



UNIVERSIDADE
FEDERAL
DE PERNAMBUCO



**Universidade Federal de Pernambuco
Centro de Informática - CIn**

Disciplina: IF969

Docente: Sérgio Ricardo de Melo Queiroz

Discente: Liosvaldo Mariano Santiago de Abreu

Relatório do projeto: Grupo #3.6

Recife, 18 de fevereiro de 2024

Contexto do problema

Nesse presente relatório, utilizaremos a base de dados aberta da rede atenção à saúde no Recife, disponibilizada pela prefeitura do Recife, para realizar um grafo de menor caminho utilizando os 120 primeiros itens da tabela de dados de pontos de saúde (PSFs, Academias da cidade, Policlínicas). A base de dados contém, além de informações de latitude e longitude dos pontos de saúde, informações como endereço, distrito sanitário e especialidade.

Implementação

Algoritmo utilizado. Algoritmo de PRIM

Desenvolvimento. Inicialmente, realizamos uma matriz de distâncias adjacentes com base na latitude e longitude de cada um dos pontos citados. Na sequência, utilizamos o algoritmo de Prim, que é um algoritmo guloso que encontra uma árvore geradora mínima de um grafo G dado um vértice inicial. para encontrar uma árvore com a menor distância entre as localidades

O algoritmo de prim inicia o processo a partir de um vértice inicial e visita os vértices vizinhos que estão dentro de uma sequência prioritária não visitados, atualizando o seu custo e seus predecessores. Feito isso, é escolhido um vértice com menor custo até que não exista vértice na fila.

Na sequência, utilizamos bibliotecas de visualização e plot para imprimir um grafo contendo todos os vértices utilizados e suas respectivas arestas.

Bibliotecas utilizadas. Inicialmente utilizamos a biblioteca de pandas para ler o CSV e realizar tratamento das informações. Para gerar a matriz adjacência, utilizamos a biblioteca Distance para calcular a distância entre os pontos solicitados. Na sequência, utilizamos a biblioteca networkx para geração de uma estrutura grafo que utilizamos durante o processo. Após isso, aplicamos o algoritmo de prim realizado de maneira autoral, utilizando a extensão da biblioteca networkx para geração de gráficos, baseada no matplotlib.pyplot.

Conclusão

Para facilitar o desenvolvimento, realizamos o download do arquivo disponível na página da prefeitura do recife Recife, Dados criamos o arquivo.

Figura 1 - Página dataset rede de atenção à saúde no Recife

Rede de atenção à saúde no Recife

URL: http://dados.recife.pe.gov.br/dataset/40638fd0-418e-4eeb-8871-b4ed92a973c2/resource/d05f6ffa-304b-4a28-bd03-1fb26cbf866/download/20220729_rede_saude_dados...

Apresenta a Rede de atenção à saúde no Recife. Os dados apresentam os tipos de serviços prestados, unidades de saúde, horário de atendimento, estando todos georreferenciados.

Explorador de Dados

Embutir

Adicionar Filtro

Grade Gráfico Mapa 1051 records « 1 - 100 »

Search data ... Go » Filtros

_id	tipo_uni...	nome_o...	rpa	distrito...	microne...	cnes	cod_nat	tipo_ser...	endereço	bairro	fone	especial...	como_u...
1	1 Acade...	Academi...	1	1	1.1	6865003	Própria	Pólo de ...	Praça 13...	Santo A...	3355382...	Prática d...	O aces...
2	1 Acade...	Academi...	5	5	5.1	7172869	Própria	Pólo de ...	Av Mano...	Mustardi...	3355382...	Prática d...	O aces...
3	1 Acade...	Academi...	2	2	2.3	6664237	Própria	Pólo de ...	Rua Tupi...	Dois Uni...	3355382...	Prática d...	O aces...
4	1 Acade...	Academi...	4	4	4.1	6664660	Própria	Pólo de ...	Praça da...	Cordeiro	3355382...	Prática d...	O aces...
5	1 Acade...	Academi...	4	4	4.1	6664679	Própria	Pólo de ...	Av Beira ...	Torre	3355382...	Prática d...	O aces...
6	1 Acade...	Academi...	6	6	6.3	6664911	Própria	Pólo de ...	Segundo...	Boa Viag...	3355382...	Prática d...	O aces...
7	1 Acade...	Academi...	6	6	6.1	6665012	Própria	Pólo de ...	Av Brasil...	Brasília ...	3355382...	Prática d...	O aces...
8	1 Acade...	Academi...	3	7	7.3	7018339	Própria	Pólo de ...	Av Norte...	Macaxeira	3355382...	Prática d...	O aces...

Após o download, executamos um algoritmo para obtermos a matriz adjacente dos 120 primeiros pontos da rede de atenção à saúde utilizando Pandas e a biblioteca Distance.

Figura 2 - Algoritmo de geração da matriz adjacente com base na latitude e longitude.

```
total_utilizado = 120

total_postos = df_postos_saude_recife.shape[0]

lista_vertices = df_postos_saude_recife['nome_oficial'].to_list()[:total_utilizado]
matriz_arestas = np.zeros((total_utilizado, total_utilizado), dtype=np.float64)

for posto_1 in np.arange(total_utilizado):
    print(posto_1)
    localizacao_1 = df_postos_saude_recife.loc[posto_1, ['latitude', 'longitude']].to_list()
    nome_posto_1 = df_postos_saude_recife.loc[posto_1, 'nome_oficial']
    if posto_1 == 120: break
    for posto_2 in np.arange(total_utilizado):
        localizacao_2 = df_postos_saude_recife.loc[posto_2, ['latitude', 'longitude']].to_list()
        nome_posto_2 = df_postos_saude_recife.loc[posto_2, 'nome_oficial']
        distancia_entre_postos = distance.distance(localizacao_1, localizacao_2).km
        if posto_2 == 120: break
        if (nome_posto_1 != nome_posto_2) and (distancia_entre_postos != 0.):
            print(f"{posto_1} {posto_2} A distancia entre: {nome_posto_1} e {nome_posto_2} é:{distancia_entre_postos}")
            matriz_arestas[posto_1, posto_2] = distancia_entre_postos
```

Feito isso, criamos um arquivo um grafo networkx para iniciar os demais passos do algoritmo de prim

Figura 3 - inicialização do grafo e execução do algoritmo de PRIM

```

matrix_pesos_utilizada = matrix_pesos.iloc[:, quantidade_vertices]
g = nx.from_pandas_adjacency(matrix_pesos_utilizada)  g: <networkx.classes.graph.Graph object>

...

t, w = functions.prim(g, 0)  t: [None, 113, 74, 85, 47, 100, 1]

```

No algoritmo, carregamos todos os pesos inicialmente com o valor infinito da lista de predecessores com o valor None. Na sequência, inicializamos o peso do vértice de início em 0 e adicionamos ele na lista de vértices a serem avaliados.

Na sequência, fazemos um laço que, enquanto a lista de vértices a serem avaliados não for vazia, ele deverá carregar a lista uma sequência de vizinhos e pesos associados a cada uma de suas ligações.

Figura 4 - Algoritmo de Prim utilizado

```

def prim(g, inicio):
    vertices_avaliables = []
    vertices_salvos = []

    pesos = []
    predecessores = []

    for n in g.nodes:
        pesos.append(float('inf'))
        predecessores.append(None)

    pesos[inicio] = 0
    vertices_avaliables.append(inicio)

    while len(vertices_avaliables) != 0:
        min = menor_vertice(vertices_avaliables, pesos)
        vertices_avaliables.remove(min)
        vertices_salvos.append(min)

        for vizinho in g.neighbors(min):
            if vizinho not in vertices_salvos and pesos[vizinho] > \
                g.adj[min][vizinho]['weight']:
                vertices_avaliables.append(vizinho)
                predecessores[vizinho] = min
                pesos[vizinho] = g.adj[min][vizinho]['weight']

    return predecessores, pesos

```

Após obtermos os predecessores e seus respectivos pesos, criamos um grafo para realizarmos a plotagem de seus resultados. Para isso, utilizamos novamente a biblioteca networkx.

Figura 5 - Algoritmo de criação da plotagem para o usuário final.

```
grafico = nx.Graph()  grafico: <networkx.classes.graph.Graph object at 0x00000220578E8060>
functions.cria_vertices(grafico, g.nodes)

for i in range(len(t)):  i: 119
    if t[i] != None:
        grafico.add_edge(i, t[i], weight=w[i])

pos = nx.spring_layout(grafico)  pos: {0: [-0.77240735 -0.4087518 ], 1: [ 0.08809987 -0.15286626], 2: [
nx.draw_networkx_nodes(grafico, pos,
                        nodelist=grafico.nodes,
                        node_size=300,
                        node_color='r')
nx.draw_networkx_edges(grafico, pos,
                        edgelist=nx.edges(grafico))

layout = {}  layout: {0: 'Academia da Cidade Pólo 13 de Maio', 1: 'Academia da Cidade Pólo ABC', 2: 'Ac
for node in grafico.nodes.keys():  node: 119
    layout[node] = lista_vertices[node]

nx.draw_networkx_labels(grafico, pos, layout, font_size=10)
```

Por fim, salvamos o arquivo na pasta diretório com o caminho “Figure/graf0”

Referências

- Recife, Dados; Rede de atenção à saúde no Recife; Dados recife;. julho 2022; Disponível em: <http://dados.recife.pe.gov.br/dataset/rede-de-atencao-a-saude-no-recife/resource/d05f6ffa-304b-4a28-bd03-1ffb26cbf866>. Visualizado em: 18 de fevereiro de 2024
- Abreu, Liosvaldo M. S; IF969_Algoritmos_e_estruturas_de_dados_projetinho_6; Github.com, fevereiro 2024; Disponível em: https://github.com/liosvaldo/IF969_Algoritmos_e_Estruturas_de_Dados. Visualizado em: 18 de fevereiro de 2024.