

Atividade Avaliativa Google Maps
Aluno: Pedro Muhamad Suleiman Craveiro
RA: 193934
Curso: Engenharia de Computação
Semestre: 4º Semestre

Processo de Desenvolvimento:

O projeto consiste na criação de um programa para encontrar o menor caminho entre dois pontos de um mapa utilizando o algoritmo de busca pelo menor caminho em grafos com pesos. Dessa forma, a ideia foi inspirada por meio de um vídeo do canal do professor Hemerson Pistori “Caminho de custo mínimo usando Dijkstra”, juntamente com os conhecimentos da biblioteca Folium, somado a aula ministrada no dia 26/09/2023 a respeito da criação de interfaces gráficas em Python. Ademais, o cenário escolhido foi a cidade de Araranguá, Santa Catarina, considerando a rota entre a UFSC Campus Araranguá, minha antiga casa e a cidade em si. Além disso, um importante fator é que os pontos de decisão foram previamente traçados, levando em consideração características como o tipo de caminho (a pé ou de carro) e os pesos associados a cada rota.

Em seguida, detalharei as seguintes funções desenvolvidas no código:

Função calcular_rota:

Esta função é responsável por calcular a rota com base no modo de transporte escolhido (carro ou a pé). Primeiro, cria um grafo usando a biblioteca networkx, adiciona vértices e arestas com pesos correspondentes às coordenadas dos pontos da cidade. Dependendo do modo de transporte escolhido, define diferentes arestas com pesos. Em seguida, utiliza a função encontrar_menor_caminho para encontrar o menor caminho entre dois pontos (casa e UFSC). Posteriormente, cria um mapa interativo usando a biblioteca folium e exibe os pontos e a rota do menor caminho. O mapa é salvo em um arquivo HTML com o caminho absoluto e aberto no Google Chrome.

Função encontrar_menor_caminho:

Esta função utiliza uma busca em largura (BFS) personalizada para encontrar o menor caminho em um grafo ponderado. A fila é utilizada para explorar os nós em camadas, e o conjunto de visitados rastreia os nós já processados. A função retorna o menor caminho entre os pontos de origem e destino com base no modo de transporte.

Função mostrar_grafo:

Esta função utiliza a biblioteca folium para criar um mapa interativo exibindo os pontos e conexões do grafo. Define coordenadas para os pontos e arestas para representar as conexões entre eles. Adiciona marcadores para os pontos e linhas

para as conexões. O mapa é salvo em um arquivo HTML chamado 'grafo-rotas.html'.

Interface Gráfica (tkinter):

O código cria uma interface gráfica simples usando a biblioteca tkinter. Permite ao usuário escolher entre os modos de transporte (carro ou a pé) por meio de botões de seleção. Além disso, há botões para calcular a rota e exibir o grafo.

Em resumo, o código integra funcionalidades de criação e visualização de grafos, cálculo de rotas mínimas e interação com o usuário por meio de uma interface gráfica.

Materiais Consultados:

- Biblioteca Python: networkx
- Biblioteca Python: folium
- Biblioteca Python: webbrowser
- Biblioteca Python: os
- Biblioteca Python: tkinter
- Vídeo do Professor Hemerson Pistori a respeito do conteúdo de Caminho de Custo Mínimo: [YouTube: Caminho de custo mínimo usando Dijkstra](#)

Ajudas Recebidas:

Dúvidas específicas foram sanadas por meio de consultas ao professor e interações com colegas, discutindo as melhores formas de implementação, questionamento sobre bibliotecas e a linguagem de programação Python. Além do mais, foram realizados testes do funcionamento do código com os colegas de laboratório.

Estimativa de Tempo de Desenvolvimento:

A estimativa de tempo foi realizada com base nos commits no GitHub, permitindo uma visão cronológica do processo de desenvolvimento. E o tempo de desenvolvimento foi em torno de 10 horas.

Código:

Foi utilizado o modelo GPT-3.5 da OpenAI (ChatGPT) como ferramenta de auxílio na elaboração do código, correção de funções e lógicas do algoritmo.

Link para o Repositório:

O código está disponível em link para o repositório no GitHub ([Github - Atividade Avaliativa Google Maps \(Pedro M. S. Craveiro\)](#)).