# Problem 1

Results of the two test cases:

Test Case 1:

1, 0, 1, 0, 0

1, 0, 1, 1, 1

1, 1, 1, 1, 0

1, 1, 0, 1, 0

Output: square size = 2, indices [2, 3];[3, 1]

Test Case 2:

1, 1, 1, 1, 1, 1

1, 1, 1, 1, 0, 0

1, 1, 1, 1, 1, 1

1, 1, 1, 0, 0, 0

1, 0, 1, 0, 1, 1

0, 0, 1, 1, 1, 1

Output: square size = 3, indices [1, 1];[1, 2];[2, 1]

# Problem 2

Results of the two test cases:

Problem 2, Part A:

Test case 1. Input: [4, 2, 7, 1, 3, 6, 9], Output: [4, 7, 2, 9, 6, 3, 1]

Test case 2. Input: [34, 24, 96, 10, null, null, null], Output: [34, 96, 24, null, null, null, 10]

Problem 2, Part B:

Test case 1. Input tree 1: [1, 2, 3, 4, 5, 6, 7], Input tree 2: [1, 3, 2, 7, 6, 5, 4] Output: "Mirror Image"

Test case 2. Input tree 1: [1, 2, 2, null, 3, null, 3], Input tree 2: [1, 2, 2, 3, null, 3, null] Output: "Mirror Image"

# Problem 3

Results of the two test cases:

Test Case 1:

Music service Cleaning service Driving service

Company A [ 108, 125, 149]

Company B [ 150, 135, 175]

Company C [ 122, 148, 250]

Expected output: $406 from ((Company A, Driving service), 149),((Company B, Cleaning service), 135),((Company C, Music service), 122))

Test Case 2:

#191 #122 #173 #121 #128 #104

Driver A [ 22, 14, 120, 21, 4, 51]

Driver B [ 19, 12, 172, 21, 28, 43]

Driver C [ 161, 122, 2, 50, 128, 39]

Driver D [ 19, 22, 90, 11, 28, 4]

Driver E [ 1, 30, 113, 14, 28, 86]

Driver F [ 60, 70, 170, 28, 68, 104]

Expected output: 51 from ((Driver A, #128), 4),((Driver B, #122), 12),((Driver C, #173), 2),((Driver D, #104), 4),((Driver E, #191), 1),((Driver F, #121), 28))

Link for the Hungarian algorithm:

<https://en.wikipedia.org/wiki/Hungarian_algorithm>

<https://github.com/wangshusen/AdvancedAlgorithms/blob/master/Slides/14_BiGraph_4.pdf>

Time complexity analysis:

Step 1:

Time complexity:

Step 2:

Time complexity:

Step 3:

Time complexity:

Step 4:

Time complexity:

Overall: