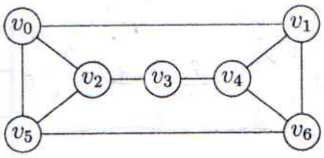
1,0

1. (1,2 ponto) Para cada uma das seguintes afirmações sobre o grafo abaixo, preencha a lacuna com **V** para indicar que a afirmação é verdadeira ou com **F** para indicar que a afirmação é falsa.
   (a) F O grau máximo é 3.
   (b) F A distância entre v_5 e v_4 é 3.
   (c) V O comprimento máximo de um caminho é 6.
   (d) V O subgrafo induzido por $S = \{v_0, v_1, v_2, v_4\}$ tem 3 arestas.
   (e) F O grafo é bipartido.



2. (1,8 ponto) Um alcano é uma molécula química C_nH_{2n+2}, com $n \geq 1$, formada por n átomos de carbono e $2n+2$ átomos de hidrogênio tendo as seguintes características:
   - cada átomo de hidrogênio está ligado a exatamente um outro átomo e este é um átomo de carbono;
   - cada átomo de carbono está ligado a exatamente quatro outros átomos e estes são átomos de hidrogênio ou de carbono.

   Por exemplo, os alcanos C_1H_4 e C_2H_6 são mostrados nas Figuras 1 e 2.

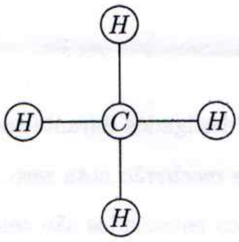


Figura 1: Alcano C_1H_4

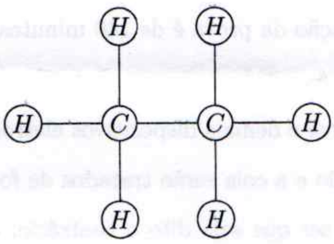


Figura 2: Alcano C_2H_6

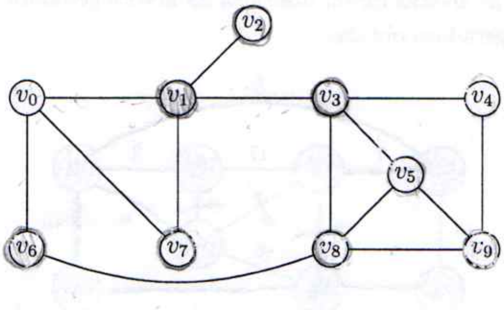
Podemos notar que um alcano corresponde a um grafo simples onde os vértices representam os átomos e as arestas representam as ligações entre os átomos. Baseado nisto, faça o que é pedido a seguir:

(a) Apresente todos os possíveis grafos que correspondem a um alcano C_5H_{12} e que não são isomorfos entre si. Não é necessário provar que os grafos não são isomorfos.

(b) De forma precisa e clara, explique por que qualquer grafo que corresponde a um alcano C_nH_{2n+2} é uma árvore.

3. (2,8 pontos) Faça o que é pedido a seguir:

(a) Considere o algoritmo de busca em profundidade executado para o grafo abaixo começando pelo **vértice v_0**. Suponha que quando os vizinhos de um vértice são inspecionados para visitação, estes vizinhos são considerados em **ordem crescente dos seus índices**. Indique a sequência de vértices visitados na busca e apresente a árvore que representa a dinâmica da busca.



(b) Considere o algoritmo de busca em largura executado para o grafo acima começando pelo **vértice v_0**. Suponha que quando os vizinhos de um vértice são inspecionados para visitação, estes vizinhos são considerados em **ordem crescente dos seus índices**. Indique a sequência de vértices visitados na busca e apresente a árvore que representa a dinâmica da busca.

O Método 1 abaixo é um método de um objeto que representa um grafo. O objeto contém um atributo `num_vertices_`, que armazena o número de vértices do grafo, e um atributo `matriz_adj_`, que armazena a representação do grafo como uma matriz de adjacências.

(c) O Método 1 implementa um algoritmo de busca no grafo. Indique que algoritmo de busca é este.

(d) Altere o Método 1 para funcionar da seguinte maneira: O método deve receber por parâmetro um vetor `pai`. Ao fim da realização da busca, o vetor `pai` deve indicar, para cada vértice alcançado pela busca, o índice do seu pai na árvore que representa a dinâmica da busca.

Método 1

```
void Grafo::busca(int v, int marcado[]) {
    marcado[v] = 1;
    for (int u = 0; u < num_vertices_; u++)
        if (matriz_adj_[v][u] != 0)
            if (marcado[u] == 0)
                busca_prof(u, marcado);
}
```

4. (2,2 pontos) Faça o que é pedido a seguir:

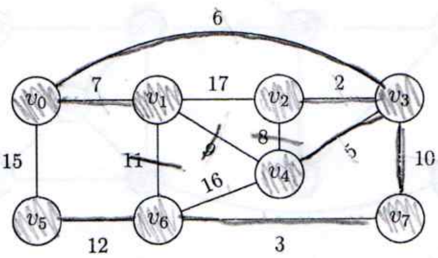
   (a) Considere os algoritmos de Prim e de Kruskal para encontrar uma árvore geradora de peso mínimo de um grafo. Execute um destes algoritmos para o grafo abaixo e informe o seguinte:

      i. O algoritmo executado;

      ii. A árvore geradora obtida;

      iii. A **ordem** em que as arestas foram inseridas na árvore geradora obtida;

      iv. O peso da árvore geradora obtida.



   (b) O Algoritmo 2 abaixo é outro algoritmo que tem como objetivo encontrar uma árvore geradora de peso mínimo de um grafo. Este algoritmo recebe como entrada um grafo conexo G com pesos nas arestas e retorna uma árvore T. Responda: Este algoritmo funciona corretamente? Se sim, explique por que e dê um exemplo que ilustra o funcionamento do algoritmo; se não, explique por que e dê um exemplo onde o algoritmo não atinge o seu objetivo.

Algoritmo 2

EncontraArvGerPesMin(G conexo)

1: Se $|V(G)| == 0$:

2:     Retorne $(\emptyset, \emptyset)$ // Retorne uma árvore vazia

3: Particione $V(G)$ em dois conjuntos de vértices V_1 e V_2 tal que os subgrafos induzidos $G[V_1]$ e $G[V_2]$ são conexos

4: T_1 = EncontraArvGerPesMin($G[V_1]$)

5: T_2 = EncontraArvGerPesMin($G[V_2]$)

6: Encontre as arestas uv de G tal que $u \in V_1$ e $v \in V_2$

7: Entre as arestas encontradas no Passo 6, selecione uma aresta de peso mínimo

8: Construa a árvore T formada por T_1, T_2 e a aresta selecionada no Passo 7

9: Retorne T

5. (2,0 pontos) O historiador que trabalha na prefeitura de uma pequena cidade foi encarregado de levantar os dados das famílias que já moraram ou ainda moram lá. Depois de passar alguns anos catalogando certidões e outros documentos, o historiador decidiu organizar os dados de maneira simplificada, através de uma longa lista de registros do seguinte tipo: *Morador(a) 1 é familiar do(a) Morador(a) 2.*

   Agora, o prefeito atribuiu ao historiador a seguinte tarefa: Determinar o maior número de pessoas que já moraram ou ainda moram na cidade e que pertencem à mesma família. Baseado nisto, responda aos itens abaixo:

   (a) Como o historiador pode construir um grafo para utilizar na realização da tarefa acima? Descreva o que representam os vértices e as arestas do grafo.

   (b) Entre os algoritmos estudados nesta disciplina, qual algoritmo pode ser utilizado na realização da tarefa acima? Descreva de forma precisa e clara como isto pode ser feito.