriano se possui um caminho euleriano mas não possui um ciclo euleriano. Determine os valores de n para os quais K_n é euleriano e os valores para os quais ele é semi-euleriano.

5)(1 ponto) É possível que um grafo bipartido com um número ímpar de vertices seja hamiltoniano (isso é, contenha um ciclo hamiltoniano)? E euleriano? Justifique sua resposta.

6)(2 pontos) Aplique o algoritmo de Dijkstra (o algoritmo está ao final da prova) ao grafo a seguir, para calcular a distância entre os vértices A e E. Mostre o vetor com os λ a cada vez que passar pelo passo 3.



2) (1.0 ponto) Para um número natural r, um grafo é r-regular se todos os vértices têm grau r. Para um grafo r-regular com n vértices e m arestas, expresse m em função de n e r.

6) (1.0 ponto) Considere um grafo G representado em uma matriz de adjacências M. Descreva um algoritmo baseado no produto matricial para verificar se o grafo G possui ciclos de comprimento igual a 3.

Primeira Prova - 05/10/2012

1)(1.0 ponto) Quantos subgrafos induzidos com 2 vértices possui um grafo de n vértices e m arestas?

4)(1.0 ponto) Seja T uma árvore com vértices $1, ..., n$. Suponha que os graus dos vértices $1, 2, 3, 4, 5, 6$ são $7, 6, 5, 4, 3, 2$ respectivamente e que os vértices $7, ..., n$ são folhas. Determine n.

6)(1.0 ponto) Considere um grafo G representado em uma matriz de adjacências M. Vimos em aula um algoritmo baseado em produto matricial para encontrar a quantidade de caminhos entre dois vértices. Descreva como o algoritmo poderia ser utilizado para verificar se o grafo G possui ciclos de comprimento igual a 3.
7)(1.0 ponto) O algoritmo de Dijkstra somente pode ser utilizado em grafos em que os pesos não positivos. Construa um grafo com pesos que podem ser negativos e no qual o algoritmo de Dijkstra devolve a resposta errada.
8)(1.0 ponto) Qual o menor número de arestas que devem ser retiradas de K_6 para que ele se torne planar?

Primeira Prova - 04/10/2013

2) (1.0 ponto) Para um número natural r, um grafo é r-regular se todos os vértices têm grau r. Para um grafo r-regular com n vértices e m arestas, expresse m em função de n e r.
4) (2.0 pontos) Um grafo valorado G com custo $c(i,j)$ entre cada aresta (i,j) satisfaz a desigualdade triangular se para todos os vértices $u, v, w \in V$:

$$c(u,w) \leq \; c(u,v) + c(v,w) \tag{1}$$

Escreva uma função (em C ou português estruturado) que receba uma matriz de adjacência e verifique se o grafo atende à desigualdade triangular, retornando:0 - Se não atende; 1 - Se atende.

8) (1 ponto) Considere o problema de, em um grafo valorado em que são atribuídos pesos às arestas, encontrar o caminho que possui o menor valor para a aresta máxima do caminho. Que alterações devem ser feitas no algoritmo de Dijkstra (mostrado a seguir) para que encontre esse caminho?

Primeira Prova - 16/10/13

2) (1 ponto) Considere um grafo com n vértices, m arestas e k componentes conexas. Quantas arestas é necessário acrescentar para torná-lo conexo?

3) (2 pontos) Uma forma de encontrar o centro de uma árvore é ir removendo, a cada iteração, todas as folhas, até que sobrem somente um ou dois vértices, que serão os pontos centrais. Faça uma função (em C ou português estruturado) que receba uma matriz de adjacências representando uma árvore de 10 vértices e escreva o número do vértice ou vértices que compõem o centro da árvore.

4) (1 ponto) Dadas as seguintes afirmações

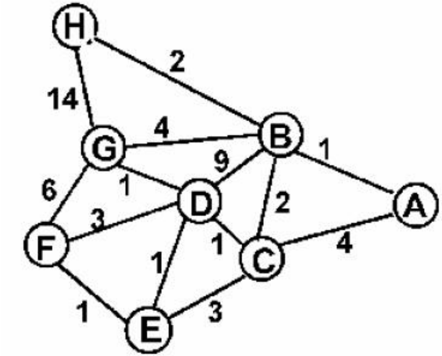
1. Qualquer grafo conexo com n vértices deve ter pelo menos $n-1$ arestas.
2. O grafo bipartido completo $K_{m,n}$ é Euleriano desde que m e n sejam ímpares.
3. Em um grafo o número de vértices de grau ímpar é sempre par.

São verdadeiras:

1. Somente a afirmaçao (1).
2. Somente as afirmações (1) e (3).
3. Somente as afirmações (2) e (3).
4. Somente as afirmações (1) e (2).
5. Todas as afirmações.

5) (1 ponto) Encontre uma expressão que relacione a quantidade de grafos possíveis para N vértices.

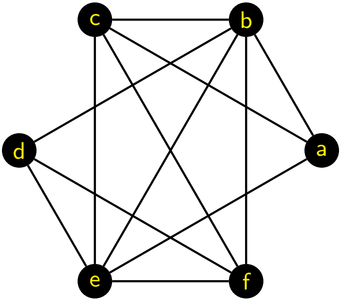
6) (2 pontos) Aplique o algoritmo de Dijkstra ao grafo a seguir, para calcular a distância entre o vértice F e todos os outros. Mostre o vetor com os λ a cada vez que passar pelo passo 3.



7) (2 pontos) O algoritmo de Floyd-Warshall encontra o caminho mínimo entre todos os pares de vértices do grafo. É baseado em programação dinâmica, de forma que a solução final vai sendo composta a partir das soluções parciais. A cada passo k o algoritmo calcula o caminho mínimo entre dois vértices que passa apenas por vértices de numeração menor ou igual a k.
Aplique o algoritmo de Floyd-Warshall no grafo representado pela matriz de adjacências a seguir, mostrando a matriz de distâncias a cada iteração.

0	3	8	∞
∞	0	∞	1
∞	4	0	∞
2	∞	5	0

4) (2.5 pontos) Aplique o teste de planaridade de demoucron ao grafo a seguir para determinar se é planar ou não. Considere o ciclo inicial como b-c-e-b. A cada iteração mostre as pontes e as faces, até o final do algoritmo.



3) (1 ponto) Seja T uma árvore com $p + q$ vértices. Suponha que p dos vértices tem grau 4 e q são folhas. Qual a relação matemática entre p e q?

3) (2 pontos) No grafo do exercício anterior, aplique o algoritmo de Tremaux, a partir do vértice A, mostrando como ficam ao final as marcações das passagens. Use como critério de escolha entre dois vértices, sempre o de menor ordem alfabética.

4)(1 pontos) Como se pode determinar quantos caminhos simples (isso é, que passam somente uma vez em cada vértice) diferentes existem entre um dado par de vértices em um grafo?

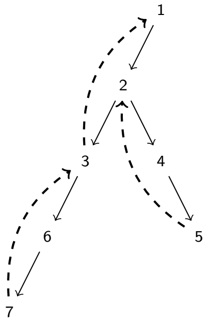
Primeira Prova - 17/10/14

1)(1 ponto) Qual o número máximo de arestas que um grafo NÃO CONEXO simples (sem laços ou arestas paralelas) com 30 vértices pode conter?
2)(1 ponto) É possível haver um grafo simples (sem laços ou arestas paralelas) com mais de um vértice em que todos os vértices tenham graus diferentes? Justifique sua resposta.
3)(1 ponto - POSCOMP 2014) Considerando que um grafo possui n vértices e m arestas, assinale a alternativa que apresenta, corretamente, um grafo planar.

a) $n = 5$, $m = 10$
b) $n = 6$, $m = 15$
c) $n = 7$, $m = 21$
d) $n = 8$, $m = 12$
e) $n = 9$, $m = 22$

4)(1 ponto) É possível construir um grafo bipartido planar com no mínimo 5 vértices de grau 3?

5)(2 pontos) A figura a seguir representa a árvore resultante da busca em profundidade e as arestas de retorno de um grafo G. Identifique o lowpt de cada vértice, as articulações, demarcadores e componentes biconexas de G.



6)(2 pontos) Dado um grafo D com arestas valoradas com pesos entre 0 e 1, representando a probabilidade de uma aresta não falhar; a probabilidade de um caminho em D estar operacional é o produto das probabilidades das arestas desse caminho (i.e., estamos supondo independência das probabilidades de não falhar). Mostre que alterações devem ser feitas no algoritmo de Dijkstra (ao final dessa prova) para descobrir o caminho mais seguro entre dois vértices.

7) (2 pontos) Em teoria dos grafos, o complemento ou inverso de um grafo G é um grafo H que tem os mesmos vértices de G, tais que dois vértices de H são adjacentes se e somente se eles não são adjacentes em G, isso é, o conjunto de arestas de H é formado por todas as arestas que não estão em G. Escreva uma função (na linguagem de sua escolha, incluindo pseudo-código) que receba duas matrizes de adjacências representando dois grafos de 10 vértices, e verifique se um grafo é o complemento do outro, retornando: 0 - Se não são complementares; 1 - Se são complementares.

Recuperação da Primeira Área - 12/12/2014

1) (1.0 ponto) Seja G um grafo com 14 vértices e 25 arestas. Se todo vértice de G tem grau 3 ou 5, quantos vértices de grau 3 o grafo G possui?
2) (1.0 ponto) Para um número natural r, um grafo é r-regular se todos os vértices têm grau r. Para um grafo r-regular com n vértices e m arestas, expresse m em função de n e r.
3) (1.0 ponto) Como é possível mostrar que um grafo não é planar?
4) (1.0 ponto) Sejam G_1 e G_2 dois grafos conexos, sem vértices em comum. Seja x um vértice de G_1 e y um vértice de G_2. Seja G o grafo que se obtém de $G_1 \cup G_2$ acrescentando a aresta (x, y). É possível que o grafo G seja euleriano? E semi-euleriano (contem um caminho euleriano mas não contem um ciclo euleriano)? Em que condições?
5) (1.0 ponto) Quantos subgrafos induzidos com 2 vértices possui um grafo de n vértices e m arestas?

6) (1 ponto) Seja T uma árvore com vértices $1, ..., n$. Suponha que os graus dos vértices $1, 2, 3, 4, 5, 6$ são $7, 6, 5, 4, 3, 2$ respectivamente e que os vértices $7, ..., n$ são folhas. Determine n.

7) (1 ponto) Um grafo é dito r-regular se todos os vértices tem grau r. Dê um exemplo de um grafo 3-regular que não seja completo.

8) (1 ponto) Uma floresta é um grafo desconexo em que todas as componentes conexas são árvores. Determine o número de arestas de uma floresta com n vértices e k componentes conexos, em função de n e k.

9) (1 ponto) Como se pode determinar quantos caminhos diferentes existem entre um dado par de vértices em um grafo?

10) (1 ponto) Quantos grafos diferentes de N vértices existem? Encontre uma expressão matemática do número de grafos em função de N.

Recuperação da Segunda Prova - 11/07/11

1) (2.0 pontos) Encontre um fluxo máximo para a rede abaixo. O número na aresta representa sua capacidade. Os números entre parênteses nos vértices *a* e *c* representam o fluxo máximo através do vértice.

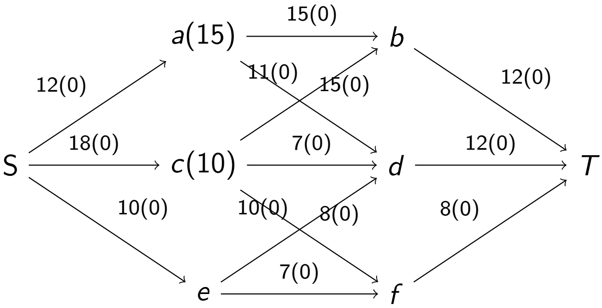


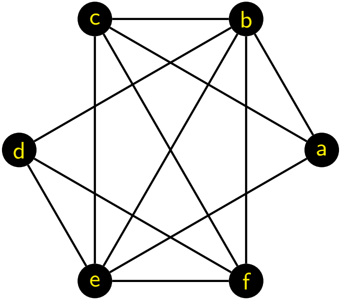
Figura: Encontre o fluxo máximo

2) (2.0 pontos) Utilizando o algoritmo por programação dinâmica visto em aula, encontre o ciclo hamiltoniano de custo mínimo no grafo a seguir:

	1	2	3	4
1	0	3	∞	5
2	6	0	4	5
3	∞	4	0	2
4	2	7	3	0

4)(2.0 pontos) Apresente 5 ordenações topológicas para o dígrafo resultante do exercício 3.

5) (2.0 pontos) Aplique o teste de planaridade de demoucron ao grafo a seguir para determinar se é planar ou não. Considere o ciclo inicial como b-c-e-b. A cada iteração mostre as pontes e as faces, até o final do algoritmo.



Recuperação da Segunda Prova - 08/12/10

1) (2.5 pontos) Encontre um fluxo máximo para a rede abaixo. O número na aresta representa sua capacidade. Os números entre parênteses nos vértices *a* e *c* representam o fluxo máximo através do vértice.

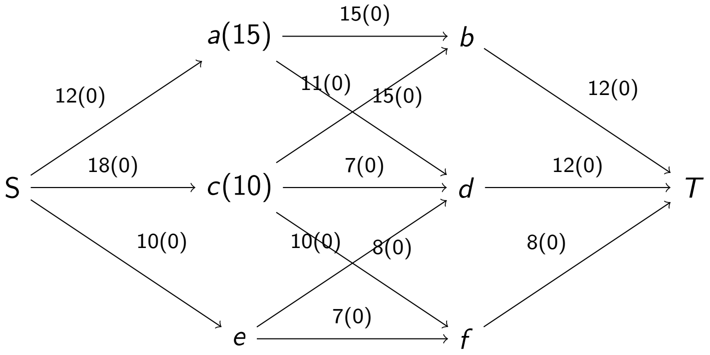
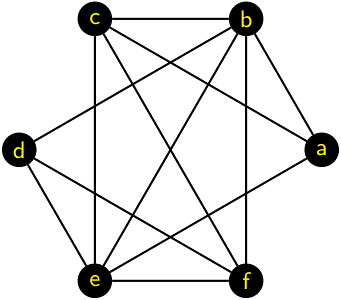


Figura: Encontre o fluxo máximo

2) (2.5 pontos) Utilizando o método de programação dinâmica encontre o ciclo hamiltoniano de custo mínimo no grafo a seguir. Mostre todas as funções calculadas.

	1	2	3	4
1	0	13	∞	5
2	6	0	14	5
3	∞	14	0	2
4	12	7	13	0

4) (2.5 pontos) Aplique o teste de planaridade de demoucron ao grafo a seguir para determinar se é planar ou não. Considere o ciclo inicial como b-c-e-b. A cada iteração mostre as pontes e as faces, até o final do algoritmo.



Segunda Prova - 29/11/13

1) (2 pontos) Para a festa de fim de ano da UCS foram montados 6 comitês, o de Decoração, o de Comidas, o de Bebidas, o de Espetáculo, o de Fundos e o de Música. João está no de Decoração e de Comidas, José está no de Bebidas e de Fundos, Juca está no de Música, Joca está no de Música e de Espetáculo, Jeca está no de Fundos e de Música, Jaca está no de Decoração e de Espetáculo e Julio está no de Bebidas e de Fundos. Deseja-se montar uma comissão de 6 pessoas de modo que cada pessoa represente um comitê do qual participe, cada pessoa representando um comitê diferente. Mostre como é possível resolver (genericamente) esse problema utilizando teoria de grafos, e encontre a solução para o caso específico descrito nessa questão.

2) (2 pontos) Você foi encarregado de montar a tabela de um campeonato com 10 times, em que todos jogam contra todos, durante 9 rodadas. Descreva como a teoria de grafos pode auxiliar você nessa tarefa.

3) (2 pontos) Utilizando o método de programação dinâmica encontre o ciclo hamiltoniano de custo mínimo no grafo a seguir. Mostre todas as funções calculadas.

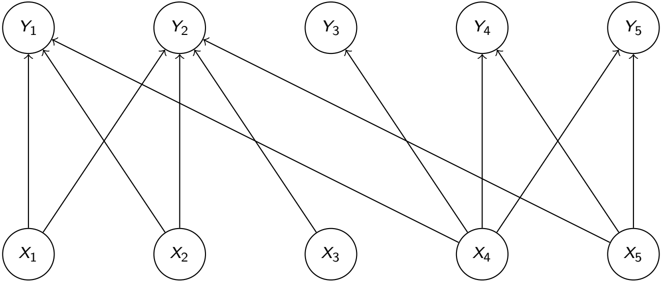
	1	2	3	4
1	0	10	∞	5
2	4	0	10	5
3	∞	12	0	2
4	12	9	13	0

5) (2 pontos) De três depósitos A, B e C, dispondo respectivamente de 20,10 e 35 toneladas de um dado produto, pretende-se fazer chegar a três destinos D, E e F, respectivamente 25, 20 e 20 toneladas do produto. As disponibilidades de transporte em caminhão entre os difentes pontos são os seguintes:

	D	E	F
A	15	10	-
B	5	-	10
C	10	5	5

Mostre como a teoria de grafos pode ser utilizada para estabelecer o melhor plano de transportes, e encontre a solução para a situação específica descrita nessa questão.

2) (2 pontos)O grafo a seguir possui uma associação perfeita? Justifique sua resposta utilizando teoria de grafos.



3)(2 pontos) Qual o número cromático do grafo K_7 após excluídas as arestas (1,2),(1,3),(1,4),(2,3) e (2,4)? Justifique sua resposta.
4) (2 pontos) Encontre o ciclo hamiltoniano de custo mínimo no grafo a seguir:

	1	2	3	4	5
1	0	3	∞	5	3
2	6	0	4	∞	∞
3	∞	4	0	2	2
4	2	∞	3	0	2
5	2	∞	3	2	0

5)(2 pontos) Considere o problema 820 da ACM, descrito a seguir. Como esse problema poderia ser modelado e resolvido com teoria de grafos? Descreva em detalhes o grafo que modelaria o problema, o que se buscaria nesse grafo e qual o algoritmo utilizado. Caso seja necessário modificar o algoritmo, descreva o que deveria ser modificado no mesmo. Qual a resposta (numérica) para a instância do problema utilizada como exemplo do problema?

Problema 820 da ACM - Largura de Banda na Internet
O Problema

Na Internet, as máquinas (nós) são interconectados, e diferentes caminhos podem existir entre um dado par de nós. A capacidade de transferência total (largura de banda) entre dois nós é dada pela quantidade máxima de dados por unidade de tempo que pode ser transmitida de um nó para outro. Usando uma técnica chamada de comutação de pacotes, estes dados podem ser transmitidos ao longo de vários caminhos ao mesmo tempo.

Por exemplo, a tabela a seguir mostra a capacidade de transferência entre cada par de nós, para uma rede de 4 nós.

	1	2	3	4
1	0	20	10	0
2	20	0	5	10
3	10	5	0	20
4	0	10	20	0

No exemplo, a largura de banda entre o nó 1 e o nó 4 é o total de pacotes que pode ir do nodo 1 até o nodo 4 por unidade de tempo.

Entrada

O arquivo de entrada contém descrições de várias redes. Toda descrição começa com uma linha contendo um único inteiro $n (2 \leq n \leq 100)$, que é o número de nós na rede. Os nós são numerados de 1 a n. A próxima linha contém três números, s, t e c. Os números s e t são os nós de origem e destino, e c é o número total de conexões na rede. Após isto há c linhas descrevendo as ligações. Cada uma dessas linhas contém três inteiros: os dois primeiros são os números dos nós conectados, e o terceiro número é a largura de banda da conexão. A largura de banda é um número não-negativo não superior a 1000.

Pode haver mais de uma conexão entre um par de nós, mas um nó não pode ser conectado a si mesmo. Todas as conexões são bidirecionais, ou seja, os dados podem ser transmitidos em ambas as direções ao longo de uma conexão, mas a soma da quantidade de dados transmitidos em ambos os sentidos deve ser menor que a largura de banda.
Uma linha contendo o número 0 segue a descrição da última rede, encerrando a entrada.

Saída

Para cada rede deve ser impresso o número da rede. Em seguida, imprima a largura de banda total entre o nó origem s e o nó destino t. Imprima uma linha em branco após cada caso de teste.

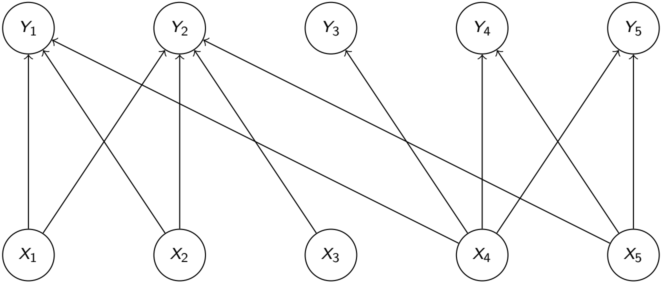
2) (2 pontos) Encontre o ciclo hamiltoniano de custo mínimo no grafo a seguir:

	1	2	3	4	5
1	0	3	∞	5	3
2	6	0	4	∞	∞
3	∞	4	0	2	2
4	2	∞	3	0	2
5	2	∞	3	2	0

3) (2 pontos) Mostre todos os passos do método de alteração estrutural para a obtenção do número cromático do grafo K_5 (vértices identificados pelas letras de a a e) após excluídas as arestas (c, d) e (b, e).

4) (2 pontos) Considere um tabuleiro de xadrez, de 8 por 8 casas. É possível cobrir todo o tabuleiro com peças de dimensão 1 x 2? E se forem removidas duas casas de cantos opostos? Esse problema pode ser resolvido através de teoria de grafos? Como?

1)(2 pontos) O grafo a seguir possui uma associação perfeita? Justifique sua resposta utilizando teoria de grafos.



2) (2 pontos) Em uma nação emergente, cada uma das seis principais cidades receberá uma estação de televisão. Diferentes estações podem usar a mesma faixa de frequência, contanto que estejam distanciadas de mais de 80 quilômetros, para evitar problemas de interferência. Suponha que as cidades tenham os seguintes nomes: A, B, C, D, E e F, e as distâncias entre elas estão descritas na tabela a seguir:

	B	C	D	E	F
A	150	135	96	76	84
B		96	147	90	102
C			72	76	51
D				69	48
E					24

Qual o número mínimo de faixas de frequência necessárias, e como pode ser feita a distribuição entre as cidades? Como isso pode ser resolvido através de Teoria de Grafos?

4) (2 pontos) Encontre o ciclo hamiltoniano de custo mínimo no grafo a seguir:

	1	2	3	4	5
1	0	3	∞	5	3
2	6	0	4	∞	∞
3	∞	4	0	2	2
4	2	∞	3	0	2
5	2	∞	3	2	0

5) (2 pontos) Considere que você é o Chefe do Departamento de Informática e você deve fazer a alocação das disciplinas aos professores para o próximo semestre. Você dispõe, para cada professor, da lista de disciplinas que ele é capaz de ministrar e do número máximo de disciplinas que o professor pode ministrar. Além disso você dispõe da lista de disciplinas que deverá ser oferecida no próximo semestre, e o número de turmas que deve ser oferecido para cada disciplina. Descreva como você pode resolver o problema, ou seja, chegar a uma atribuição de disciplinas aos professores que respeite suas competências e sua carga horária máxima de forma todas as turmas de todas as disciplinas sejam atendidas. Invente uma situação hipotética para ilustrar seu modelo, contendo ao menos 4 professores e 4 disciplinas.

Segunda Prova - 05/12/11

1)(2 pontos) O curso de inglês AILÓVIU pretende enviar 20 alunos/alunas para fazer um intercâmbio de um mês na Inglaterra. Os alunos ficarão em casas de família, dois alunos em cada casa. Cada aluno forneceu à coordenação do curso uma lista com suas preferências de quem gostaria de dividir residência. Observe, entretanto, que a relação de "preferência" não é necessariamente simétrica, isso é, o Juca gostaria de dividir residência com o Zé, mas o Zé não quer dividir residência com o Juca. Explique, DETALHADAMENTE, como o problema poderia ser modelado e resolvido com teoria de grafos. Se for necesário, invente um exemplo.

3)(2 pontos) Encontre o ciclo hamiltoniano de custo mínimo no grafo a seguir:

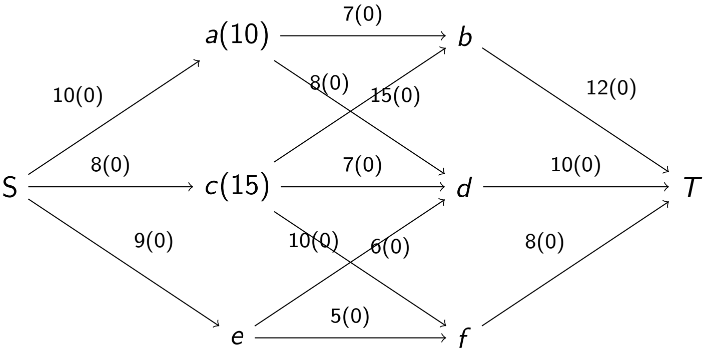
	1	2	3	4	5
1	0	3	∞	5	3
2	6	0	4	∞	∞
3	∞	4	0	2	2
4	2	∞	3	0	2
5	2	∞	3	2	0

4)(2 pontos) Descreva um algoritmo para obter uma ordenação topológica dos vértices de um dígrafo. Dê 5 ordenações topológicas para o dígrafo do exercício 2.

4)(2 pontos) Usando o algoritmo de programação dinâmica visto em aula, encontre o ciclo hamiltoniano de custo mínimo no grafo a seguir:

	1	2	3	4
1	0	7	6	5
2	4	0	5	7
3	∞	2	0	2
4	2	7	3	0

5) (2 pontos) Encontre um fluxo máximo para a rede abaixo. O número na aresta representa sua capacidade. Os números entre parênteses nos vértices *a* e *c* representam o fluxo máximo através do vértice. Mostre pelo menos uma iteração do algoritmo de fluxo máximo, mostrando a rede residual gerada.

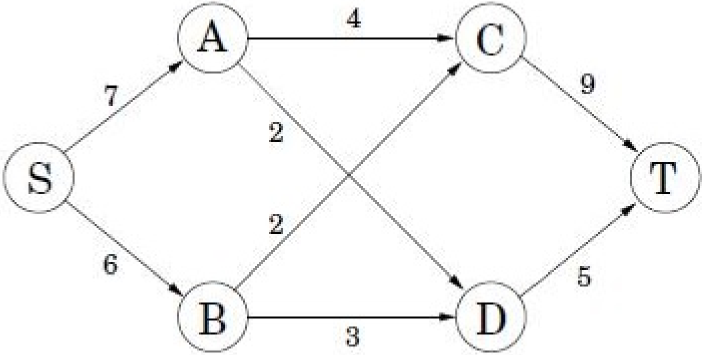


Segunda Prova - 05/12/14

1)(2 pontos) Considere o problema a seguir: Um hospital possui 10 médicos, cada um com uma lista de noites no mês (considere um mês de 4 semanas) em que ele tem disponibilidade para plantões. Como se poderia fazer a alocação dos médicos de modo que haja 2 médicos de plantão a cada noite? Mostre em detalhes como esse problema poderia ser modelado e resolvido utilizando teoria de grafos. Como se poderia acrescentar nesse modelo a restrição de que cada médico não pode atender mais de dois plantões na semana?

3)(2 pontos) Um conjunto dominante é um subconjunto de vértices tal que todo vértice do grafo está no conjunto ou é adjacente a um dos vértices. Escreva uma função (pode ser pseudo-código) que receba uma matriz de adjacências G[10][10] e um vetor Dom[10] representando um subconjunto de vértices de G (os vértices pertencentes ao subconjunto estão marcados com 1 no vetor, os que não pertencem estão marcados com 0) e verifique se o subconjunto descrito em Dom[10] é um Conjunto Dominante de Vértices, retornando: 1 - Se Dom é um conjunto dominante; 0 - Se Dom não é um conjunto dominante.

4)(2 pontos) Utilizando o algoritmo de Ford-Fulkerson visto em aula, calcule o fluxo máximo da rede a seguir, mostrando a rede residual e o novo fluxo a cada passo. Os valores representados nas arestas são as capacidades das arestas.



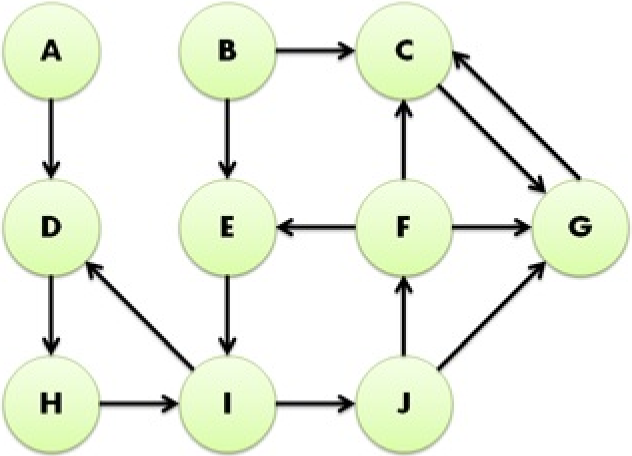
5) (2 pontos) Um vendedor de brinquedos educativos possui em estoque brinquedos de várias formas geométricas (cubos, pirâmides, etc.), cada qual fabricado em várias cores. O vendedor quer carregar consigo o menor número de objetos tal que cada cor e cada forma estejam representadas pelo menos uma vez. Explique como esse problema pode ser modelado e resolvido com teoria de grafos.

Recuperação da Segunda Área - 12/12/14

3) (2 pontos) Utilizando o método de programação dinâmica encontre o ciclo hamiltoniano de custo mínimo no grafo a seguir. Mostre todas as funções calculadas.

	1	2	3	4
1	0	10	∞	5
2	4	0	10	5
3	∞	12	0	2
4	12	9	13	0

5) (2 pontos) Identifique as componentes fortemente conexas do grafo abaixo:

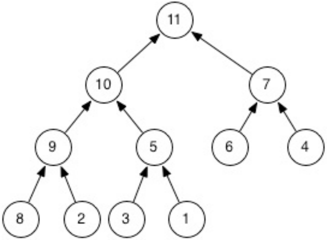


Exercícios de Grafos do URI Online Judge

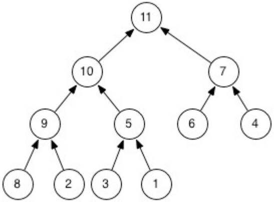
- Prim com fila de prioridades - 1152 - Estradas Escuras

Heaps, priority queues e Heapsort

- Um **heap** é uma estrutura de dados organizada como árvore binária em que todos os níveis até o penúltimo estão completos, e o último nível é preenchido "em ordem", da esquerda para a direita.
- Além disso, o valor da chave em cada nodo deve ser maior (maxheap) ou menor (minheap) que o valor da chave nos dois filhos, e a ordem da chave entre os filhos não interessa.



- Um heap pode ser implementado sobre um vetor, onde cada elemento v[i] é pai dos elementos nas posições v[2*i+1] e v[2*i+2] e a raiz da árvore ocupa a primeira posição (v[0]).
- Assim, dado o índice de um nodo no vetor, os índices do pai e dos filhos à esquerda e à direita podem ser obtidos por:
  - Pai(i) : (i-1)/2
  - Esquerda(i) : i * 2 + 1
  - Direita(i) : i * 2 + 2
- Assim, o heap abaixo poderia ser representado pelo vetor [11 10 7 9 5 6 4 8 2 3 1]



- As principais operações sobre heaps são:
  - cria-heap: cria um heap vazio
  - heapify: cria um heap a partir de um conjunto de elementos
  - busca-maior (em um heapmax) ou busca-menor(em um heapmin)
  - remove-maior: remove a raiz de um heapmax
  - altera-valor-chave
  - insere: adiciona elemento a um heap

Inserção em Heap

- Pode ser implementada adicionando o novo elemento no final do heap e depois "empurrando-o" para cima enquanto ele for maior que o pai (no caso de um Heap Max) até alcançar sua posição.
- Qual o custo?

```
void corrigeSubindo(int index)
{
// Se index não é a raiz e o valor do index for maior do que o valor de seu pai,
// troca seus valores (index e pai(index)) e corrige para o pai
if (index > 0 && heap[index] > heap[pai(index)])
{
troca(heap[index], heap[pai(index)]);
corrigeSubindo(pai(index));
}
}
```

Defines

- No código anterior, uma forma de implementar o cálculo do índice do pai é através de uma função. Uma forma melhor ainda é através de um define.
- Normalmente defines são utilizados para melhorar a legibilidade e manutenção do código. Coisas como o código abaixo, para definir o tamanho de um vetor, de modo que, ao alterar o tamanho do vetor, não seja necessário inspecionar todo o código:
- #define TAM 10

- defines podem receber parâmetros, o que possibilida o seu uso no lugar de funções, com a mesma legibilidade de funções mas sem o custo de chamadas e retornos (o código do define é expandido no lugar do uso do define).
- Os parâmetros vão entre parênteses, como o define abaixo, que calcula o maior valor entre duas expressões
- #define max(A,B) ((A)>(B)?(A):(B))
- ou os defines abaixo, para o cálculo da posição do pai, filho esquerdo e filho direito no heap.
  - #define pai(i) ((i-1)/2)
  - #define fesq(i) (i*2+1)
  - #define fdir(i) (i*2+2)
- Pode-se passar expressões, mas o que resultaria no exemplo acima ao usar-se "fesq(1+2)"?

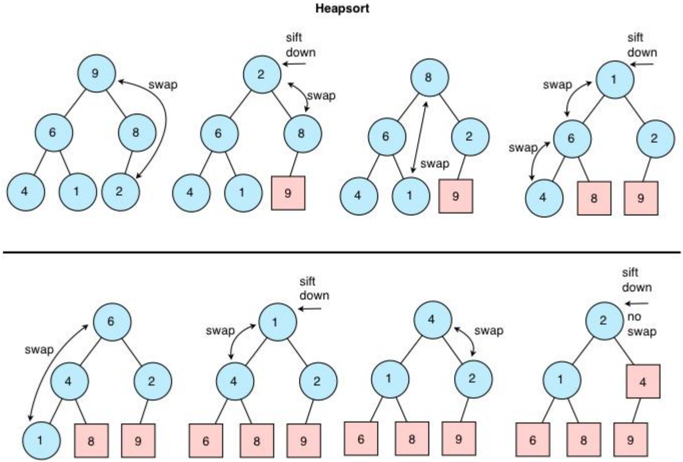
Remoção

- A remoção do maior valor da raiz (no caso do Heap Max) pode ser implementada removendo a raiz e substituindo-a pelo último elemento do Heap e "empurrando-o" para baixo na árvore até que o heap esteja corrigido.
- Qual o custo?

```
void corrigeDescendo(int index)
   {
   if (filhoE(index) < heap.size())     // Se index tem filho
      {
      // Seleciona o filho com menor valor (esquerda ou direita?)
      int filho = filhoE(index);
      if (filhoD(index) < heap.size() && heap[filhoD(index)] > heap[filhoE(index)])
         filho = filhoD(index);
      // Se o valor do pai é maior do que o valor do maior filho, terminamos
      if (heap[index] > heap[filho])
         return;
      // Caso contrário, trocamos o pai com o filho no heap e corrigimos agora para o fi
      troca(heap[index], heap[filho]);
      corrigeDescendo(filho);
      }
   }
```

- Heaps podem ser utilizados para implementar eficientemente filas de prioridades, já que o custo das principais operações (inserção, remoção) é $O(\log n)$ e o custo para obter o maior elemento é $O(1)$.
- Filas de prioridade são utilizadas em diversos algoritmos clássicos tais como:
  - Algoritmo de Dijkstra para caminho mínimo
  - Algoritmos de Prim e Kruskal para árvore geradora mínima
  - Algoritmo de Branch-and-bound para buscar nodo de melhores possibilidades

- Heaps também são utilizados para implementar o algoritmo de ordenação Heapsort que basicamente é:
  - Coloque os dados a serem ordenados em um heap (custo O(n log n))
  - A cada iteração remova a raiz do heap (maior valor) e coloque no final do vetor, e reorganize o heap (O(log n))



Heapsort
swap
sift down
swap
swap
swap
sift down
swap
swap
swap
sift down
swap
sift down
no swap

- Diversas linguagens tem bibliotecas prontas para a implementação de heaps. Em python há uma implementação no pacote **heapq**.
- O heap é implementado sobre uma lista. É um min-heap. Para implementar um max-heap pode-se inverter o sinal dos valores ao inserir e retirar, de modo que o maior passará a ser o menor.

  import heapq
- As principais funções são:
  - heappush(heap,item) - Coloca o item no heap
  - heappop(heap) - retorna o menor elemento do heap removendo-o do mesmo. Se heap estiver vazio, gera um IndexError.
  - heappushpop(heap,item) - Coloca o item no heap e retorna o menor elemento.
  - heapify(x) - Transforma a lista x em um heap em tempo linear.

- Para acessar o menor elemento sem removê-lo do heap basta acessar a posição 0.
- Os itens do heap podem ser tuplas. Nesse caso a comparação é feita pelo primeiro valor e, em caso de empate, pelo segundo valor (e pelo terceiro, etc.)
- O código a seguir recebe um iterador e monta um min_heap a partir dele:

```
>>> def heapsort(iterable):
...     h = []
...     for value in iterable:
...         heappush(h, value)
...     return [heappop(h) for i in range(len(h))]
...
>>> heapsort([1, 3, 5, 7, 9, 2, 4, 6, 8, 0])
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

Priority_queue (fila de prioridade) em C++

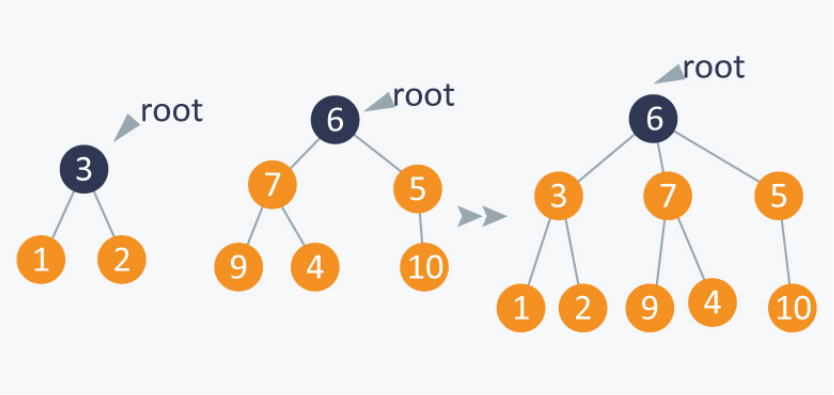
- O container **priority_queue** implementa uma fila de prioridade. Essa estrutura se caracteriza por permitir identificar o maior (ou menor) elemento em tempo O(1), removê-lo em tempo O(log n) e inserir novos elementos em tempo O(log n).
- É útil em algoritmos como o de Prim ou o de Dijkstra, em que um passo importante, e custoso, do algoritmo é a identificação, a cada passo do elemento de menor valor em um conjunto (no caso, a distância de cada vértice ao vértice inicial, no Dijkstra, ou à sub-árvore já formada, no Prim)
- Ele está definido no módulo queue (include queue)

- A versão default implementa uma fila de prioridade max-heap, devolvendo a cada vez o maior valor da fila.
  - priority_queue <int> g;
- Os principais métodos para manipulação de filas de prioridade são :
  - empty() - retorna true se a fila está vazia
  - push(val) - empilha um valor
  - top() - retorna o valor no topo da pilha
  - pop() - remove (e descarta) o valor no topo da pilha
- Para implementar um min-heap, em que o MENOR elemento é mantido no topo, usa-se a sintaxe enigmática abaixo:
  - priority_queue $< int, vector < int >, greater < int >>$ nome;

Disjoint-sets (Union-finds)

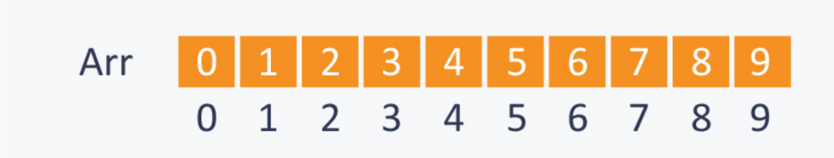
- Um disjoint-set (também conhecido como union-find ou união-busca) é uma estrutura de dados que mantém o controle de um conjunto de elementos particionados em subconjuntos disjuntos.
- Ele fornece de forma eficiente operações para adicionar novos conjuntos, para mesclar conjuntos existentes e para determinar se os elementos estão no mesmo conjunto.
- Além de muitos outros usos, os disjoint-sets desempenham um papel fundamental no algoritmo de Kruskal de busca de árvore geradora mínima.

- As principais operações sobre um disjoint-set são duas:
  - União: união de dois subconjuntos em um único;
  - Busca: determina em qual subconjunto um elemento em particular está. Esta operação também pode ser utilizada para determinar se dois elementos estão em um mesmo subconjunto.
- Um disjoint-set é representado como uma floresta, onde cada um dos subconjuntos forma uma árvore, na qual cada elemento aponta para seu pai, e a raiz da subárvore contem o elemento representativo do subconjunto.


root
3
1
2
root
6
7
5
9
4
10
root
6
3
7
5
1
2
9
4
10

- Essa floresta pode ser representada por um vetor, onde o número da posição representa o vértice, e o conteúdo de cada posição contem o índice do pai. A posição do elemento que está na raiz da árvore contém seu próprio índice, permitindo identificar o elemento como raiz.
- Inicialmente cada elemento representa um subconjunto, sendo raiz da árvore de seu subconjunto, apontando para si mesmo
- Considere o conjunto de valores a seguir:

- Inicialmente cada elemento aponta para si mesmo e o conteúdo do vetor de "pais" é:

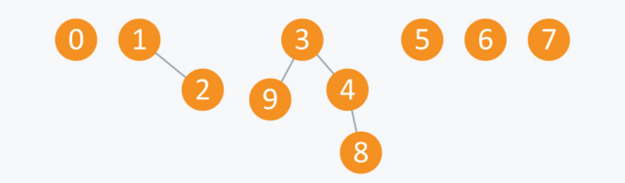


- Suponha que se faça a união dos conjuntos 1 e 2. Isso pode ser implementado fazendo com que o pai do conjunto 2 seja o nodo 1.

- E o vetor resultante após essa operação será:

- Supondo que se faça em seguida as operações união(4,3), união(8,4) e união(9,3),os subconjuntos resultantes serão:

- E o vetor resultante será:



- A função abaixo recebe o índice de um elemento e retorna o índice do elemento representativo de seu subconjunto, atualizando, no caminho, a informação do pai de cada nodo no caminho para a raiz:

```
int find(int x)
{
if (pai[x]!=x)
    pai[x]=find(pai[x]);
return pai[x];
}
```

- E o código abaixo efetua a união dos subconjuntos dos elementos x e y. Se eles já fazem parte do mesmo subconjunto, a função não faz nada:

```
void union(int x, int y)
   {
int xset = find(x);
    int yset = find(y);
    if(xset==yset) return;
    pai[xset]=yset;
   }
```

- A função anterior pode resultar em árvores bem desbalanceadas. Uma forma de melhorar o desempenho é armazenar junto à raiz da subárvore a altura da mesma, e "pendurar"a árvore mais baixa na árvore mais alta. Esse algoritmo é chamado de union-rank ("rank"é a altura da subárvore).
- O código do union-rank é mostrado na lâmina seguinte. Nele, cada elemento contem o índice do pai, e naqueles elementos que são raízes de árvores, o rank contem a altura da árvore:

```
typedef struct {int pai;int rank;} tnodo;
tnodo ln[100001];
void union_rank(int x, int y)
   {
int xset = find(x);
    int yset = find(y);
    if(xset==yset) return;
    if (ln[xset].rank<ln[yset].rank)
        ln[xset].pai=yset;
    else
        ln[yset].pai=xset;
    if (ln[xset].rank==ln[yset].rank)
        (ln[xset].rank)++;
    }
```

A STL (Standard Templates Library) do C++

- A **STL** é uma biblioteca que contem um conjunto de classes úteis para implementar alguns tipos abstratos de dados. Entre essas classes estão **containers**, que são tipos estruturados com métodos para executar diversas operações sobre eles. A STL ofere 5 containers básicos, 3 adaptadores de containers e alguns containers associativos, como dicionários e conjuntos.
- Vantagens do uso dos containers é que simplificam a codificação e algumas operações sobre eles são implementadas de forma muito eficiente.
- Para usar os componentes da stl é necessário colocar no início do código o comando "using namespace std;" para informar ao compilador que serão usadas funções ou classes da biblioteca padrão.
- Ou colocar "std::" em cada uso de elemento da biblioteca.

- Os containers se diferenciam pela estrutura sobre a qual são implementados (vetor, lista encadeada, lista duplamente encadeada) e as operações que podem ser executadas sobre eles, definidas pelos métodos que oferecem.
- Os 5 containers básicos são:
    - vector
    - list
    - deque
    - arrays (a partir do C++11)
    - forward_list (a partir do C++11)

- Além dos tipos básicos, a STL provê também adaptadores de containers (container adaptors), implementados sobre outros containers, oferecendo operações adicionais.
- São eles:
  - Stack (pilha)
  - Queue (fila)
  - Priority Queue (fila de prioridade)

vector

- O vector é o container mais eficiente em relação a acesso a memória.
- Ele é implementado sobre uma área contínua de memória e permite acesso direto aos elementos, como um vetor, e inserção e remoção no final.
- A declaração de um vector é:
  - **vector** <tipo> nome;
- Ex:
  - **vector** <int> var1, var2;
- Para utilizar um vector é necessário colocar #include <vector>

- O acesso aos elementos de um vector pode ser feito diretamente como se fosse um vetor (ex: vet[i]), e nesse caso não é feita nenhuma verificação se o elemento existe no vector ou não.
- Ou pode ser feito pelo método at(), e no caso de ser feito um acesso a um elemento ainda não inserido é gerada uma exceção.

- Alguns dos principais métodos do vector são:
  - front() - devolve o valor do primeiro elemento. Se estiver vazio, resulta um comportamento indefinido.
  - back() - devolve o valor do último elemento
  - empty() - testa se o container está vazio
  - push_back() - insere um elemento no final
  - pop_back() - remove um elemento do final
  - assign(int **n**, int **v**) - preenche **n** posições do vetor com o valor **v**

Iteradores

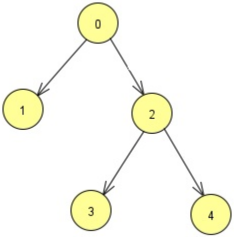
- Os containers possuem também a figura dos **iteradores**, que são objetos utilizados para percorrer sequencialmente os elementos de um container.
- A declaração de um iterador de nome **it** para um vector é feita por:
  - **vector**<int>::iterator it;
- em que o operador :: é usado para referir um elemento de uma classe (no caso, a classe vect<int>.

- e os métodos begin() e end() retornam iteradores apontando para o primeiro elemento do vector, e para a posição seguinte ao último elemento do vector. Assim, um uso comum para um iterador sobre um vector chamado vec seria:

  for (it = vec.begin(); it!=vec.end(); ++it)
- E o acesso ao elemento apontado pelo iterador é feito com o operador *, como se fosse um pointer.

- O código abaixo cria uma lista de adjacências para o grafo à esquerda:

```
#include <vector>
// declara vetor de
// listas de adjacências
using namespace std;
vector <int> adj[5];
int main()
    {
    adj[0].push_back(1);
    adj[0].push_back(2);
    adj[2].push_back(3);
    adj[2].push_back(4);
    }
```



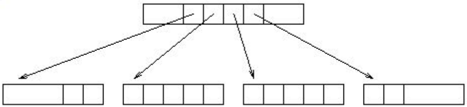
- E o código abaixo implementa uma DFS sobre o grafo:

```
int visit[5]={0};
vector <int> adj[5];

void dfs(int i)
    {
    printf("%d\n",i);
    visit[i]=1;
    vector <int>::iterator it;
    for (it=adj[i].begin();it!=adj[i].end();it++)
        if (!visit[*it])
            dfs(*it);
}
```

Deque

- Operações de inserção no início ou no meio não são muito eficientes em um vector, já que todos os elementos após o ponto de inserção devem ser deslocados.
- Uma estrutura mais eficiente para inserção nas duas pontas é o deque (double-ended queue), que é implementado na forma a seguir.



- Essa estrutura de memória permite a inserção e remoção de elementos em qualquer posição sem precisar deslocar todos os posteriores.

list

- O tipo list implementa uma lista duplamente encadeada com as operações de inclusão e remoção nas duas pontas.
- Ao contrário do vector e deque, ela não permite acesso indexado a todos os elementos.
- A declaração de uma variável do tipo list é:
  - **list** <tipo> nome;
- Ex:
  - **list** <int> lista1, lista2;
- Para utilizar listas é necessário colocar #include $< list >$

- As principais operações sobre uma lista são:
  - front() - Retorna o valor do primeiro elemento da lista.
  - back() - Retorna o valor do último elemento da lista.
  - push_front(g) - Adiciona um novo elemento 'g' no início da lista.
  - push_back(g) - Adiciona um novo elemento 'g' no final da lista.
  - pop_front() - Remove e descarta o primeiro elemento da lista.
  - pop_back() - Remove e descarta o último elemento da lista

Queue (fila)

- O container **Queue** implementa uma fila, com as operações padrão de filas, de inserção e remoção em ambas as pontas.
- A queue é implementada sobre um deque ou uma list.
- A criação de uma fila é feita como qualquer outro container:
  - **queue** <tipo> nome;
- Não esquecer do #include <queue>

- Os principais métodos para trabalhar com queues são:
  - empty() - retorna true se a fila está vazia
  - push(valor) - insere elemento no final da fila
  - front() - retorna o primeiro elemento elemento na fila
  - back() - retorna o último elemento da fila
  - pop() - remove (e descarta) o próximo elemento da fila
- O código a seguir implementa uma BFS sobre o grafo do exemplo anterior, representado por um vetor de listas de adjacências:

```
void bfs(int i)
  {
  int visit[5]={0};
  queue <int> fila;
  fila.push(i);
  visit[i]=1;
  while (!fila.empty())
    {
    int v=fila.front();
    printf("%d",v);
    fila.pop();
    vector <int>::iterator it;
    for (it=adj[v].begin();it!=adj[v].end();it++)
      if (!visit[*it])
        {
        visit[*it]=1;
        fila.push(*it);
        }
    } // do while
  } // da função
```

Stack (pilha)

- O container **Stack** implementa uma pilha, com as operações padrão de pilhas de inserção e remoção no topo.
- Os principais métodos para manipulação de pilhas são:
  - empty() - retorna true se a fila está vazia
  - push(val) - empilha um valor
  - top() - retorna o valor no topo da pilha
  - pop() - remove (e descarta) o valor no topo da pilha
- O código a seguir implementa uma ordenação topológica no grafo, usando o método da DFS

```
#include <stdio.h>
#include <stack>
#include <vector>
using namespace std;
stack<int> pilha;
int visit[5]={0};
vector <int> adj[5];
void dfs(int i)
  {
  visit[i]=1;
  vector <int >::iterator it;
  for (it=adj[i].begin();it!=adj[i].end();it++)
     if (!visit[*it])
        dfs(*it);
  pilha.push(i);
}
int main(){
  adj[0].push_back(1);
  adj[0].push_back(2);
  adj[2].push_back(3);
  adj[2].push_back(4);
  for (int i=0;i<5;i++)
     if (!visit[i])
        dfs(i);
  while (!pilha.empty()){
     printf("%d\n",pilha.top());
     pilha.pop();}
  }
```

Priority_queue (fila de prioridade)

- O container **priority_queue** implementa uma fila de prioridade. Essa estrutura se caracteriza por permitir identificar o maior (ou menor) elemento em tempo O(1), removê-lo em tempo O(log n) e inserir novos elementos em tempo O(log n).
- É útil em algoritmos como o de Prim ou o de Dijkstra, em que um passo importante, e custoso, do algoritmo é a identificação, a cada passo do elemento de menor valor em um conjunto (no caso, a distância de cada vértice ao vértice inicial, no Dijkstra, ou à sub-árvore já formada, no Prim)
- Ele está definido no módulo queue (include queue)

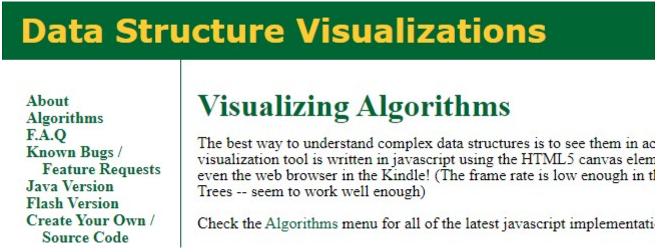
- A versão default implementa uma fila de prioridade max-heap, devolvendo a cada vez o maior valor da fila.
  - priority_queue <int> g;
- Os principais métodos para manipulação de filas de prioridade são :
  - empty() - retorna true se a fila está vazia
  - push(val) - empilha um valor
  - top() - retorna o valor no topo da pilha
  - pop() - remove (e descarta) o valor no topo da pilha
- Para implementar um min-heap, em que o MENOR elemento é mantido no topo, usa-se a sintaxe enigmática abaixo:
  - priority_queue $< int, vector < int >, greater < int >>$ nome;

- Containers podem conter tipos escalares, como int e float, mas também podem conter tipos estruturados, como structs.
- Quando o container deve armazenar pares de valores, pode-se utilizar o template pair $< tipo1, tipo2 >$ que agrupa dois valores como se fosse uma tupla.
- Ex: vector $< pair < int, int >>$ v;

- Na implementação do algoritmo de Dijkstra utilizando uma fila de prioridades para recuperar a cada passo o vértice mais próximo do vértice inicial, deve-se armazenar, na fila de prioridades, o número do vértice é a distância associada a ele, ou seja, cada posição da fila de prioridades conterá 2 valores, o número e o peso do vértice.
- Isso pode ser feito definindo um par de valores, pelo

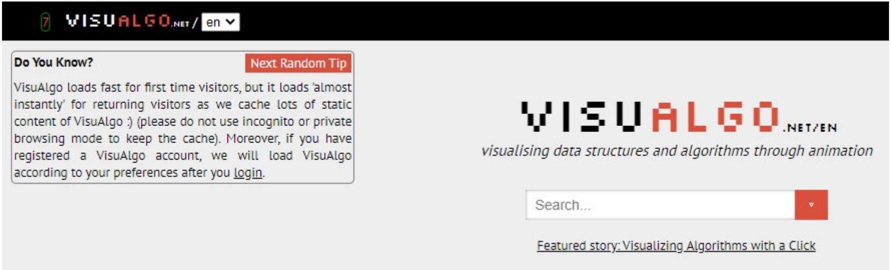
Recursos online

- O site Data Structures Visualization (Botão) apresenta animações para uma variedade grande de algoritmos e estruturas de dados, incluindo árvores, ordenação, heaps, programação dinâmica, backtracking, grafos e outros.



- Especificamente de grafos, o site contem animações para:
  - Breadth-First Search
  - Depth-First Search
  - Connected Components
  - Dijkstra's Shortest Path
  - Árvore geradora mínima por Prim e Kruskal
  - Topological Sort (Using Indegree array)
  - Topological Sort (Using DFS)
  - Floyd-Warshall (all pairs shortest paths)

- O site VisuAlgo (Botão) apresenta também animações para uma variedade grande de algoritmos e estruturas de dados.
- Um diferencial é que ele mostra um pseudo-código do algoritmo durante a animação, destacando a linha em execução a cada passo.



Lista de Problemas NP-Completos em Grafos

- Cobertura de vértices
- Caminho hamiltoniano
- Problema do caixeiro viajante (como problema de decisão)
- Isomorfismo de subgrafos
- Clique
- Conjunto independente máximo
- Conjunto dominante
- Coloração
- Lista mais completa ▸ Aqui