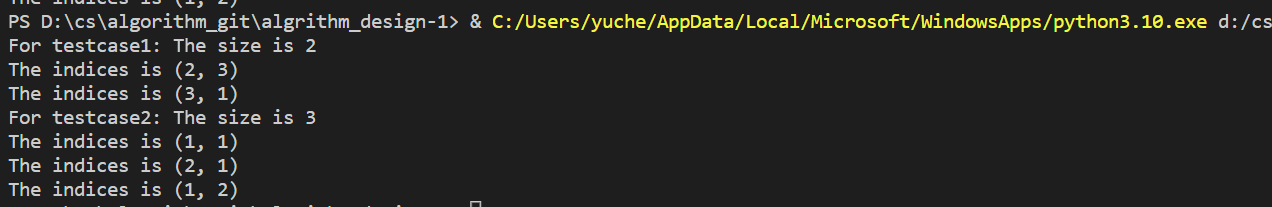
Report

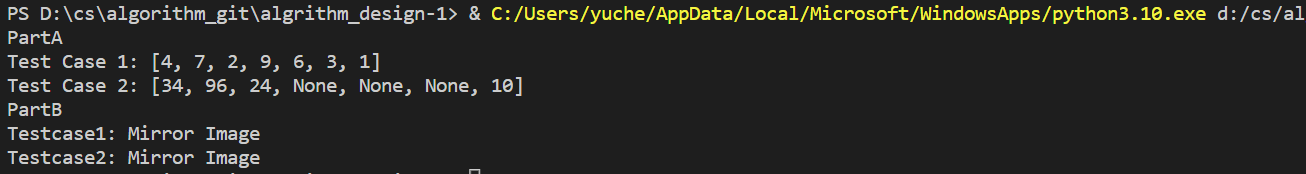
4591774

Yuchen Zhang

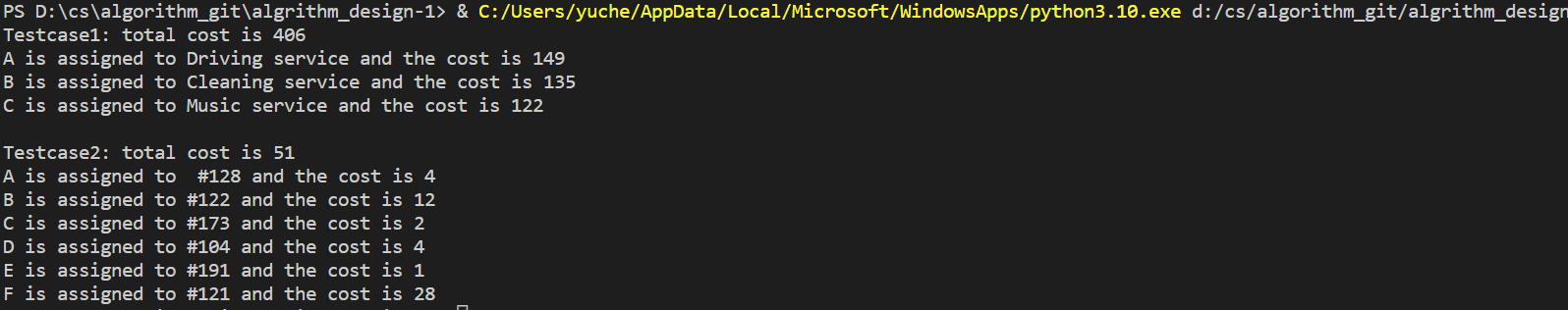
Problem 1:



Problem2:



Problem3:



The time complexity is O(n^3)

The first step and the second step scan and update the elements in the matrix. Since there are a total of m \* n(It can also be approximated as the maximum value of n among m and n, so m \* n become n^2) elements in the matrix, the time complexity of both steps is O(n^2)

The third step is to cover all zero elements with a straight line, so this step needs to visit all zero elements, where the number of zero elements is up to n^2.

The fourth step requires scanning and updating the elements of the matrix by up to n^2.

Thus the third step, and the fourth step are both O(n^2).

But the third and fourth steps need to be iterated until the minimum number of straight lines is equal to n when the iteration is stopped. At each iteration, the minimum number of straight lines increases by at least 1, so the maximum number of iterations is n.

Therefore, the third step, the fourth step has the highest time complexity of O(n^3).

The fifth step yields the final assignment scheme with a time complexity of O(n).

Thus the time complexity of the Hungarian algorithm is O(n^3).

# The resource is from <https://python.plainenglish.io/hungarian-algorithm-introduction-python-implementation-93e7c0890e15>, <https://brilliant.org/wiki/hungarian-matching/#citation-3> and <https://blog.csdn.net/u014754127/article/details/78086014>