Questão 1: Decidiu-se utilizar três classes para representar o grafo. Uma classe para o próprio grafo, uma classe para os vértices e outra para as arestas. O grafo possui uma lista de vértices e uma lista de arestas, enquanto a aresta possui uma lista com dois vértices e o seu peso. Essa representação é mais compatível com a orientação a objetos e ao alto nível da linguagem python, além de ser mais fácil de ser implementada, porém, é mais lenta do que a utilização de uma matriz de adjacências, por exemplo. Além dos métodos requisitados pelo enunciado da questão, foram adicionados os métodos inserirVertice(v), que insere um vértice v no grafo; inserirAresta(u, v, peso), que insere uma aresta (u, v) com respectivo peso; obterVertice(i), que retorna o objeto vértice com índice i; e arestasVizinhos(v), que retorna as arestas ligadas a v.

Questão 2: No algoritmo de busca em largura são produzidas três listas que armazenam as informações sobre os vértices já encontrados, a distância da origem e o ancestral direto do vértice na árvore gerada, sendo que esta última lista não é utilizada no contexto deste código. Para fazer a busca, uma estrutura de fila é usada, enfileirando os vértices vizinhos do vértice atual, extraído dessa mesma fila. Para cada vértice processado e ainda não conhecido, se atualiza seu status para conhecido, incrementa a distância com base no vértice ancestral e o define como ancestral direto, para então adicionar esse vértice que está sendo processado na fila. Isso acontece até que todos os vértices alcançáveis sejam conhecidos e, portanto, a fila fique vazia.

Questão 3: O algoritmo do ciclo euleriano segue o mesmo padrão do algoritmo das anotações da disciplina, há três funções: CicloEuleriano, que recebe o grafo, chama a função Hierholzer e imprime os resultados; Hierholzer, que, a partir do grafo, cria uma lista de arestas, seleciona um vértice inicial, que decidiu-se escolher o primeiro da lista de vértices do grafo, e busca um ciclo euleriano através da função buscarSubcicloEuleriano; e por último, buscarSubcicloEuleriano, que recebe o grafo, um vértice inicial e uma lista C de arestas que já foram visitadas e retorna um ciclo que é uma lista de vértices Ciclo. Essa última função utiliza o método extra arestaVizinhos(v) do Grafo para obter as arestas ligadas a v.

Questão 4: Nesse algoritmo de BellmanFord o objetivo é encontrar o menor caminho para cada um dos vértices partindo de uma origem. Para isso são utilizadas apenas duas listas contendo as distâncias da origem até o vértice de um determinado índice e o ancestral direto desse vértice. Através de um processo de relaxamento, se percorre por todas as arestas analisando se a distância do caminho atual é menor que a distância anteriormente encontrada, se sim, o novo valor é atualizado na lista. Isso acontece até que todos os caminhos possíveis sejam percorridos. Ademais há uma função listarAncestrais que auxiliam na impressão dos resultados.

Questão 5: O algoritmo de FloydWarshall é bastante simples. Há duas funções, FloydWarshall, que implementa o próprio algoritmo, e a função printSolution, que imprime o resultado conforme solicitado na questão. A função FloydWarshall começa montando uma matriz de adjacências D e em seguida, executa o algoritmo sobre ela. Para essa implementação, esse algoritmo é ineficiente porque é necessário montar a matriz de adjacências, seria mais adequado utilizar o algoritmo de Dijsktra ou Bellman-Ford para cada vértice e montar a matriz D.