Atividade Prática A1 - Grafos

João Pedro Santana, 15204129
Gustavo Vicente Moser, 16204938
Stefano Bergamin Poletto, 16100745

Para facilitar o entendimento partes que estão presentes no código serão representadas com “”. Exemplo “vizinho = []”

Questão 1

A questão 1 tem como objetivo realizar a criação de um Grafo não-dirigido e ponderado G(V, E, w), além de operações auxiliares para este grafo.

Para criação do grafo se utilizou de dois dicionários para representar os vértices e as arestas, o grafo foi definido como classe, para realizar sua inicialização se faz a leitura dos documentos disponibilizados pelo professor, utilizando da função criada “ler(self)”, para realizar os procedimentos necessários.

Para maioria das operações auxiliares se utilizou de funções presentes na linguagem python3 para ser feita de maneira simples e eficiente.

A operação qtdVertices() retorna o tamanho da lista que é armazenado em “vertices”. A operação qtdArestas() segue a mesma ideia porém tem que utilizar de um for para percorrer todas as suas arestas somando assim seu valor total.

Para a operação grau(v) se pega a o número de arestas que saem do vértice desejado e em seguida se percorre os outros vértices para encontrar arestas que conectam no vértice desejado.

Rotulo(v) é feito com um simples “.get” passando a vértice desejada, retornando assim seu rótulo (nome).

A operação vizinhos(v) realiza as mesmas operações que o grau(v) com a única diferença sendo que “vizinhos” é uma lista e o “grau” um simples int.

Para a operação haArestas() se utiliza da função .get para realizar a checagem se existe ou não arestas entre os vértices desejados retornando o True e False dependendo se foi encontrado ou não.

Peso(u, v) assim como na haArestas utiliza da função get(), porém o retorno é o valor encontrado por essa função.

Ler(arquivo) é a operação fundamental para o programa já que sem ela não é possivel realizar os testes com os documentos disponibilizados pelo professor. Nela é feita a abertura do documento x desejado utilizando do “open(x)” e para resgatar os valores desejados desse documento se utiliza uma junção de diversos “search(x, y)” onde x é o padrão que se deseja resgatar do texto y. São realizados algumas alterações para então ser realizada a atualização das arestas e dos updates.

Questão 2

Para realizar a construção do código da busca em largura se baseou no algoritmo discutido em sala e disponível na apostila, para as variáveis C, D e A presentes no algoritmo se utilizou um dicionário onde pode ser alterado de maneira mais fácil, chamado de “CDA”. O restante do código só segue a base do algoritmo, transformando para linguagem Python. Para a entrega de resultado se utiliza de um dicionário “result” onde neles são alterados os resultados para retornar o resultado de maneira correta.

Questão 3

No Ciclo Euleriano novamente se utilizou do algoritmo disponibilizado na apostila da matéria, para o bom funcionamento do Algoritmo de Hierholzer se utilizou de dicionários para facilitar, no “buscaCicloEuleriano”, o “C” foi criado assim. Utilizando do algoritmo como base se fez as buscas em C utilizando do “get()” e alterações no mesmo utilizado do “update()”. O Algoritmo de buscar Subciclo Euleriano também foi criado, Ciclo é recebe um vetor com valores de “v” que seria o vértice desejado da pesquisa. O restante do código foi transferir o algoritmo para a linguagem Python.

Questão 4

Novamente se utilizou o algoritmo disponibilizado na apostila como base, assim como na questão 2 o C, D e A foram feitos em um dicionário para fazer as alterações de maneira rápida e simples. Foi criado um dicionário “unvisited” para marcar as vértices nao visitados. O restante do código segue o algoritmo como base transformando para Python. Para realizar a entrega dos resultados de maneira desejada se cria o dicionário “paths” e então se faz o print desejado.

Questão 5

Novamente se utilizou do dicionário para facilitar, “dist” foi a variável escolhida, por se tratar de um algoritmo que utiliza de matrizes muitos “for”s foram utilizados, “update()” foram tambem muito utilizado para realizar as alterações dos valores dentro da matriz. O algoritmo base, assim como nas outras questões, é o disponível na apostila da matéria, o Floyd-Warshall. Nessa questão também foi necessário a criação de um vetor “path” para realizar a entrega dos resultados de maneira pedida na atividade.