



UNIVERSIDADE
FEDERAL
DE PERNAMBUCO



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

Disciplina: Algoritmos e Estruturas de Dados

Docente: Sérgio Ricardo de Melo Queiroz

Discente: Leonardo Bezerra de Oliveira

Relatório do projeto

Recife, 27 de abril de 2023

Contexto do problema

Diante da minha escolha do Algoritmo de Prim para realização desse projeto, tentei buscar uma base de dados que fizesse sentido para mim tendo em vista a utilização desse algoritmo, que é baseado na busca pelo menor caminho através das arestas e vértices. Pensando bastante nesse assunto, acabei tendo a ideia de utilizar a distância entre países, estados, cidades, por ser algo mais palpável, utilizaria as distâncias como os pesos e os nomes das cidades como os vértices para criar o grafo baseado nesse algoritmo.

Pesquisando bases de dados tive dificuldade de encontrar esse tipo de base em sites como o kaggle, que usei de exemplo por ser um bastante conhecido, então resolvi procurar no Google mesmo e encontrei a base de dados fornecida pela PUCRS que indica a distância entre algumas cidades e capitais.

A base de dados utilizada conta com a distância rodoviária entre algumas das principais cidades brasileiras, como é o caso por exemplo da distância entre Anápolis, a cidade do estado de Goiás e a cidade de Campinas em São Paulo.

Implementação

Algoritmo utilizado.

O algoritmo escolhido foi o algoritmo de Prim. A princípio tinha a ideia de utilizar o algoritmo de Dijkstra, mas já não haviam vagas disponíveis para utilização desse na planilha de indicação. Julgando os processos do algoritmo de Bellman-Ford, pensei em optar por esse, mas resolvi pesquisar mais sobre os outros algoritmos disponíveis e mudei de ideia e acabei optando por realizar o projeto com o algoritmo de Prim, pois parecia interessante e para mim fazia mais sentido.

Desenvolvimento.

Primeiramente resolvi criar um planejamento para que fossem bem divididas as atividades que ia realizar de acordo com os dias e a minha rotina. Optei por estar fazendo o trabalho individualmente para que pudesse aprender o máximo possível, tendo em vista que não tive tanto contato com esse tipo de algoritmo e estava com dificuldades em relação às listas e criar os códigos em geral, achei que seria uma oportunidade de me desenvolver, tendo em vista também o uso de bibliotecas e a organização possível de dados que esse algoritmo geraria, que traria bons aprendizados para mim.

Mesmo criando esse planejamento, tive algumas dificuldades, até mesmo para seguir o planejamento, por conta de motivos pessoais acabei tendo que reduzir meu escopo e realizar mais atividades nos dias que estivesse livre, para adiantar o processo, tendo em vista que em alguns dos dias não consegui uma grande ou alguma evolução do processo, realizar o projeto sozinho também se torna mais complexo.

Para iniciar o processo criei primeiramente o algoritmo de prim básico, para que pudesse ter uma ideia do que iria fazer a seguir. Após ter encontrado a base de dados que me agradou e estava dentro do esperado em relação a quantidade de vértices e arestas, comecei na tentativa de interligar essa base ao algoritmo para que o grafo esperado fosse gerado, tive bastante dificuldade nesse sentido.

Como já tinha utilizado outras vezes o pandas e achava uma ferramenta mais simples de entender, decidi utilizá-lo, então fiz inicialmente a ligação com o link online mesmo, pensando eu que daria certo, mas não funcionou.

```
import pandas as pd
url = "https://www.inf.pucrs.br/~danielc/peng1a/outros/distancias.xls"
df = pd.read_excel(url)
vertices = list(df.columns)
graph = {vertex: {} for vertex in vertices}
```

como não deu certo, alterei a forma de chamada da base:

```
import pandas as pd
file_path = '/content/distancias.xls'
df = pd.read_excel(file_path)
```

e por fim:

```
import pandas as pd
file_path = 'distancias.xls'
df = pd.read_excel(file_path, header=None, skiprows=2)
```

A implementação se inicia com a chamada das bibliotecas utilizadas, pandas, networkx e matplotlib.

import, início, utilização do pandas para ligar o código com a base de dados.

```
1  import pandas as pd
2  import networkx as nx
3  import matplotlib.pyplot as plt
4
5  file_path = 'distancias.xls'
6  df = pd.read_excel(file_path, header=None, skiprows=2)
7
8  vertices = df.iloc[:, 0].tolist()
9
10 graph = {vertex: {} for vertex in vertices}
```

Preenchimento do grafo, alterações

gerar árvore mínima:

```

55     def find_min_vertex(distances, visited):
56         min_vertex = None
57         min_distance = float('inf')
58
59         for vertex in distances:
60             if not visited[vertex] and distances[vertex] < min_distance:
61                 min_vertex = vertex
62                 min_distance = distances[vertex]
63
64         return min_vertex
65

```

Criar grafo no NetworkX, printar árvore mínima e distância, print da quantidade de vértices e arestas da árvore mínima:

```

66 G = nx.Graph()
67 for vertex, neighbors in graph.items():
68     for neighbor, weight in neighbors.items():
69         G.add_edge(vertex, neighbor, weight=weight)
70
71 minimum_spanning_tree = prim(graph)
72
73
74 for vertex, neighbors in minimum_spanning_tree.items():
75     for neighbor, weight in neighbors.items():
76         print(f"{vertex} - {neighbor}: {weight}")
77
78 num_vertices = len(minimum_spanning_tree)
79 num_arestas = sum(len(neighbors) for neighbors in minimum_spanning_tree.values()) // 2
80 print(f"Número de vértices na Árvore Geradora Mínima: {num_vertices}")
81 print(f"Número de arestas na Árvore Geradora Mínima: {num_arestas}")
82

```

Criação do grafo final do NetworkX, salvando o grafo como arquivo:

```

83 G = nx.Graph()
84 for vertex, neighbors in graph.items():
85     for neighbor, weight in neighbors.items():
86         G.add_edge(vertex, neighbor, weight=weight)
87
88 pos = nx.spring_layout(G)
89 nx.draw(G, pos, with_labels=True, font_weight='bold')
90 edge_labels = {(u, v): d['weight'] for u, v, d in G.edges(data=True)}
91 nx.draw_networkx_edge_labels(G, pos, edge_labels=edge_labels)
92 plt.show()

```

No começo não sabia bem como dividir para que as colunas e linhas que constavam os nomes das cidades não entrassem como valores das distâncias, para que somente a partir de uma linha fosse contabilizado isso. Pesquisando mais sobre a biblioteca utilizada e usando de seus facilitadores, depois de incontáveis tentativas, consegui um resultado mediano, ainda havia erros, mas ao menos as cidades estavam sendo “printadas”, o que me deu alguma esperança.

```

↳  QUADRO DE DISTÂNCIAS RODOVIÁRIAS ENTRE AS PRINCIPAIS CIDADES BRASILEIRAS  \
0      NaN
1      AMERICANA
2      ANÁPOLIS
3      ARACAJÚ
4      ARAÇATUBA

      Unnamed: 1 Unnamed: 2      Unnamed: 3 Unnamed: 4 Unnamed: 5 Unnamed: 6  \
0  ARACAJÚ    BELÉM  BELO HORIZONTE  BOA VISTA  BRASÍLIA  CAMPINAS
1    2216    2808          595    4633    887    42
2    1885    1964          870    4123    160    882
3     NaN    2079          1578    6000    1652   2182
4    2330    2662          854    4487    854    456

      Unnamed: 7      Unnamed: 8 Unnamed: 9  ... Unnamed: 28      Unnamed: 29  \
0  CAMPO GRANDE  CAXIAS DO SUL    CUIABÁ  ...  RIO BRANCO  RIO DE JANEIRO
1          979          1084    1489  ...    3479          545
2          982          1767    981  ...    2971          1296
3         2765          3169    2775  ...    4763          1855
4          490          1187    1343  ...    3333          942

      Unnamed: 30 Unnamed: 31      Unnamed: 32 Unnamed: 33 Unnamed: 34 Unnamed: 35  \
0  SALVADOR    SANTOS  S.J.DOS CAMPOS    SÃO LUÍS    SÃO PAULO    TERESINA
1    1971    205          208    2845    133    2664
2    1679    1045          1052    2005    973    1937
3     356    2249          2086    1578    2187    1142
4    2130    604          595    2699    532    2631

```

Nesse momento percebi que havia alguns problemas em relação aos dados que não constavam na tabela (distâncias vazias) da base de dados. Observando a imagem acima também pude perceber o Unnamed que não fazia sentido com o resultado esperado, que na primeira linha do print consta a primeira linha do nome da tabela, além de “NaN” como nome do índice 0, o que não deveria acontecer, por esse ser na tabela somente uma imagem.

Então fiz alterações no código, para que ele excluísse as linhas que indicavam nomes na tabela da contagem das distâncias e onde o espaço estivesse vazio não fosse contabilizado a distância. Além de alterações no final do código, então foi possível visualizar a ligação entre os vértices e as distâncias(pesos das arestas).

Para a criação de interface utilizei as bibliotecas NetworkX e Matplotlib, pesquisando consegui entender melhor a funcionalidade e pude gerar o grafo através do próprio código. Para ter outra visualização do grafo coloquei ele no aplicativo Gephi e obtive uma melhor imagem, trazida na conclusão.

Bibliotecas utilizadas.

Pandas, <<https://pandas.pydata.org/>>;

NetworkX <<https://networkx.org/>>

Matplotlib <<https://matplotlib.org/>>

Apps utilizados.

Gephi para gerar interface mais visualizável.

<<https://gephi.org/>>

Conclusão

Através desse projeto pude perceber como é importante ter um planejamento de desenvolvimento, tive algumas dificuldades porque não consegui seguir bem o planejamento que tinha feito, por conta de alguns contratempos. Foi interessante o projeto também para aprender um pouco mais sobre o pandas, além do uso do VSCode e do github em si.

O algoritmo de prim é interessante para fazer esse tipo de filtragem, analisando os pesos e na questão escolhida faz muito sentido, tendo em vista que se pode escolher, por exemplo, que cidades passar em uma viagem para economizar tempo ou a depender do que foi objetificado pelo viajante.

Percebi durante esse projeto também que um dos meios mais interessantes para me auxiliar foi o Youtube. Assisti vídeos e algo interessante foi como o vídeo mais curto que assisti foi o que me trouxe mais informação sobre o algoritmo e me fez entender a sua funcionalidade. O vídeo citado é do canal no youtube "Papa Thalyson", intitulado Algoritmo Prim em 3 minutos - Ideia e exemplos, além desse outros me trouxeram muito auxílio em relação ao projeto, para entender as funcionalidades das bibliotecas, por exemplo.

A questão da interface gráfica também parecia complicada para mim, mas utilizei o Gephi e as bibliotecas network e matplotlib.pyplot para que fosse gerado o grafo mostrado a seguir:

Os outros imports tem relação com a geração final do grafo, utilizei o NetworkX e a Matplotlib, utilizando das informações trazidas no próprio site sobre essas ferramentas e as que encontrei em algumas das referências citadas.

Em seguida está o código relacionado ao algoritmo de prim, mas em relação às ligações totais do grafo, em seguida fiz um caminho para a geração da árvore mínima, buscando o caminho com menor peso entre os vértices, fazendo comparações.

Prints de parte do output:

AMERICAANA -> {'ANAPOLIS': 2216.0, 'ARACAJU': 2808.0, 'ARACATUBA': 595.0, 'ARARAQUARA': 4633.0, 'BAGÉ': 887.0, 'BARRETOS': 42.0, 'BARBACENA': 979.0, 'BARRA CARUARA' -> {'AMERICANA': 2216.0, 'ANAPOLIS': 1885.0, 'ARACAJU': 1964.0, 'ARACATUBA': 2330.0, 'ARARAQUARA': 2188.0, 'BAGÉ': 3681.0, 'BARRETOS': 2086.0, 'BARBACENA': 1755.0, 'BARBACENA' -> {'AMERICANA': 2808.0, 'ANAPOLIS': 1964.0, 'ARACAJU': 2079.0, 'ARACATUBA': 2662.0, 'ARARAQUARA': 2657.0, 'BAGÉ': 4182.0, 'BARRETOS': 2501.0, 'BARBACENA': 2985.0, 'BARREIRAS' -> {'AMERICANA': 595.0, 'ANAPOLIS': 2330.0, 'ARACAJU': 2662.0, 'ARACATUBA': 854.0, 'ARARAQUARA': 606.0, 'BAGÉ': 2097.0, 'BARRETOS': 630.0, 'BARBACENA': 169.0, 'BARRA CARUARA' -> {'AMERICANA': 4633.0, 'ANAPOLIS': 2188.0, 'ARACAJU': 2657.0, 'ARACATUBA': 606.0, 'ARARAQUARA': 4480.0, 'BAGÉ': 5585.0, 'BARRETOS': 4324.0, 'BARBACENA': 4894.0, 'BAGÉ' -> {'AMERICANA': 887.0, 'ANAPOLIS': 3681.0, 'ARACAJU': 4182.0, 'ARACATUBA': 2097.0, 'ARARAQUARA': 5585.0, 'BAGÉ': 2355.0, 'BARRETOS': 693.0, 'BARBACENA': 883.0, 'BARREIRAS' -> {'AMERICANA': 42.0, 'ANAPOLIS': 2086.0, 'ARACAJU': 2501.0, 'ARACATUBA': 630.0, 'ARARAQUARA': 4324.0, 'BAGÉ': 693.0, 'BARRETOS': 344.0, 'BARBACENA': 547.0, 'BARREIRAS' -> {'AMERICANA': 979.0, 'ANAPOLIS': 1755.0, 'ARACAJU': 2985.0, 'ARACATUBA': 169.0, 'ARARAQUARA': 4894.0, 'BAGÉ': 883.0, 'BARRETOS': 547.0, 'BARBACENA': 1457.0, 'BARREIRAS' -> {'AMERICANA': 1084.0, 'ANAPOLIS': 1089.0, 'ARACAJU': 2078.0, 'ARACATUBA': 1352.0, 'ARARAQUARA': 4911.0, 'BAGÉ': 648.0, 'BARRETOS': 1557.0, 'BARBACENA': 1770.0, 'BAURUR' -> {'AMERICANA': 1489.0, 'ANAPOLIS': 2312.0, 'ARACAJU': 2727.0, 'ARACATUBA': 731.0, 'ARARAQUARA': 4550.0, 'BAGÉ': 919.0, 'BARRETOS': 269.0, 'BARBACENA': 680.0, 'BARREIRAS' -> {'AMERICANA': 510.0, 'ANAPOLIS': 2079.0, 'ARACAJU': 2595.0, 'ARACATUBA': 2824.0, 'ARARAQUARA': 6083.0, 'BAGÉ': 2120.0, 'BARRETOS': 2140.0, 'BARBACENA': 2942.0, 'BARRA CARUARA' -> {'AMERICANA': 1855.0, 'ANAPOLIS': 1578.0, 'ARACAJU': 2824.0, 'ARACATUBA': 2120.0, 'ARARAQUARA': 4736.0, 'BAGÉ': 716.0, 'BARRETOS': 740.0, 'BARBACENA': 1453.0, 'BLUMENAU' -> {'AMERICANA': 807.0, 'ANAPOLIS': 2843.0, 'ARACAJU': 3451.0, 'ARACATUBA': 1252.0, 'ARARAQUARA': 5079.0, 'BAGÉ': 1624.0, 'BARRETOS': 724.0, 'BARBACENA': 1249.0, 'BOA VISTA' -> {'AMERICANA': 3076.0, 'ANAPOLIS': 6000.0, 'ARACAJU': 6083.0, 'ARACATUBA': 4736.0, 'ARARAQUARA': 3071.0, 'BAGÉ': 4275.0, 'BARRETOS': 4665.0, 'BARBACENA': 3836.0, 'BRASILIA' -> {'AMERICANA': 801.0, 'ANAPOLIS': 1650.0, 'ARACAJU': 2140.0, 'ARACATUBA': 741.0, 'ARARAQUARA': 4275.0, 'BAGÉ': 2175.0, 'BARRETOS': 921.0, 'BARBACENA': 1134.0, 'BRASILIA' -> {'AMERICANA': 2809.0, 'ANAPOLIS': 1544.0, 'ARACAJU': 3244.0, 'ARACATUBA': 550.0, 'ARARAQUARA': 5363.0, 'BAGÉ': 1343.0, 'BARRETOS': 844.0, 'BARBACENA': 177.0, 'CALDAS NOVAS' -> {'AMERICANA': 637.0, 'ANAPOLIS': 1837.0, 'ARACAJU': 2174.0, 'ARACATUBA': 638.0, 'ARARAQUARA': 4139.0, 'BAGÉ': 280.0, 'BARRETOS': 655.0, 'BARBACENA': 860.0, 'CAMPO GRANDE' -> {'AMERICANA': 562.0, 'ANAPOLIS': 541.0, 'ARACAJU': 2031.0, 'ARACATUBA': 2101.0, 'ARARAQUARA': 6523.0, 'BAGÉ': 2175.0, 'BARRETOS': 2705.0, 'BARBACENA': 3285.0, 'CAMPO GRANDE' -> {'AMERICANA': 514.0, 'ANAPOLIS': 2182.0, 'ARACAJU': 2842.0, 'ARACATUBA': 601.0, 'ARARAQUARA': 4665.0, 'BAGÉ': 921.0, 'BARRETOS': 466.0, 'BARBACENA': 1012.0, 'CAMPO GRANDE' -> {'AMERICANA': 2453.0, 'ANAPOLIS': 2764.0, 'ARACAJU': 2942.0, 'ARACATUBA': 1453.0, 'ARARAQUARA': 3836.0, 'BAGÉ': 1134.0, 'BARRETOS': 1012.0, 'BARBACENA': 2031.0, 'CAMPOS DE GOITACAZES' -> {'AMERICANA': 3848.0, 'ANAPOLIS': 1647.0, 'ARACAJU': 3347.0, 'ARACATUBA': 532.0, 'ARARAQUARA': 5294.0, 'BAGÉ': 1249.0, 'BARRETOS': 770.0, 'BARBACENA': 2031.0, 'CARUARU' -> {'AMERICANA': 2947.0, 'ANAPOLIS': 407.0, 'ARACAJU': 1948.0, 'ARACATUBA': 1967.0, 'ARARAQUARA': 6389.0, 'BAGÉ': 2041.0, 'BARRETOS': 2571.0, 'BARBACENA': 3153.0, 'CAVALO' -> {'AMERICANA': 1690.0, 'ANAPOLIS': 2950.0, 'ARACAJU': 3242.0, 'ARACATUBA': 1369.0, 'ARARAQUARA': 4467.0, 'BAGÉ': 1434.0, 'BARRETOS': 906.0, 'BARBACENA': 637.0, 'CAXIAS DO SUL' -> {'AMERICANA': 1474.0, 'ANAPOLIS': 3169.0, 'ARACAJU': 3727.0, 'ARACATUBA': 1585.0, 'ARARAQUARA': 5355.0, 'BAGÉ': 1900.0, 'BARRETOS': 1057.0, 'BARBACENA': 1525.0, 'CHUI' -> {'AMERICANA': 1211.0, 'ANAPOLIS': 3811.0, 'ARACAJU': 4369.0, 'ARACATUBA': 2227.0, 'ARARAQUARA': 5863.0, 'BAGÉ': 2542.0, 'BARRETOS': 1699.0, 'BARBACENA': 2033.0, 'CORUMBA' -> {'AMERICANA': 2945.0, 'ANAPOLIS': 3161.0, 'ARACAJU': 3339.0, 'ARACATUBA': 1850.0, 'ARARAQUARA': 4233.0, 'BAGÉ': 1531.0, 'BARRETOS': 1409.0, 'BARBACENA': 403.0, 'CRICIUMA' -> {'AMERICANA': 2660.0, 'ANAPOLIS': 3078.0, 'ARACAJU': 3672.0, 'ARACATUBA': 1487.0, 'ARARAQUARA': 5314.0, 'BAGÉ': 1859.0, 'BARRETOS': 959.0, 'BARBACENA': 1484.0, 'CUIABA' -> {'AMERICANA': 204.0, 'ANAPOLIS': 2773.0, 'ARACAJU': 2941.0, 'ARACATUBA': 1594.0, 'ARARAQUARA': 3142.0, 'BAGÉ': 1133.0, 'BARRETOS': 1523.0, 'BARBACENA': 694.0, 'CURITIBA' -> {'AMERICANA': 3479.0, 'ANAPOLIS': 2595.0, 'ARACAJU': 3193.0, 'ARACATUBA': 1004.0, 'ARARAQUARA': 4821.0, 'BAGÉ': 1366.0, 'BARRETOS': 476.0, 'BARBACENA': 991.0, 'DOURADOS' -> {'AMERICANA': 545.0, 'ANAPOLIS': 3032.0, 'ARACAJU': 3235.0, 'ARACATUBA': 1449.0, 'ARARAQUARA': 4054.0, 'BAGÉ': 1352.0, 'BARRETOS': 1008.0, 'BARBACENA': 224.0, 'FEIRA DE SANTANA' -> {'AMERICANA': 1971.0, 'ANAPOLIS': 322.0, 'ARACAJU': 1984.0, 'ARACATUBA': 1256.0, 'ARARAQUARA': 5678.0, 'BAGÉ': 1415.0, 'BARRETOS': 1866.0, 'BARBACENA': 25.0, 'FLORIANOPOLIS' -> {'AMERICANA': 205.0, 'ANAPOLIS': 2892.0, 'ARACAJU': 3500.0, 'ARACATUBA': 1301.0, 'ARARAQUARA': 5128.0, 'BAGÉ': 1673.0, 'BARRETOS': 773.0, 'BARBACENA': 1298.0, 'FORTALEZA' -> {'AMERICANA': 208.0, 'ANAPOLIS': 1183.0, 'ARACAJU': 1611.0, 'ARACATUBA': 2528.0, 'ARARAQUARA': 6548.0, 'BAGÉ': 2208.0, 'BARRETOS': 3133.0, 'BARBACENA': 3407.0, 'FORTALEZA' -> {'AMERICANA': 208.0, 'ANAPOLIS': 1183.0, 'ARACAJU': 1611.0, 'ARACATUBA': 2528.0, 'ARARAQUARA': 6548.0, 'BAGÉ': 2208.0, 'BARRETOS': 3133.0, 'BARBACENA': 3407.0, 'FORTALEZA' -> {'AMERICANA': 208.0, 'ANAPOLIS': 1183.0, 'ARACAJU': 1611.0, 'ARACATUBA': 2528.0, 'ARARAQUARA': 6548.0

GUAIRAMA -> {'AMERICANA': 133.0, 'ANÁPOLIS': 204.0, 'ARACAJU': 269.0, 'ARACATUBA': 46.0, 'ARARAQUARA': 4432.0, 'BAGÉ': 676.0, 'BARRETOS': 310.0, 'BARBACEN' ↕ ↘ ↙ ↗ ⚙️ 📄 🗑️
GRANHUNHS -> {'AMERICANA': 2664.0, 'ANÁPOLIS': 390.0, 'ARACAJU': 1914.0, 'ARACATUBA': 1860.0, 'ARARAQUARA': 6282.0, 'BAGÉ': 1934.0, 'BARRETOS': 2464.0, 'BARBACENA': 3040.0, 'BARREIRAS':
GOIÂNIA -> {'AMERICANA': 465.0, 'ANÁPOLIS': 1849.0, 'ARACAJU': 2017.0, 'ARACATUBA': 906.0, 'ARARAQUARA': 4076.0, 'BAGÉ': 209.0, 'BARRETOS': 835.0, 'BARBACENA': 935.0, 'BARREIRAS': 1.
GOV. VALADARES -> {'AMERICANA': 993.0, 'ANÁPOLIS': 1260.0, 'ARACAJU': 2921.0, 'ARACATUBA': 324.0, 'ARARAQUARA': 5064.0, 'BAGÉ': 1041.0, 'BARRETOS': 931.0, 'BARBACENA': 1781.0, 'BARRI
GUARATINGUETA -> {'ANÁPOLIS': 2004.0, 'ARACAJU': 3088.0, 'ARACATUBA': 556.0, 'ARARAQUARA': 4913.0, 'BAGÉ': 1167.0, 'BARRETOS': 254.0, 'BARBACENA': 1188.0, 'BARREIRAS': 1169.0, 'BAURI
ILHEUS -> {'ANÁPOLIS': 690.0, 'BAGÉ': 1378.0, 'BELO HORIZONTE': 384.0, 'BRÁSÍLIA': 1587.0, 'CRICIÚMA': 1173.0, 'FEIRA DE SANTANA': 462.0}
IMPERATRIZ -> {'ANÁPOLIS': 1909.0, 'ARACAJU': 609.0, 'ARACATUBA': 2231.0, 'ARARAQUARA': 5484.0, 'BAGÉ': 1521.0, 'BARRETOS': 2243.0, 'BARBACENA': 2343.0, 'BARREIRAS': 3128.0, 'BAURU':
IPATINGA -> {'ANÁPOLIS': 1361.0, 'ARACAJU': 3019.0, 'ARACATUBA': 217.0, 'ARARAQUARA': 4958.0, 'BAGÉ': 935.0, 'BARRETOS': 828.0, 'BARBACENA': 1675.0, 'BARREIRAS': 1897.0, 'BAURU': 181
ITABIRA -> {'ANÁPOLIS': 1463.0, 'ARACAJU': 2918.0, 'ARACATUBA': 99.0, 'ARARAQUARA': 4839.0, 'BAGÉ': 816.0, 'BARRETOS': 704.0, 'BARBACENA': 1556.0, 'BARREIRAS': 1688.0, 'BAURU': 1697.
ITABUNA -> {'ANÁPOLIS': 660.0, 'ARACAJU': 2360.0, 'ARACATUBA': 1104.0, 'ARARAQUARA': 5849.0, 'BAGÉ': 1283.0, 'BARRETOS': 1710.0, 'BARBACENA': 2566.0, 'BARREIRAS': 2625.0, 'BAURU': 2
ITAJAI -> {'ANÁPOLIS': 2804.0, 'ARACAJU': 3412.0, 'ARACATUBA': 1213.0, 'ARARAQUARA': 5040.0, 'BAGÉ': 1585.0, 'BARRETOS': 685.0, 'BARBACENA': 1210.0, 'BARREIRAS': 526.0, 'BAURU': 1898
ITAPARICA -> {'ANÁPOLIS': 356.0, 'BAGÉ': 1620.0, 'BELO HORIZONTE': 201.0, 'BRÁSÍLIA': 1828.0, 'CRICIÚMA': 839.0}
ITUUMBURA -> {'ANÁPOLIS': 2039.0, 'ARACAJU': 2217.0, 'ARACATUBA': 702.0, 'ARARAQUARA': 4040.0, 'BAGÉ': 409.0, 'BARRETOS': 643.0, 'BARBACENA': 848.0, 'BARREIRAS': 1516.0, 'BAURU': 898
JOÃO PESSOA -> {'ANÁPOLIS': 611.0, 'ARACAJU': 2161.0, 'ARACATUBA': 2171.0, 'ARARAQUARA': 6539.0, 'BAGÉ': 2245.0, 'BARRETOS': 2775.0, 'BARBACENA': 3357.0, 'BARREIRAS': 3762.0, 'BAURU'
JOINVILLE -> {'ANÁPOLIS': 2722.0, 'ARACAJU': 3330.0, 'ARACATUBA': 1131.0, 'ARARAQUARA': 4958.0, 'BAGÉ': 1503.0, 'BARRETOS': 603.0, 'BARBACENA': 1128.0, 'BARREIRAS': 606.0, 'BAURU': 1
JUAZEIRO DO NORTE -> {'ANÁPOLIS': 702.0, 'ARACAJU': 1530.0, 'ARACATUBA': 2047.0, 'ARARAQUARA': 6273.0, 'BAGÉ': 2010.0, 'BARRETOS': 2657.0, 'BARBACENA': 3132.0, 'BARREIRAS': 3638.0, '
LUIZ DE FORA -> {'ANÁPOLIS': 1718.0, 'ARACAJU': 3100.0, 'ARACATUBA': 272.0, 'ARARAQUARA': 4978.0, 'BAGÉ': 992.0, 'BARRETOS': 541.0, 'BARBACENA': 1512.0, 'BARREIRAS': 1498.0, 'BAURU':
LAGES -> {'ANÁPOLIS': 2959.0, 'ARACAJU': 3517.0, 'ARACATUBA': 1368.0, 'ARARAQUARA': 5181.0, 'BAGÉ': 1690.0, 'BARRETOS': 847.0, 'BARBACENA': 1315.0, 'BARREIRAS': 226.0, 'BAURU': 2003.
LIMEIRA -> {'ANÁPOLIS': 2151.0, 'ARACAJU': 2783.0, 'ARACATUBA': 570.0, 'ARARAQUARA': 4608.0, 'BAGÉ': 862.0, 'BARRETOS': 65.0, 'BARBACENA': 954.0, 'BARREIRAS': 1107.0, 'BAURU': 1464.0
LONDRIINA -> {'ANÁPOLIS': 2584.0, 'ARACAJU': 2891.0, 'ARACATUBA': 1002.0, 'ARARAQUARA': 4445.0, 'BAGÉ': 1083.0, 'BARRETOS': 526.0, 'BARBACENA': 615.0, 'BARREIRAS': 913.0, 'BAURU': 136
MACÉIO -> {'ANÁPOLIS': 294.0, 'ARACAJU': 2173.0, 'ARACATUBA': 1854.0, 'ARARAQUARA': 6276.0, 'BAGÉ': 1928.0, 'BARRETOS': 2458.0, 'BARBACENA': 3040.0, 'BARREIRAS': 3445.0, 'BAURU': 304
MANAUS -> {'ANÁPOLIS': 5215.0, 'ARACAJU': 5298.0, 'ARACATUBA': 3951.0, 'ARARAQUARA': 785.0, 'BAGÉ': 3490.0, 'BARRETOS': 3880.0, 'BARBACENA': 3051.0, 'BARREIRAS': 4570.0, 'BAURU': 235
MARFÍA -> {'ANÁPOLIS': 2388.0, 'ARACAJU': 2695.0, 'ARACATUBA': 805.0, 'ARARAQUARA': 4484.0, 'BAGÉ': 887.0, 'BARRETOS': 367.0, 'BARBACENA': 654.0, 'BARREIRAS': 1038.0, 'BAURU': 1342.
Maringá -> {'ANÁPOLIS': 2678.0, 'ARACAJU': 2970.0, 'ARACATUBA': 1097.0, 'ARARAQUARA': 4471.0, 'BAGÉ': 1162.0, 'BARRETOS': 634.0, 'BARBACENA': 641.0, 'BARREIRAS': 954.0, 'BAURU': 1325
MOSSORÓ CLAROS -> {'ANÁPOLIS': 1328.0, 'ARACAJU': 2796.0, 'ARACATUBA': 418.0, 'ARARAQUARA': 4819.0, 'BAGÉ': 694.0, 'BARRETOS': 1017.0, 'BARBACENA': 1523.0, 'BARREIRAS': 1996.0, 'BAURU'
MONTE SORÓ -> {'ANÁPOLIS': 878.0, 'ARACAJU': 1831.0, 'ARACATUBA': 2438.0, 'ARARAQUARA': 6617.0, 'BAGÉ': 2239.0, 'BARRETOS': 3042.0, 'BARBACENA': 3391.0, 'BARREIRAS': 4029.0, 'BAURU': 3
NATAL -> {'ANÁPOLIS': 788.0, 'ARACAJU': 2108.0, 'ARACATUBA': 2348.0, 'ARARAQUARA': 6770.0, 'BAGÉ': 2422.0, 'BARRETOS': 2952.0, 'BARBACENA': 3537.0, 'BARREIRAS': 3939.0, 'BAURU': 354
NOVO HAMBURGHO -> {'ANÁPOLIS': 3259.0, 'ARACAJU': 3817.0, 'ARACATUBA': 1675.0, 'ARARAQUARA': 5351.0, 'BAGÉ': 1990.0, 'BARRETOS': 1147.0, 'BARBACENA': 1521.0, 'BARREIRAS': 94.0, 'BAURU'
OURINHOS -> {'ANÁPOLIS': 2435.0, 'ARACAJU': 2788.0, 'ARACATUBA': 853.0, 'ARARAQUARA': 4477.0, 'BAGÉ': 980.0, 'BARRETOS': 369.0, 'BARBACENA': 647.0, 'BARREIRAS': 943.0, 'BAURU': 1335.
PARANAGUÁ -> {'ANÁPOLIS': 2629.0, 'ARACAJU': 3295.0, 'ARACATUBA': 1031.0, 'ARARAQUARA': 4923.0, 'BAGÉ': 1420.0, 'BARRETOS': 510.0, 'BARBACENA': 1093.0, 'BARREIRAS': 662.0, 'BAURU': 1
PARNAÍBA -> {'ANÁPOLIS': 1469.0, 'ARACAJU': 1107.0, 'ARACATUBA': 2658.0, 'ARARAQUARA': 6408.0, 'BAGÉ': 2060.0, 'BARRETOS': 3054.0, 'BARBACENA': 3267.0, 'BARREIRAS': 4033.0, 'BAURU':
PASSO FUNDO -> {'ANÁPOLIS': 3153.0, 'ARACAJU': 3627.0, 'ARACATUBA': 1562.0, 'ARARAQUARA': 5126.0, 'BAGÉ': 1800.0, 'BARRETOS': 1053.0, 'BARBACENA': 1296.0, 'BARREIRAS': 226.0, 'BAURU'
PATOS DE MINAS -> {'ANÁPOLIS': 1671.0, 'ARACAJU': 2589.0, 'ARACATUBA': 417.0, 'ARARAQUARA': 4412.0, 'BAGÉ': 549.0, 'BARRETOS': 711.0, 'BARBACENA': 1116.0, 'BARREIRAS': 169.0, 'BAURU'

Número de vértices no Grafo: 111

Número de arestas no Grafo: 6551

AMERICANA - BARRETOS: 42.0

ANÁPOLIS - MACEIÓ: 294.0

ANÁPOLIS - FEIRA DE SANTANA: 322.0

ANÁPOLIS - PAULO AFONSO: 307,0

ARACAJÚ - IMPERATRIZ: 609,0

ARACAJÚ - TUCURUÍ: 350,0

ARACATUBA - BARBACENA: 169,0

ARACATUBA - ARARAQUARA: 606.0

ARACATUBA - GOV. VALADARES: 324.0

ARACATUBA - IPATINGA: 217.0

ARACATUBA - ITABIRA: 99,0

ARACATUBA - MONTES CLAROS: 418.0

ARARAQUARA - ARACATUBA: 606,0

ARARAQUARA - MANAUS: 785.0

BAGÉ - GOIÂNIA: 209,0

BAGÉ - STA. MARTA DA VITÓRIA: 597,0

BARRETOS - AMERICANA: 42.0

BARRETOS - LTMETRA: 65,0

BARRETOS - POCOS DE CALDAS: 150.0

BARRETOS - RIBEIRÃO PRETO: 238-0

BARRETOS - SÃO PAULO: 99.0

BARRETOS - SOBOCABA: 85,0

BARRACENA - ARACATUBA: 169,0

BARBACENA - DOUBRADOS: 224,0

BARRACENA - COBUMBÁ: 403 0

BARBACENA - PONTA DOBÃ: 335,0

BARBACENA - FONTE FORA.

BARREIRAS - LAJES: 220,0

BARREIRAS - NOVO HAMBURGO: 94.
BARREIRAS - BAIXO FUNDO: 226.0

SANTOS - FRANCA: 72.0
SÃO CARLOS - CUIABÁ: 116.0
S. JOSÉ DO R. PRETO - CUIABÁ: 189.0
S. JOSÉ DOS CAMPOS - FRANCA: 97.0
SÃO LUÍS - GUARANHUNS: 446.0
SÃO LOURENÇO - CAMPINA GRANDE: 208.0
SÃO LOURENÇO - FORTALEZA: 182.0
SÃO PAULO - FLORIANÓPOLIS: 72.0
SÃO PAULO - FORTALEZA: 97.0
SÃO PAULO - BARRETOS: 99.0
STA. MARIA DA VITÓRIA - BAGÉ: 597.0
SINOP - BAURU: 488.0
SOBRAL - BOA VISTA: 238.0
SOROCABA - FRANCA: 87.0
SOROCABA - BARRETOS: 85.0
TAUBATÉ - FORTALEZA: 43.0
TERESINA - FOZ DO IGUAÇÚ: 446.0
TUBARÃO - BLUMENAU: 144.0
TUCURUÍ - ARACAJÚ: 350.0
UBERABA - GOIÂNIA: 113.0
UBERABA - CUIABÁ: 178.0
UBERLANDIA - CUIABÁ: 281.0
URUGUAIANA - CAXIAS DO SUL: 604.0
VALENÇA (BA) - BELO HORIZONTE: 194.0
VALENÇA (BA) - FEIRA DE SANTANA: 105.0
VILHENA - CORUMBÁ: 704.0
VITÓRIA - CAMPINA GRANDE: 519.0
V. DA CONQUISTA - BELO HORIZONTE: 411.0
VOLTA REDONDA - DOURADOS: 112.0
Número de vértices na Árvore Geradora Mínima: 111
Número de arestas na Árvore Geradora Mínima: 110

As características da tabela me trouxeram algumas dificuldades, tendo em vista essas especificidades da posição que deveria começar o grafo, por exemplo, tendo que especificar linha e coluna, além da contabilização dos pesos (as distâncias entre as cidades, as arestas), o início mesmo do grafo e as ligações entre as cidades(vértices).

Mesmo com todos os problemas que tive durante o processo consegui fazer essa ligação entre as cidades e organizar as distâncias em si, até para ter uma visualização melhor e pensei na possibilidade que esse resultado poderia trazer caso fosse fazer algum tipo de análise de dados, o que faz sentido tendo em vista o uso das funcionalidades da biblioteca pandas dentro dessa área.

Referências

Guia Básico de Git - Como fazer commit no GitHub. Blog Coder 2021 , disponível em:
<<https://blog.cod3r.com.br/guia-basico-de-git-como-fazer-commit-no-github/>>

Introdução Pandas Python. Hashtag Treinamentos. 2021, disponível em:
<<https://www.hashtagtreinamentos.com/introducao-pandas-python>>

Pandas o que é e para quê serve, Alura. 2021, disponível em:<<https://www.alura.com.br/artigos/pandas-o-que-e-para-que-serve-como-instalar>>

Gephi, guia básico, disponível em:<<https://gephi.org/users/>>

O que é commit e como usar commits semânticos, Geek Hunter, disponível em:
<<https://blog.geekhunter.com.br/o-que-e-commit-e-como-usar-commits-semanticos/>>

Algoritmo de Prim em 3 minutos - Ideia e exemplos, Papa Thalyson, 2021. disponível em:
<https://www.youtube.com/watch?v=XXgMkLSNpg&ab_channel=PapaThalyson>

Algoritmo de Prim - Teoria dos grafos (Exemplos práticos), Industrial 21, 2021. disponível em: <https://www.youtube.com/watch?v=Pgf-AL1vYb4&ab_channel=INDUSTRIAL21>

Manipulação de grafos em Python com NetworkX, Professor Alexandre Nevada, disponível em:
<https://www.youtube.com/watch?v=PfT8_2sKReo&ab_channel=Prof.AlexandreLevada>

Como sair do ZERO em Gráficos no Python [Matplotlib], Hashtag Programação, 2021, disponível em:
<https://www.youtube.com/watch?v=FDU-D8ddTU4&ab_channel=HashtagPrograma%C3%A7%C3%A3o>

Python - Introdução a Biblioteca Matplotlib e Numpy pythonaltas, 2022, disponível em:
<https://www.youtube.com/watch?v=QLACZgfUAhY&ab_channel=Pythonaltas>

Algoritmo de Prim - Árvores Geradoras Mínimas - Algoritmos em Grafos, aulas de computação, 2021, disponível em:
<https://www.youtube.com/watch?v=vXdtCJmkoEU&ab_channel=AulasdeComputa%C3%A7%C3%A3o>

Gephi e análise baseada em grafos, André Santachè, 2015, disponível em:
<https://www.youtube.com/watch?v=4tAOA7Skqtw&t=204s&ab_channel=Andr%C3%A9Santanch%C3%A8>

Grafos com Python - Matriz de Adjacências - Aula 15 de Grafos, Professor Douglas Maioli, disponível em:
<https://www.youtube.com/watch?v=lzG9l_7GaZM&ab_channel=ProfessorDouglasMaioli>

