Name: Jon Abrahamson
ID: 107084898

CSCI 3104, Algorithms **Profs. Grochow & Layer**
Explain-It-Back 11 **Spring 2019, CU-Boulder**

A startup has hired you as the chief technology officer (i.e., the only one who knows how to program). After the founders (all MBAs) finish explaining their vision for changing the world, you realize that what they describe can be reduced to the traveling salesman problem. No worries, you develop a solution that is a 1.5 approximation. The founders are devastated that they cannot use the word "optimal" in their next VC pitch, and wonder out loud if they need to get a new CTO who can do better. Convince them that an efficient optimal solution is unlikely (i.e., P probably does not equal NP) and that your solution is quite good.

Name: Jon Abrahamson
ID: 107084898

CSCI 3104, Algorithms **Profs. Grochow & Layer**
Explain-It-Back 11 **Spring 2019, CU-Boulder**

No matter what CTO that you hire, you will never be able to call your solution "optimal". Our problem boils down to one similar to the traveling salesman problem. The problem asks the following question: Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city and returns to the origin city?

To solve this problem, one merely need to compute the length of all possible routes and keep the shortest one. The issue is that as the number of cities the salesman has to visit grows, the number of routes between cities grows much faster. If there are n cities then this number is factorial of n-1 = (n-1)(n-2)...3*2*1. We can select arbitrarily one city to start from (the starting point doesn't matter much given one must end at the same place). Then we have n-1 different choices for the second city to be visited, n-2 choices for the third city, and so on. For 16 cities there are more than a trillion different routes.

In order to check that a proposed tour is a solution of the TSP in NP we need to check two things, namely: 1. That each city is is visited only once 2. That there is no shorter tour than the one we are checking

For the first, we merely have to check that each city is visited once. This can be done in polynomial time, hence the TSP is P. The second condition is what makes the problem difficult to solve. As of today, no one has found a way to check condition 2 in polynomial time. It means that the TSP isn't in NP, as far as we know. Therefore, TSP isn't NP as far as we know.