Test Documentation of Team 6

Warehouse Application Test Cases

Algorithms

Algorithm 1 Greedy Algorithm

1) Choose an unpicked item whose location is nearest to start location or current location.
2) Pick that item and test if all the items are picked, if so, terminate the program and output the path, otherwise, return to the first step.
3) The upper bound of our greedy algorithm is O(n^2).

Algorithm 2 Parallelized Genetic Algorithm

1) Initialize the population which means randomly generate a series of paths. (When the number of items is less than 5, we generate all the possible paths, otherwise we generate 500 paths), and separate the population into two sub populations.
2) Using multiprocessing to parallelized the initial population part and greedy algorithm, put the result of greedy algorithm into two sub populations to be the upper bound, and each process will process one sub population.
3) In Crossover, we have a crossover rate 0.7, we will generate a random number between 0 and 1, if it is less than 0.7, we randomly choose two paths to crossover to generate new paths.
4) In Mutation, we have a mutation rate 0.02, we will generate a random number between 0 and 1, if it is less than 0.02, we randomly choose one paths to do the mutation function, which means randomly change two items' positions in the path to generate new paths.
5) In our algorithm, the iteration is set to 1500 times. In every iteration, we only keep the best result of the last generation.
6) After reaching the iteration times, two processes will give two best paths, choose the better one of these two paths and return it.

Algorithm 3 Brute force

1) Get the order list.
2) Find all possible combinations of the items.
3) Iterate through every combination, add start and end points to the list and calculate total distance travelled for each possible sequence.
4) Return the best sequence which has the shortest travelled distance.
5) Since the brute force cannot give a result after the order size is larger than 10, so only test case 1 and test case2 in test group B include algorithm 3.

Test Cases

Test Group A

All test cases here have the same start/end point of 0,0. All test cases in this group will be labeled A1, A2, A3. All the algorithms tested in this group are genetic algorithms with different crossover/mutation rate, initial population, and iteration times. In test case A1, algorithm 1 has 200 initial population, algorithm 2 has 500 initial population, and algorithm 3 has 800 initial population, other parameters are same. In test case A2, algorithm 1 has 0.4 crossover rate, 0.01 mutation rate, algorithm 2 has 0.7 crossover rate, 0.02 mutation rate and algorithm 3 has 1 crossover rate and 0.04 mutation rate, other parameters are same. In test case A3, algorithm 1 has 1000 iterations, algorithm 2 has 1500 iterations and algorithm 1 has 2000 iterations, other parameters are same.

Test Group B

All test cases here have the same start/end point of 0,0. All test cases in this group will be labeled A1, B2, 3B, B4, B5, B6. In this test group, greedy algorithm, genetic algorithm and brute-force algorithm will be tested.

Test Case A1

Input: [427230, 372539, 396879, 391680, 208660, 105912, 332555, 227534, 68048, 188856, 736830, 736831, 479020, 103313, 365797]

Start Point: [0,0]

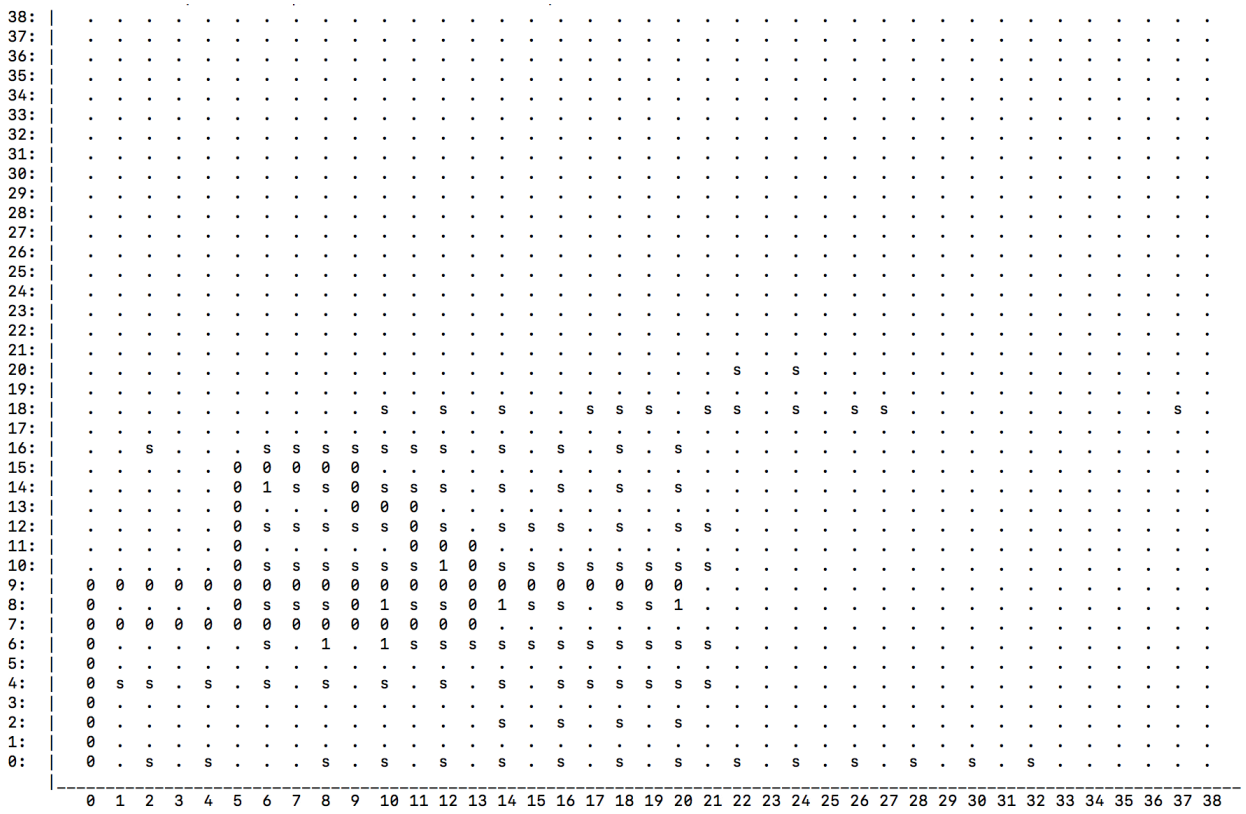
End Point: [0,0]

Distance: 114

Algorithm 1

Output: ['427230', '391680', '736831', '736830', '68048', '372539', '188856', '396879', '332555', '227534', '103313', '105912', '479020', '365797', '208660']

Distance: 71



Running Time: 4.80 sec

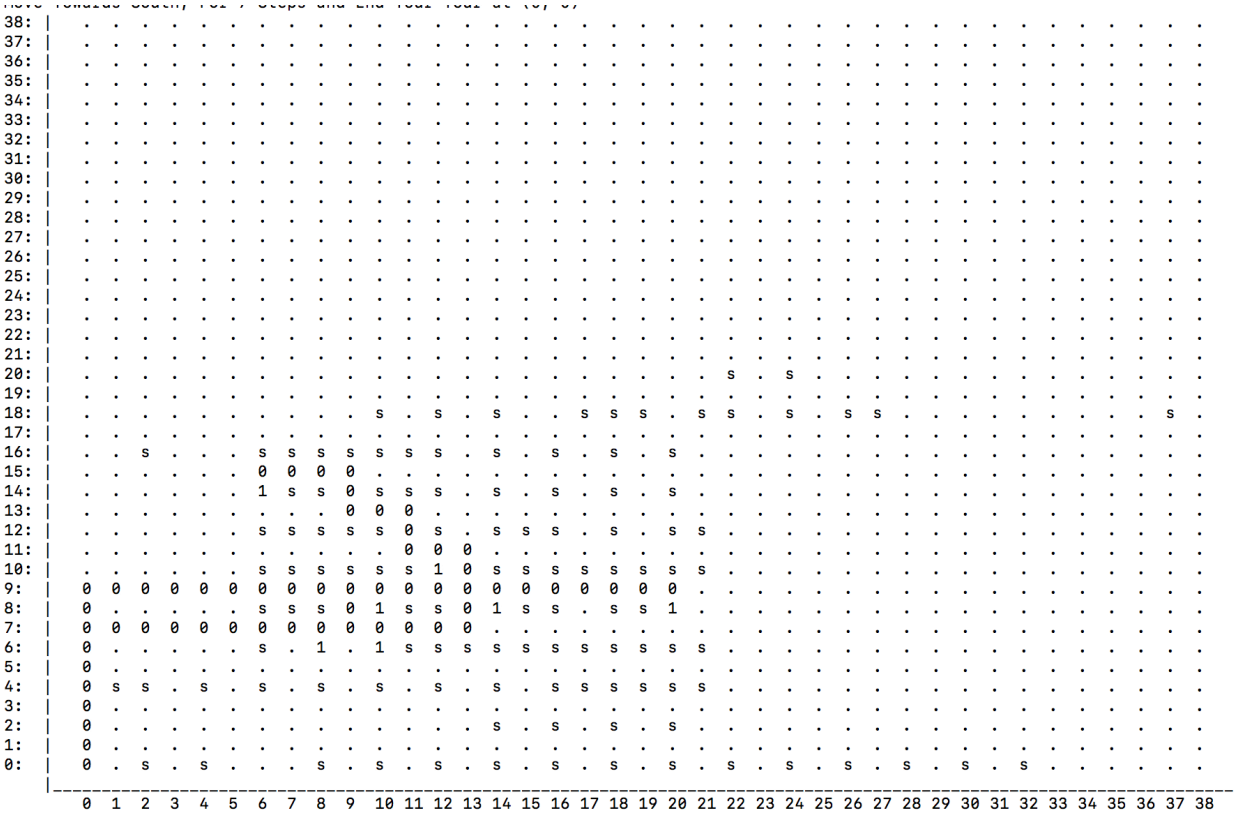
Memory Used: 62.1 MB

Comments: This algorithm takes shortest time and give worst result.

Algorithm 2

Output: ['427230', '372539', '396879', '391680', '188856', '68048', '736830', '736831', '332555', '227534', '103313', '105912', '479020', '365797', '208660']

Distance: 65



Running Time: 7.69 sec

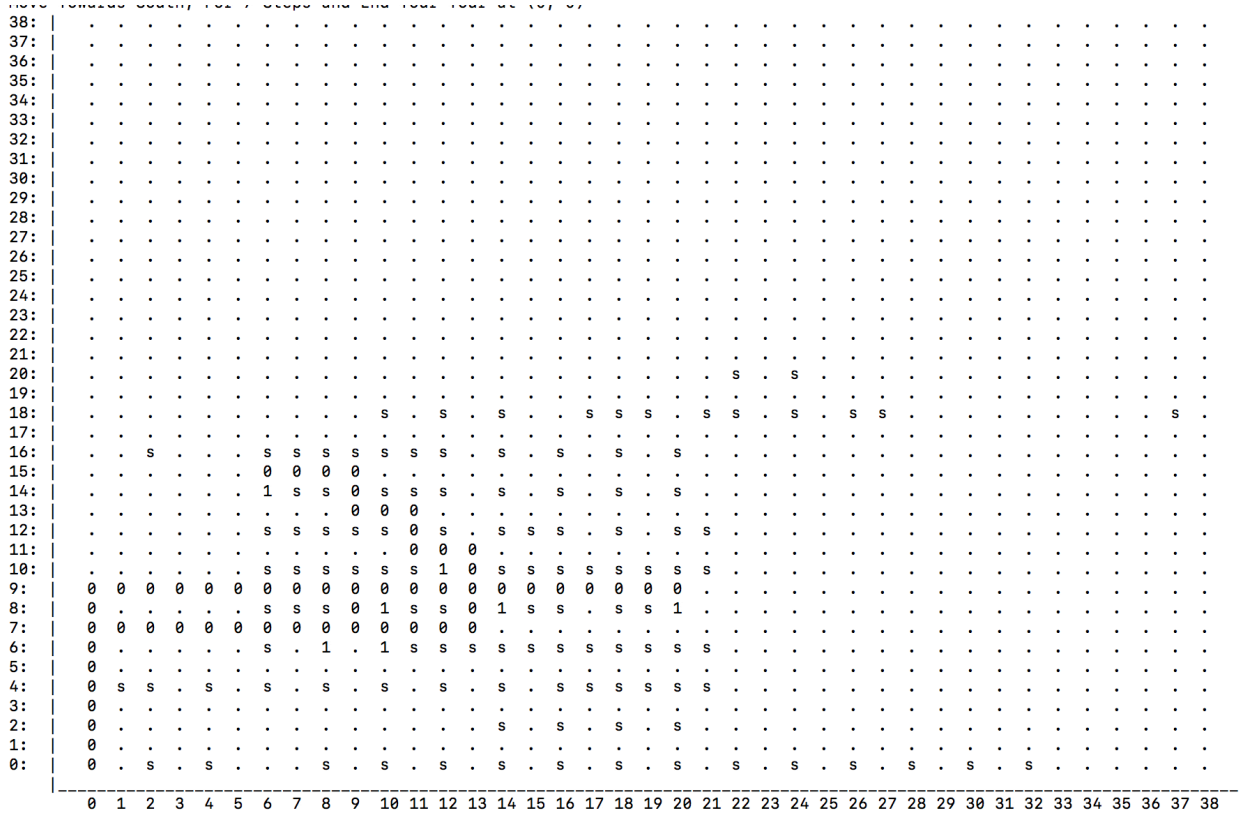
Memory Used: 61.73 MB

Comments: This algorithm takes second shortest time and give best result.

Algorithm 3

Output: ['427230', '372539', '396879', '391680', '188856', '68048', '736830', '736831', '332555', '227534', '103313', '105912', '479020', '365797', '208660']

Distance: 65



Running Time: 11.11 sec

Memory Used: 61.35 MB

Comments: This algorithm takes longest time and give best result.

Test Case A2

Input: [427230, 372539, 396879, 391680, 208660, 105912, 332555, 227534, 68048, 188856, 736830, 736831, 479020, 103313, 365797]

Start Point: [0,0]

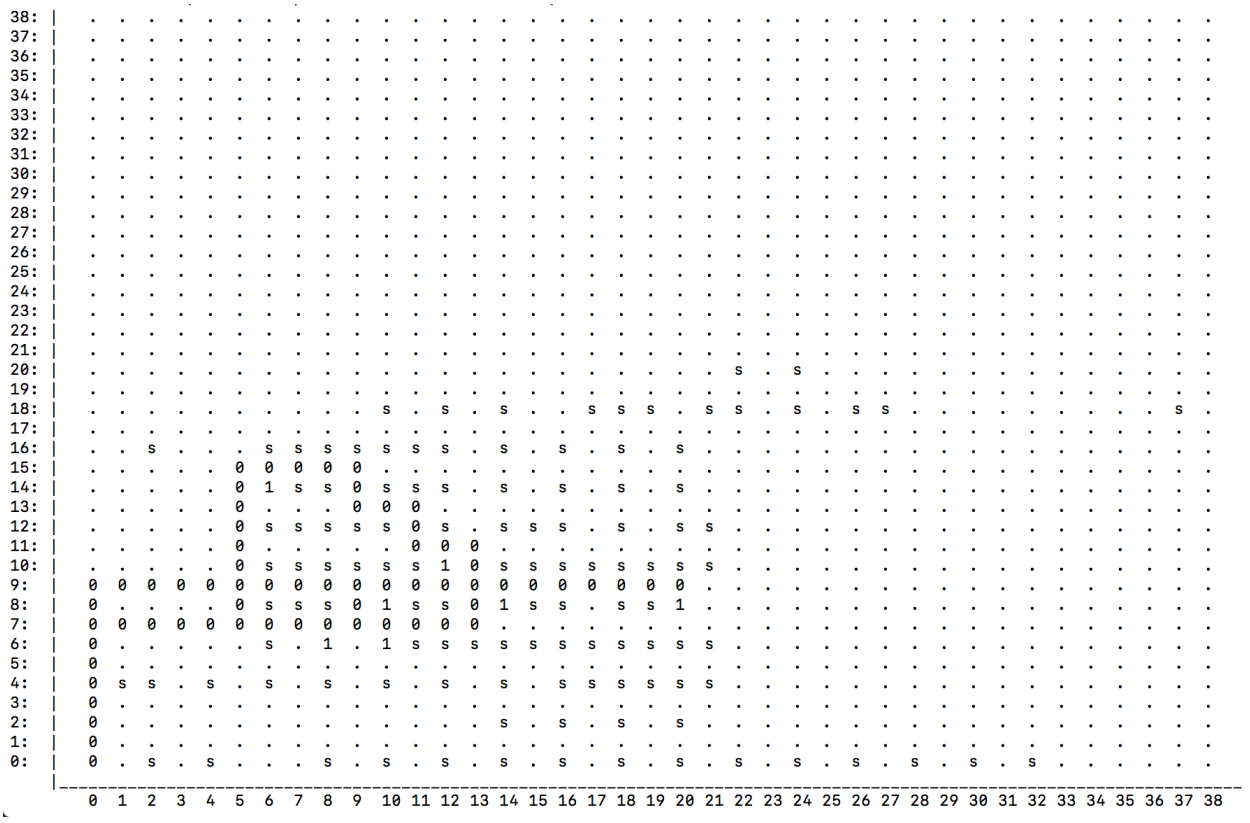
End Point: [0,0]

Distance: 114

Algorithm 1

Output: ['208660', '365797', '479020', '227534', '103313', '332555', '105912', '391680', '372539', '188856', '396879', '736831', '736830', '68048', '427230']

Distance: 67



Running Time: 7.18 sec

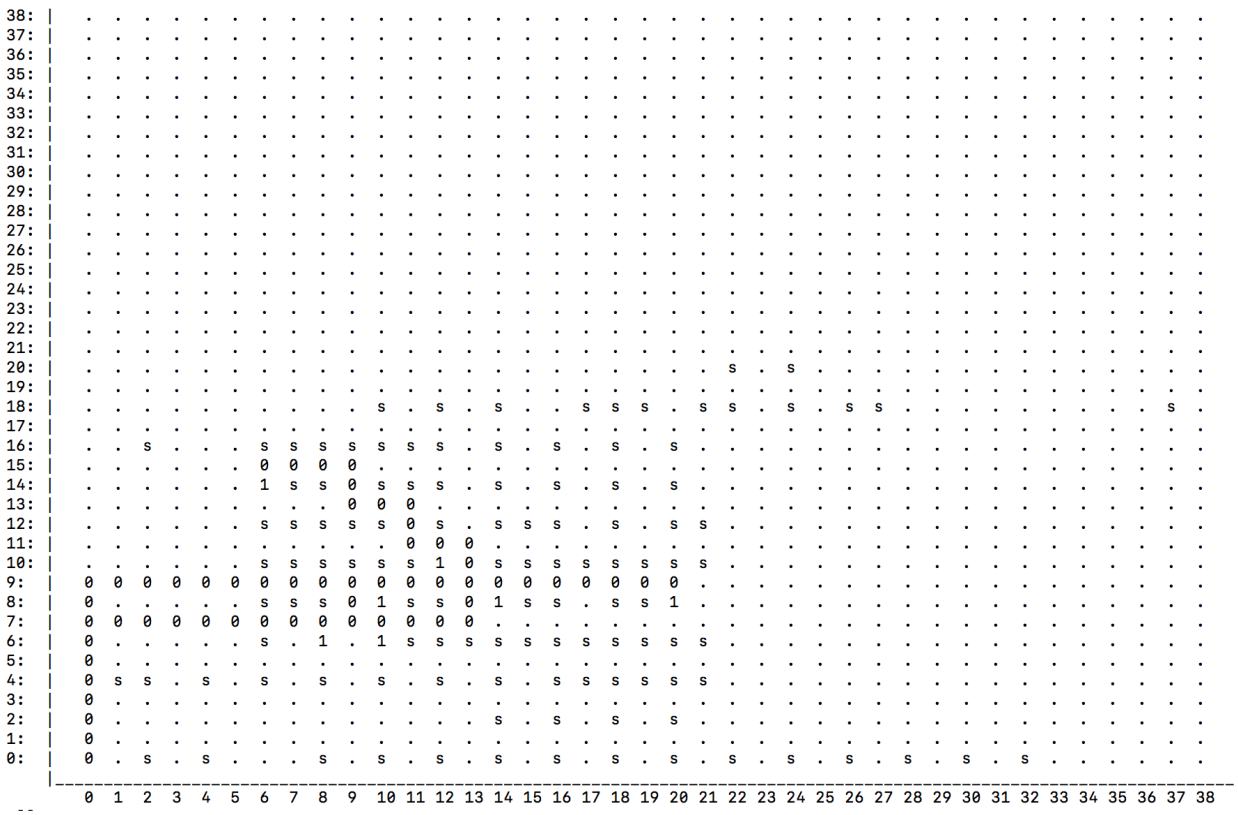
Memory Used: 62.1 MB

Comments: This algorithm takes shortest time and give worst result.

Algorithm 2

Output: ['427230', '372539', '396879', '391680', '188856', '68048', '736830', '736831', '105912', '332555', '227534', '103313', '479020', '365797', '208660']

Distance: 65



Running Time: 7.77 sec

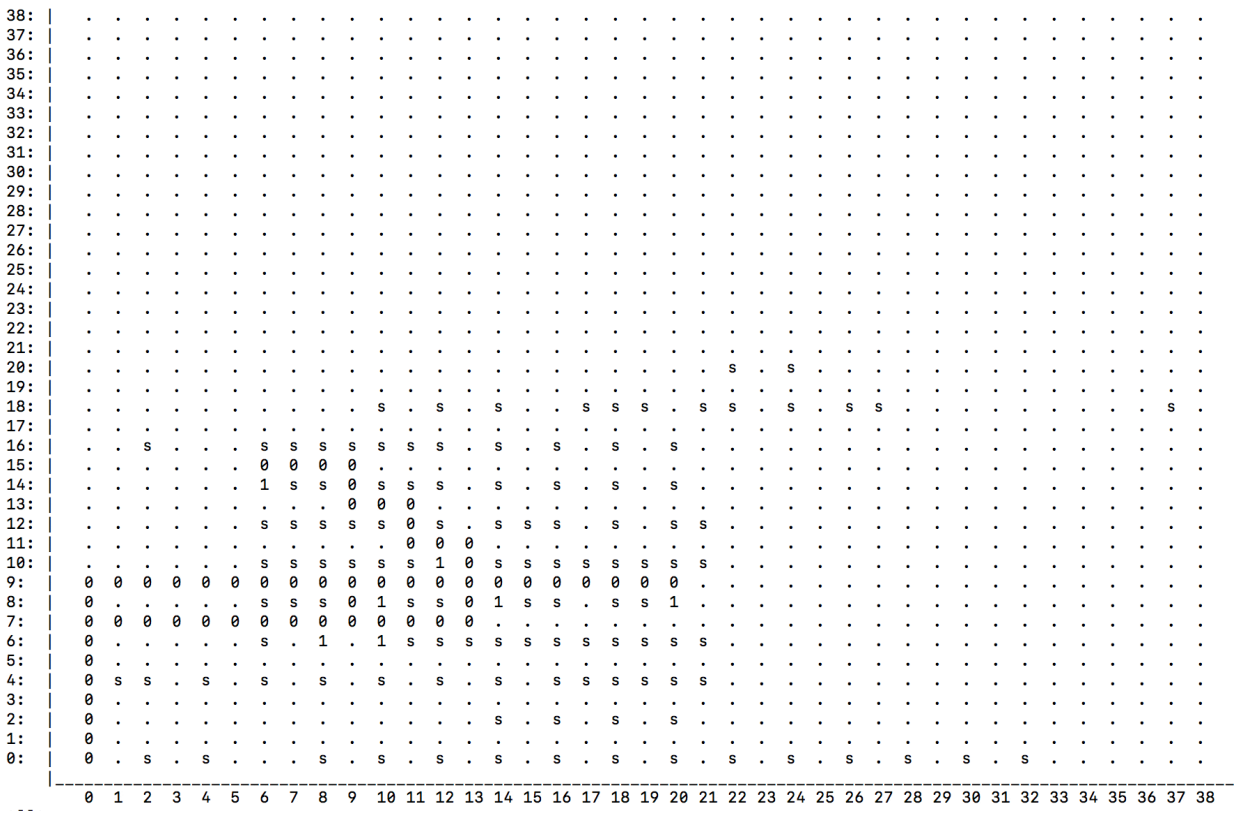
Memory Used: 61.73 MB

Comments: This algorithm takes second shortest time and give best result.

Algorithm 3

Output: ['427230', '372539', '396879', '391680', '188856', '68048', '736830', '736831', '105912', '332555', '227534', '103313', '479020', '365797', '208660']

Distance: 65



Running Time: 8.83 sec

Memory Used: 61.35 MB

Comments: This algorithm takes longest time and give best result.

Test Case A3

Input: [427230, 372539, 396879, 391680, 208660, 105912, 332555, 227534, 68048, 188856, 736830, 736831, 479020, 103313, 365797]

Start Point: [0,0]

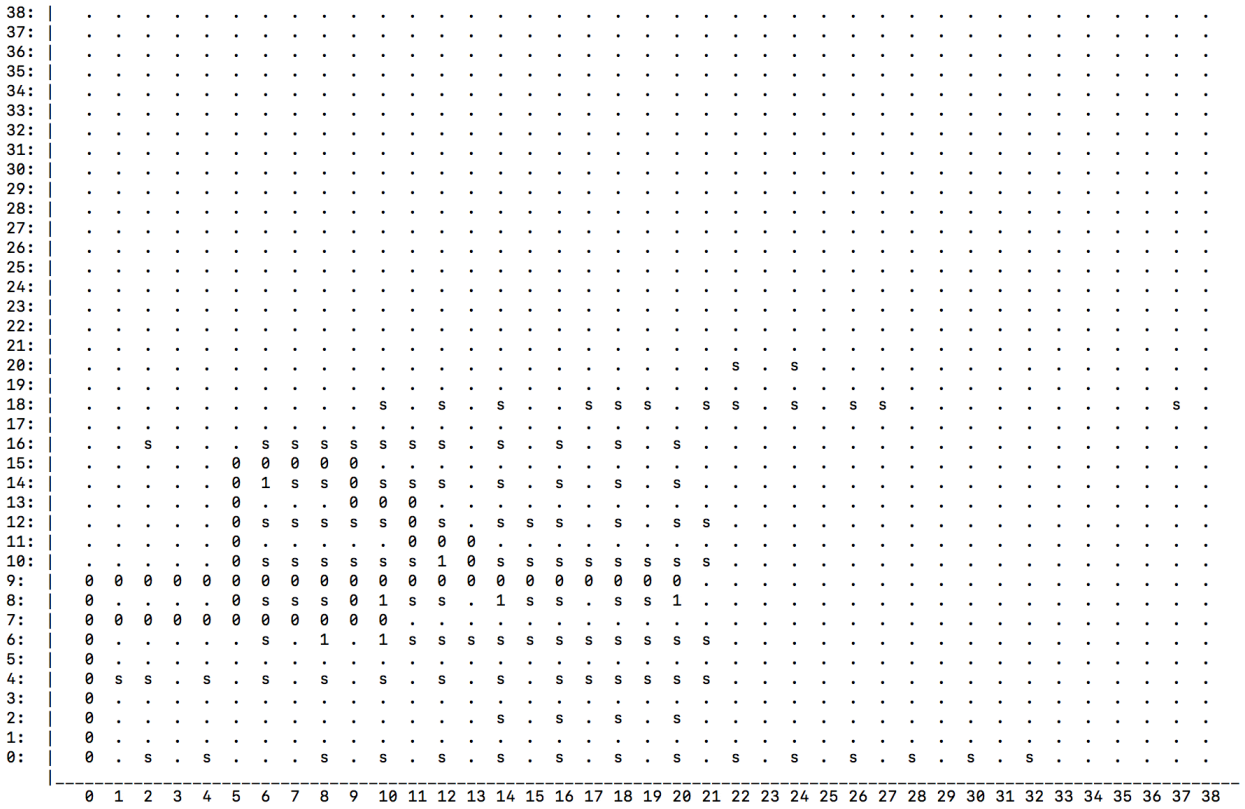
End Point: [0,0]

Distance: 114

Algorithm 1

Output: ['427230', '188856', '391680', '396879', '372539', '103313', '332555', '227534', '105912', '736830', '736831', '68048', '479020', '365797', '208660']

Distance: 69



Running Time: 6.84 sec

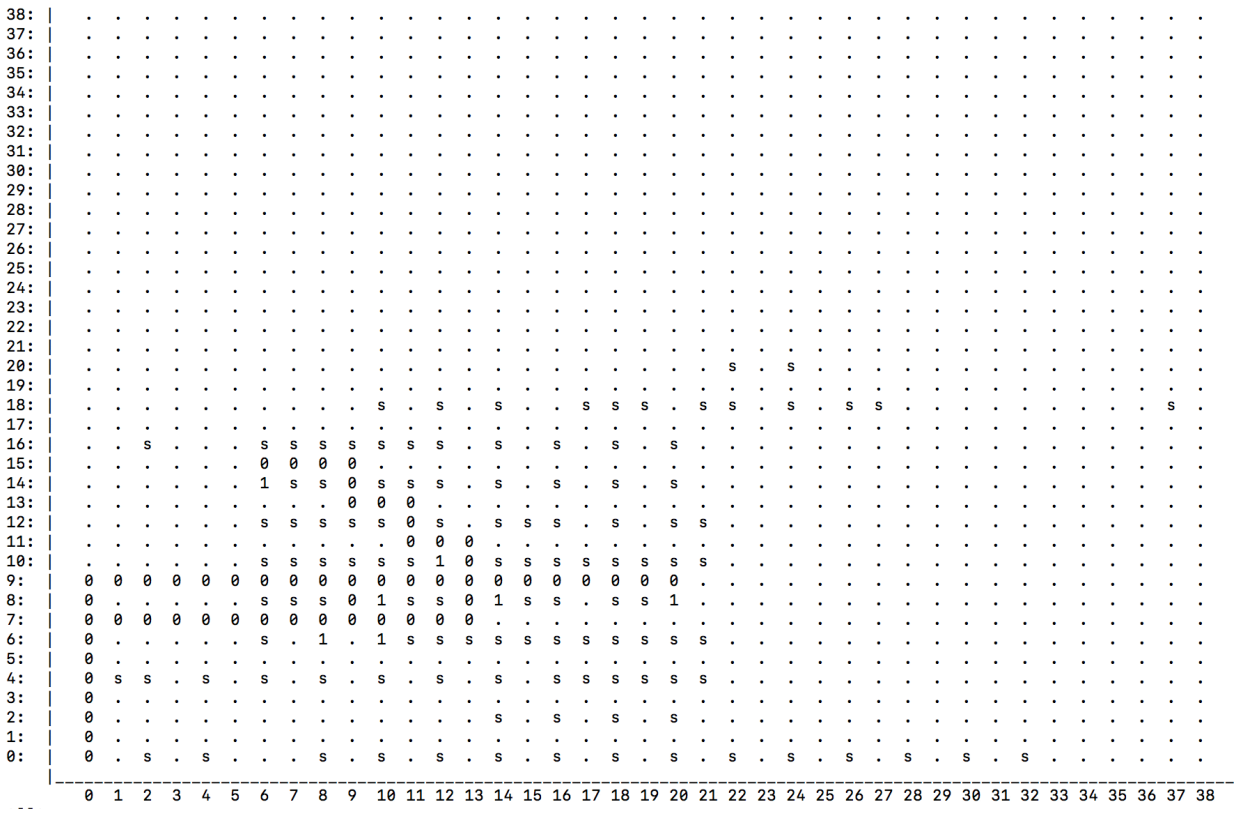
Memory Used: 62.1 MB

Comments: This algorithm takes shortest time and give worst result.

Algorithm 2

Output: ['427230', '372539', '396879', '391680', '188856', '68048', '736830', '736831', '332555', '227534', '103313', '105912', '479020', '365797', '208660']

Distance: 65



Running Time: 7.83 sec

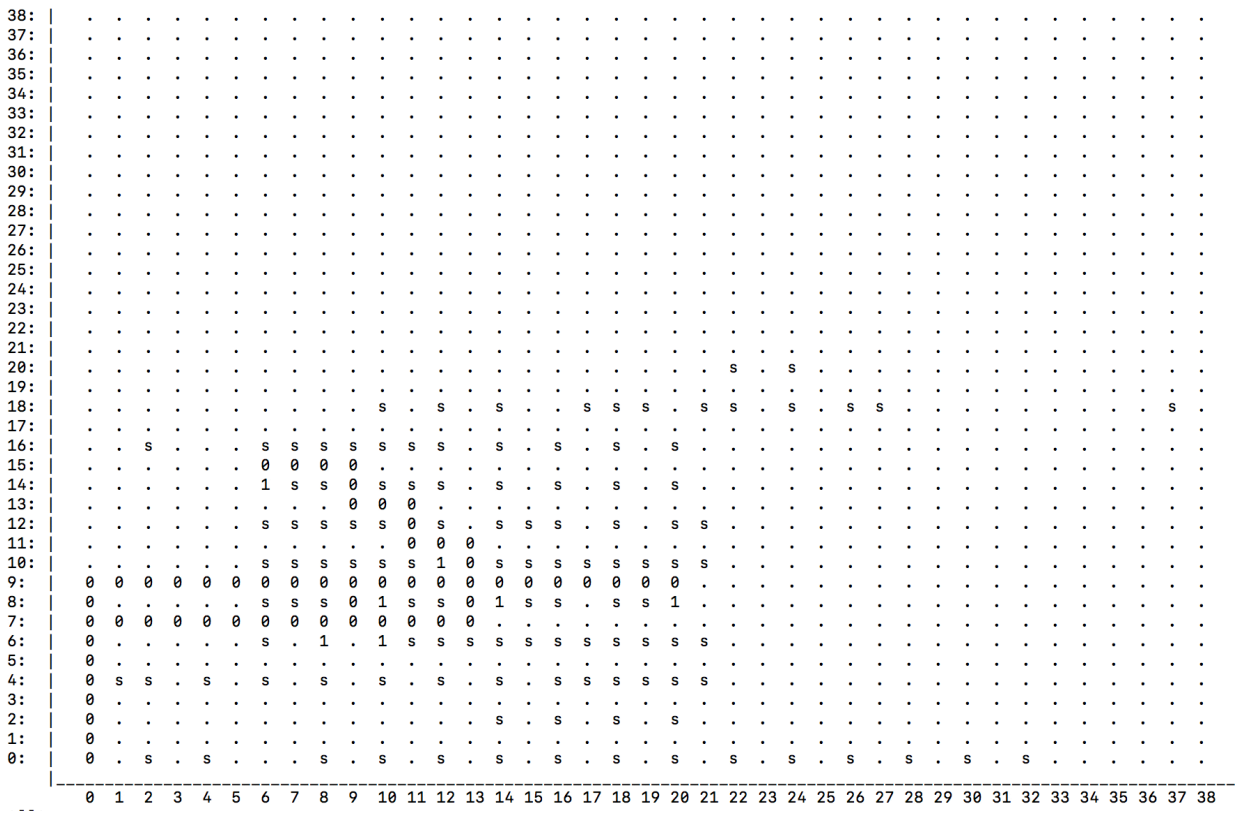
Memory Used: 61.73 MB

Comments: This algorithm takes second shortest time and give best result.

Algorithm 3

Output: ['427230', '372539', '396879', '391680', '188856', '68048', '736830', '736831', '105912', '332555', '227534', '103313', '479020', '365797', '208660']

Distance: 65



Running Time: 9.65 sec

Memory Used: 61.35 MB

Comments: This algorithm takes longest time and give best result.

Test Case B1

Input: [108335, 391825, 340367, 286457, 661741]

Start Point: [0,0]

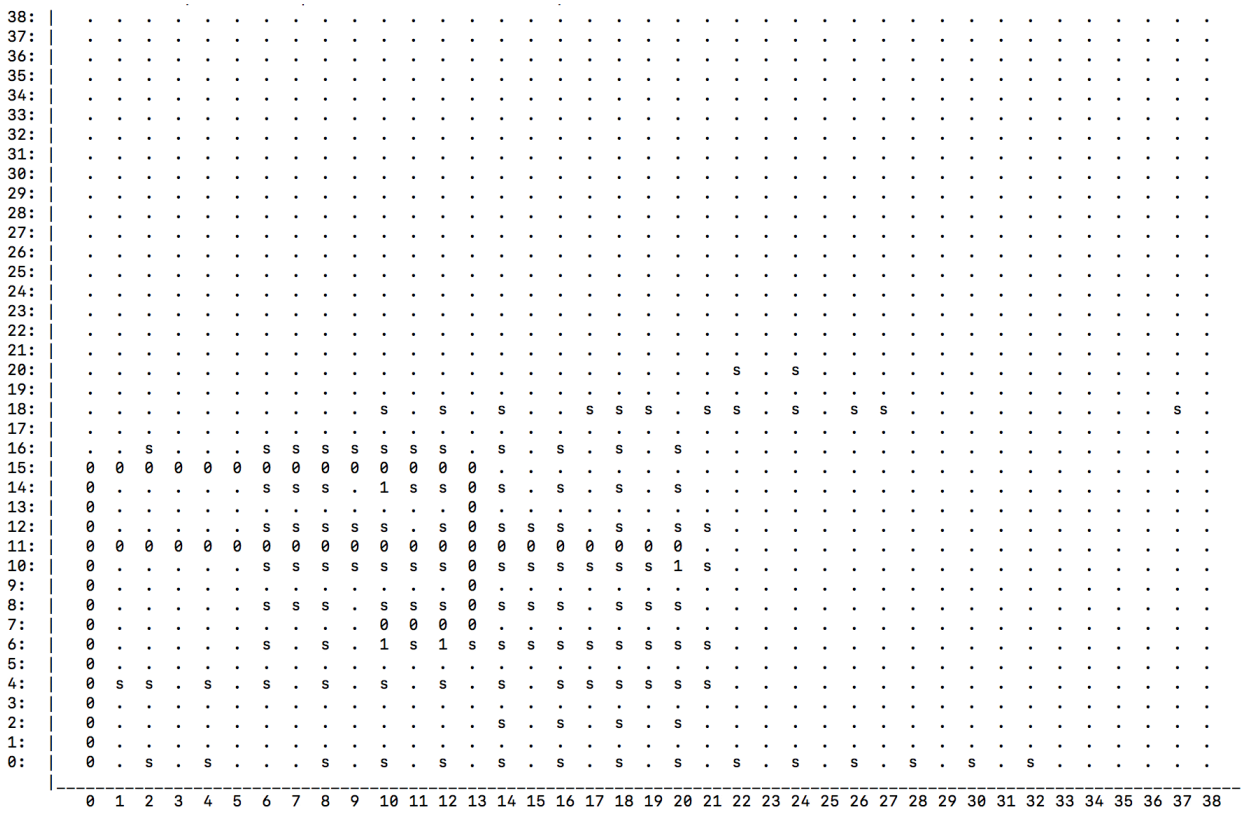
End Point: [0,0]

Distance: 62

Algorithm 1

Output: ['108335', '286457', '340367', '661741', '391825']

Distance: 55



Running Time: 4.32 sec

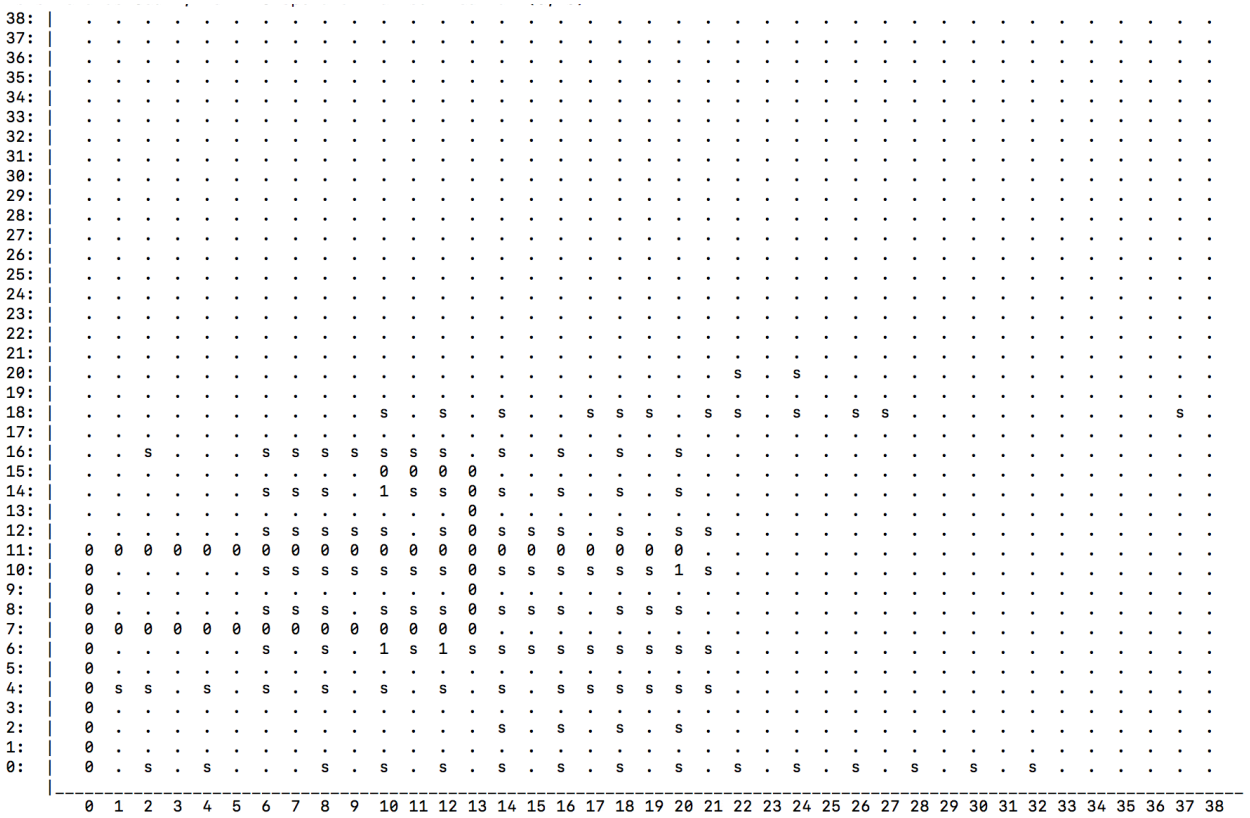
Memory Used: 62 MB

Comments: This algorithm takes shorter time but give a worse result.

Algorithm 2

Output: ['108335', '391825', '661741', '340367', '286457']

Distance: 52



Running Time: 5.17 sec

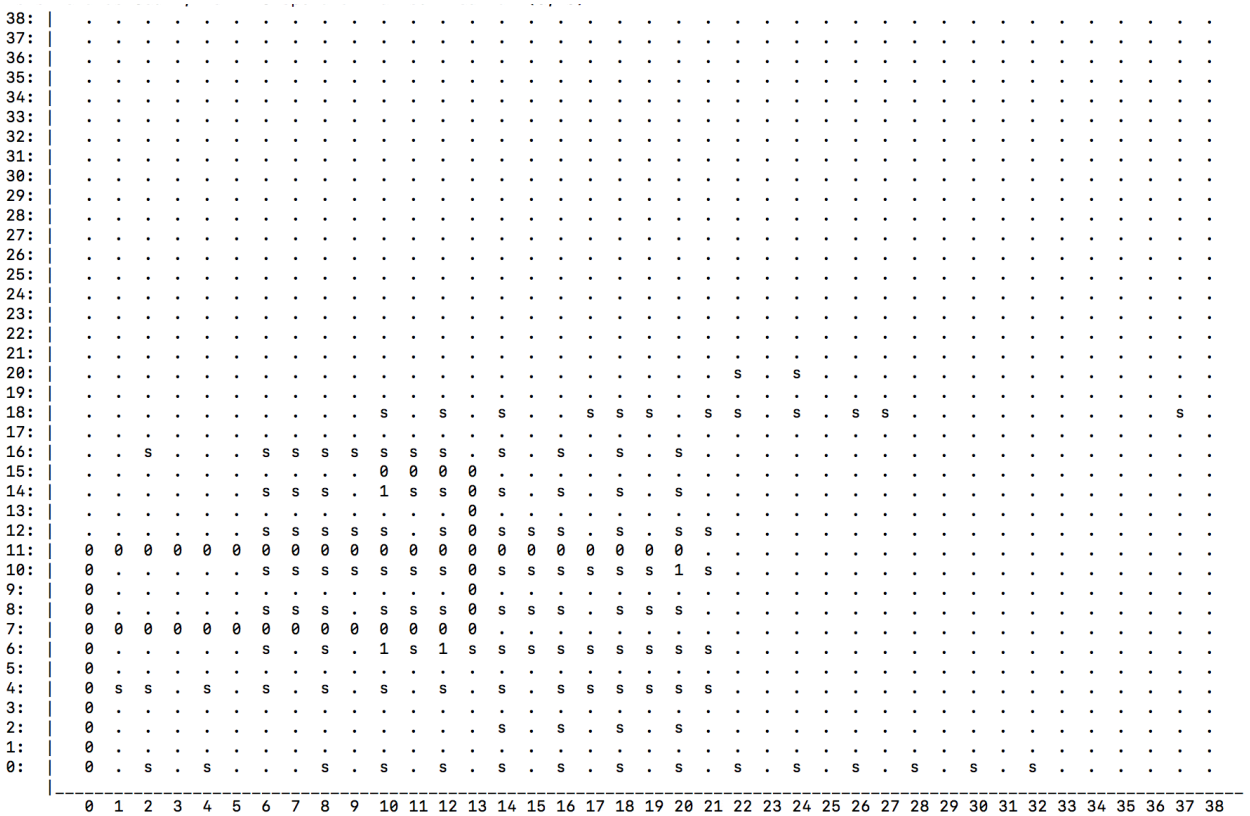
Memory Used: 61.93 MB

Comments: This algorithm takes proper time and give the same result with brute force.

Algorithm 3

Output: ['108335', '391825', '661741', '340367', '286457']

Distance: 52



Running Time: 5.51 sec

Memory Used: 61.58 MB

Comments: This algorithm takes longest time but give the best result.

Test Case B2

Input: [281610, 342706, 111873, 198029, 366109, 287261, 76283, 254489, 258540, 286457]

Start Point: [0,0]

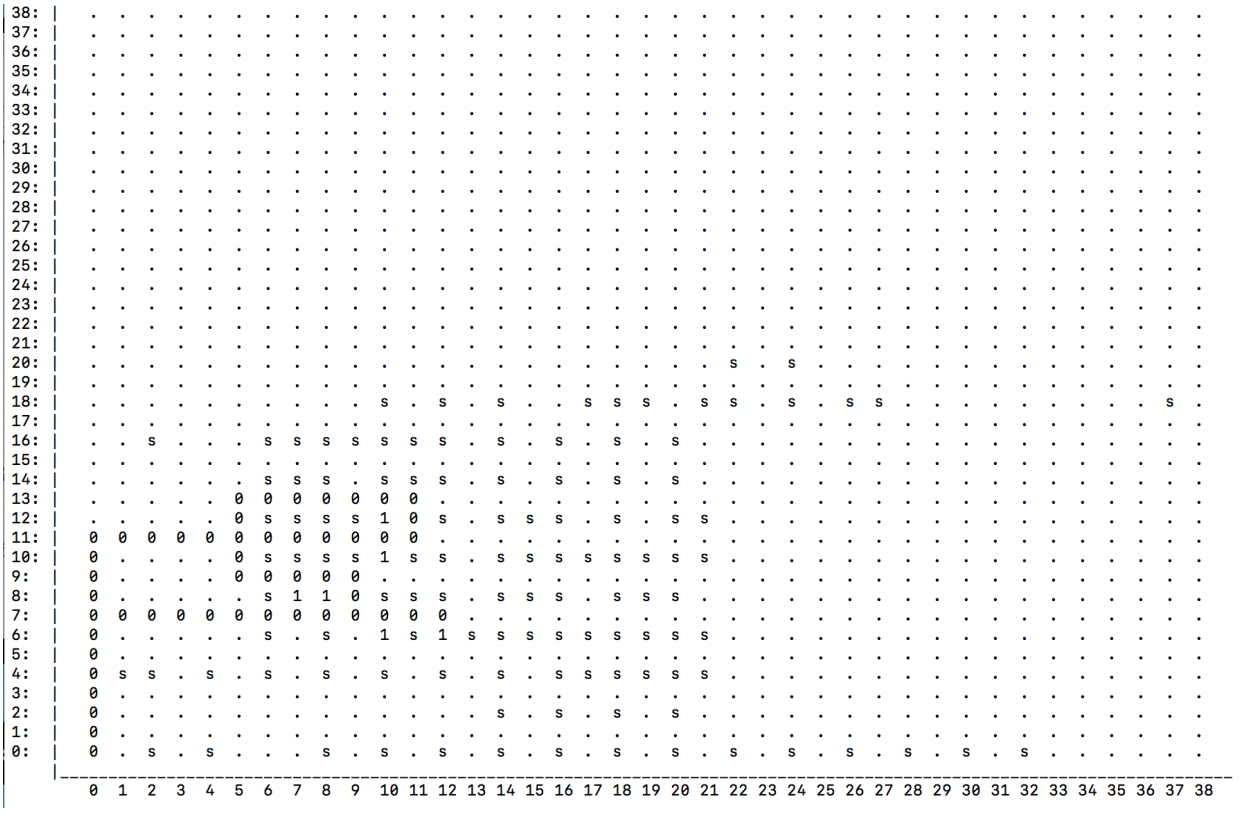
End Point: [0,0]

Distance: 94

Algorithm 1

Output: ['281610', '366109', '76283', '198029', '342706', '111873', '287261', '254489', '258540', '286457']

Distance: 50



Running Time: 4.99

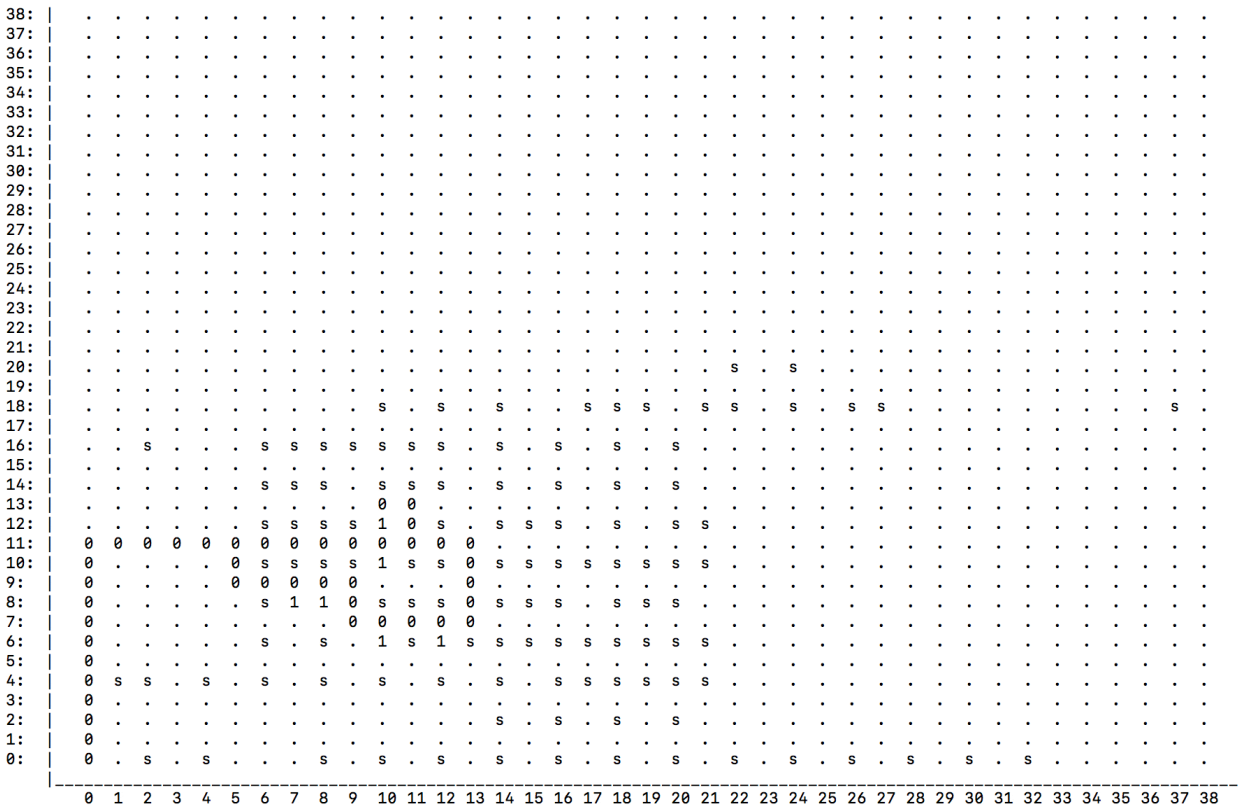
Memory Used: 61.85 MB

Comments: This algorithm takes shorter time but give a worse result.

Algorithm 2

Output: ['281610', '342706', '254489', '111873', '287261', '258540', '286457', '198029', '366109', '76283']

Distance: 44



Running Time: 9.56 sec

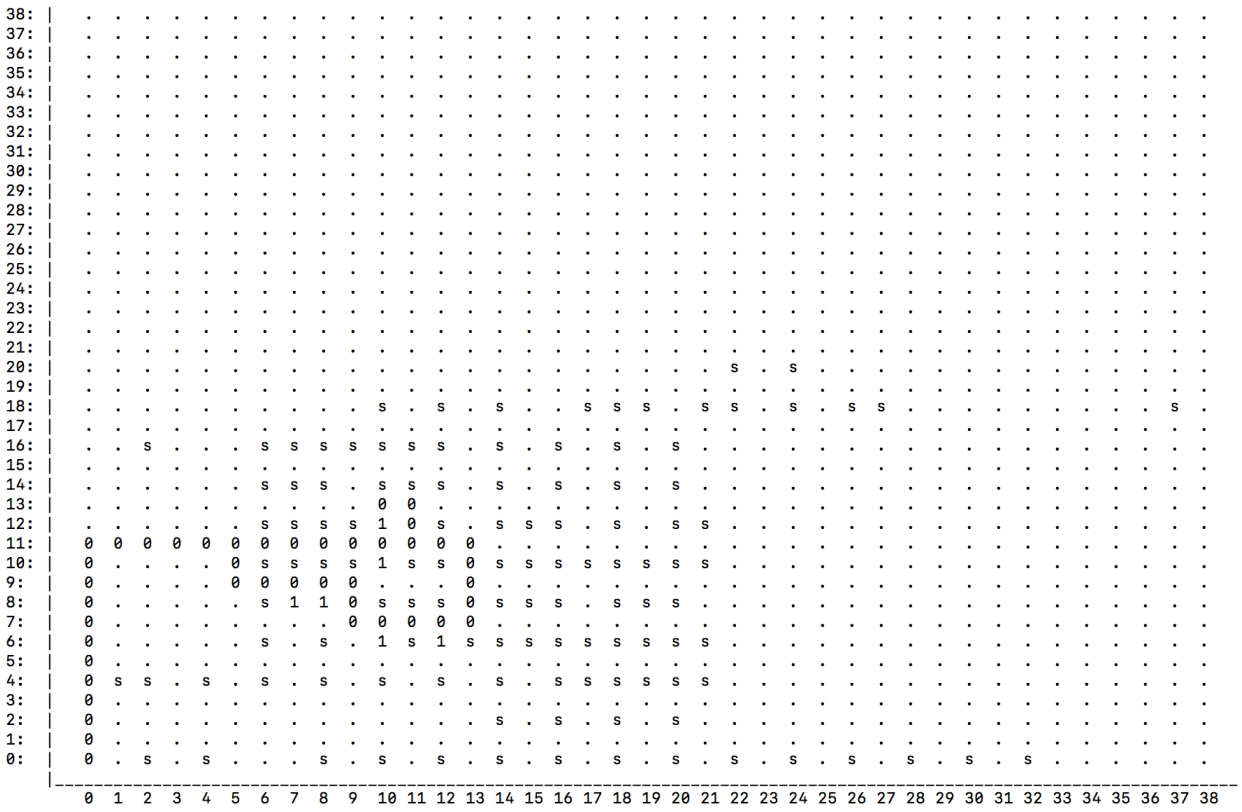
Memory Used: 61.35 MB

Comments: This algorithm takes proper time and give the same result with brute force.

Algorithm 3

Output: ['281610', '342706', '254489', '111873', '287261', '258540', '286457', '198029', '366109', '76283']

Distance: 44



Running Time: 18.14 sec

Memory Used: 509 MB

Comments: This algorithm takes longest time but give the best result.

Test Case B3

Input: [427230, 372539, 396879, 391680, 208660, 105912, 332555, 227534, 68048, 188856, 736830, 736831, 479020, 103313, 365797]

Start Point: [0,0]

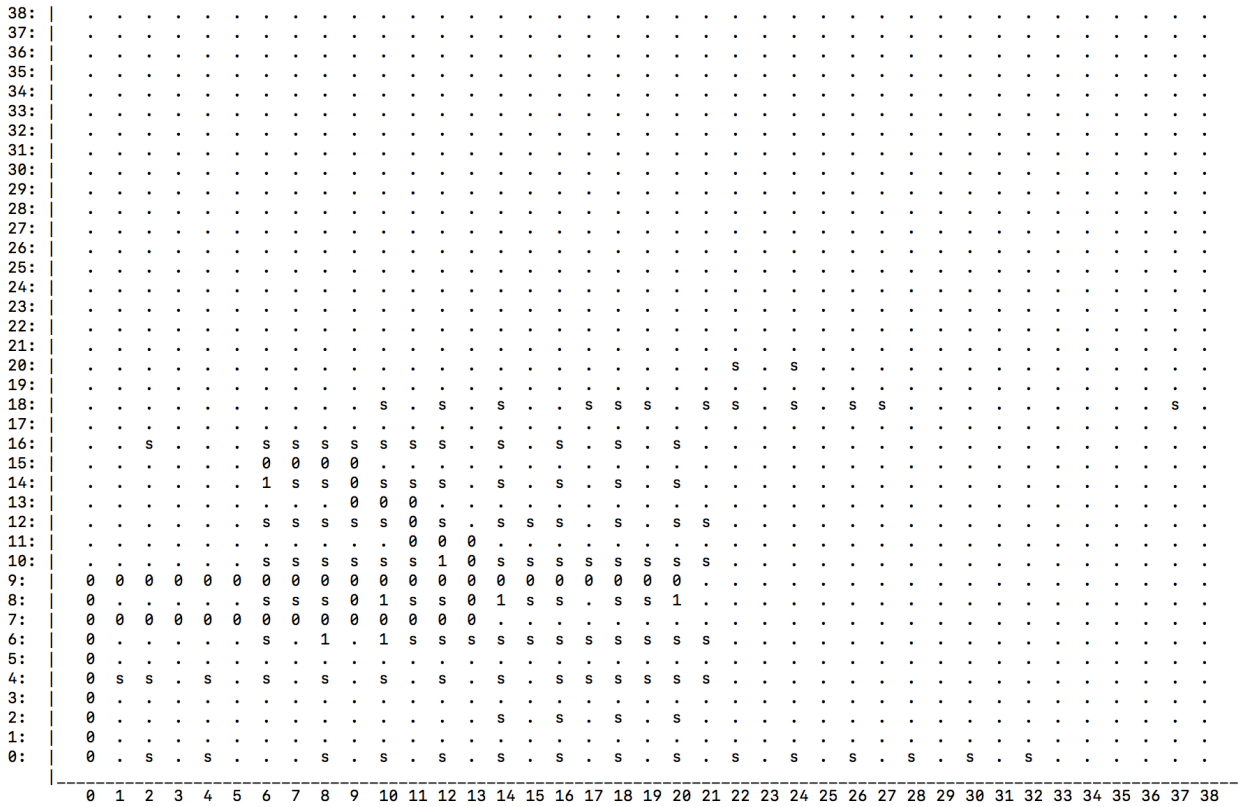
End Point: [0,0]

Distance: 114

Algorithm 1

Output: ['208660', '427230', '372539', '396879', '391680', '188856', '68048', '736830', '736831', '479020', '105912', '332555', '227534', '103313', '365797']

Distance: 86



Running Time: 6.85 sec

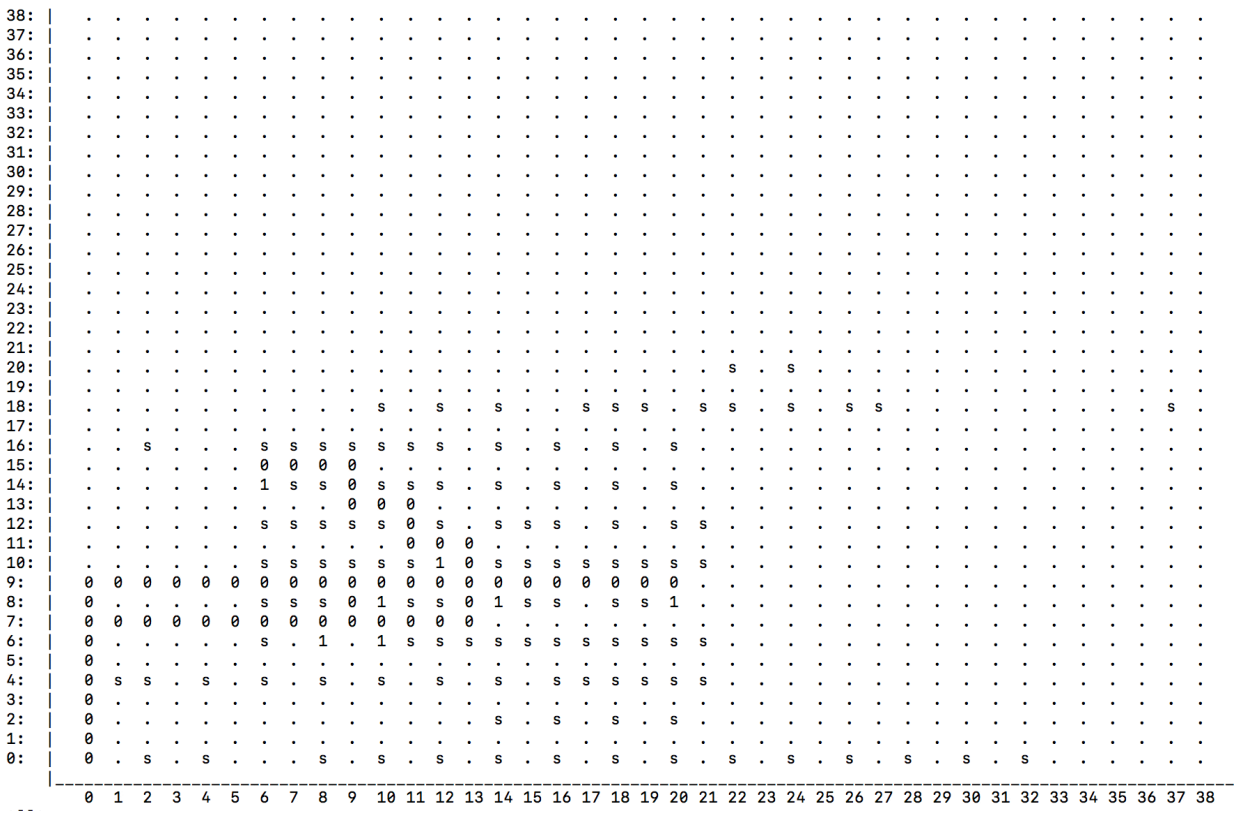
Memory Used: 62.1 MB

Comments: This algorithm takes shorter time but give a worse result.

Algorithm 2

Output: ['427230', '372539', '396879', '391680', '188856', '68048', '736830', '736831', '105912', '332555', '227534', '103313', '479020', '365797', '208660']

Distance: 65



Running Time: 11.09 sec

Memory Used: 61.73 MB

Comments: This algorithm takes longer time and give the better result.

Test Case B4

Input: [108335, 365285, 98888, 352109, 383599, 176484, 245818, 208299, 376926, 1651829, 1622191, 379184, 392028, 391231, 373389, 289088, 2625629, 204257, 380057, 179949]

Start Point: [0,0]

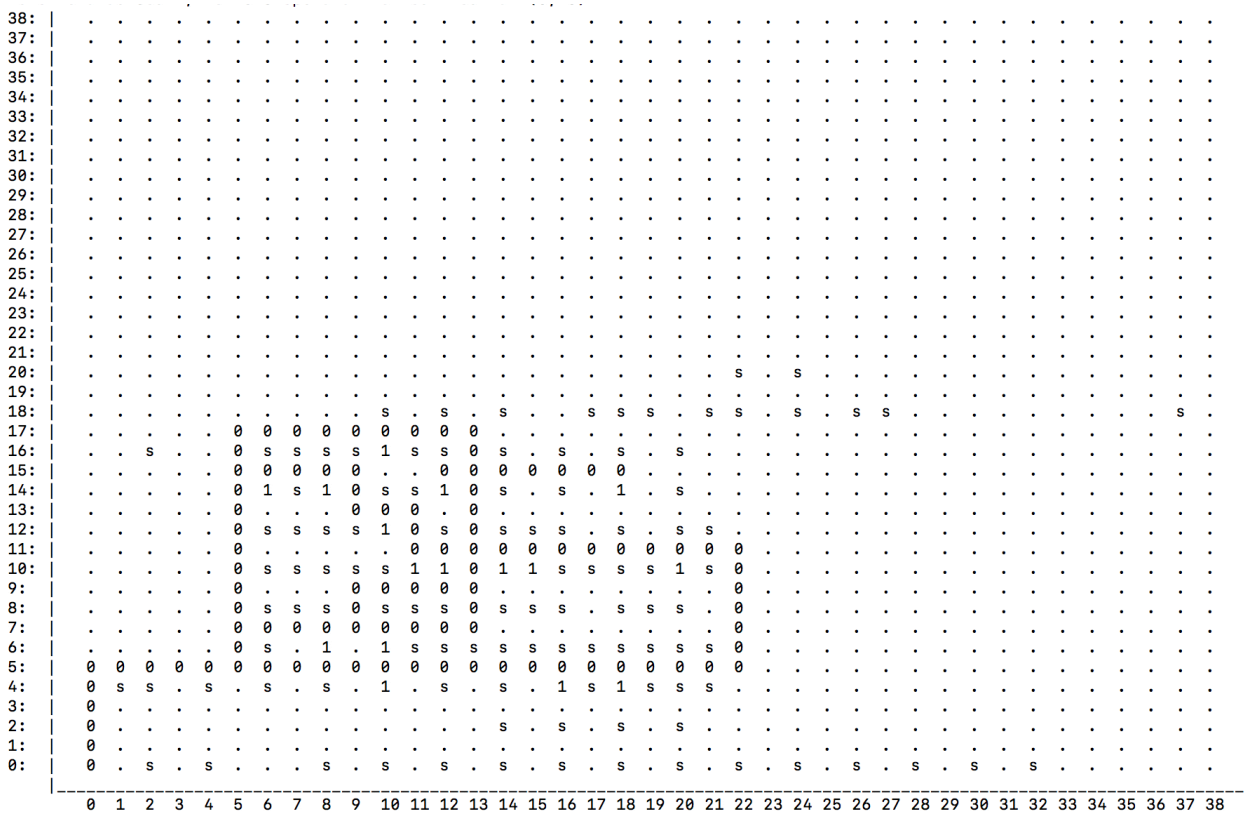
End Point: [0,0]

Distance: 226

Algorithm 1

Output: ['176484', '289088', '376926', '245818', '352109', '373389', '392028', '365285', '208299', '98888', '2625629', '380057', '179949', '391231', '1651829', '1622191', '204257', '108335', '383599', '379184']

Distance: 101



Running Time: 8.87 sec

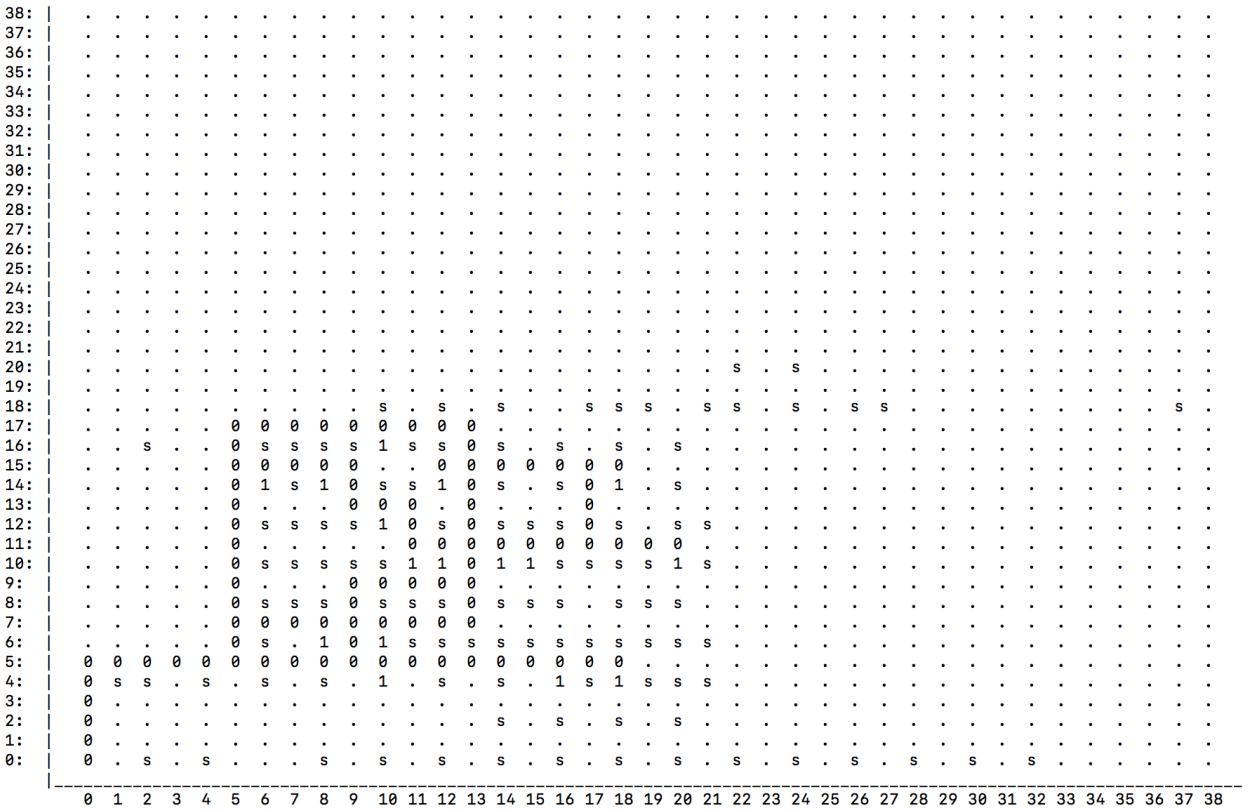
Memory Used: 62.3 MB

Comments: This algorithm takes shorter time but give a worse result.

Algorithm 2

Output: ['379184', '208299', '383599', '108335', '204257', '1622191', '1651829', '391231', '179949', '380057', '2625629', '98888', '365285', '392028', '373389', '352109', '245818', '376926', '289088', '176484']

Distance: 94



Running Time: 15 sec

Memory Used: 61.76 MB

Comments: This algorithm takes longer time and give the better result.

Test Case B5

Input: ['365285', '388082', '2625629', '392028', '373389', '31378', '260033', '208299', '1622191', '1651829', '204257', '179949', '324086', '380057', '180023', '176484', '36346', '379184', '391231', '289088', '205427', '245818', '352109', '2489041', '89769', '108335', '383599', '376926', '98888', '41100']

Start Point: [0,0]

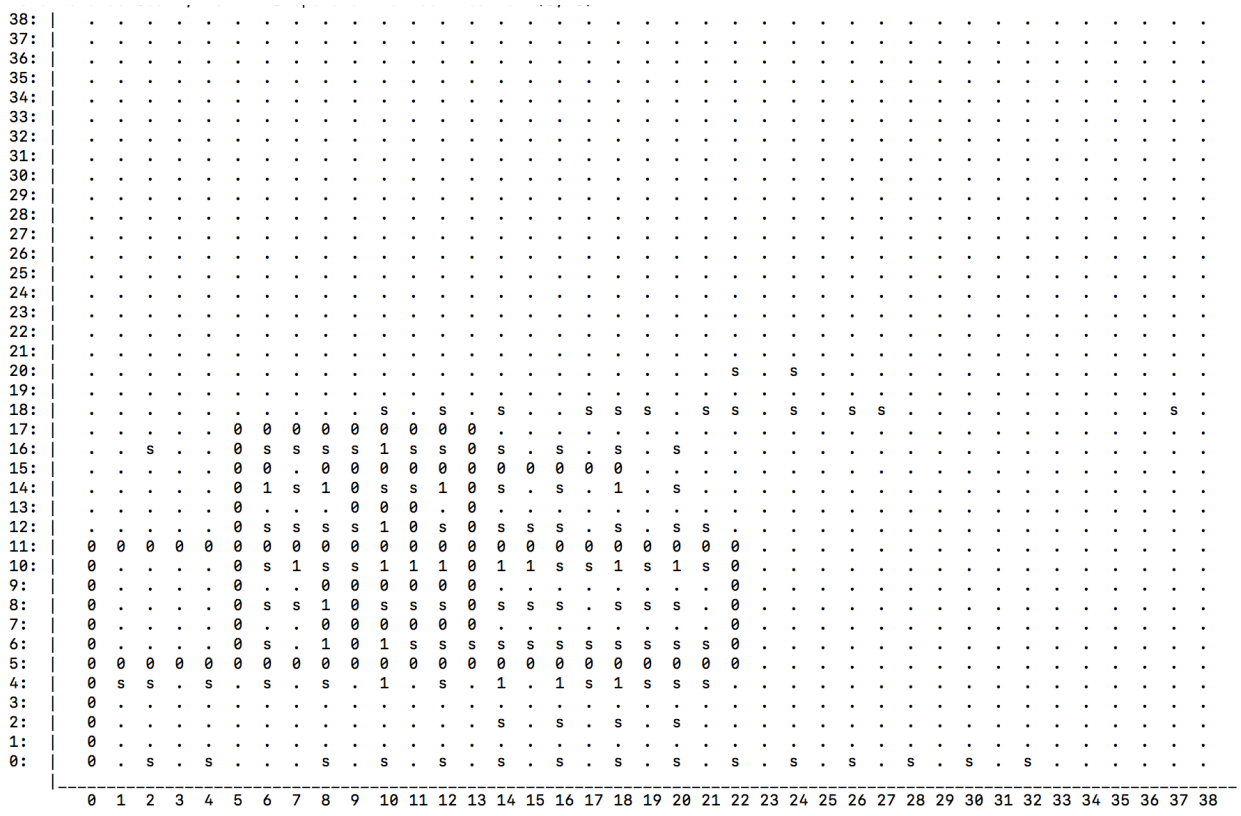
End Point: [0,0]

Distance: 321

Algorithm 1

Output: ['176484', '289088', '376926', '180023', '1651829', '1622191', '204257', '391231', '392028', '260033', '365285', '388082', '352109', '373389', '31378', '383599', '208299', '245818', '205427', '41100', '2625629', '98888', '380057', '179949', '324086', '36346', '379184', '108335', '89769', '2489041']

Distance: 133



Running Time: 10.56 sec

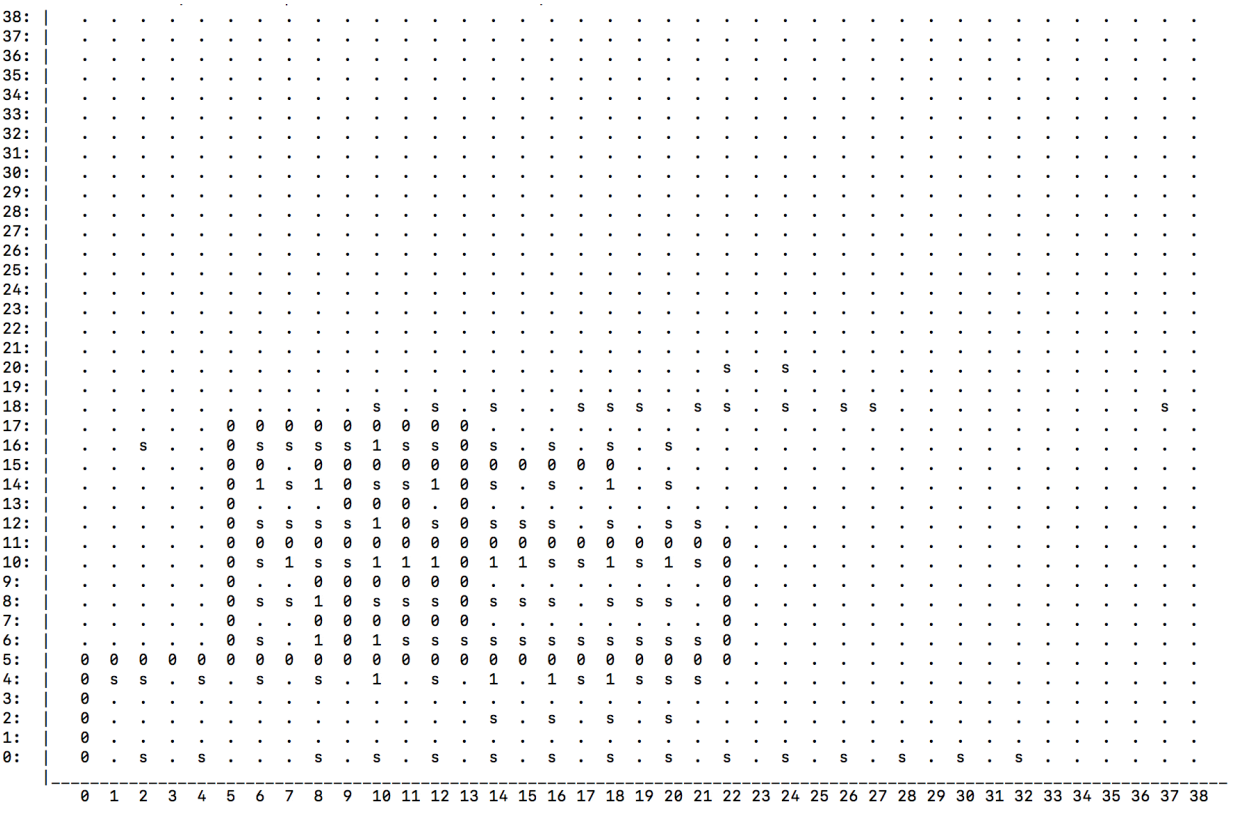
Memory Used: 62.3 MB

Comments: This algorithm takes shorter time but give a worse result.

Algorithm 2

Output: ['176484', '289088', '376926', '180023', '1651829', '1622191', '204257', '391231', '392028', '260033', '365285', '388082', '352109', '373389', '31378', '383599', '208299', '245818', '205427', '41100', '2625629', '98888', '380057', '179949', '324086', '108335', '89769', '2489041', '36346', '379184']

Distance: 125



Running Time: 19.25 sec

Memory Used: 61.83 MB

Comments: This algorithm takes longer time and give the better result.

Test Case B6

Input: ['944527', '180023', '8659', '176484', '289088', '317225', '387238', '388082', '392028', '260033', '352109', '204257', '1651829', '324086', '365285', '1327769', '168753', '205427', '1399145', '373389', '298112', '391231', '379184', '2489041', '365028', '208299', '36346', '98888', '376926', '2625629', '380057', '2217454', '17112', '135359', '245818', '366339', '2217451', '392819', '2217453', '219501', '190046', '208974', '179949', '89769', '1400584', '108335', '1622191', '31378', '383599', '41100']

Start Point: [0,0]

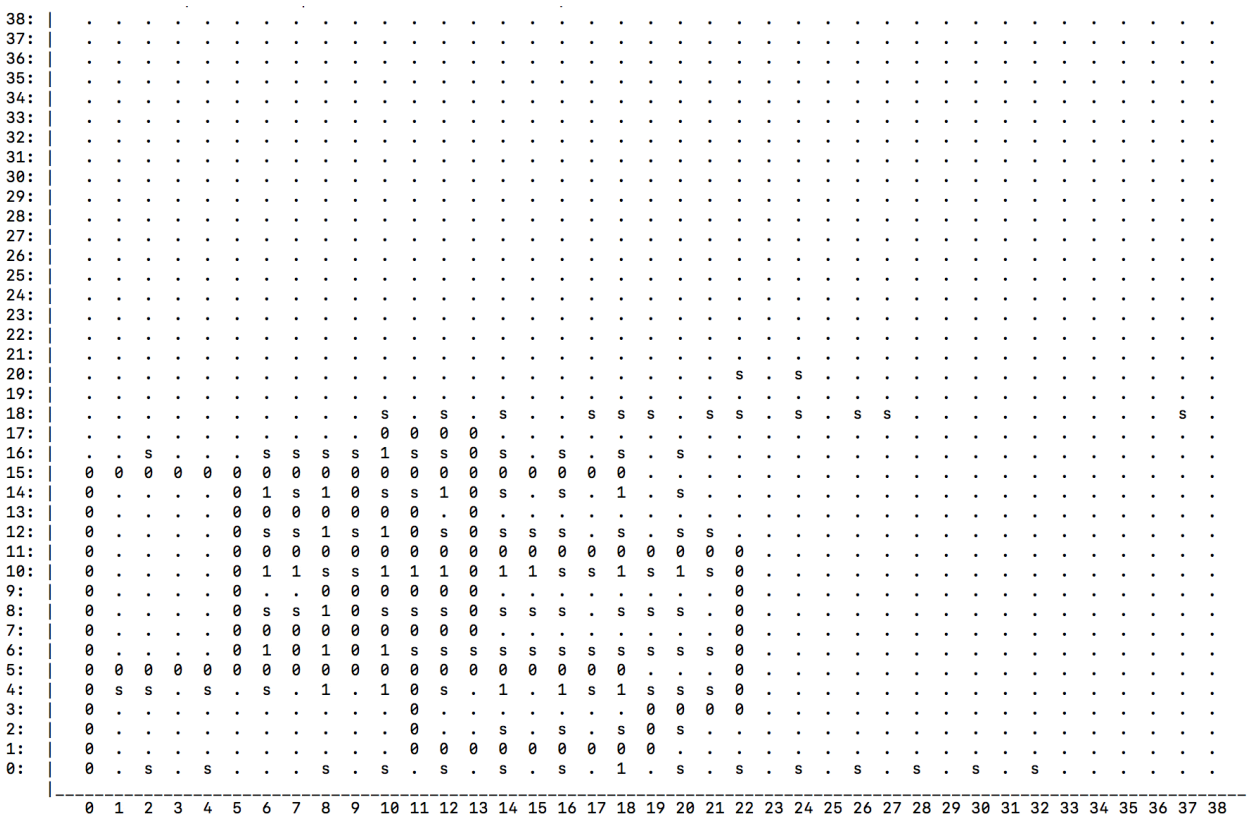
End Point: [0,0]

Distance: 550

Algorithm 1

Output: ['176484', '289088', '8659', '17112', '379184', '391231', '1651829', '1622191', '204257', '190046', '180023', '380057', '179949', '324086', '219501', '168753', '208974', '392028', '260033', '365028', '366339', '387238', '365285', '388082', '352109', '373389', '298112', '1327769', '1399145', '31378', '944527', '2217451', '2217453', '2217454', '392819', '41100', '205427', '135359', '317225', '245818', '208299', '36346', '383599', '108335', '89769', '2489041', '1400584', '376926', '2625629', '98888']

Distance: 162



Running Time: 15.22 sec

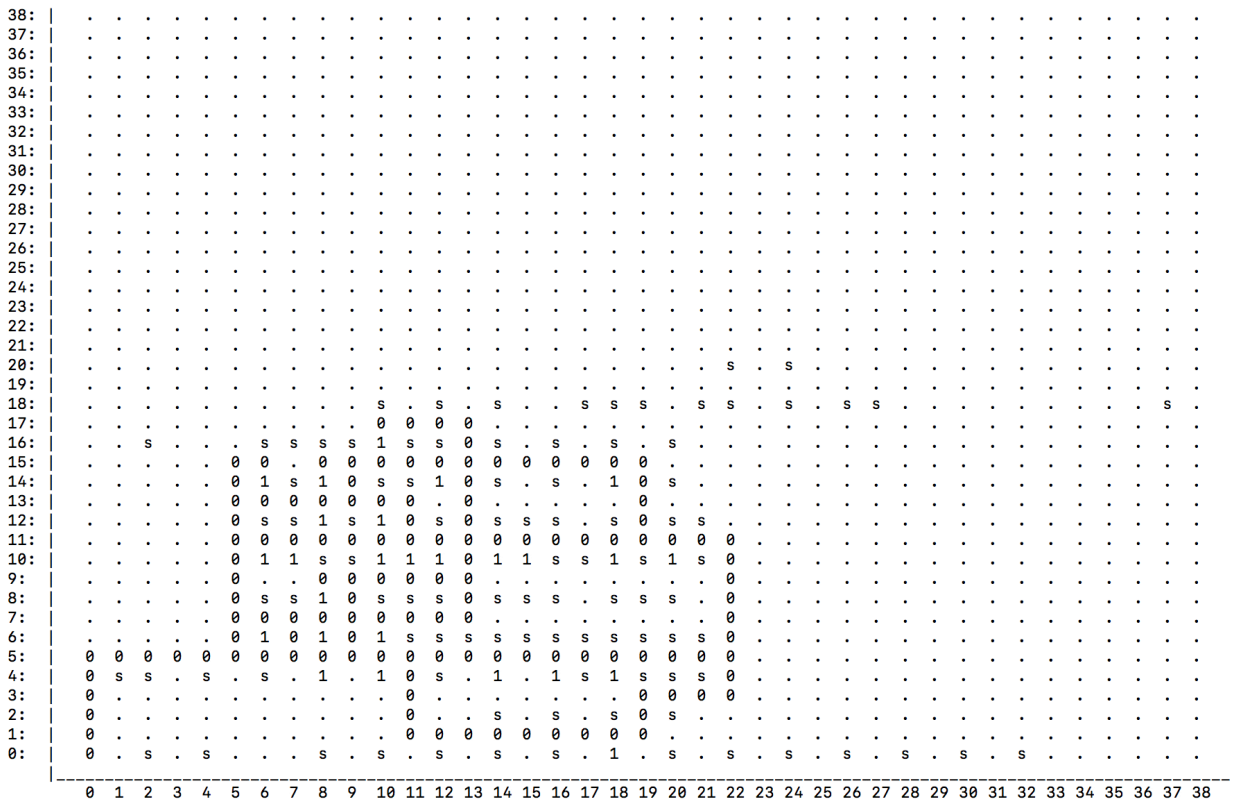
Memory Used: 62.55 MB

Comments: This algorithm takes shorter time but give a worse result.

Algorithm 2

Output: ['36346', '98888', '2625629', '376926', '1400584', '2489041', '89769', '108335', '383599', '208299', '245818', '317225', '135359', '205427', '41100', '392819', '2217454', '2217453', '2217451', '944527', '31378', '1399145', '1327769', '298112', '373389', '352109', '388082', '365285', '387238', '366339', '365028', '260033', '392028', '208974', '168753', '219501', '324086', '179949', '380057', '180023', '190046', '204257', '1622191', '1651829', '391231', '379184', '17112', '8659', '289088', '176484']

Distance: 139



Running Time: 28.42 sec

Memory Used: 62.51MB

Comments: This algorithm takes longer time and give the better result.

Overall Results/Comments

In the test Group A, in test case A1 we can find that algorithm one takes shortest time but give the worst answer; algorithm 2 take the second shortest time but give the best answer; algorithm 3 takes longest time but give the same answer with algorithm 2. Therefore, we take algorithm two, which has 500 initial population. In test case A2 we can find that algorithm one takes shortest time but give the worst answer; algorithm 2 take the second shortest time but give the best answer; algorithm 3 takes longest time but give the same answer with algorithm 2. Therefore, we take algorithm two, which has 0.07 crossover rate and 0.02 mutation rate. In test case A3 we can find that algorithm one takes shortest time but give the worst answer; algorithm 2 take the second shortest time but give the best answer; algorithm 3 takes longest time but give the same answer with algorithm 2. Therefore, we take algorithm two, which has 1500 iterations. Consequently, we finally used the Genetic algorithm with 500 initial population, 0.7 crossover arte, 0.02 mutation rate and 1500 iterations to compare with greedy algorithm and brute force algorithm in test Group B.

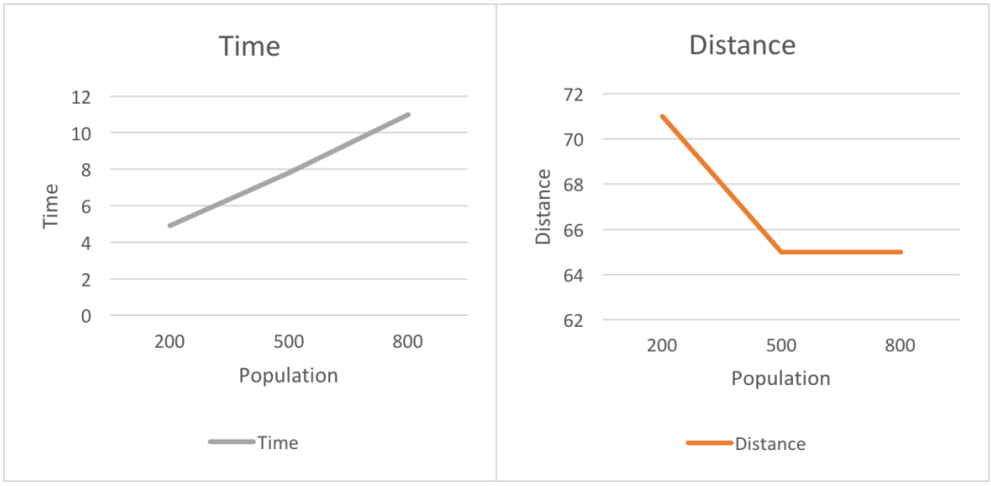


Figure 1. Distance and Time comparison among different initial population

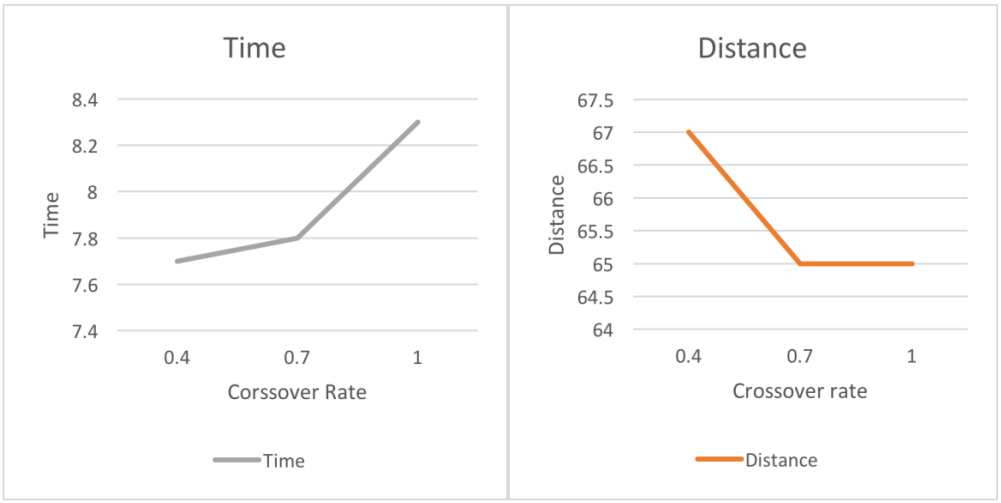


Figure 1. Distance and Time comparison among different Crossover rate

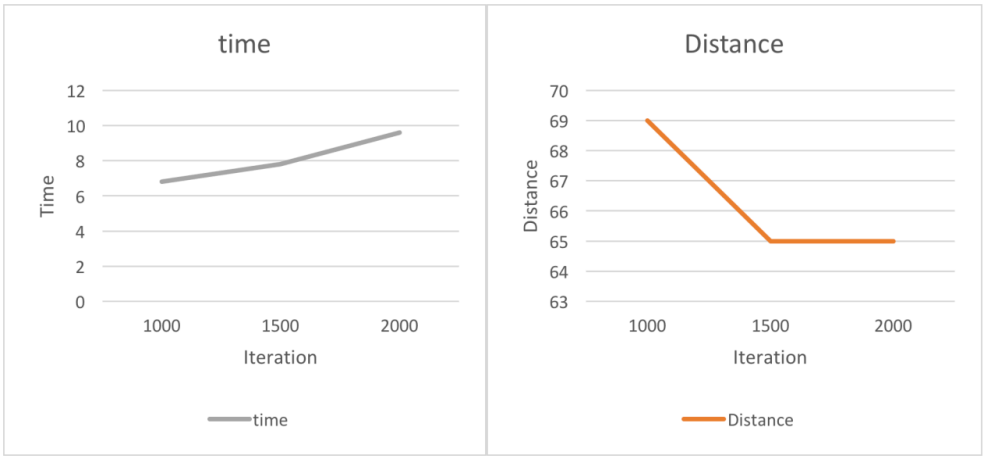


Figure 1. Distance and Time comparison among different iteration

In the test Group B, we tested genetic algorithm, greedy algorithm and brute force algorithm. Brute force can only finish when the order size is less than 10; greedy could spend shortest time but give the worst answer; however, genetic algorithm takes longer time but give the same result with brute force. Since our time out is set to 25 seconds, so when the order size is less than 50 we will use Genetic algorithm, otherwise we will use greedy.

There are two charts showing the comparison between greedy algorithm and genetic algorithm. The first chart shows the distance comparison and the second shows the time comparison.

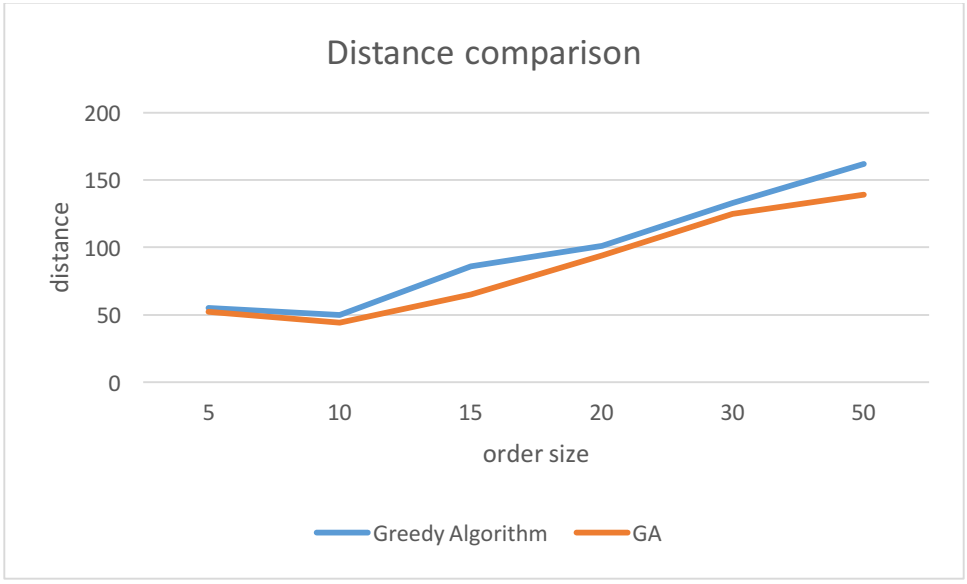


Figure 4. Distance comparison between greedy algorithm and genetic algorithm

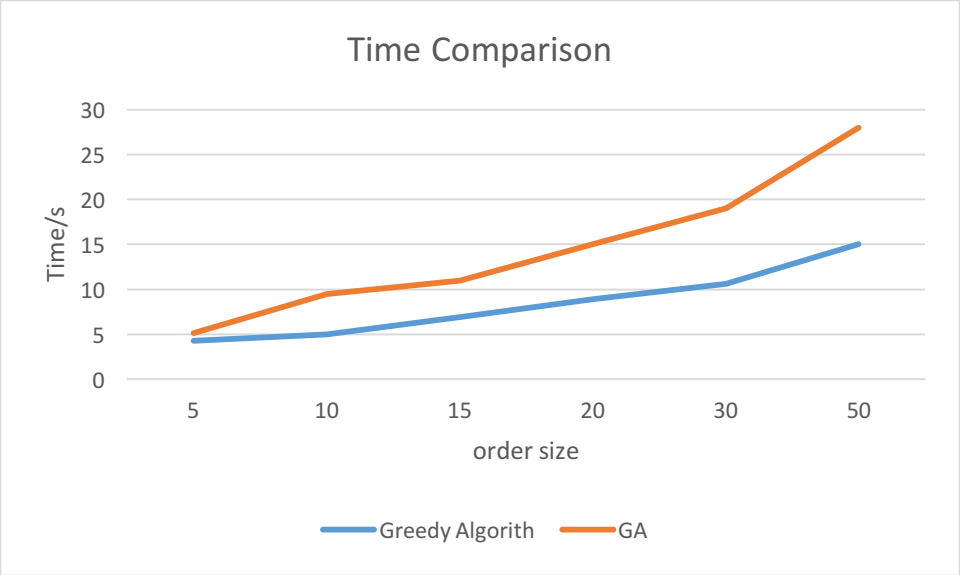


Figure 5. Time comparison between greedy algorithm and genetic algorithm