Algorithm Design Document

General Strategy:
The algorithm is based on an Partial Randomized Strategy. We first assume the order of dropping TAs is the same, or at least similar, to the TSP order of the graph. After we solve the TSP order, we will then apply dynamic programming to get the optimized route with this dropping order. Then, we could allow some small variation from the previous TSP order and generate some variant order and apply the dynamic programing to solve the optimized rout again. In the end, we will return the solution with the smallest cost.

TSP Order Calculation
First, We will solve the classical Traveling Salesman Problem (TSP) on the original graph, using the Minimal Spanning Tree (MST) based approximation algorithm, which could be solved in polynomial time. It will first computing the minimal spanning tree of the Graph. Then, it will convert the MST to a Euclidean Graph and find the Euclidean Tour of the graph. In the end, it will convert the Euclidean tour back to TSP order and we will output the Home order base on this TSP order.

Find the Optimal Route from TSP Order
With this dropping order, the original problem will become a dynamic programing problem. The professor could start from the starting vertex and go to a random location. At each location the first TA still in the car should decide if he want to get off and then walk home. After the TAs getting off the car if there is one the professor will go to another location. After all the TA is dropped the professor would like to go back to the original vertex in shortest path. We will record the route and TAs' dropping order in a Hash-Table and output the optimized order with the minimal cost.

Relaxation and Generalization
The solution that is output by the previous steps might not be optimal, since the dropping order generated by the approximated TSP could not be change. Now, we will allow some variation (i.e. N=20) by reshuffle the order in a small range (i.e. each vertex could not deviate from the original order by 3, etc). This relaxation method will give us a higher probability to migrate to other local optimal solution, which might be closer to the global optimal. After getting the new orders, we will calculate the optimized route again. Finally, we will output the solution with the smallest cost.