Design Doc

IDEA

We first considered that the algorithm should be a greedy algorithm, because our goal is to find the minimum cost. And the cost will be maximum if the car goes nowhere and everyone has to walk home. So the most greedy method is to drive to a new location which minimizes the total cost every time. Since we have made the best choice each time, the total cost should be the minimum, so we believe that our algorithm can work properly. However, it still remains unknown that where to drop off the TAs, so it is hard to compute the cost. To solve this, we decide to store the distance between each home vertex and all other location vertices, which will be described in detail below.

ALGORITHM

We will construct a table for each home vertex, storing the name of every location vertex and distance between the location vertex and the home vertex in ascending order, so that no matter which vertices our path contains, we can always determine the best locations to drop off every TA by scanning these tables.

Then we can use the greedy algorithm. The initial situation is that the car goes nowhere and everyone has to walk home. Then we scan all the neighbor vertices of the s vertex(start), (because now we have tables constructed above, we know where to drop off all the TAs properly, it is practical to evaluate the cost of every possible path), and find the vertex that minimizes the current cost, add it to the path, then do scanning on its neighbors and repeat the algorithm. The loop of the algorithm will never stop until there is no neighbor vertex of the current vertex which could make the total cost smaller. After that, we can run the shortest path algorithm(like Dijkstra)   from the current vertex to the vertex s, and add the cost of this shortest path to the total cost, which is now the cost of a cycle.

TESTING

Also, we design another algorithm for testing. The algorithm is simple, we just drive to every home vertex, and try to find the minimum Hamilton cycle from s. Our algorithm above should work better than the testing algorithm, and the result of the test will be covered in Phase 2.