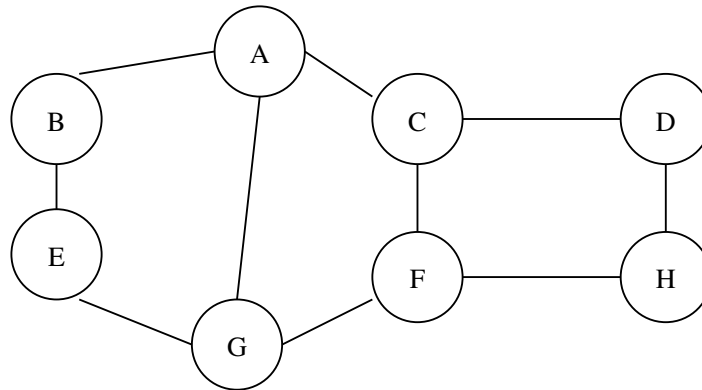


Lista de Exercícios – Grafos

1. Considere o grafo abaixo:



É solicitado o seguinte:

- Todos os caminhos não cíclicos de a até h
- Todos os caminhos não cíclicos de c até e
- Todos os caminhos não cíclicos de b até f

2. Ainda considerando a figura do exercício 1, ache os todos os nós adjacentes para os nós a, f, g.

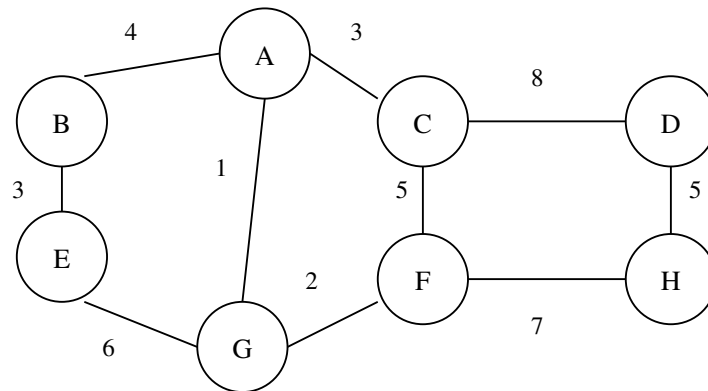
3. Dê o resultado do passeio com o algoritmo de profundidade (*depth-first*) sobre o gráfico do exercício 1, iniciando do vértice A

4. Dê o resultado do passeio com o algoritmo de largura (*breadth-first*) sobre o gráfico do exercício 1, iniciando do vértice A

5. Desenhe a matriz de adjacência do grafo do exercício 1

6. Desenhe a lista de adjacência do grafo do exercício 1

7. Ache o caminho mais curto entre o nó A e todos os outros nós do grafo abaixo:



8. Dê a matriz adjacência do grafo do exercício 7

9. Dê a lista de adjacência do grafo do exercício 7

10. Dada a matriz de adjacência abaixo, faça o respectivo desenho do grafo orientado.

	A	B	C	D	E	F
A	0	3	4	0	2	1
B	0	0	2	0	0	3
C	0	0	0	2	6	1
D	2	6	1	0	1	2
E	0	0	0	0	0	3
F	0	0	0	0	0	0

11. Escreva um algoritmo que ache o grau de um nó usando a representação de lista de adjacência.

12. Um grafo pode ser utilizado para mostrar relacionamentos. Por exemplo, dada a seguinte lista de pessoas pertencentes ao mesmo clube (vértices) e suas relações de amizade (arestas), determine:

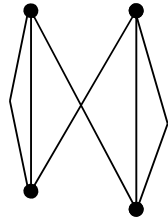
```

Pessoas= { Jorge, Jaime, Jose, Francisco, Frederico, João, Suzana }
Amizades  = { (Jorge, Jaime),          (Francisco, Frederico),
              (Jorge, João),           (Jaime, Frederico),
              (Jaime, Francisco),       (Jaime, Suzana),
              (Suzana, Francisco) }
  
```

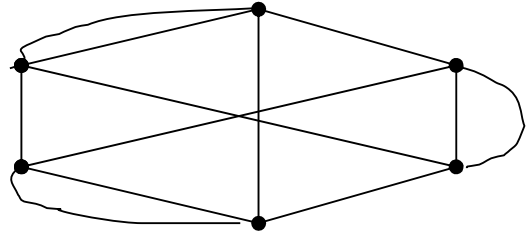
- Todos os amigos de João
- Todos os amigos de Suzana
- Todos os amigos de Jaime

13. Considerando o teorema dos caminhos Eulerianos, responda as seguintes perguntas:

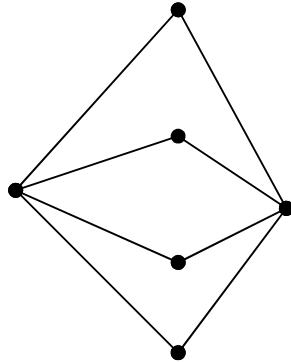
- 1) Existe o caminho Euleriano?
- 2) Se a resposta para a pergunta anterior for positiva, encontre um caminho Euleriano



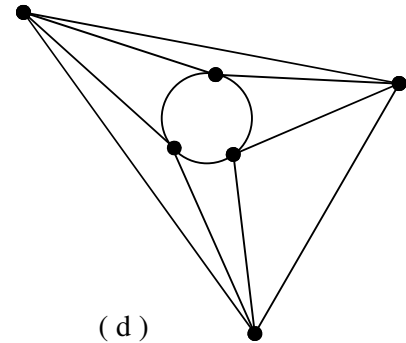
(a)



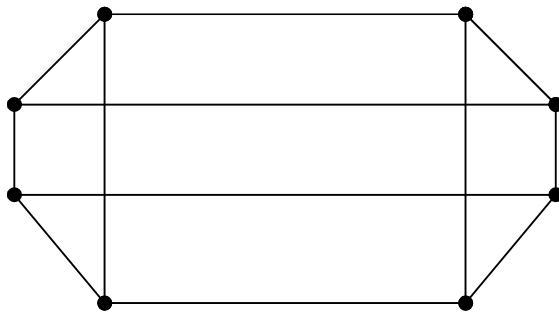
(b)



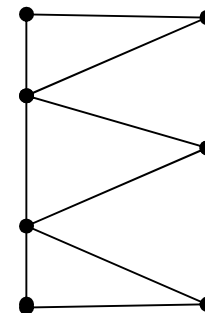
(c)



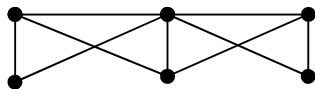
(d)



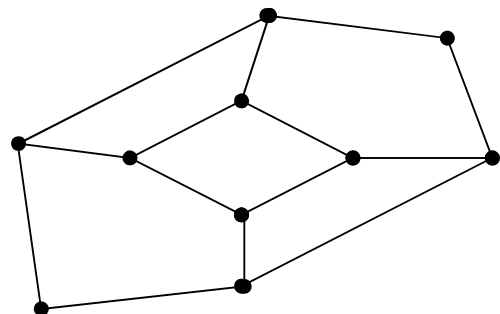
(e)



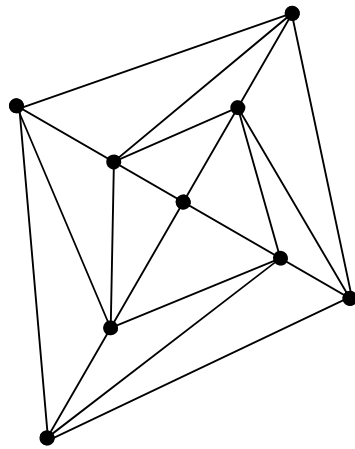
(f)



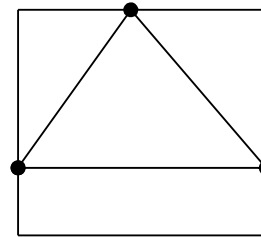
(g)



(h)

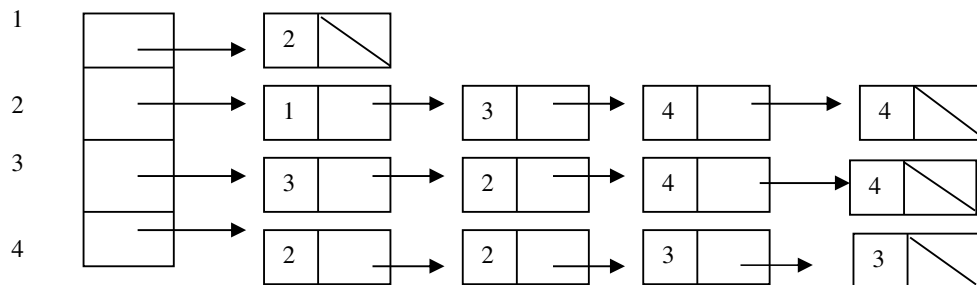


(i)

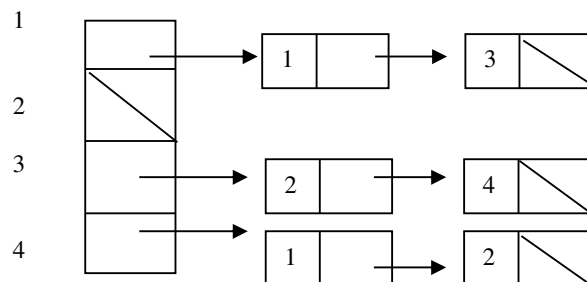


(j)

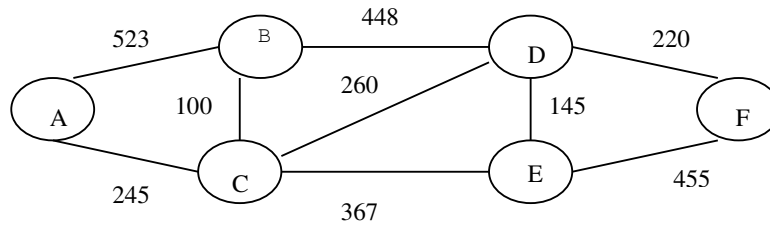
14. Desenhe o grafo não-direcionado representado pela lista de adjacências da figura a seguir



15. Desenhe o grafo direcionado representado pela lista de adjacências da figura dada



16. Abaixo está uma rede representando as distâncias entre várias cidades. Solicita-se que você faça a matriz e lista de adjacência dessa rede.



17. Desenvolva um algoritmo capaz de verificar se um grafo é Euleriano.

18. Desenvolva um algoritmo capaz de encontrar um caminho Euleriano em um grafo.

19. Sabe-se que encontrar um caminho Hamiltoniano envolve grande esforço computacional. Ignorando essa limitação de tempo, desenvolva um algoritmo que procura um caminho Hamiltoniano em um grafo.

20. Escreva uma função para encontrar o caminho mínimo entre dois pontos.

21. Uma maneira interessante de encontrar um caminho mínimo é realizar uma busca bidirecional. Parte-se do início até o fim e, ao mesmo tempo, do final até o início. Quando as duas buscas encontram um nó em comum, pode-se traçar o caminho entre elas. Escreva uma função (ou conjunto de funções) capaz de realizar a busca bidirecional.

22. Considere o enunciado do problema lógico “Ponte Escura” a seguir:

Você precisa atravessar 5 pessoas em uma ponte escura. Para atravessar, é preciso utilizar uma lanterna. A lanterna tem bateria para 30 minutos e cada pessoa leva um determinado tempo (descrito a seguir) para atravessar a ponte. As duas pessoas que atravessam a ponte sempre gastam no tempo da pessoa mais devagar. As pessoas são:

1. Corredor (1 minuto)
2. Moleque (3 minutos)
3. Programador (6 minutos)
4. Dama (8 minutos)
5. Idosa (12 minutos)

Você pretende resolver este problema com um algoritmo de busca. Para isso:

- a) Modele o nó que representaria um estado deste problema.
- b) Crie uma função que retorna todas as ações possíveis dado um estado
- c) Crie uma função sucessos que recebe um estado e uma ação e retorna um novo estado
- d) Crie uma função para verificar a parada da busca

23. Considere um problema de lógica simples, de enunciado a seguir:

“Um agricultor está com um problema!

Precisa atravessar o rio para transportar sua carga que são dois fardos de capim e um carneiro. O problema é que ele só pode transportar uma coisa de cada vez no seu pequeno

barco. Pior ainda, se ele deixar o carneiro e o capim juntos, o carneiro vai comer o mesmo.

Como fazer então para que ele leve sua carga para a outra margem sem prejuízo algum? "

Você decidiu montar um algoritmo de busca para resolver este problema.

- a) Modele um nó capaz de representar a situação deste problema em um dado instante.
- b) Crie um método que retorna todas as ações possíveis dado um nó.
- c) Crie um método que retorna um nó sucessor dada uma ação possível e o nó atual.