

# Prova 3 AEDS - Problema do Caixeiro Viajante

Autores: Amanda da Costa Vasconcelos de Moura e Diego Ascânio Santos

## Parte 1: Método da Força Bruta

O método da força bruta verifica cada uma das  $(n - 1)!$  permutações possíveis em um grafo de  $n$  cidades para se percorrer um caminho mínimo passando por todas as cidades com início e fim na cidade de origem.

Aqui geraremos 11 grafos completos (de 2 a 12 vértices) com pesos inteiros aleatórios entre 1 e 100, representados por matrizes de adjacências simétricas, com diagonais principais nulas (ou seja, nenhum vértice possui arestas para si próprio)

Imprimiremos estes grafos e depois, executaremos a resolução do caixeiro viajante pelo método da força bruta, retornando o caminho mínimo encontrado, seu custo e quantos passos foram necessários para calculá-lo.

Em seguida plotaremos um gráfico relacionando o número de cidades  $n$  com as respectivas quantidade de passos necessárias para calcular o caminho mínimo

```
In [1]: %matplotlib inline
from matplotlib import pyplot
from prova_3_utils import gerar_adjacencia_simetrica_aleatoria, repr_matriz, carregar_grafo_tsp
from TSP import G, TSP
import numpy as np
import time
custos_computacionais = []

for N in range(2, 13):
    begin = time.time()
    adjacencias = gerar_adjacencia_simetrica_aleatoria(N)
    g = G(grafo = adjacencias)
    tsp = TSP(g = g)
    print('N: {}Adjacencias:\n{}'.format(N, repr_matriz(adjacencias)))
    caminho_minimo, custo_caminho_minimo, passos = tsp.solve(heuristic = False)
    custos_computacionais.append(passos)
    print('Caminho Mínimo: {}nCusto do Caminho: {}nCusto Computacional: {}'.format(caminho_minimo,
                                                                                    custo_caminho_minimo,
                                                                                    passos))

    print('Tempo de execução: {:.2f}s\n'.format(time.time() - begin))

pyplot.title('Cidades x Custo Computacional Para Calcular o Caminho Mínimo\n' +
             'Eixo Y em Escala Logarítmica')
pyplot.yscale('log')
pyplot.plot(custos_computacionais, 'bo');
```

N: 2

Adjacencias:

	0	1
0	0	60
1	60	0

Caminho Mínimo: [0 1 0]

Custo do Caminho: 120

Custo Computacional: 1

Tempo de execução: 0.00s

N: 3

Adjacencias:

	0	1	2
0	0	97	41
1	97	0	1
2	41	1	0

Caminho Mínimo: [0 1 2 0]

Custo do Caminho: 139

Custo Computacional: 2

Tempo de execução: 0.00s

N: 4

Adjacencias:

	0	1	2	3
0	0	93	49	38
1	93	0	13	47
2	49	13	0	1
3	38	47	1	0

Caminho Mínimo: [0 1 2 3 0]

Custo do Caminho: 145  
Custo Computacional: 6  
Tempo de execução: 0.00s

N: 5

Adjacencias:

	0	1	2	3	4
0	0	20	15	51	16
1	20	0	79	61	83
2	15	79	0	48	50
3	51	61	48	0	68
4	16	83	50	68	0

Caminho Mínimo: [0 1 3 2 4 0]  
Custo do Caminho: 195  
Custo Computacional: 24  
Tempo de execução: 0.00s

N: 6

Adjacencias:

	0	1	2	3	4	5
0	0	91	61	61	99	29
1	91	0	5	33	56	92
2	61	5	0	83	24	99
3	61	33	83	0	87	14
4	99	56	24	87	0	17
5	29	92	99	14	17	0

Caminho Mínimo: [0 3 1 2 4 5 0]  
Custo do Caminho: 169  
Custo Computacional: 120  
Tempo de execução: 0.00s

N: 7

Adjacencias:

	0	1	2	3	4	5	6
0	0	49	2	41	38	99	34
1	49	0	23	61	56	3	55
2	2	23	0	27	70	39	86
3	41	61	27	0	50	95	86
4	38	56	70	50	0	72	67
5	99	3	39	95	72	0	27
6	34	55	86	86	67	27	0

Caminho Mínimo: [0 2 3 4 1 5 6 0]  
Custo do Caminho: 199  
Custo Computacional: 720  
Tempo de execução: 0.01s

N: 8

Adjacencias:

	0	1	2	3	4	5	6	7
0	0	98	42	8	38	41	98	47
1	98	0	92	4	45	91	12	97
2	42	92	0	83	96	93	28	2
3	8	4	83	0	80	51	71	48
4	38	45	96	80	0	82	64	60
5	41	91	93	51	82	0	8	24
6	98	12	28	71	64	8	0	51
7	47	97	2	48	60	24	51	0

Caminho Mínimo: [0 3 1 6 5 7 2 4 0]  
Custo do Caminho: 192  
Custo Computacional: 5040  
Tempo de execução: 0.04s

N: 9

Adjacencias:

	0	1	2	3	4	5	6	7	8
0	0	30	73	33	27	88	18	86	66
1	30	0	51	89	71	65	91	86	89
2	73	51	0	30	94	11	55	49	56
3	33	89	30	0	47	74	67	98	99
4	27	71	94	47	0	88	33	80	12
5	88	65	11	74	88	0	40	28	66
6	18	91	55	67	33	40	0	21	36
7	86	86	49	98	80	28	21	0	5
8	66	89	56	99	12	66	36	5	0

Caminho Mínimo: [0 1 5 2 3 4 8 7 6 0]

Custo do Caminho: 239

Custo Computacional: 40320

Tempo de execução: 0.33s

N: 10

Adjacencias:

	0	1	2	3	4	5	6	7	8	9
0	0	24	57	47	63	39	41	21	44	35
1	24	0	81	20	90	26	51	27	98	60
2	57	81	0	79	37	44	87	52	21	1
3	47	20	79	0	53	41	50	77	29	63
4	63	90	37	53	0	78	20	70	13	46
5	39	26	44	41	78	0	85	20	67	7
6	41	51	87	50	20	85	0	66	44	69
7	21	27	52	77	70	20	66	0	76	65
8	44	98	21	29	13	67	44	76	0	30
9	35	60	1	63	46	7	69	65	30	0

Caminho Mínimo: [0 1 3 6 4 8 2 9 5 7 0]

Custo do Caminho: 197

Custo Computacional: 362880

Tempo de execução: 3.07s

N: 11

Adjacencias:

	0	1	2	3	4	5	6	7	8	9	10
0	0	2	56	82	11	32	87	36	95	17	96
1	2	0	95	61	39	49	39	72	90	37	75
2	56	95	0	28	58	31	85	16	99	44	97
3	82	61	28	0	50	24	93	89	36	86	44
4	11	39	58	50	0	41	39	33	43	25	64
5	32	49	31	24	41	0	93	4	32	31	41
6	87	39	85	93	39	93	0	63	94	63	26
7	36	72	16	89	33	4	63	0	96	51	56
8	95	90	99	36	43	32	94	96	0	34	84
9	17	37	44	86	25	31	63	51	34	0	99
10	96	75	97	44	64	41	26	56	84	99	0

Caminho Mínimo: [0 1 6 10 3 2 7 5 8 9 4 0]

Custo do Caminho: 261

Custo Computacional: 3628800

Tempo de execução: 33.56s

N: 12

Adjacencias:

	0	1	2	3	4	5	6	7	8	9	10	11
0	0	73	68	19	74	30	61	92	36	84	91	86
1	73	0	76	20	90	79	64	58	36	14	7	35
2	68	76	0	50	92	4	85	47	48	32	92	67

3	19	20	50	0	14	41	24	92	28	94	69	12
4	74	90	92	14	0	3	21	24	26	5	15	61
5	30	79	4	41	3	0	62	53	67	99	1	54
6	61	64	85	24	21	62	0	70	67	39	81	1
7	92	58	47	92	24	53	70	0	72	87	67	55
8	36	36	48	28	26	67	67	72	0	31	59	19
9	84	14	32	94	5	99	39	87	31	0	5	39
10	91	7	92	69	15	1	81	67	59	5	0	33
11	86	35	67	12	61	54	1	55	19	39	33	0

Caminho Mínimo: [ 0 3 1 9 10 5 2 7 4 6 11 8 0]

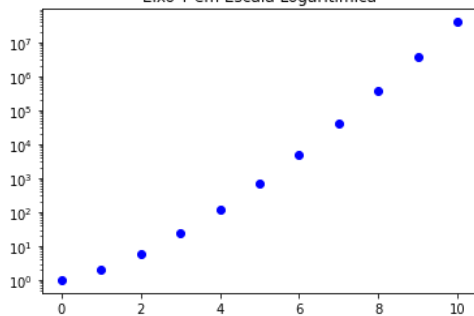
Custo do Caminho: 211

Custo Computacional: 39916800

Tempo de execução: 427.01s

Cidades x Custo Computacional Para Calcular o Caminho Mínimo

Eixo Y em Escala Logarítmica



## Parte 2 - Solução Heurística

1. Implementar uma solução heurística para resolver o problema do Caixeiro Viajante, a Heurística fica à nossa escolha.
2. Aplicar a heurística escolhida aos problemas **si535.tsp**, **pa561.tsp** e **si1032.tsp**

A heurística escolhida para resolver o problema foi a heurística nearest neighbor (vizinho mais próximo) que consiste em:

1. Defina u como o vértice de origem do caixeiro viajante
2. Enquanto Houverem Vértices a Serem Visitados:
  - A. Visite o vértice v mais próximo adjacente a u
  - B. Adicione v aos vértices visitados
  - C. Atualize u para o ultimo vértice v visitado

Essa heurística não é ótima, segundo JONHSON e MCGEOCH(1995) a heurística retorna em média caminhos 25% maiores do que o caminho mínimo, mas, é uma heurística fácil de ser implementada e que retorna um caminho candidato em custo  $O(n^2)$ .

Por isso, é a heurística escolhida para resolver o problema do caixeiro viajante.

Assim, a solução heurística foi implementada (disponível no método **\_\_nearest\_neighbour\_heuristic** em **TSP.py**) e abaixo a aplicação da heurística aos problemas **si535.tsp**, **pa561.tsp** e **si1032.tsp**

```
In [11]: %matplotlib inline
from matplotlib import pyplot
from prova_3_utils import gerar_adjacencia_simetrica_aleatoria, repr_matriz, carregar_grafo_tsp
from TSP import G, TSP
import numpy as np
import time

filepaths = ['tsp_files/si535.tsp', 'tsp_files/pa561.tsp', 'tsp_files/si1032.tsp']
custos_computacionais = []

for filepath in filepaths:
    begin = time.time()
    g = G(grafo = carregar_grafo_tsp(filepath))
    tsp = TSP(g = g)
    caminho_minimo, custo_caminho_minimo, passos = tsp.solve()
    custos_computacionais.append((g.adjacencias.shape[0], passos))
    print('Quantidade de Cidades: {}'.format(g.adjacencias.shape[0], caminho_minimo))
    print('Custo do Caminho: {}'.format(custo_caminho_minimo, passos))
    print('Tempo de execução: {:.2f}s'.format(time.time() - begin))

pyplot.title('Cidades x Custo Computacional da Heurística de Caminho Mínimo' +
            'Eixo Y em Escala Logarítmica')
pyplot.yscale('log')
for N, c in custos_computacionais:
    pyplot.plot(N, c, 'ro');
```

Quantidade de Cidades: 535

Caminho Mínimo: [0, 8, 20, 135, 486, 502, 509, 521, 529, 510, 497, 481, 478, 491, 511, 518, 531, 201, 226, 240, 246, 257, 234, 228, 212, 247, 213, 227, 238, 244, 263, 237, 224, 210, 242, 259, 527, 515, 514, 490, 485, 488, 499, 513, 516, 532, 52, 4, 534, 279, 319, 146, 341, 288, 317, 337, 287, 310, 338, 290, 321, 334, 291, 309, 340, 137, 139, 141, 143, 136, 138, 140, 142, 144, 145, 171, 172, 173, 174, 148, 149, 327, 316, 280, 273, 271, 278, 320, 328, 272, 281, 318, 332, 277, 270, 308, 15, 0, 151, 333, 284, 307, 329, 254, 248, 232, 216, 207, 206, 217, 236, 251, 258, 218, 204, 25, 533, 26, 520, 27, 507, 496, 48, 3, 23, 22, 21, 492, 476, 468, 10, 28, 428, 422, 409, 437, 32, 447, 11, 454, 469, 457, 445, 429, 411, 414, 435, 443, 12, 3, 6, 453, 471, 470, 465, 433, 415, 24, 9, 475, 503, 31, 508, 30, 523, 29, 528, 209, 215, 241, 243, 262, 233, 221, 211, 245, 264, 33, 526, 34, 519, 35, 505, 501, 489, 487, 494, 39, 506, 38, 517, 37, 530, 13, 40, 371, 375, 373, 372, 364, 2, 362, 36, 8, 376, 363, 374, 410, 430, 417, 436, 412, 440, 449, 451, 466, 459, 464, 458, 448, 439, 421, 418, 431, 461, 463, 480, 498, 479, 495, 477, 493, 202, 222, 239, 252, 261, 230, 220, 214, 203, 219, 229, 250, 260, 249, 256, 205, 225, 235, 253, 255, 23, 1, 223, 208, 268, 282, 313, 331, 285, 269, 312, 330, 289, 314, 335, 286, 311, 147, 339, 283, 315, 336, 525, 522, 512, 504, 484, 482, 500, 419, 427, 441, 450, 460, 456, 444, 438, 416, 420, 432, 446, 455, 472, 452, 442, 434, 413, 462, 467, 473, 36, 5, 366, 370, 367, 369, 1, 3, 4, 180, 179, 5, 360, 378, 380, 377, 381, 379, 388, 389, 395, 6, 382, 383, 387, 386, 384, 401, 403, 402, 404, 391, 390, 392, 393, 394, 405, 406, 407, 408, 361, 396, 400, 397, 398, 399, 358, 357, 359, 425, 426, 124, 12, 1, 120, 123, 122, 192, 119, 196, 197, 125, 195, 128, 198, 199, 188, 131, 189, 126, 191, 190, 127, 130, 133, 129, 132, 423, 424, 114, 110, 106, 102, 101, 105, 100, 104, 108, 111, 193, 107, 103, 99, 161, 163, 165, 115, 194, 112, 116, 118, 113, 10, 9, 117, 95, 91, 96, 93, 88, 85, 89, 94, 98, 97, 86, 81, 80, 84, 90, 83, 79, 87, 92, 157, 153, 82, 74, 69, 64, 59, 54, 55, 60, 65, 70, 75, 66, 61, 56, 57, 62, 67, 63, 58, 68, 73, 78, 72, 77, 71, 76, 18, 51, 50, 52, 53, 17, 46, 47, 48, 16, 44, 4, 5, 42, 43, 49, 185, 184, 182, 181, 183, 186, 187, 14, 41, 15, 296, 297, 299, 300, 301, 322, 323, 344, 343, 342, 324, 325, 326, 345, 152, 304, 303, 306, 305, 302, 349, 350, 348, 347, 346, 295, 294, 293, 292, 298, 353, 354, 352, 351, 355, 356, 27, 5, 170, 267, 265, 266, 274, 276, 134, 200, 474, 19, 385, 7, 155, 158, 159, 154, 156, 160, 162, 164, 166, 167, 168, 169, 17, 8, 177, 176, 175, 0]

Custo do Caminho: 50144  
Custo Computacional: 285156  
Tempo de execução: 4.22s

Quantidade de Cidades: 561  
Caminho Mínimo: [0, 323, 330, 324, 329, 331, 328, 326, 325, 327, 399, 157, 158, 480, 481, 159, 160, 163, 161, 155, 162, 11, 2, 111, 116, 119, 115, 118, 122, 121, 125, 184, 185, 127, 126, 128, 129, 130, 131, 123, 124, 120, 164, 138, 165, 168, 170, 141, 144, 133, 139, 140, 135, 134, 132, 186, 189, 187, 188, 94, 96, 97, 95, 93, 92, 72, 90, 91, 69, 52, 53, 54, 49, 46, 4, 2, 41, 171, 172, 173, 47, 174, 175, 113, 109, 106, 107, 108, 153, 105, 152, 154, 156, 479, 167, 169, 143, 145, 146, 147, 2, 27, 228, 229, 230, 506, 508, 521, 520, 509, 519, 513, 512, 511, 495, 497, 498, 514, 515, 518, 517, 458, 459, 460, 445, 44, 4, 437, 436, 416, 413, 410, 409, 492, 486, 485, 408, 405, 407, 411, 406, 423, 422, 403, 402, 397, 417, 334, 419, 336, 335, 350, 337, 352, 355, 338, 340, 339, 341, 420, 421, 425, 426, 428, 429, 432, 434, 431, 435, 412, 430, 424, 418, 398, 401, 40, 0, 404, 482, 483, 484, 487, 488, 489, 490, 491, 494, 496, 493, 414, 499, 516, 415, 462, 461, 466, 446, 451, 447, 387, 448, 391, 390, 369, 389, 388, 385, 386, 442, 384, 440, 441, 438, 439, 382, 380, 383, 363, 364, 365, 368, 367, 366, 370, 371, 37, 3, 372, 374, 375, 376, 377, 378, 396, 395, 393, 394, 449, 450, 453, 454, 455, 456, 457, 478, 468, 452, 476, 477, 475, 472, 471, 470, 469, 467, 465, 464, 463, 543, 542, 541, 524, 523, 522, 231, 507, 529, 531, 532, 251, 268, 269, 270, 267, 248, 24, 9, 250, 232, 233, 234, 236, 237, 235, 238, 247, 265, 245, 244, 243, 242, 241, 240, 255, 256, 253, 258, 257, 259, 260, 317, 316, 315, 314, 313, 318, 312, 320, 302, 303, 301, 305, 306, 308, 304, 321, 319, 322, 311, 276, 275, 273, 272, 271, 287, 28, 8, 279, 291, 290, 289, 294, 293, 298, 292, 295, 296, 297, 299, 300, 307, 310, 309, 264, 261, 262, 263, 274, 266, 239, 246, 222, 221, 151, 148, 142, 136, 150, 137, 215, 216, 214, 191, 192, 193, 195, 194, 196, 209, 208, 207, 206, 205, 204, 102, 9, 8, 99, 79, 80, 100, 81, 82, 87, 86, 85, 84, 83, 73, 64, 61, 60, 59, 58, 57, 67, 66, 75, 77, 89, 68, 51, 44, 31, 40, 36, 3, 5, 4, 5, 7, 8, 6, 18, 15, 16, 17, 19, 11, 12, 28, 29, 21, 33, 32, 34, 22, 30, 23, 24, 25, 26, 13, 14, 20, 27, 10, 9, 2, 1, 38, 37, 3, 39, 43, 45, 50, 48, 55, 70, 71, 56, 180, 181, 182, 183, 179, 177, 176, 117, 178, 114, 110, 190, 218, 219, 225, 197, 200, 201, 203, 199, 210, 211, 212, 103, 88, 104, 101, 76, 78, 74, 65, 62, 63, 342, 343, 344, 345, 348, 347, 346, 332, 333, 349, 351, 353, 357, 354, 358, 356, 361, 359, 360, 362, 379, 427, 433, 381, 443, 392, 549, 548, 473, 474, 550, 551, 55, 2, 553, 554, 556, 555, 547, 546, 525, 526, 538, 539, 540, 557, 558, 559, 560, 536, 537, 530, 527, 528, 534, 533, 535, 281, 286, 285, 284, 283, 282, 280, 277, 278, 254, 223, 224, 220, 217, 202, 198, 252, 226, 213, 149, 500, 501, 502, 503, 504, 50, 5, 510, 544, 545, 166, 0]

Custo do Caminho: 3422  
Custo Computacional: 313600  
Tempo de execução: 4.33s

Quantidade de Cidades: 1032  
Caminho Mínimo: [0, 1, 2, 72, 73, 74, 144, 145, 146, 147, 240, 241, 242, 243, 336, 337, 338, 423, 451, 461, 480, 526, 529, 488, 456, 454, 425, 341, 340, 339, 247, 246, 245, 244, 151, 150, 149, 148, 77, 76, 75, 5, 4, 3, 6, 7, 8, 78, 79, 80, 152, 153, 154, 155, 248, 249, 250, 251, 342, 343, 344, 427, 452, 460, 501, 520, 535, 492, 464, 449, 430, 347, 346, 345, 255, 25, 4, 253, 252, 159, 158, 157, 156, 83, 82, 81, 11, 10, 9, 12, 13, 14, 84, 85, 86, 160, 161, 162, 163, 256, 257, 258, 259, 34, 8, 349, 350, 420, 447, 467, 493, 522, 549, 495, 479, 453, 418, 353, 352, 351, 263, 262, 261, 260, 167, 166, 165, 164, 89, 88, 87, 17, 16, 15, 18, 19, 20, 90, 91, 92, 168, 169, 170, 171, 264, 265, 266, 267, 354, 355, 356, 415, 448, 476, 502, 52, 1, 543, 489, 477, 450, 413, 359, 358, 357, 271, 270, 269, 268, 175, 174, 173, 172, 95, 94, 93, 23, 22, 21, 24, 25, 26, 96, 97, 98, 176, 177, 178, 179, 272, 273, 274, 275, 360, 361, 362, 410, 446, 475, 483, 518, 532, 485, 466, 455, 428, 365, 364, 363, 279, 278, 277, 276, 183, 182, 181, 180, 101, 100, 99, 29, 28, 27, 30, 31, 32, 102, 103, 104, 184, 185, 186, 187, 280, 281, 282, 283, 366, 367, 368, 419, 443, 478, 482, 504, 545, 487, 474, 442, 409, 371, 370, 369, 287, 286, 285, 284, 191, 19, 0, 189, 188, 107, 106, 105, 35, 34, 33, 506, 544, 519, 542, 525, 541, 523, 546, 527, 540, 524, 533, 750, 740, 745, 739, 74, 3, 748, 741, 742, 744, 746, 747, 749, 751, 731, 738, 722, 753, 772, 775, 781, 780, 791, 782, 785, 771, 755, 752, 759, 765, 767, 757, 827, 822, 819, 824, 829, 831, 825, 821, 820, 823, 826, 828, 818, 816, 817, 793, 794, 795, 796, 797, 798, 799, 80, 0, 801, 802, 803, 792, 805, 806, 807, 808, 809, 810, 811, 812, 813, 814, 815, 804, 760, 763, 761, 766, 768, 762, 769, 758, 756, 754, 764, 783, 788, 786, 787, 789, 790, 784, 770, 779, 778, 777, 776, 774, 773, 721, 720, 852, 855, 858, 850, 863, 86, 6, 869, 840, 832, 836, 839, 842, 845, 847, 830, 901, 904, 909, 903, 899, 907, 911, 902, 905, 900, 910, 897, 898, 882, 877, 881, 880, 879, 878, 872, 876, 875, 874, 873, 883, 885, 886, 887, 888, 889, 890, 891, 892, 893, 894, 884, 895, 896, 844, 84, 3, 841, 837, 838, 849, 835, 833, 834, 848, 851, 871, 868, 867, 865, 864, 862, 870, 860, 859, 861, 856, 854, 853, 857, 927, 932, 933, 935, 936, 937, 938, 940, 930, 951, 950, 942, 947, 945, 931, 928, 916, 914, 915, 929, 919, 918, 921, 922, 924, 92, 5, 975, 964, 974, 973, 972, 971, 970, 969, 968, 967, 966, 965, 952, 962, 961, 960, 959, 958, 957, 956, 955, 954, 963, 953, 977, 976, 991, 990, 979, 988, 987, 986, 978, 985, 984, 983, 982, 981, 980, 923, 920, 912, 913, 917, 943, 946, 949, 944, 93, 9, 941, 934, 948, 995, 583, 1027, 584, 1006, 585, 999, 586, 1001, 587, 1023, 588, 1015, 589, 1022, 573, 1014, 569, 1016, 5, 71, 1018, 570, 1020, 553, 1021, 554, 1024, 555, 1026, 556, 1028, 557, 1029, 558, 1011, 559, 1010, 560, 994, 552, 996, 562, 998, 563, 992, 564, 1002, 565, 1003, 566, 1005, 567, 1007, 561, 1009, 568, 615, 604, 613, 612, 611, 610, 609, 618, 608, 60, 7, 606, 605, 602, 603, 593, 594, 595, 596, 592, 598, 599, 600, 597, 601, 617, 614, 629, 630, 619, 628, 624, 622, 620, 626, 621, 623, 625, 627, 631, 574, 1004, 591, 1008, 1000, 575, 997, 576, 993, 577, 1030, 582, 1031, 578, 1025, 579, 1012, 580, 1017, 581, 1013, 572, 1019, 590, 635, 651, 652, 654, 655, 657, 658, 660, 650, 671, 670, 668, 662, 665, 667, 653, 648, 634, 636, 637, 640, 642, 643, 646, 641, 644, 632, 616, 687, 686, 681, 685, 684, 683, 682, 673, 680, 679, 678, 677, 676, 675, 67, 4, 672, 691, 690, 694, 711, 709, 708, 707, 706, 705, 704, 703, 710, 693, 701, 700, 692, 699, 698, 702, 697, 696, 695, 638, 647, 645, 639, 649, 633, 663, 666, 669, 664, 659, 656, 661, 718, 719, 712, 717, 716, 715, 714, 713, 688, 689, 989, 846, 72, 5, 734, 733, 732, 730, 729, 728, 727, 726, 723, 724, 735, 736, 737, 906, 908, 926, 528, 511, 490, 459, 438, 426, 404, 403, 402, 331, 330, 329, 328, 235, 234, 233, 232, 140, 139, 138, 68, 67, 66, 63, 64, 65, 135, 136, 137, 228, 229, 230, 231, 32, 4, 325, 326, 327, 399, 400, 401, 422, 434, 457, 491, 516, 530, 481, 469, 445, 429, 398, 397, 396, 323, 322, 321, 320, 227, 226, 225, 224, 134, 133, 132, 62, 61, 60, 57, 58, 59, 129, 130, 131, 220, 221, 222, 223, 316, 317, 318, 319, 393, 394, 39, 5, 431, 436, 462, 503, 514, 534, 500, 470, 435, 408, 392, 391, 390, 315, 314, 313, 312, 219, 218, 217, 216, 128, 127, 126, 56, 55, 54, 51, 52, 53, 123, 124, 125, 212, 213, 214, 215, 308, 309, 310, 311, 387, 388, 389, 421, 439, 465, 497, 512, 53, 1, 496, 468, 433, 417, 386, 385, 384, 307, 306, 305, 304, 211, 210, 209, 208, 122, 121, 120, 50, 49, 48, 45, 46, 47, 117, 118, 119, 204, 205, 206, 207, 300, 301, 302, 303, 381, 382, 383, 416, 432, 458, 486, 510, 550, 498, 471, 444, 414, 380, 37, 9, 378, 299, 298, 297, 296, 203, 202, 201, 200, 116, 115, 114, 44, 43, 42, 39, 40, 41, 111, 112, 113, 196, 197, 198, 199, 292, 293, 294, 295, 375, 376, 377, 412, 437, 473, 499, 508, 547, 494, 472, 440, 411, 374, 373, 372, 291, 290, 289, 288, 19, 5, 194, 193, 192, 110, 109, 108, 38, 37, 36, 505, 551, 509, 548, 507, 539, 513, 536, 515, 538, 406, 405, 335, 334, 333, 33, 2, 239, 238, 237, 236, 143, 142, 141, 71, 70, 69, 407, 424, 441, 463, 484, 517, 537, 0]

Custo do Caminho: 94571

Custo Computacional: 1062961

Tempo de execução: 17.28s

Cidades x Custo Computacional da Heurística de Caminho MínimoEixo Y em Escala Logarítmica

