

**Universidade Federal do Pará**

**Instituto de Tecnologia**

**Faculdade de Engenharia da Computação**

**Teoria da Computação II**

**Aluno: Otavio Augusto**

**Matricula: 201206840012**

**1. Defina o que é um subgrupo.**

R: Um subgrupo de um grafo *G* é um grafo cujo conjunto de vértices é um subconjunto do conjunto de vértices *G* e o conjunto de arestas é um subconjunto do conjunto de arestas de *G*, ou seja, cuja relação de adjacência é um subconjunto de *G* restrita a esse subconjunto.

**2. Defina o que e um grafo bipartido.**

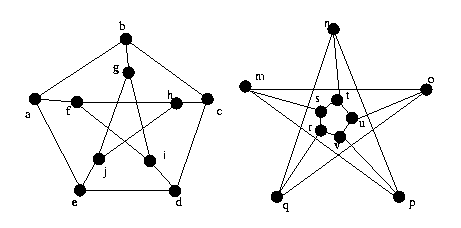
R: Um grafo bipartido ou biógrafo é um grafo cujos vértices podem ser divididos em dois conjuntos disjuntos *U* e *V* tais que toda aresta conecta um vértice em *U* a um vértice em *V*ou seja, *U* e *V* são conjuntos independentes. Equivalentemente, um grafo bipartido é um grafo que não contém qualquer ciclo de comprimento ímpar.

**3. Defina o que e um grafo conexo. E um desconexo?**

R: Um grafo G=(V, E) é conexo se existir um caminho entre qualquer par de vértices. Caso Contrário é desconexo – se há pelo menos um par de vértices que não está ligado a nenhuma cadeia (caminho).

**4. O que são grafos isomorfos? Desenhe um exemplo.**

R: Dois grafos G1(V1,E1) e G2(V2,E2) são ditos *isomorfos entre si* se existe uma correspondência entre os seus vértices e arestas de tal maneira que a relação de incidência seja preservada. Em outros termos, temos |V1|= |V2| e existe uma função unívoca f: V1-->V2, tal que (i,j) é elemento de E1 se e somente se (ƒ(i),f(j)) é elemento de E2.



**5. Defina o que e um grafo Hamiltoniano.**

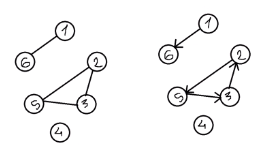
R: Um grafo G é dito ser hamiltoniano se existe um ciclo em G que contenha todos os seus vértices, sendo que cada vértice só aparece uma vez no ciclo. Este ciclo é chamado de ciclo hamiltoniano. Sendo assim. um grafo é hamiltoniano se ele contiver um ciclo hamiltoniano.

**6. Defina o que e um grafo Euleriano.**

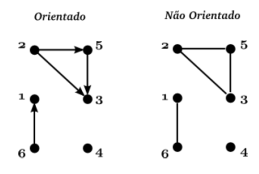
R: Um grafo G é dito ser euleriano se há um ciclo em G que contenha todas as suas arestas. Este ciclo é dito ser um ciclo euleriano. O grafo da figura ao lado, por exemplo, é euleriano já que ele contém o ciclo: (u1, u2, u3, u4, u5, u3, u1, u6, u2, u7, u3, u6, u7, u1), que é euleriano.

**7. Desenhe as versões não orientadas e orientadas do grafo G(V, E), onde V = {1, 2, 3, 4, 5, 6} e E = {(2, 5), (6, 1), (5, 3), (2, 3)}.**

R:



**8. Defina os grafos ilustrados abaixo.**



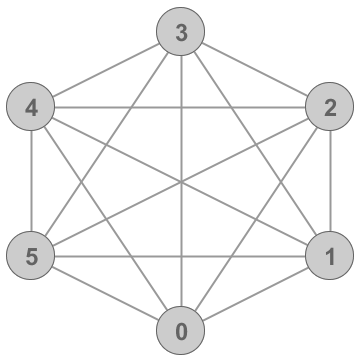
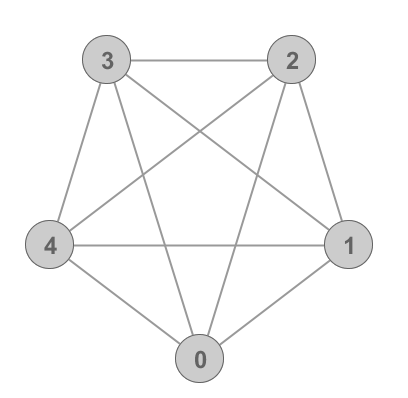
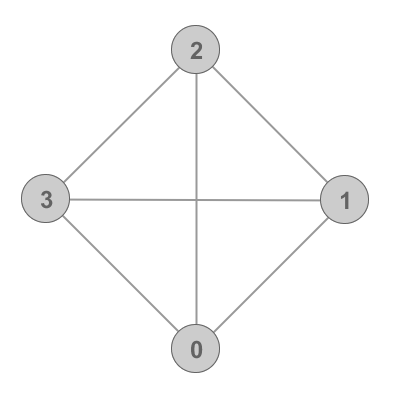
R: G(V,E) onde V = {1,2,3,4,5,6} e E = {(2,5), (2,3), (6,1)}

**9. Defina e desenhe os grafos não orientados completos com 4, 5 e 6 vértices.**

R: V = {0,1,2,3} e E = {(0,1), (0,2), (0,3), (1,2), (1,3), (2,3)}

V = {0,1,2,3,4} e E = {(0,1), (0,2), (0,3), (0,4) (1,2), (1,3), (1,4), (2,3), (2,4), (3,4)}

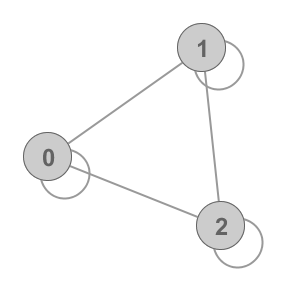
V = {0,1,2,3,4,5} e E = {(0,1), (0,2), (0,3), (0,4) , (0,5), (1,2), (1,3), (1,4), (1,5), (2,3), (2,4), (2,5), (3,4),(3,5),(4,5)}



**10. Dê um exemplo de um grafo em que cada vértice é adjacente a dois outros vértices e**

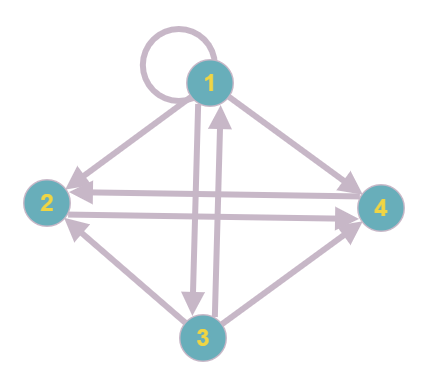
**cada aresta é adjacente a duas outras arestas.**

R:



**11. Quantas arestas tem um grafo com 3 vértices de grau 3 e um vértice de grau 5?**

R:



**12. Em um grafo com *n* vértices e *m* arestas, qual a soma dos graus de todos os vértices?   
Observe que, em um grafo não orientado, cada aresta soma 1 ao grau de cada vértice   
em que incide e cada aresta incide somente sobre dois vértices. Em um grafo orientado,   
por outro lado, cada aresta soma 1 ao grau de cada vértice em que incide, porém, cada   
aresta incide somente sobre um vértice.**

R: A soma dos graus de todos os vértices de um grafo G é duas vezes o número de suas arestas.

**13. Sabendo que cada vértice tem pelo menos grau 3, qual o maior número possível de  
vértices em um grafo com 35 arestas? Lembre-se que a soma dos graus dos vértices é  
igual a duas vezes o numero de arestas. Se cada aresta liga dois vértices teríamos 70  
vértices de grau 1.**

R:

Vertices \* Grau = 2 \* Número de Arrestas

3\*V = 2\*35

V = 70/3

V = 23

Maior quantidade de vértices possíveis é 23.

**14. Quantas arestas possui um grafo completo com n vertices? E um grafo orientado completo com n vertices?**

R:

Completo não orientado

Completo orientado

**15. Faça uma função para obter todos os nós adjacentes (vizinhos) a um nó do grafo, dado que o grafo é representado por uma matriz de adjacências.**

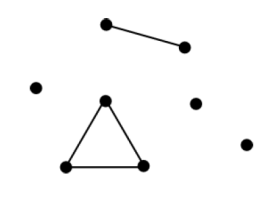
R: Utiliza uma matriz N x N para armazenar o grafo, onde N é o número de vértices. Uma aresta é representada por uma marca na posição (i , j) da matriz.



**16. Faça uma função para obter todos os nós adjacentes (vizinhos) a um nó do grafo, dado que o grafo e representado por uma lista de adjacências.**

R: Um grafo contendo N vértices utiliza um array de ponteiros de tamanho N para armazenar os vértices do grafo. Para cada vértice é criada uma lista de arestas, onde cada posição da lista armazena o índice do vértice a qual aquele vértice se conecta.

**17. Quantas componentes conexas tem o seguinte grafo?**



R: Cinco vértices estão conectados.

**18. Descreva com suas palavras o funcionamento de um algoritmo de busca em profundidade. De dois exemplos de aplicação real desse algoritmo.**

R: Partindo de um vértice inicial, a busca explora o máximo possível cada um dos vizinhos de um vértice antes de retroceder (*backtracking*). O mecanismo de *backtracking* faz com que a busca retorna pelo mesmo caminho percorrido com o objetivo de encontrar um caminho alternativo*.*

* Achar componentes conectados.
* Achar componentes fortemente conectados.

**19. Descreva com suas palavras o funcionamento de um algoritmo de busca em largura. De dois exemplos de aplicação real desse algoritmo.**

R: Partindo de um vértice inicial, a busca explora todos os vizinhos de um vértice. Em seguida, para cada vértice vizinho, ela repete esse processo, visitando os vértices ainda inexplorados. Em outras palavras, esse tipo de busca se inicia em um vértice e então visita todos os seus vizinhos antes de se aprofundar na busca. Esse processo continua até que o alvo da busca seja encontrado ou não existam mais vértices a serem visitados.

* Achar todos os nódulos conectados a apenas um componente.
* Achar o menor caminho entre um nó raiz e os outros nós do grafo.

**20. Descreva com suas palavras o funcionamento de um algoritmo de busca pelo menor  
caminho. De dois exemplos de aplicação real desse algoritmo.**

R: O algoritmo de busca de menor caminho escolhido foi o Dijkstra. O algoritmo considera um conjunto S de menores caminhos, iniciado com um vértice inicial I. A cada passo do algoritmo busca-se nas adjacências dos vértices pertencentes a S aquele vértice com menor distância relativa a I e adiciona-o a S e, então, repetindo os passos até que todos os vértices alcançáveis por I estejam em S. Arestas que ligam vértices já pertencentes a S são desconsideradas.

**21. Dado o dígrafo G = (V, E) sendo V = {M, N, O, P, Q, R, S} e E = {(M, S), (N, O), (P, R), (N, S), (O, M), (N, Q), (O, M), (P, P), (S, M), (O, N), (S, M), (N, R), (P, M), (M, S)}**

**a) Especifique, caso exista, um caminho simples desde o vértice M até o vértice S.**

R: M-S

**b) Especifique, caso exista, um ciclo simples, envolvendo pelo menos 4 nós.**

R: M-O-N-S

**c) O dígrafo é conexo ou não conexo?**

R: É conexo.

**d) Qual o grau dos vértices N e R.**

R: Vértice N possui grau 4 de saída e grau 1 de entrada. Vértice R possui grau 0 de saída e grau 2 de entrada.

**e) Represente o dígrafo utilizando representação por lista de adjacência.**

R:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| M | > | S |  |  |  |  |  |  |
| N | > | O | > | Q | > | R | > | S |
| O | > | M | > | N | > | R |  |  |
| P | > | M | > | P |  |  |  |  |
| Q |  |  |  |  |  |  |  |  |
| R |  |  |  |  |  |  |  |  |
| S | > | M |  |  |  |  |  |  |

**f) Represente o dígrafo utilizando representação por matriz de adjacência.**

R:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | M | N | O | P | Q | R | S |
| M | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| N | 0 | 0 | 1 | 0 | 1 | 1 | 1 |
| O | 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| P | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| Q | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| S | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

**22. Implemente um algoritmo para verificar se um grafo é acíclico utilizando o algoritmo de busca em profundidade.**

R:

Começo

Verifica todos os nós dentro do grafo

Guarda informação do nó dentro de uma estrutura (Array)

Guarda nó inicial

Guarda informação do inicio

Travessia até o nó final

Verificando se os nós do Array são os mesmo

Verifica se ao chegar no ultimo nó esse nó é o inicial

Se sim

É um grafo ciclico

Se não

É um grafo áciclico

Fim

**23. Escreva uma versão não recursiva do algoritmo de busca em profundidade.**

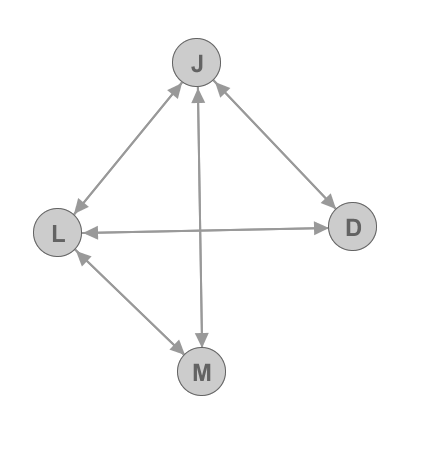
Caso o nó filho não possua informação do seu pai, não há como fazer uma busca em profundidade de forma iterativa (Não recursiva) sem tal conhecimento. Porém caso, o filho possua informação da profundidade e do seu pai, basta o algoritmo implementar um método para continuar buscando enquanto o seu próximo ponteiro não seja um valor nulo.

**24. Exemplifique com algumas situações de uso dos grafos e justifique.**

R: Os grafos são utilizados em várias estruturadas de dados presentes em várias aplicações. Por exemplo, tabelas hash, listas encadeadas, arvores binárias, listas adjacentes e assim sucessivamente. Sendo assim, os grafos são indispensáveis para realizar a hierarquização, alocação e organização de dados presentes em uma aplicação qualquer. Caso não houvesse, seria extremamente difícil implementar tais estruturas e assim consequentemente as aplicações que utilizamos nos dias de hoje.

**25. Os Turistas Jensen, Leuzingner, Dufour e Medeiros se encontram em um bar de Paris e começam a conversar. As línguas disponíveis são o inglês, o francês, o português e o alemão. Jensen fala todas. Leuzingner não fala apenas o português. Dufour fala francês e alemão. Medeiros fala inglês e português. Represente por meio de um dígrafo todas as possibilidades de um deles dirigir a palavra a outro, sendo compreendido.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Turista** | **Inglês** | **Francês** | **Português** | **Alemão** |
| **Jensen** | **1** | **1** | **1** | **1** |
| **Leuzingner** | **1** | **1** | **0** | **1** |
| **Dufour** | **0** | **1** | **0** | **1** |
| **Medeiros** | **1** | **0** | **1** | **0** |



**26. Você usaria uma lista de adjacência ou uma matriz de adjacência em cada um dos casos abaixo? Justifique sua escolha.**

**a) O grafo tem 10.000 vértices e 20.000 arestas, e importante usar tão pouco espaço quanto possível.**

R: Lista adjacente, já que não possuímos tantos vértices.

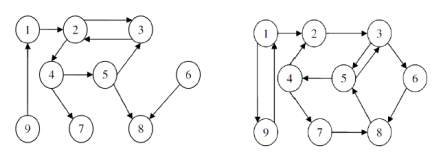
**b) O grafo tem 10.000 vértices e 20.000.000 arestas, e importante usar tão pouco espaço quanto possível.**

R: Matriz adjacente, já que temos muitos vértices.

**c) Você deve ter a aresta adjacente tão rápido quanto possível, sem se importar quanto  
espaço você usa.**

R: Lista adjacente.

**27. Dado os grafos abaixo, mostre o resultado da busca em largura e em profundidade.**



Busca em largura

Primeiro grafo: 1-> 2-> 3-> 4-> 5-> 8-> 7

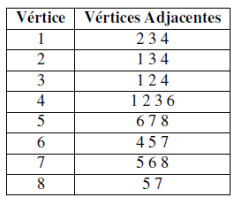
Segundo grafo: 1-> 2-> 3-> 5-> 4-> 7-> 8-> 6->9

Busca em profundidade

Primeiro grafo: 1-> 2 -> 3 -> 4-> 7 -> 5 -> 8

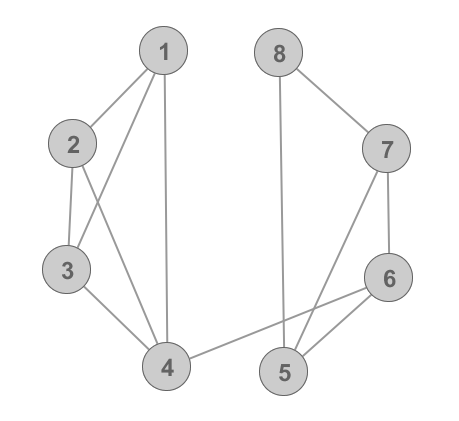
Segundo grafo: 1-> 9 -> 2 -> 3 -> 5 -> 6 -> 4 -> 7 -> 8

**28. Seja um grafo G cujos vértices são os inteiros de 1 a 8 e os vértices adjacentes a cada vértice são dados pela tabela abaixo:**



**a) Desenhe o grafo G.**

R:



**b) Represente o grafo por meio de uma matriz de adjacência.**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 2 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 3 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 4 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 6 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| 7 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |
| 8 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |

**29. Dada a matriz de adjacências de uma grafo de *N* vértices, faca um algoritmo que determine se esse grafo e orientado ou não orientado.**

R: Verificando-se a existência de um numero não nulo em xi,yi e não em yi, xi mostra que o grafo é possivelmente orientado, sendo assim conseguimos estabelecer se um grafo é ou não orientado.

**30. Por que, em uma matriz de adjacências, verificar a existência de uma aresta e *O* (1).**

R: Uma matriz nada mais é que um vetor bidimensional como o vetor possui notação O(1), a matriz possui também já que a forma de indexação é praticamente a mesma, mudando somente a adição de um índice a mais.

**31. Qual método usa mais espaço, listas de adjacência ou matriz de adjacência e por quê?**

R: Listas de adjacência, porque dependendo do número de arestas, caso exista duplicidade de arestas a mesma ira se repetir, sendo assim é necessário a criação de uma instancia para essa aresta, já na matriz só ocorre uma vez.

**32. Escreva um algoritmo que verifique se dois grafos G1 e G2 não são isomorfos com base   
no numero de vértices e arestas e, também, comparando a lista ordenada dos graus de  
seus vértices.**

Manuel Blum e Sampath Kannan ([1995](https://en.wikipedia.org/wiki/Graph_isomorphism_problem#CITEREFBlumKannan1995)) mostraram um programa de verificação para o isomorfismo de grafos. Suponha que P é dito como um procedimento de tempo polinomial que verifica se dois gráficos são isomorfos, mas não confiáveis. Para verificar se G e H são isomorfos:

1. Pergunte a P se G e H são isomorfos.
   1. Se a resposta for "sim":
      1. Tentar construir um isomorfismo usando P como sub-rotina. Marcando um vértice u em G e v em H e modificar os gráficos para torná-los distintos (com uma pequena mudança local). Perguntar a P se os grafos modificados são isomorfos. Se não, mudar v para um vértice diferente. Continue procurando.
      2. Ou o isomorfismo será encontrado (e pode ser verificado) ou P irá contradizer a si mesmo..
2. Se a resposta for "não":
   1. Execute o seguinte 100 vezes. Escolha aleatoriamente G ou H, e permutar aleatoriamente seus vértices. Perguntar a P se o grafo é isomorfo a G e H. (como no protocolo AM para o não isomorfismo do grafo).
   2. Se algum dos testes falharem, julga P como um programa inválido. Caso contrário, responda "não".

**33. Escreva um algoritmo que recebe um caminho e verifica se ele e um ciclo.**

R:

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | para cada vértice  r  faça  seja  T  o território de  r  para cada  v  em  T  faça  se existe um arco da forma  (v,r)  então devolva  1  devolva 0 |

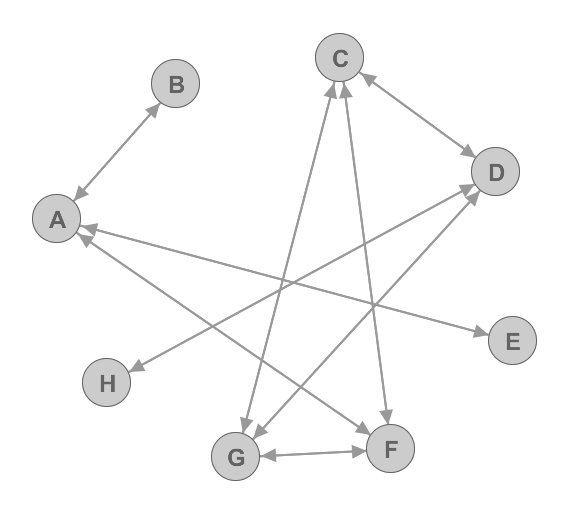
**34. Escreva um algoritmo que recebe um caminho e verifica se ele e um ciclo simples**

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | para cada vértice  r  faça  seja  T  o território de  r  para cada  v  em  T  faça  se existe um arco da forma  (v,r)  então devolva  1  devolva 0 |

**35. Considere a seguinte representação de um grafo com oito vértices e nove arestas usando listas de adjacências**

**A: E F B  
B: A  
C: G D F  
D: H G C  
E: A  
F: A G C  
G: D F C  
H: D**

**Mostre o resultado da busca em largura e em profundidade a partir do vértice A. Mostre também a distancia de cada vértice ao vértice A.**



A -> B -> C - > D -> G -> F -> H -> E

A = 0

B = 1

C = 2

D = 3

E = 1

F = 1

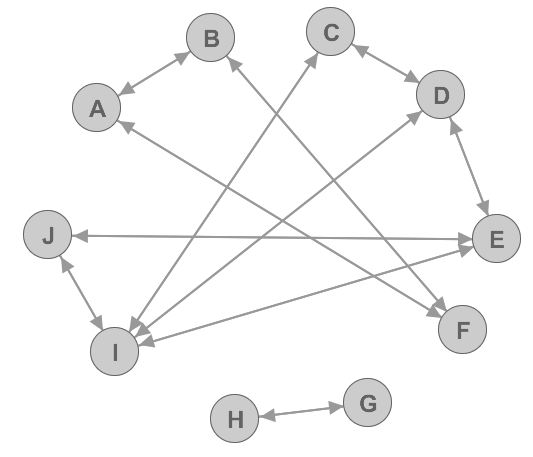
G = 2

H = 4

**36. Considere a seguinte representação de um grafo usando listas de adjacências:**

**A: F B  
B: A F  
C: D I  
D: E C I  
E: D J I  
F: A B  
G: H  
H: G  
I: J E C D  
J: I E**

**Obtenha os componentes conectados de um grafo usando o algoritmo de busca em  
profundidade.**



A - >B -> F