



Relatório do Projeto – Parte 1

Nome do Integrante	TIA
Mateus Fernandes	32137141
Victor Hugo Antonio Couto	32173482

Conteúdo do Relatório

Representação da Rota da Seda durante o século XIV a partir de Grafo

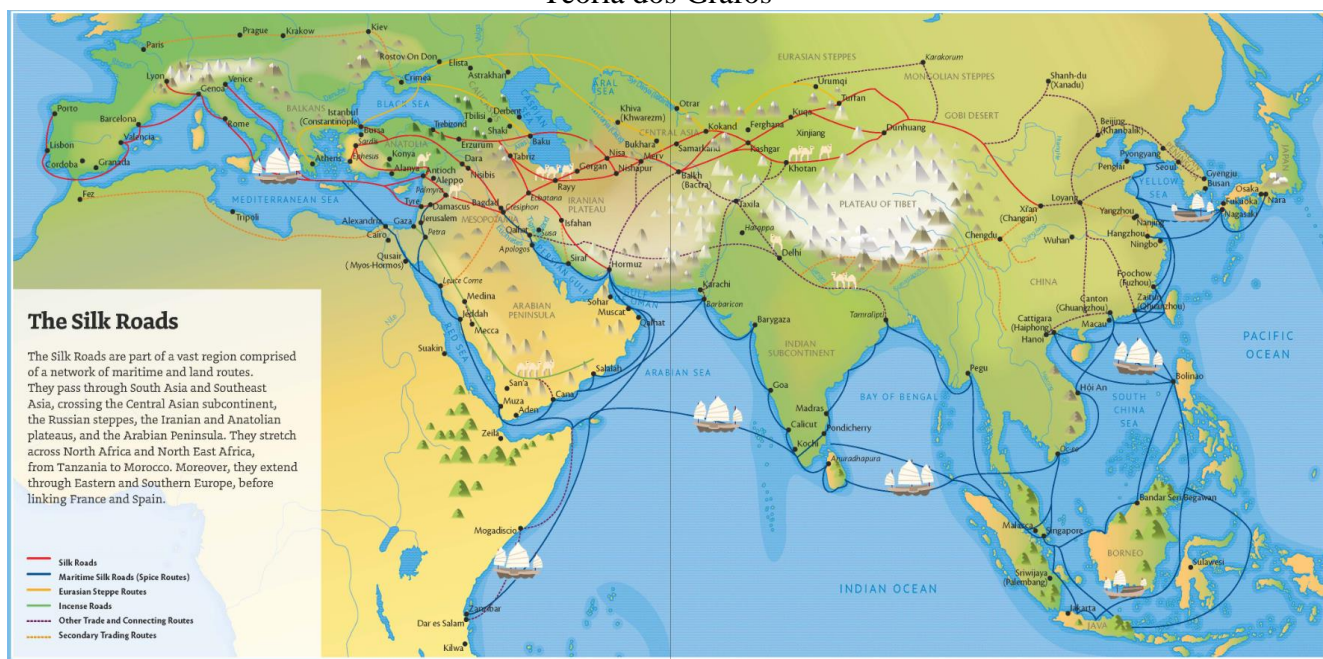
As Rotas da Seda, antigas redes comerciais, desempenharam um papel fundamental na história. Além de facilitar o comércio de bens valiosos, elas serviram como canais para a transmissão de conhecimento, ideias e culturas entre diversas civilizações da Eurásia. Os viajantes ao longo dessas rotas não buscavam apenas lucro, mas também oportunidades de intercâmbio intelectual e cultural. Isso resultou no compartilhamento de ciência, artes, literatura, tecnologia e muito mais, influenciando o desenvolvimento de línguas, religiões e culturas. As Rotas da Seda atuaram como pontes culturais, deixando um legado duradouro que se manifesta nas diversas culturas, línguas e religiões que se desenvolveram ao longo dos milênios. Elas continuaram a promover uma interação cultural contínua à medida que mercadores e viajantes de diferentes origens se encontravam. Desde suas origens exploratórias, as Rotas da Seda evoluíram para desempenhar um papel fundamental na formação de sociedades em toda a Eurásia e além dela.

O projeto possui como objetivo principal fornecer conhecimento a respeito da Rota da Seda durante o século XIV a partir da representação em forma de grafo utilizando os conceitos de teoria dos grafos.

A rota da seda representada no projeto é um grafo não direcionado e conexo, implementado em lista de adjacência, constituído de 123 cidades, os vértices do grafo, e 156 arestas com respectivos pesos que representam a distância aproximada em quilômetros entre as respectivas cidades.

Para a primeira parte do projeto foi focalizado apenas a representação geográfica das cidades e as distâncias entre as mesmas, informações sobre determinadas cidades, conhecimentos específicos e curiosidades serão implementadas posteriormente.

Imagem da rota da seda:



Modelagem no *Graph Online*:

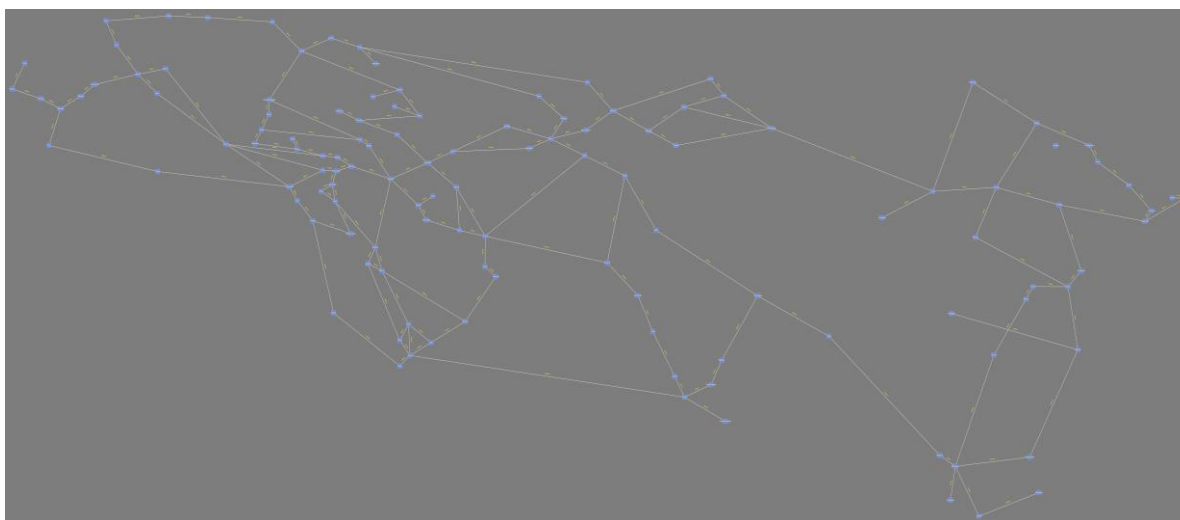


Imagem do grafo:

Objetivos do Desenvolvimento Sustentável (ODS)

O projeto contribui indiretamente para o ODS 4 de Educação de Qualidade ao fornecer recursos que podem contribuir para a educação e pesquisa educacional. Isso inclui o uso de gráficos e visualizações para facilitar a compreensão da rota da seda, a criação de recursos educacionais interativos, e a promoção da conscientização sobre a herança cultural e histórica das Rotas da Seda.



Imagens dos Testes

Menu:

<

Mapeamento da rota da seda durante o século XIV

>

Opções de operações:

1. Ler os dados do arquivo.

2. Gravar dados no arquivo.

3. Inserir vértice no Grafo.

4. Inserir aresta no Grafo.

5. Remover vértice do grafo.

6. Remover aresta do grafo.

7. Mostrar conteúdo do arquivo.

8. Mostrar grafo.

9. Conexidade do grafo.

10. Coloração.

11. Grau dos Vértices.

12. Grafo Euleriano.

13. Percurso Euleriano.

14. Grafo Hamiltoniano.

15. Encerrar.

<

Escolha a opção:

>

Opção 01: Ler os dados do arquivo.



<

Mapeamento da rota da seda durante o século XIV

>

Opções de operações:

1. Ler os dados do arquivo.

2. Gravar dados no arquivo.

3. Inserir vértice no Grafo.

4. Inserir aresta no Grafo.

5. Remover vértice do grafo.

6. Remover aresta do grafo.

7. Mostrar conteúdo do arquivo.

8. Mostrar grafo.

9. Conexidade do grafo.

10. Coloração.

11. Grau dos Vértices.

12. Grafo Euleriano.

13. Percurso Euleriano.

14. Grafo Hamiltoniano.

15. Encerrar.

<

Escolha a opção: 1

>

Arquivo lido com sucesso.

Opção 02: Gravar dados no arquivo

Mapamento da rota da seda durante o século XIV	
Opções de operações:	
1. Ler os dados do arquivo.	
2. Gravar dados no arquivo.	
3. Inserir vértice no Grafo.	
4. Inserir aresta no Grafo.	
5. Remover vértice do grafo.	
6. Remover aresta do grafo.	
7. Mostrar conteúdo do arquivo.	
8. Mostrar grafo.	
9. Condição do grafo.	
10. Coloração.	
11. Grau dos Vértices.	
12. Grafo Euleriano.	
13. Percorso Euleriano.	
14. Grafo Hamiltoniano.	
15. Encerrar.	

Exibir a opção 2	
3	3 Kiev, 14 Crimeia, 15 Athens, 16 Eria, 17 Irpina, 18 Alexandria, 19 Cairo, 20 Quair, 21 Sum, 22 Zila, 23 Constantinople, 24 Bursa, 25 Sardia, 26 Ephesus, 27 Alanya, 28 Antioch, 29 Aleppo, 31 Palmyra, 33 Damascus, 35 Gazala, 36 Petra, 37 Leont, 38 Baran, 39 Mishin, 40 Bamada, 41 Tadmor, 42 Erzurum, 43 Sam, 44 Sam, 45 Baniyas, 46 Aleppo, 47 Hama, 48 Hama, 49 Hama, 50 Hama, 51 Hama, 52 Hama, 53 Hama, 54 Hama, 55 Hama, 56 Hama, 57 Hama, 58 Hama, 59 Hama, 60 Hama, 61 Hama, 62 Hama, 63 Hama, 64 Hama, 65 Hama, 66 Hama, 67 Hama, 68 Hama, 69 Hama, 70 Hama, 71 Hama, 72 Hama, 73 Hama, 74 Hama, 75 Hama, 76 Hama, 77 Hama, 78 Hama, 79 Hama, 80 Hama, 81 Hama, 82 Hama, 83 Hama, 84 Hama, 85 Hama, 86 Hama, 87 Hama, 88 Hama, 89 Hama, 90 Hama, 91 Hama, 92 Hama, 93 Hama, 94 Hama, 95 Hama, 96 Hama, 97 Hama, 98 Hama, 99 Hama, 100 Hama, 101 Hama, 102 Hama, 103 Hama, 104 Hama, 105 Hama, 106 Hama, 107 Hama, 108 Hama, 109 Hama, 110 Hama, 111 Hama, 112 Hama, 113 Hama, 114 Hama, 115 Hama, 116 Hama, 117 Hama, 118 Hama, 119 Hama, 120 Hama, 121 Hama, 122 Hama, 123 Hama, 124 Hama, 125 Hama, 126 Hama, 127 Hama, 128 Hama, 129 Hama, 130 Hama, 131 Hama, 132 Hama, 133 Hama, 134 Hama, 135 Hama, 136 Hama, 137 Hama, 138 Hama, 139 Hama, 140 Hama, 141 Hama, 142 Hama, 143 Hama, 144 Hama, 145 Hama, 146 Hama, 147 Hama, 148 Hama, 149 Hama, 150 Hama, 151 Hama, 152 Hama, 153 Hama, 154 Hama, 155 Hama, 156 Hama, 157 Hama, 158 Hama, 159 Hama, 160 Hama, 161 Hama, 162 Hama, 163 Hama, 164 Hama, 165 Hama, 166 Hama, 167 Hama, 168 Hama, 169 Hama, 170 Hama, 171 Hama, 172 Hama, 173 Hama, 174 Hama, 175 Hama, 176 Hama, 177 Hama, 178 Hama, 179 Hama, 180 Hama, 181 Hama, 182 Hama, 183 Hama, 184 Hama, 185 Hama, 186 Hama, 187 Hama, 188 Hama, 189 Hama, 190 Hama, 191 Hama, 192 Hama, 193 Hama, 194 Hama, 195 Hama, 196 Hama, 197 Hama, 198 Hama, 199 Hama, 200 Hama, 201 Hama, 202 Hama, 203 Hama, 204 Hama, 205 Hama, 206 Hama, 207 Hama, 208 Hama, 209 Hama, 210 Hama, 211 Hama, 212 Hama, 213 Hama, 214 Hama, 215 Hama, 216 Hama, 217 Hama, 218 Hama, 219 Hama, 220 Hama, 221 Hama, 222 Hama, 223 Hama, 224 Hama, 225 Hama, 226 Hama, 227 Hama, 228 Hama, 229 Hama, 230 Hama, 231 Hama, 232 Hama, 233 Hama, 234 Hama, 235 Hama, 236 Hama, 237 Hama, 238 Hama, 239 Hama, 240 Hama, 241 Hama, 242 Hama, 243 Hama, 244 Hama, 245 Hama, 246 Hama, 247 Hama, 248 Hama, 249 Hama, 250 Hama, 251 Hama, 252 Hama, 253 Hama, 254 Hama, 255 Hama, 256 Hama, 257 Hama, 258 Hama, 259 Hama, 260 Hama, 261 Hama, 262 Hama, 263 Hama, 264 Hama, 265 Hama, 266 Hama, 267 Hama, 268 Hama, 269 Hama, 270 Hama, 271 Hama, 272 Hama, 273 Hama, 274 Hama, 275 Hama, 276 Hama, 277 Hama, 278 Hama, 279 Hama, 280 Hama, 281 Hama, 282 Hama, 283 Hama, 284 Hama, 285 Hama, 286 Hama, 287 Hama, 288 Hama, 289 Hama, 290 Hama, 291 Hama, 292 Hama, 293 Hama, 294 Hama, 295 Hama, 296 Hama, 297 Hama, 298 Hama, 299 Hama, 300 Hama, 301 Hama, 302 Hama, 303 Hama, 304 Hama, 305 Hama, 306 Hama, 307 Hama, 308 Hama, 309 Hama, 310 Hama, 311 Hama, 312 Hama, 313 Hama, 314 Hama, 315 Hama, 316 Hama, 317 Hama, 318 Hama, 319 Hama, 320 Hama, 321 Hama, 322 Hama, 323 Hama, 324 Hama, 325 Hama, 326 Hama, 327 Hama, 328 Hama, 329 Hama, 330 Hama, 331 Hama, 332 Hama, 333 Hama, 334 Hama, 335 Hama, 336 Hama, 337 Hama, 338 Hama, 339 Hama, 340 Hama, 341 Hama, 342 Hama, 343 Hama, 344 Hama, 345 Hama, 346 Hama, 347 Hama, 348 Hama, 349 Hama, 350 Hama, 351 Hama, 352 Hama, 353 Hama, 354 Hama, 355 Hama, 356 Hama, 357 Hama, 358 Hama, 359 Hama, 360 Hama, 361 Hama, 362 Hama, 363 Hama, 364 Hama, 365 Hama, 366 Hama, 367 Hama, 368 Hama, 369 Hama, 370 Hama, 371 Hama, 372 Hama, 373 Hama, 374 Hama, 375 Hama, 376 Hama, 377 Hama, 378 Hama, 379 Hama, 380 Hama, 381 Hama, 382 Hama, 383 Hama, 384 Hama, 385 Hama, 386 Hama, 387 Hama, 388 Hama, 389 Hama, 390 Hama, 391 Hama, 392 Hama, 393 Hama, 394 Hama, 395 Hama, 396 Hama, 397 Hama, 398 Hama, 399 Hama, 400 Hama, 401 Hama, 402 Hama, 403 Hama, 404 Hama, 405 Hama, 406 Hama, 407 Hama, 408 Hama, 409 Hama, 410 Hama, 411 Hama, 412 Hama, 413 Hama, 414 Hama, 415 Hama, 416 Hama, 417 Hama, 418 Hama, 419 Hama, 420 Hama, 421 Hama, 422 Hama, 423 Hama, 424 Hama, 425 Hama, 426 Hama, 427 Hama, 428 Hama, 429 Hama, 430 Hama, 431 Hama, 432 Hama, 433 Hama, 434 Hama, 435 Hama, 436 Hama, 437 Hama, 438 Hama, 439 Hama, 440 Hama, 441 Hama, 442 Hama, 443 Hama, 444 Hama, 445 Hama, 446 Hama, 447 Hama, 448 Hama, 449 Hama, 450 Hama, 451 Hama, 452 Hama, 453 Hama, 454 Hama, 455 Hama, 456 Hama, 457 Hama, 458 Hama, 459 Hama, 460 Hama, 461 Hama, 462 Hama, 463 Hama, 464 Hama, 465 Hama, 466 Hama, 467 Hama, 468 Hama, 469 Hama, 470 Hama, 471 Hama, 472 Hama, 473 Hama, 474 Hama, 475 Hama, 476 Hama, 477 Hama, 478 Hama, 479 Hama, 480 Hama, 481 Hama, 482 Hama, 483 Hama, 484 Hama, 485 Hama, 486 Hama, 487 Hama, 488 Hama, 489 Hama, 490 Hama, 491 Hama, 492 Hama, 493 Hama, 494 Hama, 495 Hama, 496 Hama, 497 Hama, 498 Hama, 499 Hama, 500 Hama, 501 Hama, 502 Hama, 503 Hama, 504 Hama, 505 Hama, 506 Hama, 507 Hama, 508 Hama, 509 Hama, 510 Hama, 511 Hama, 512 Hama, 513 Hama, 514 Hama, 515 Hama, 516 Hama, 517 Hama, 518 Hama, 519 Hama, 520 Hama, 521 Hama, 522 Hama, 523 Hama, 524 Hama, 525 Hama, 526 Hama, 527 Hama, 528 Hama, 529 Hama, 530 Hama, 531 Hama, 532 Hama, 533 Hama, 534 Hama, 535 Hama, 536 Hama, 537 Hama, 538 Hama, 539 Hama, 540 Hama, 541 Hama, 542 Hama, 543 Hama, 544 Hama, 545 Hama

Opção 03: Inserir vértice no Grafo



UNIVERSIDADE PRESBITERIANA MACKENZIE

Faculdade de Computação e Informática

Prof. Dr. Ivan Carlos Alcântara de Oliveira
Teoria dos Grafos



```
< ----- >
Mapeamento da rota da seda durante o século XIV
-----
Opções de operações:
-----
1. Ler os dados do arquivo.
-----
2. Gravar dados no arquivo.
-----
3. Inserir vértice no Grafo.
-----
4. Inserir aresta no Grafo.
-----
5. Remover vértice do grafo.
-----
6. Remover aresta do grafo.
-----
7. Mostrar conteúdo do arquivo.
-----
8. Mostrar grafo.
-----
9. Conexidade do grafo.
-----
10. Coloração.
-----
11. Grau dos Vértices.
-----
12. Grafo Euleriano.
-----
13. Percurso Euleriano.
-----
14. Grafo Hamiltoniano.
-----
15. Encerrar.
-----
< ----- >

Escolha a opção: 3
Informe o nome do vértice: Yhorm

Vértice adicionado com sucesso.
```



UNIVERSIDADE PRESBITERIANA MACKENZIE

Faculdade de Computação e Informática

Prof. Dr. Ivan Carlos Alcântara de Oliveira
Teoria dos Grafos



```
< ----- >
Mapeamento da rota da seda durante o século XIV
-----
Opções de operações:
-----
1. Ler os dados do arquivo.
-----
2. Gravar dados no arquivo.
-----
3. Inserir vértice no Grafo.
-----
4. Inserir aresta no Grafo.
-----
5. Remover vértice do grafo.
-----
6. Remover aresta do grafo.
-----
7. Mostrar conteúdo do arquivo.
-----
8. Mostrar grafo.
-----
9. Conexidade do grafo.
-----
10. Coloração.
-----
11. Grau dos Vértices.
-----
12. Grafo Euleriano.
-----
13. Percurso Euleriano.
-----
14. Grafo Hamiltoniano.
-----
15. Encerrar.
-----
< ----- >

Escolha a opção: 3
Informe o nome do vértice: Creta

Vértice adicionado com sucesso.
```

Opção 04: Inserir aresta no Grafo



UNIVERSIDADE PRESBITERIANA MACKENZIE

Faculdade de Computação e Informática

Prof. Dr. Ivan Carlos Alcântara de Oliveira
Teoria dos Grafos



```
<----->
Mapeamento da rota da seda durante o século XIV
-----
Opções de operações:
-----
1. Ler os dados do arquivo.
-----
2. Gravar dados no arquivo.
-----
3. Inserir vértice no Grafo.
-----
4. Inserir aresta no Grafo.
-----
5. Remover vértice do grafo.
-----
6. Remover aresta do grafo.
-----
7. Mostrar conteúdo do arquivo.
-----
8. Mostrar grafo.
-----
9. Conexidade do grafo.
-----
10. Coloração.
-----
11. Grau dos Vértices.
-----
12. Grafo Euleriano.
-----
13. Percurso Euleriano.
-----
14. Grafo Hamiltoniano.
-----
15. Encerrar.
----->

Escolha a opção: 4
Vértice de origem: Yhorm
Vértice de destino: Athens
Peso da aresta: 1560
Aresta entre Yhorm e Athens inserida com sucesso.
```

```
Crimea <--860--> Kiev <--600--> Constantinople <--860--> Rostov <--1500--> Derbent
Athens <--1550--> Rome <--1900--> Venice <--1100--> Alexandria <--1300--> Antioch <--1200--> Tyre <--1560--> Yhorm
Fez <--470--> Granada <--2100--> Tripoli
```

```
Sulawesi <--220--> Jakarta
Yhorm <--1560--> Athens
```

Opção 05: Remover vértice do Grafo



UNIVERSIDADE PRESBITERIANA MACKENZIE

Faculdade de Computação e Informática

Prof. Dr. Ivan Carlos Alcântara de Oliveira
Teoria dos Grafos



```
<----->
Mapeamento da rota da seda durante o século XIV
-----
Opções de operações:
-----
1. Ler os dados do arquivo.
-----
2. Gravar dados no arquivo.
-----
3. Inserir vértice no Grafo.
-----
4. Inserir aresta no Grafo.
-----
5. Remover vértice do grafo.
-----
6. Remover aresta do grafo.
-----
7. Mostrar conteúdo do arquivo.
-----
8. Mostrar grafo.
-----
9. Conexidade do grafo.
-----
10. Coloração.
-----
11. Grau dos Vértices.
-----
12. Grafo Euleriano.
-----
13. Percurso Euleriano.
-----
14. Grafo Hamiltoniano.
-----
15. Encerrar.
----->

Escolha a opção: 5
Informe o nome do vértice: Yhorm
Vértice Yhorm removido com sucesso.
```

```
Crimea <--860--> Kiev <--600--> Constantinople <--860--> Rostov <--1500--> Derbent
Athens <--1550--> Rome <--1900--> Venice <--1100--> Alexandria <--1300--> Antioch <--1200--> Tyre
Fez <--470--> Granada <--2100--> Tripoli
```




UNIVERSIDADE PRESBITERIANA MACKENZIE

Faculdade de Computação e Informática

Prof. Dr. Ivan Carlos Alcântara de Oliveira
Teoria dos Grafos



```
<----->
Mapeamento da rota da seda durante o século XIV
-----
Opções de operações:
-----
1. Ler os dados do arquivo.
-----
2. Gravar dados no arquivo.
-----
3. Inserir vértice no Grafo.
-----
4. Inserir aresta no Grafo.
-----
5. Remover vértice do grafo.
-----
6. Remover aresta do grafo.
-----
7. Mostrar conteúdo do arquivo.
-----
8. Mostrar grafo.
-----
9. Conexidade do grafo.
-----
10. Coloração.
-----
11. Grau dos Vértices.
-----
12. Grafo Euleriano.
-----
13. Percurso Euleriano.
-----
14. Grafo Hamiltoniano.
-----
15. Encerrar.
----->

Escolha a opção: 5
Informe o nome do vértice: Granada
Vértice Granada removido com sucesso.
```

```
Porto <--313--> Lisbon
Lisbon <--313--> Porto <--502--> Cordoba
Cordoba <--502--> Lisbon <--206--> Granada
Granada <--206--> Cordoba <--451--> Valencia <--470--> Fez
Valencia <--451--> Granada <--350--> Barcelona
Barcelona <--350--> Valencia <--680--> Genoa
Genoa <--680--> Barcelona <--460--> Lyon <--500--> Rome <--370--> Venice
```

```
Porto <--313--> Lisbon
Lisbon <--313--> Porto <--502--> Cordoba
Cordoba <--502--> Lisbon
Valencia <--350--> Barcelona
Barcelona <--350--> Valencia <--680--> Genoa
Genoa <--680--> Barcelona <--460--> Lyon <--500--> Rome <--370--> Venice
```

Opção 06: Remover Aresta do Grafo



UNIVERSIDADE PRESBITERIANA MACKENZIE

Faculdade de Computação e Informática

Prof. Dr. Ivan Carlos Alcântara de Oliveira
Teoria dos Grafos



```
< ----- >
Mapeamento da rota da seda durante o século XIV
-----
Opções de operações:
-----
1. Ler os dados do arquivo.
-----
2. Gravar dados no arquivo.
-----
3. Inserir vértice no Grafo.
-----
4. Inserir aresta no Grafo.
-----
5. Remover vértice do grafo.
-----
6. Remover aresta do grafo.
-----
7. Mostrar conteúdo do arquivo.
-----
8. Mostrar grafo.
-----
9. Conexidade do grafo.
-----
10. Coloração.
-----
11. Grau dos Vértices.
-----
12. Grafo Euleriano.
-----
13. Percurso Euleriano.
-----
14. Grafo Hamiltoniano.
-----
15. Encerrar.
-----
< ----- >
```

Escolha a opção: 6
Vértice de origem: Valencia
Vértice de destino: Barcelona
Aresta entre Valencia e Barcelona removida com sucesso.

```
número de vértices:123
número de arestas:155

Porto <--313--> Lisbon
Lisbon <--313--> Porto <--502--> Cordoba
Cordoba <--502--> Lisbon <--206--> Granada
Granada <--206--> Cordoba <--451--> Valencia <--470--> Fez
Valencia <--451--> Granada
Barcelona <--680--> Genoa
Genoa <--680--> Barcelona <--460--> Lyon <--500--> Rome <--370--> Venice
```



UNIVERSIDADE PRESBITERIANA MACKENZIE

Faculdade de Computação e Informática

Prof. Dr. Ivan Carlos Alcântara de Oliveira
Teoria dos Grafos



```
< ----- >
Mapeamento da rota da seda durante o século XIV
-----
Opções de operações:
-----
1. Ler os dados do arquivo.
-----
2. Gravar dados no arquivo.
-----
3. Inserir vértice no Grafo.
-----
4. Inserir aresta no Grafo.
-----
5. Remover vértice do grafo.
-----
6. Remover aresta do grafo.
-----
7. Mostrar conteúdo do arquivo.
-----
8. Mostrar grafo.
-----
9. Conexidade do grafo.
-----
10. Coloração.
-----
11. Grau dos Vértices.
-----
12. Grafo Euleriano.
-----
13. Percurso Euleriano.
-----
14. Grafo Hamiltoniano.
-----
15. Encerrar.
-----
< ----- >
```

Escolha a opção: 6
Vértice de origem: Genoa
Vértice de destino: Lyon
Aresta entre Genoa e Lyon removida com sucesso.

```
número de vértices:123
número de arestas:154
```

```
Porto <--313--> Lisbon
Lisbon <--313--> Porto <--502--> Cordoba
Cordoba <--502--> Lisbon <--206--> Granada
Granada <--206--> Cordoba <--451--> Valencia <--470--> Fez
Valencia <--451--> Granada
Barcelona <--680--> Genoa
Genoa <--680--> Barcelona <--500--> Rome <--370--> Venice
Lyon <--440--> Paris
```

Opção 07: Mostrar Conteúdo do Arquivo



Opção 08: Mostrar grafo

```

n:124 m:157
Porto <--313--> Lisbon <--340--> Cordoba
Lisbon <--313--> Porto <--502--> Cordoba
Cordoba <--502--> Lisbon <--206--> Granada <--340--> Porto
Granada <--206--> Cordoba <--451--> Valencia <--470--> Fez
Valencia <--451--> Granada <--350--> Barcelona
Barcelona <--350--> Valencia <--680--> Genoa
Genoa <--680--> Barcelona <--460--> Lyon <--500--> Rome <--370--> Venice
Lyon <--460--> Genoa <--440--> Paris
Rome <--500--> Genoa <--1550--> Athens
Venice <--370--> Genoa <--1900--> Athens
Paris <--440--> Lyon <--1050--> Prague
Prague <--1050--> Paris <--530--> Krakow
Krakow <--530--> Prague <--860--> Kiev
Kiev <--860--> Krakow <--860--> Crimea
Crimea <--860--> Kiev <--600--> Constantinople <--860--> Rostov <--1500--> Derbent
Athens <--1550--> Rome <--1900--> Venice <--1100--> Alexandria <--1300--> Antioch <--1200--> Tyre
Fez <--470--> Granada <--2100--> Tripoli
Tripoli <--2100--> Fez <--1800--> Alexandria
Alexandria <--1100--> Athens <--1800--> Tripoli <--220--> Cairo <--350--> Tyre
Cairo <--220--> Alexandria <--600--> Qusair
Qusair <--600--> Cairo <--2100--> Suakin <--800--> Leuce
Suakin <--2100--> Qusair <--2100--> Zeila
Zeila <--2100--> Suakin <--250--> Aden
Constantinople <--600--> Crimea <--110--> Bursa <--1350--> Dara
Bursa <--110--> Constantinople <--290--> Sardis
Sardis <--290--> Bursa <--110--> Ephesus <--1400--> Konya
Ephesus <--110--> Sardis <--520--> Alanya
Alanya <--520--> Ephesus <--230--> Konya <--560--> Antioch
Konya <--1400--> Sardis <--230--> Alanya
Antioch <--1300--> Athens <--560--> Alanya <--100--> Aleppo
Aleppo <--100--> Antioch <--280--> Palmyra
Palmyra <--280--> Aleppo <--242--> Damascus <--650--> Bagdad
Damascus <--242--> Palmyra <--144--> Tyre <--270--> Jerusalem
Tyre <--1200--> Athens <--350--> Alexandria <--144--> Damascus

```



UNIVERSIDADE PRESBITERIANA MACKENZIE

Faculdade de Computação e Informática

Prof. Dr. Ivan Carlos Alcântara de Oliveira

Teoria dos Grafos



Tyre <--1200--> Athens <--350--> Alexandria <--144--> Damascus
Jerusalem <--270--> Damascus <--150--> Gaza <--260--> Petra
Gaza <--150--> Jerusalem <--250--> Petra
Petra <--260--> Jerusalem <--250--> Gaza <--1280--> Leuce <--1600--> Medina
Leuce <--800--> Qusair <--1280--> Petra
Dara <--1350--> Constantinople <--30--> Nisibis
Nisibis <--30--> Dara <--650--> Bagdad
Bagdad <--650--> Palmyra <--650--> Nisibis <--452--> Ecbatana <--513--> Qalhat <--1900--> Medina
Trebizond <--240--> Erzurum
Erzurum <--240--> Trebizond <--930--> Baku <--580--> Tabriz
Baku <--930--> Erzurum <--450--> Shaki <--260--> Derbent
Shaki <--450--> Baku
Tbilisi <--560--> Derbent
Derbent <--1500--> Crimea <--260--> Baku <--560--> Tbilisi
Tabriz <--580--> Erzurum <--550--> Ecbatana
Ecbatana <--452--> Bagdad <--550--> Tabriz <--340--> Rayy <--430--> Isfahan
Rostov <--860--> Crimea <--270--> Elista
Elista <--270--> Rostov <--320--> Astrakhan <--2800--> Otrar <--1800--> Khiva
Astrakhan <--320--> Elista
Rayy <--340--> Ecbatana <--420--> Gorgan <--1240--> Nishapur
Gorgan <--420--> Rayy <--880--> Merv
Nishapur <--1240--> Rayy <--510--> Merv
Merv <--880--> Gorgan <--510--> Nishapur <--350--> Bukhara <--600--> Samarkand <--600--> Balkh
Khiva <--1800--> Elista <--390--> Bukhara
Bukhara <--350--> Merv <--390--> Khiva
Otrar <--2800--> Elista <--470--> Kokand
Kokand <--470--> Otrar <--400--> Samarkand <--450--> Kashgar <--1700--> Urumqi
Samarkand <--600--> Merv <--400--> Kokand
Medina <--1600--> Petra <--1900--> Bagdad <--450--> Jeddah <--510--> Mecca
Jeddah <--450--> Medina <--100--> Mecca <--1200--> Muza
Mecca <--510--> Medina <--100--> Jeddah <--1100--> San'a <--2500--> Salalah
San'a <--1100--> Mecca <--400--> Muza <--380--> Aden <--800--> Cana
Muza <--1200--> Jeddah <--400--> San'a <--340--> Aden
Aden <--250--> Zeila <--380--> San'a <--340--> Muza <--540--> Cana <--3600--> Kochi
Cana <--800--> San'a <--540--> Aden <--760--> Salalah
Salalah <--2500--> Mecca <--760--> Cana <--1200--> Muscat
Muscat <--1200--> Salalah <--210--> Sohar
Sohar <--210--> Muscat <--300--> Hormuz
Hormuz <--300--> Sohar <--510--> Siraf <--930--> Isfahan <--1450--> Karachi <--2200--> Balkh
Hormuz <--300--> Sohar <--510--> Siraf <--930--> Isfahan <--1450--> Karachi <--2200--> Balkh
Siraf <--510--> Hormuz <--580--> Apologos
Apologos <--580--> Siraf <--80--> Qalhat
Qalhat <--513--> Bagdad <--80--> Apologos <--90--> Susa
Susa <--90--> Qalhat
Isfahan <--430--> Ecbatana <--930--> Hormuz
Karachi <--1450--> Hormuz <--1000--> Barygaza <--1300--> Taxila
Barygaza <--1000--> Karachi <--880--> Goa
Goa <--880--> Barygaza <--600--> Calicut
Calicut <--600--> Goa <--170--> Kochi
Kochi <--3600--> Aden <--170--> Calicut <--480--> Anuradhapura <--540--> Pondicherry
Anuradhapura <--480--> Kochi
Pondicherry <--540--> Kochi <--150--> Madras
Madras <--150--> Pondicherry <--2400--> Tamralipti
Tamralipti <--2400--> Madras <--1460--> Delhi <--1800--> Pegu
Delhi <--1460--> Tamralipti <--755--> Taxila
Taxila <--1300--> Karachi <--755--> Delhi <--940--> Balkh
Balkh <--600--> Merv <--2200--> Hormuz <--940--> Taxila
Pegu <--1800--> Tamralipti <--2100--> Malacca
Malacca <--2100--> Pegu <--320--> Singapore
Kashgar <--450--> Kokand <--500--> Khotan <--450--> Kuqa
Khotan <--500--> Kashgar <--1800--> Dunhuang
Kuqa <--450--> Kashgar <--670--> Turfan <--1200--> Dunhuang
Turfan <--670--> Kuqa <--240--> Urumqi <--890--> Dunhuang
Urumqi <--1700--> Kokand <--240--> Turfan
Dunhuang <--1800--> Khotan <--1200--> Kuqa <--890--> Turfan <--1900--> Xi'an
Chengu <--870--> Xi'an
Xi'an <--1900--> Dunhuang <--870--> Chengu <--1100--> Shan-du <--490--> Loyang
Shan-du <--1100--> Xi'an <--370--> Beijing
Beijing <--370--> Shan-du <--800--> Loyang <--900--> Pyongyang
Loyang <--490--> Xi'an <--800--> Beijing <--900--> Nanjing <--510--> Wuhan
Pyongyang <--900--> Beijing <--200--> Seoul
Seoul <--200--> Pyongyang <--350--> Busan
Wuhan <--510--> Loyang <--1100--> Zaitun
Nanjing <--900--> Loyang <--900--> Nagasaki <--1000--> Foochow
Busan <--350--> Seoul <--310--> Fukuoka
Fukuoka <--310--> Busan <--150--> Nagasaki
Nagasaki <--900--> Nanjing <--150--> Fukuoka <--700--> Nara
Nara <--700--> Nagasaki <--40--> Osaka



```
Nagasaki <--900--> Nanjing <--130--> Fukuoka <--700--> Nara
Nara <--700--> Nagasaki <--40--> Osaka
Osaka <--40--> Nara
Foochow <--1000--> Nanjing <--180--> Zaitun
Zaitun <--180--> Foochow <--1100--> Wuhan <--780--> Canton <--900--> Bolinao
Canton <--780--> Zaitun <--90--> Macau
Macau <--90--> Canton <--900--> Hoi
Bolinao <--900--> Zaitun <--1300--> Cattigara <--2300--> Bandar
Cattigara <--1300--> Bolinao
Hoi <--900--> Macau <--2700--> Singapore
Bandar <--2300--> Bolinao <--2700--> Singapore
Singapore <--320--> Malacca <--2700--> Hoi <--2700--> Bandar <--600--> Sriwijaya <--110--> Jakarta
Sriwijaya <--600--> Singapore
Jakarta <--110--> Singapore <--220--> Sulawesi
Sulawesi <--220--> Jakarta

fim da impressao do grafo.
```

Opção 09: Conexidade do grafo

<

>

Mapeamento da rota da seda durante o século XIV

Opções de operações:

1. Ler os dados do arquivo.

2. Gravar dados no arquivo.

3. Inserir vértice no Grafo.

4. Inserir aresta no Grafo.

5. Remover vértice do grafo.

6. Remover aresta do grafo.

7. Mostrar conteúdo do arquivo.

8. Mostrar grafo.

9. Conexidade do grafo.

10. Coloração.

11. Grau dos Vértices.

12. Grafo Euleriano.

13. Percurso Euleriano.

14. Grafo Hamiltoniano.

15. Encerrar.

<

>

Escolha a opção: 9

Conexidade do grafo: Grafo é conexo!

Opção 10: Coloração do Grafo



Roteamento da rota da seda durante o século XIV	
Opções de operações	
1. Ler os dados do arquivo.	
2. Gravar dados no arquivo.	
3. Inserir vértice no grafo.	
4. Inserir aresta no grafo.	
5. Remover vértice do grafo.	
6. Remover aresta do grafo.	
7. Mostrar conteúdo do arquivo.	
8. Mostrar grafo.	
9. Conectividade do grafo.	
10. Coloração.	
11. Grau dos Vértices.	
12. Grafo Euleriano.	
13. Percursos Euleriano.	
14. Grafo Hamiltoniano.	
15. Incorpar.	

Selecione a opção: 11

Vértice: 0 Grau: 1, Vértice: 1 Grau: 2, Vértice: 2 Grau: 2, Vértice: 3 Grau: 3, Vértice: 4 Grau: 2, Vértice: 5 Grau: 2, Vértice: 6 Grau: 4, Vértice: 7 Grau: 2, Vértice: 8 Grau: 2, Vértice: 9 Grau: 2, Vértice: 10 Grau: 2, Vértice: 11 Grau: 2, Vértice: 12 Grau: 2, Vértice: 13 Grau: 2, Vértice: 14 Grau: 3, Vértice: 15 Grau: 5, Vértice: 16 Grau: 2, Vértice: 17 Grau: 2, Vértice: 18 Grau: 4, Vértice: 19 Grau: 2, Vértice: 20 Grau: 3, Vértice: 21 Grau: 2, Vértice: 22 Grau: 3, Vértice: 23 Grau: 3, Vértice: 24 Grau: 3, Vértice: 25 Grau: 3, Vértice: 26 Grau: 2, Vértice: 27 Grau: 3, Vértice: 28 Grau: 2, Vértice: 29 Grau: 2, Vértice: 30 Grau: 2, Vértice: 31 Grau: 3, Vértice: 32 Grau: 3, Vértice: 33 Grau: 3, Vértice: 34 Grau: 3, Vértice: 35 Grau: 2, Vértice: 36 Grau: 4, Vértice: 37 Grau: 2, Vértice: 38 Grau: 2, Vértice: 39 Grau: 2, Vértice: 40 Grau: 2, Vértice: 41 Grau: 2, Vértice: 42 Grau: 3, Vértice: 43 Grau: 2, Vértice: 44 Grau: 2, Vértice: 45 Grau: 2, Vértice: 46 Grau: 3, Vértice: 47 Grau: 2, Vértice: 48 Grau: 2, Vértice: 49 Grau: 2, Vértice: 50 Grau: 4, Vértice: 51 Grau: 3, Vértice: 52 Grau: 3, Vértice: 53 Grau: 3, Vértice: 54 Grau: 2, Vértice: 55 Grau: 5, Vértice: 56 Grau: 2, Vértice: 57 Grau: 2, Vértice: 58 Grau: 2, Vértice: 59 Grau: 4, Vértice: 60 Grau: 2, Vértice: 61 Grau: 4, Vértice: 62 Grau: 3, Vértice: 63 Grau: 2, Vértice: 64 Grau: 2, Vértice: 65 Grau: 3, Vértice: 66 Grau: 3, Vértice: 67 Grau: 2, Vértice: 68 Grau: 2, Vértice: 69 Grau: 2, Vértice: 70 Grau: 2, Vértice: 71 Grau: 5, Vértice: 72 Grau: 2, Vértice: 73 Grau: 2, Vértice: 74 Grau: 3, Vértice: 75 Grau: 1, Vértice: 76 Grau: 2, Vértice: 77 Grau: 3, Vértice: 78 Grau: 2, Vértice: 79 Grau: 2, Vértice: 80 Grau: 2, Vértice: 81 Grau: 4, Vértice: 82 Grau: 4, Vértice: 83 Grau: 1, Vértice: 84 Grau: 3, Vértice: 85 Grau: 3, Vértice: 86 Grau: 2, Vértice: 87 Grau: 2, Vértice: 88 Grau: 3, Vértice: 89 Grau: 2, Vértice: 90 Grau: 2, Vértice: 91 Grau: 3, Vértice: 92 Grau: 2, Vértice: 93 Grau: 3, Vértice: 94 Grau: 3, Vértice: 95 Grau: 2, Vértice: 96 Grau: 4, Vértice: 97 Grau: 1, Vértice: 98 Grau: 4, Vértice: 99 Grau: 2, Vértice: 100 Grau: 3, Vértice: 101 Grau: 3, Vértice: 102 Grau: 2, Vértice: 103 Grau: 2, Vértice: 104 Grau: 2, Vértice: 105 Grau: 3, Vértice: 106 Grau: 2, Vértice: 107 Grau: 2, Vértice: 108 Grau: 3, Vértice: 109 Grau: 3, Vértice: 110 Grau: 3, Vértice: 111 Grau: 3, Vértice: 112 Grau: 3, Vértice: 113 Grau: 2, Vértice: 114 Grau: 2, Vértice: 115 Grau: 2, Vértice: 116 Grau: 2, Vértice: 117 Grau: 2, Vértice: 118 Grau: 2, Vértice: 119 Grau: 3, Vértice: 120 Grau: 3, Vértice: 121 Grau: 3, Vértice: 122 Grau: 3, Vértice: 123 Grau: 3, Vértice: 124 Grau: 2, Vértice: 125 Grau: 2, Vértice: 126 Grau: 2, Vértice: 127 Grau: 2, Vértice: 128 Grau: 2, Vértice: 129 Grau: 2, Vértice: 130 Grau: 2, Vértice: 131 Grau: 2, Vértice: 132 Grau: 2, Vértice: 133 Grau: 2, Vértice: 134 Grau: 2, Vértice: 135 Grau: 2, Vértice: 136 Grau: 2, Vértice: 137 Grau: 2, Vértice: 138 Grau: 2, Vértice: 139 Grau: 2, Vértice: 140 Grau: 2, Vértice: 141 Grau: 2, Vértice: 142 Grau: 2, Vértice: 143 Grau: 2, Vértice: 144 Grau: 2, Vértice: 145 Grau: 2, Vértice: 146 Grau: 2, Vértice: 147 Grau: 2, Vértice: 148 Grau: 2, Vértice: 149 Grau: 2, Vértice: 150 Grau: 2, Vértice: 151 Grau: 2, Vértice: 152 Grau: 2, Vértice: 153 Grau: 2, Vértice: 154 Grau: 2, Vértice: 155 Grau: 2, Vértice: 156 Grau: 2, Vértice: 157 Grau: 2, Vértice: 158 Grau: 2, Vértice: 159 Grau: 2, Vértice: 160 Grau: 2, Vértice: 161 Grau: 2, Vértice: 162 Grau: 2, Vértice: 163 Grau: 2, Vértice: 164 Grau: 2, Vértice: 165 Grau: 2, Vértice: 166 Grau: 2, Vértice: 167 Grau: 2, Vértice: 168 Grau: 2, Vértice: 169 Grau: 2, Vértice: 170 Grau: 2, Vértice: 171 Grau: 2, Vértice: 172 Grau: 2, Vértice: 173 Grau: 2, Vértice: 174 Grau: 2, Vértice: 175 Grau: 2, Vértice: 176 Grau: 2, Vértice: 177 Grau: 2, Vértice: 178 Grau: 2, Vértice: 179 Grau: 2, Vértice: 180 Grau: 2, Vértice: 181 Grau: 2, Vértice: 182 Grau: 2, Vértice: 183 Grau: 2, Vértice: 184 Grau: 2, Vértice: 185 Grau: 2, Vértice: 186 Grau: 2, Vértice: 187 Grau: 2, Vértice: 188 Grau: 2, Vértice: 189 Grau: 2, Vértice: 190 Grau: 2, Vértice: 191 Grau: 2, Vértice: 192 Grau: 2, Vértice: 193 Grau: 2, Vértice: 194 Grau: 2, Vértice: 195 Grau: 2, Vértice: 196 Grau: 2, Vértice: 197 Grau: 2, Vértice: 198 Grau: 2, Vértice: 199 Grau: 2, Vértice: 200 Grau: 2, Vértice: 201 Grau: 2, Vértice: 202 Grau: 2, Vértice: 203 Grau: 2, Vértice: 204 Grau: 2, Vértice: 205 Grau: 2, Vértice: 206 Grau: 2, Vértice: 207 Grau: 2, Vértice: 208 Grau: 2, Vértice: 209 Grau: 2, Vértice: 210 Grau: 2, Vértice: 211 Grau: 2, Vértice: 212 Grau: 2, Vértice: 213 Grau: 2, Vértice: 214 Grau: 2, Vértice: 215 Grau: 2, Vértice: 216 Grau: 2, Vértice: 217 Grau: 2, Vértice: 218 Grau: 2, Vértice: 219 Grau: 2, Vértice: 220 Grau: 2, Vértice: 221 Grau: 2, Vértice: 222 Grau: 2, Vértice: 223 Grau: 2, Vértice: 224 Grau: 2, Vértice: 225 Grau: 2, Vértice: 226 Grau: 2, Vértice: 227 Grau: 2, Vértice: 228 Grau: 2, Vértice: 229 Grau: 2, Vértice: 230 Grau: 2, Vértice: 231 Grau: 2, Vértice: 232 Grau: 2, Vértice: 233 Grau: 2, Vértice: 234 Grau: 2, Vértice: 235 Grau: 2, Vértice: 236 Grau: 2, Vértice: 237 Grau: 2, Vértice: 238 Grau: 2, Vértice: 239 Grau: 2, Vértice: 240 Grau: 2, Vértice: 241 Grau: 2, Vértice: 242 Grau: 2, Vértice: 243 Grau: 2, Vértice: 244 Grau: 2, Vértice: 245 Grau: 2, Vértice: 246 Grau: 2, Vértice: 247 Grau: 2, Vértice: 248 Grau: 2, Vértice: 249 Grau: 2, Vértice: 250 Grau: 2, Vértice: 251 Grau: 2, Vértice: 252 Grau: 2, Vértice: 253 Grau: 2, Vértice: 254 Grau: 2, Vértice: 255 Grau: 2, Vértice: 256 Grau: 2, Vértice: 257 Grau: 2, Vértice: 258 Grau: 2, Vértice: 259 Grau: 2, Vértice: 260 Grau: 2, Vértice: 261 Grau: 2, Vértice: 262 Grau: 2, Vértice: 263 Grau: 2, Vértice: 264 Grau: 2, Vértice: 265 Grau: 2, Vértice: 266 Grau: 2, Vértice: 267 Grau: 2, Vértice: 268 Grau: 2, Vértice: 269 Grau: 2, Vértice: 270 Grau: 2, Vértice: 271 Grau: 2, Vértice: 272 Grau: 2, Vértice: 273 Grau: 2, Vértice: 274 Grau: 2, Vértice: 275 Grau: 2, Vértice: 276 Grau: 2, Vértice: 277 Grau: 2, Vértice: 278 Grau: 2, Vértice: 279 Grau: 2, Vértice: 280 Grau: 2, Vértice: 281 Grau: 2, Vértice: 282 Grau: 2, Vértice: 283 Grau: 2, Vértice: 284 Grau: 2, Vértice: 285 Grau: 2, Vértice: 286 Grau: 2, Vértice:

Opção 12: Grafo Euleriano



UNIVERSIDADE PRESBITERIANA MACKENZIE

Faculdade de Computação e Informática

Prof. Dr. Ivan Carlos Alcântara de Oliveira
Teoria dos Grafos



```
< ----- >
Mapeamento da rota da seda durante o século XIV
-----
Opções de operações:
-----
1. Ler os dados do arquivo.
-----
2. Gravar dados no arquivo.
-----
3. Inserir vértice no Grafo.
-----
4. Inserir aresta no Grafo.
-----
5. Remover vértice do grafo.
-----
6. Remover aresta do grafo.
-----
7. Mostrar conteúdo do arquivo.
-----
8. Mostrar grafo.
-----
9. Conexidade do grafo.
-----
10. Coloração.
-----
11. Grau dos Vértices.
-----
12. Grafo Euleriano.
-----
13. Percurso Euleriano.
-----
14. Grafo Hamiltoniano.
-----
15. Encerrar.
-----
> ----- <

Escolha a opção: 12
O grafo não é Euleriano.
```

Opção 13: Percurso Euleriano



UNIVERSIDADE PRESBITERIANA MACKENZIE

Faculdade de Computação e Informática

Prof. Dr. Ivan Carlos Alcântara de Oliveira
Teoria dos Grafos



<

Mapeamento da rota da seda durante o século XIV

>

Opções de operações:

1. Ler os dados do arquivo.

2. Gravar dados no arquivo.

3. Inserir vértice no Grafo.

4. Inserir aresta no Grafo.

5. Remover vértice do grafo.

6. Remover aresta do grafo.

7. Mostrar conteúdo do arquivo.

8. Mostrar grafo.

9. Conexidade do grafo.

10. Coloração.

11. Grau dos Vértices.

12. Grafo Euleriano.

13. Percurso Euleriano.

14. Grafo Hamiltoniano.

15. Encerrar.

<

Escolha a opção: 13

0 grafo não possui percurso Euleriano.

>

Opção 14: Grafo Hamiltoniano



UNIVERSIDADE PRESBITERIANA MACKENZIE

Faculdade de Computação e Informática

Prof. Dr. Ivan Carlos Alcântara de Oliveira
Teoria dos Grafos



```
< ----- >
Mapeamento da rota da seda durante o século XIV
-----
Opções de operações:
-----
1. Ler os dados do arquivo.
-----
2. Gravar dados no arquivo.
-----
3. Inserir vértice no Grafo.
-----
4. Inserir aresta no Grafo.
-----
5. Remover vértice do grafo.
-----
6. Remover aresta do grafo.
-----
7. Mostrar conteúdo do arquivo.
-----
8. Mostrar grafo.
-----
9. Conexidade do grafo.
-----
10. Coloração.
-----
11. Grau dos Vértices.
-----
12. Grafo Euleriano.
-----
13. Percurso Euleriano.
-----
14. Grafo Hamiltoniano.
-----
15. Encerrar.
-----
< ----- >

Escolha a opção: 14
O grafo não é Hamiltoniano.
```

Opção 15: Encerrar



UNIVERSIDADE PRESBITERIANA MACKENZIE

Faculdade de Computação e Informática

Prof. Dr. Ivan Carlos Alcântara de Oliveira
Teoria dos Grafos



```
< ----- >
Mapeamento da rota da seda durante o século XIV
-----
Opções de operações:
-----
1. Ler os dados do arquivo.
-----
2. Gravar dados no arquivo.
-----
3. Inserir vértice no Grafo.
-----
4. Inserir aresta no Grafo.
-----
5. Remover vértice do grafo.
-----
6. Remover aresta do grafo.
-----
7. Mostrar conteúdo do arquivo.
-----
8. Mostrar grafo.
-----
9. Conexidade do grafo.
-----
10. Coloração.
-----
11. Grau dos Vértices.
-----
12. Grafo Euleriano.
-----
13. Percurso Euleriano.
-----
14. Grafo Hamiltoniano.
-----
15. Encerrar.
-----
>

Escolha a opção: 15
Fim do programa
```



Arquivos Fontes

Conteúdo arquivo Main

"""

Mateus Fernandes Castanharo - 32137141

Victor Hugo Antonio Couto - 32173482

O projeto consiste em representar a rota comercial da seda, uma das rotas mais famosas durante a idade média,

durante o século XIV, utilizando os conceitos de teoria dos grafos.

O grafo modulado contém as principais cidades que faziam parte, que contemplam os continentes: Europa e Ásia,

e as respectivas distância aproximadas entre as cidades.

O principal objetivo do projeto é de fornecer informações sobre a rota da seda de uma maneira dinâmica e intuitiva.

"""

```
from grafolmp import Grafo
import time
```

```
def grafoArq(file):
    with open(file) as f:
        content = f.readlines() # Salva arquivo inteiro
        tipoGrafo = int(content[0]) # Primeira linha do arquivo
        qtdVertices = int(content[1]) # Segunda linha do arquivo
        qtdArestas = int(content[qtdVertices+2]) # Arestas
        vertices = [] # Todos os vertices
        arestas = [] # Todas as arestas

        # loop do primeiro vertice até o ultimo
        for line in range(2, qtdVertices+2):
            vertices.append(content[line].split(" ")) # Guarda em uma lista
```



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática

Prof. Dr. Ivan Carlos Alcântara de Oliveira

Teoria dos Grafos



```
# loop do final dos vertices até o final do arquivo
for line in range(qtdVertices+3, len(content)):
    arestas.append(content[line].split(" ")) # Guarda em uma lista
```

```
# Cria o grafo com todos os vertices
g = Grafo(qtdVertices)
```

```
for i in range(qtdVertices): # armazena os nomes
    g.nomes[i] = vertices[i][1].rstrip('\n')
```

```
for i in range(qtdArestas): # Adiciona todas as arestas e pesos
```

```
g.insereAresta(int(arestas[i][0]), int(
    arestas[i][1]), int(arestas[i][2]), 1)
```

```
g.typeGraph = tipoGrafo
print("Arquivo lido com sucesso.")
```

```
return g
```

```
def writeFile(graph):
    graph.escreverArquivo("grafo.txt")
```

```
def inserirVertice(graph):
    nomeVertice = input("Informe o nome do vértice: ")
    graph.insereVertice(nomeVertice)
```

```
def inserirAresta(graph):
    verticeOrigem = input("Vértice de origem: ")
    verticeDestino = input("Vértice de destino: ")
    peso = int(input("Peso da aresta: "))
    graph.insereAresta(verticeOrigem, verticeDestino, peso, 4)
```

```
def removerVertice(graph):
```



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática

Prof. Dr. Ivan Carlos Alcântara de Oliveira
Teoria dos Grafos



```
nomeVertice = input("Informe o nome do vértice: ")  
graph.removeVertice(nomeVertice)
```

```
def removerAresta(graph):  
    verticeOrigem = input("Vértice de origem: ")  
    verticeDestino = input("Vértice de destino: ")  
    graph.removeAresta(verticeOrigem, verticeDestino)
```

```
def showContent(graph):  
    print(f"Conteúdo do arquivo:\nNúmero de vértices: {graph.n}.\nNúmero de arestas:  
{graph.m}.\nCidades: {graph.nomes}.\nDistância entre as cidades: {graph.pesos}.\n\n")
```

```
match graph.typeGraph:  
    case 0:  
        print("Grafo não orientado sem peso!")  
    case 1:  
        print("Grafo não orientado com peso no vértice!")  
    case 2:  
        print("Grafo não orientado com peso na aresta!")  
    case 3:  
        print("Grafo não orientado com peso nos vértices e arestas!")  
    case 4:  
        print("Grafo orientado sem peso!")  
    case 5:  
        print("Grafo orientado com peso no vértice!")  
    case 6:  
        print("Grafo orientado com peso na aresta!")  
    case 7:  
        print("Grafo orientado com peso nos vértices e arestas!")  
    case _:  
        print("Não é uma opção valida")  
        exit()
```

```
def conexidadeGrafo(graph):  
    print("Conexidade do grafo: ", end="")  
    if len(graph.busca_profundidade(1)) != graph.n:  
        print("Grafo é desconexo!")  
    else:  
        print("Grafo é conexo!")
```



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática

Prof. Dr. Ivan Carlos Alcântara de Oliveira
Teoria dos Grafos



```
def coloracaoGrafo(graph):  
    print("Coloração do grafo:\n")  
    graph.coloracao_sequencial()
```

```
def euleriano(graph):  
    print("O grafo é Euleriano" if graph.grafoEuleriano() == 1 else "O grafo não é Euleriano.")
```

```
def perEuleriano(graph):  
    print("O grafo possui percurso Euleriano" if graph.percursoEuleriano() == 1 else "O grafo  
    não possui percurso Euleriano.")
```

```
def grauDosvertices(graph):  
    graph.imprimirGrauVertices()
```

```
def hamiltoniano(graph):  
    print("O grafo é Hamiltoniano" if graph.grafoHamiltoniano() == 1 else "O grafo não é  
    Hamiltoniano.")
```

```
def menu():  
    graph = 0  
    while(True):  
        print('\n\n', "<", "-*60, ">")  
        print(" |", " *5, "Mapeamento da rota da seda durante o século XIV", " *6, "|", "\n", "|",  
        "-*60, "|", "\n",  
        "|", " *20, "Opções de operações:", " " *  
        17, "|", "\n", "|", "-*60, "|", "\n",  
        "|", " *15, "1. Ler os dados do arquivo.", " " *  
        16, "|", "\n", "|", "-*60, "|", "\n",  
        "|", " *15, "2. Gravar dados no arquivo.", " " *  
        16, "|", "\n", "|", "-*60, "|", "\n",  
        "|", " *15, "3. Inserir vértice no Grafo.", " " *  
        15, "|", "\n", "|", "-*60, "|", "\n",  
        "|", " *15, "4. Inserir aresta no Grafo.", " " *  
        16, "|", "\n", "|", "-*60, "|", "\n",  
        "|", " *15, "5. Remover vértice do grafo.", " " *  
        15, "|", "\n", "|", "-*60, "|", "\n",  
        "|", " *15, "6. Remover aresta do grafo.", " " *  
        16, "|", "\n", "|", "-*60, "|", "\n",
```



UNIVERSIDADE PRESBITERIANA MACKENZIE

Faculdade de Computação e Informática

Prof. Dr. Ivan Carlos Alcântara de Oliveira

Teoria dos Grafos



```
"|", " "*15, "7. Mostrar conteúdo do arquivo.", " " *  
12, "|", "\n", "|", "-"*60, "|", "\n",  
"|", " "*15, "8. Mostrar grafo.", " "*26, "|", "\n", "|", "-"*60, "|", "\n",  
"|", " "*15, "9. Conexidade do grafo.", " " *  
20, "|", "\n", "|", "-"*60, "|", "\n",  
"|", " "*15, "10. Coloração.", " " *  
29, "|", "\n", "|", "-"*60, "|", "\n",  
"|", " "*15, "11. Grau dos Vértices.", " " *  
21, "|", "\n", "|", "-"*60, "|", "\n",  
"|", " "*15, "12. Grafo Euleriano.", " " *  
23, "|", "\n", "|", "-"*60, "|", "\n",  
"|", " "*15, "13. Percurso Euleriano.", " " *  
20, "|", "\n", "|", "-"*60, "|", "\n",  
"|", " "*15, "14. Grafo Hamiltoniano.", " " *  
20, "|", "\n", "|", "-"*60, "|", "\n",  
"|", " "*15, "15. Encerrar.", " "*30, "|", "\n", "<", "-"*60, ">", "\n",  
)
```

```
opcao = int(input("Escolha a opção: "))
```

match opcao:

case 1:

```
graph = grafoArq("grafo.txt")  
time.sleep(1)
```

case 2:

```
writeFile(graph)  
time.sleep(1)
```

case 3:

```
inserirVertice(graph)  
time.sleep(1)
```

case 4:

```
inserirAresta(graph)  
time.sleep(1)
```

case 5:

```
removerVertice(graph)  
time.sleep(1)
```

case 6:

```
removerAresta(graph)  
time.sleep(1)
```

case 7:

```
showContent(graph)  
time.sleep(1)
```

case 8:

```
print("\n\n", "-"*20, " Apresentação do grafo ", "-"*20)  
graph.show()  
time.sleep(3)
```

case 9:



UNIVERSIDADE PRESBITERIANA MACKENZIE

Faculdade de Computação e Informática

Prof. Dr. Ivan Carlos Alcântara de Oliveira
Teoria dos Grafos



```
    conexidadeGrafo(graph)
    time.sleep(1)
case 10:
    coloracaoGrafo(graph)
    time.sleep(1)
case 11:
    grauDosvertices(graph)
    time.sleep(1)
case 12:
    euleriano(graph)
    time.sleep(1)
case 13:
    perEuleriano(graph)
    time.sleep(1)
case 14:
    hamiltoniano(graph)
    time.sleep(1)
case 15:
    print("Fim do programa")
    time.sleep(1)
    break
case _:
    print("Opção inválida!")
    time.sleep(1)
```

```
def main():
    menu()
```

```
main()
```

Conteúdo arquivo graphImp

```
"""
```

```
    Mateus Fernandes Castanharo - 32137141
    Victor Hugo Antonio Couto - 32173482
```

```
"""
```

```
class Grafo:
```



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática

Prof. Dr. Ivan Carlos Alcântara de Oliveira
Teoria dos Grafos



```
def __init__(self, n):
    self.n = n
    self.m = 0
    self.pesos = {}
    self.nomes = {}
    self.listaAdj = [[] for i in range(self.n)]
```

```
def escreverArquivo(self, arq):
    with open(arq, 'r') as f:
        content = []
        content.append(str(2)+'\n')
        content.append(str(self.n) + '\n')

        for i in self.nomes:
            content.append(str(i) + ' ' + self.nomes[i] + '\n')
```

```
        content.append(str(self.m) + '\n')
```

```
        listaAux = []
        indexAux = self.n + 3
        print(indexAux)
        print(content)
```

```
        for i, valor in self.pesos.items():
            if i[1] > i[0]:
                listaAux.append((i, valor))
                content.append(str(listaAux[0][0]).replace("(", "").replace(")", "").replace(",", "") +
                ' ' + str(listaAux[0][1]) + '\n')
                indexAux += 1
                listaAux.clear()
```

```
        with open(arq, 'w') as f:
            f.writelines(content)
```

```
def insereVertice(self, v):
    lista = []
    idVertice = self.n
```

```
    for i in self.nomes:
        if v == self.nomes[i]:
            print("Vértice já existente.")
            return
```



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática

Prof. Dr. Ivan Carlos Alcântara de Oliveira
Teoria dos Grafos



```
self.nomes[idVertice] = v
self.listaAdj.append(lista)
self.n += 1
```

```
print(f"\nVértice adicionado com sucesso.")
```

```
def removerVertice(self, v):
    if v not in self.nomes.values():
        print(f"Vértice {v} não encontrado.")
        return

    for i in range(self.n):
        if self.nomes[i] == v:
            for adjacente in self.listaAdj[i]:
                self.listaAdj[adjacente].remove(i)
            del self.pesos[i, adjacente]
            del self.pesos[adjacente, i]
```

```
indice_v = None
for i in range(self.n):
    if self.nomes[i] == v:
        indice_v = i
        break
```

```
chaves = list(self.nomes.keys())
```

```
id_removido = None
```

```
for chave in chaves:
    id_vertice = self.nomes[chave]
    if id_vertice == v:
        id_removido = chave
        break

if id_removido is not None:
    for chave in chaves:
        if chave > id_removido:
            novo_id = chave - 1
            self.nomes[novo_id] = self.nomes[chave]
            del self.nomes[chave]
    del self.listaAdj[id_removido]
```

```
for i in range(len(self.listaAdj)):
```



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática

Prof. Dr. Ivan Carlos Alcântara de Oliveira
Teoria dos Grafos



```
if i >= id_removido:
    for j in range(len(self.listaAdj[i])):
        novo = self.listaAdj[i][j] - 1
        self.listaAdj[i][j] = novo
```

```
else:
    print(f"Vértice {v} não encontrado.")
```

```
if indice_v is not None:
    chaves = list(self.pesos.keys())
    for chave in chaves:
        (valor1, valor2) = chave
        if valor1 > indice_v or valor2 > indice_v:
            novo_valor1 = valor1 if valor1 <= indice_v else valor1 - 1
            novo_valor2 = valor2 if valor2 <= indice_v else valor2 - 1
            valor = self.pesos[chave]
            del self.pesos[chave]
            nova_chave = (novo_valor1, novo_valor2)
            self.pesos[nova_chave] = valor
```

```
self.n -= 1
print(f"Vértice {v} removido com sucesso.")
```

```
def insereAresta(self, v, w, peso, category):
    if category == 1:
        self.listaAdj[v].append(w)
        self.listaAdj[w].append(v)
        self.pesos[v, w] = peso
        self.pesos[w, v] = peso
        self.m += 1
    else:
        for i in range(self.n):
            if self.nomes[i] == v:
                indexV = i
            if self.nomes[i] == w:
                indexW = i

        self.listaAdj[indexV].append(indexW)
        self.listaAdj[indexW].append(indexV)
        self.pesos[indexV, indexW] = peso
        self.pesos[indexW, indexV] = peso
        self.m += 1
        print(f"Aresta entre {v} e {w} inserida com sucesso.")
```

```
def removeAresta(self, v, w):
```



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática

Prof. Dr. Ivan Carlos Alcântara de Oliveira

Teoria dos Grafos



```
if (v not in self.nomes.values()) or (w not in self.nomes.values()):  
    print(f"Vértice {v} ou {w} não encontrado.")  
    return
```

```
for i in range(self.n):  
    if self.nomes[i] == v:  
        indexV = i  
    if self.nomes[i] == w:  
        indexW = i
```

```
self.listaAdj[indexV].remove(indexW)  
self.listaAdj[indexW].remove(indexV)  
del self.pesos[indexV, indexW]  
del self.pesos[indexW, indexV]  
self.m -= 1  
print(f"Aresta entre {v} e {w} removida com sucesso.")
```

```
def show(self):  
    print(f"\n número de vértices:{self.n:2d}", end="")  
    print(f"\n número de arestas:{self.m:2d}")
```

```
vertices_validos = [i for i in range(self.n)]  
index = 0  
for i in vertices_validos:  
    vertex_name = self.nomes[i]  
    print(f"\n{vertex_name}", end="")  
    for neighbor in self.listaAdj[index]:  
        if (neighbor >= self.n) or (neighbor < 0):  
            continue  
        neighbor_name = self.nomes[neighbor]  
        if index == neighbor:  
            continue  
        print(f" <-- {self.pesos[i, neighbor]} --> {neighbor_name}", end="")  
    index += 1
```

```
print("\n\nfim da impressao do grafo.")
```

```
def busca_profundidade(self, vertice_inicial):  
    visitados = set()
```

```
def dfs(vertice):  
    visitados.add(vertice)  
    for vizinho in self.listaAdj[vertice]:
```



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática

Prof. Dr. Ivan Carlos Alcântara de Oliveira
Teoría dos Grafos



```
if vizinho not in visitados:  
    dfs(vizinho)
```

```
dfs(vertice_inicial)  
return visitados
```

```
def coloracao_sequencial(self):  
    cores = [-1] * self.n # Inicializa todas as cores como -1 (não atribuídas)
```

```
for vertice in range(self.n):  
    cores_disponiveis = set(range(self.n))
```

```
for vizinho in self.listaAdj[vertice]:  
    if cores[vizinho] != -1:  
        cores_disponiveis.discard(cores[vizinho])
```

```
cor = min(cores_disponiveis)  
cores[vertice] = cor
```

```
self.imprimir_coloracao(cores)
```

```
def imprimir_coloracao(self, cores):  
    mesmasCores = {}  
    for vertice, cor in enumerate(cores):  
        mesmasCores.setdefault(cor, []).append(vertice)
```

```
for cor, vertices in mesmasCores.items():  
    print(f"\n\nCor {cor}:")  
    for vertice in vertices:  
        nome_vertice = self.nomes.get(vertice, f"Vértice {vertice}")  
        print(f" {nome_vertice}", end="")
```

```
def grauVertice(self, vertice):  
    return len(self.listaAdj[vertice])
```

```
def imprimirGrauVertices(self):  
    for vertice in range(self.n):  
        grau = self.grauVertice(vertice)  
        print(f"Vértice: {vertice} Grau: {grau}, ", end="")
```

```
def grafoEuleriano(self):
```



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática

Prof. Dr. Ivan Carlos Alcântara de Oliveira
Teoria dos Grafos



graulmpar = 0

```
for vertice in self.listaAdj:
    if len(vertice) % 2 != 0:
        graulmpar += 1
```

return graulmpar == 0 or graulmpar == 2

```
def percursoEuleriano(self):
    if not self.busca_profundidade(0):
        return False
```

```
    graus_impares = 0
    for vertice in range(self.n):
        if self.grauVertice(vertice) % 2 != 0:
            graus_impares += 1
```

return graus_impares == 0 or graus_impares == 2

```
def grafoHamiltoniano(self):
    if self.n < 3:
        return True
```

```
    if not self.busca_profundidade(0):
        return False
```

```
    for i in range(self.n): # Teorema de Ore
        for j in range(i + 1, self.n):
            if i not in self.listaAdj[j] and self.grauVertice(i) + self.grauVertice(j) < self.n:
                return False
    return True
```

Conteúdo do grafo.txt:

2
123
0 Porto
1 Lisbon
2 Cordoba
3 Granada
4 Valencia



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática

Prof. Dr. Ivan Carlos Alcântara de Oliveira
Teoria dos Grafos



- 5 Barcelona
- 6 Genoa
- 7 Lyon
- 8 Rome
- 9 Venice
- 10 Paris
- 11 Prague
- 12 Krakow
- 13 Kiev
- 14 Crimea
- 15 Athens
- 16 Fez
- 17 Tripoli
- 18 Alexandria
- 19 Cairo
- 20 Qusair
- 21 Suakin
- 22 Zeila
- 23 Constantinople
- 24 Bursa
- 25 Sardis
- 26 Ephesus
- 27 Alanya
- 28 Konya
- 29 Antioch
- 30 Aleppo
- 31 Palmyra
- 32 Damascus
- 33 Tyre
- 34 Jerusalem
- 35 Gaza
- 36 Petra
- 37 Leuce
- 38 Dara
- 39 Nisibis
- 40 Bagdad
- 41 Trebizond
- 42 Erzurum
- 43 Baku
- 44 Shaki
- 45 Tbilisi
- 46 Derbent
- 47 Tabriz
- 48 Ecbatana
- 49 Rostov
- 50 Elista
- 51 Astrakhan
- 52 Rayy



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática

Prof. Dr. Ivan Carlos Alcântara de Oliveira
Teoria dos Grafos



- 53 Gorgan
- 54 Nishapur
- 55 Merv
- 56 Khiva
- 57 Bukhara
- 58 Otrar
- 59 Kokand
- 60 Samarkand
- 61 Medina
- 62 Jeddah
- 63 Mecca
- 64 San'a
- 65 Muza
- 66 Aden
- 67 Cana
- 68 Salalah
- 69 Muscat
- 70 Sohar
- 71 Hormuz
- 72 Siraf
- 73 Apologos
- 74 Qalhat
- 75 Susa
- 76 Isfahan
- 77 Karachi
- 78 Barygaza
- 79 Goa
- 80 Calicut
- 81 Kochi
- 82 Anuradhapura
- 83 Pondicherry
- 84 Madras
- 85 Tamralipti
- 86 Delhi
- 87 Taxila
- 88 Balkh
- 89 Pegu
- 90 Malacca
- 91 Kashgar
- 92 Khotan
- 93 Kuqa
- 94 Turfan
- 95 Urumqi
- 96 Dunhuang
- 97 Chengu
- 98 Xi'an
- 99 Shan-du
- 100 Beijing



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática

Prof. Dr. Ivan Carlos Alcântara de Oliveira
Teoria dos Grafos



101 Loyang
102 Pyongyang
103 Seoul
104 Wuhan
105 Nanjing
106 Busan
107 Fukuoka
108 Nagasaki
109 Nara
110 Osaka
111 Foochow
112 Zaitun
113 Canton
114 Macau
115 Bolinao
116 Cattigara
117 Hoi
118 Bandar
119 Singapore
120 Sriwijaya
121 Jakarta
122 Sulawesi
156
0 1 313
1 2 502
2 3 206
3 4 451
3 16 470
4 5 350
5 6 680
6 7 460
6 8 500
6 9 370
7 10 440
8 15 1550
9 15 1900
10 11 1050
11 12 530
12 13 860
13 14 860
14 23 600
14 49 860
14 46 1500
15 18 1100
15 29 1300
15 33 1200
16 17 2100
17 18 1800



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática

Prof. Dr. Ivan Carlos Alcântara de Oliveira
Teoría dos Grafos



18 19 220
18 33 350
19 20 600
20 21 2100
20 37 800
21 22 2100
22 66 250
23 24 110
23 38 1350
24 25 290
25 26 110
25 28 1400
26 27 520
27 28 230
27 29 560
29 30 100
30 31 280
31 32 242
31 40 650
32 33 144
32 34 270
34 35 150
34 36 260
35 36 250
36 37 1280
36 61 1600
38 39 30
39 40 650
40 48 452
40 74 513
40 61 1900
41 42 240
42 43 930
42 47 580
43 44 450
43 46 260
45 46 560
47 48 550
48 52 340
48 76 430
49 50 270
50 51 320
50 58 2800
50 56 1800
52 53 420
52 54 1240
53 55 880
54 55 510



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática

Prof. Dr. Ivan Carlos Alcântara de Oliveira
Teoria dos Grafos



55 57 350
55 60 600
55 88 600
56 57 390
58 59 470
59 60 400
59 91 450
59 95 1700
61 62 450
61 63 510
62 63 100
62 65 1200
63 64 1100
63 68 2500
64 65 400
64 66 380
64 67 800
65 66 340
66 67 540
66 81 3600
67 68 760
68 69 1200
69 70 210
70 71 300
71 72 510
71 76 930
71 77 1450
71 88 2200
72 73 580
73 74 80
74 75 90
77 78 1000
77 87 1300
78 79 880
79 80 600
80 81 170
81 82 480
81 83 540
83 84 150
84 85 2400
85 86 1460
85 89 1800
86 87 755
87 88 940
89 90 2100
90 119 320
91 92 500
91 93 450



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática

Prof. Dr. Ivan Carlos Alcântara de Oliveira
Teoria dos Grafos



92 96 1800
93 94 670
93 96 1200
94 95 240
94 96 890
96 98 1900
97 98 870
98 99 1100
98 101 490
99 100 370
100 101 800
100 102 900
101 105 900
101 104 510
102 103 200
103 106 350
105 108 900
105 111 1000
106 107 310
107 108 150
108 109 700
109 110 40
111 112 180
104 112 1100
112 113 780
112 115 900
113 114 90
114 117 900
115 116 1300
115 118 2300
117 119 2700
118 119 2700
119 120 600
119 121 110
121 122 220

Link do GitHub: <https://github.com/MateusFernandesCastanhara/Teoria-de-Grafos>