The Traveling SalesmanProblem
Group Work, CS24
Prof. Ostheimer

Learning Objectives:

1. get to know each other;
2. practice solving unfamiliar problems independently;
3. practice close reading in the understanding of definitions and problem specifications;
4. practice analytical thinking by analyzing the time complexity of an algorithm;
5. develop an appreciation for the way in which mathematics helps us analyze computer programs;
6. develop an appreciation for the challenges inherent in finding fast solutions to problems.

Problem Specification:

- **Input:** a list of cities, and a table giving the distances between every pair of cities;
- **Output:** a shortest tour through all the cities.

Note that by a "tour" we mean a way of visiting each city once and only once and then winding up back at the first city visited.

Warm-up Question: Why do I say *a* tour rather than *the* tour?

Algorithm Description: One possible algorithm for this problem is the so-called "Exhaustive Algorithm" in which we list out all the tours, calculate the length of each tour, and choose a tour with the shortest length.

Algorithm Analysis: How long do you think it will take a computer to find the shortest tour using the exhaustive algorithm? This is the question that you will explore in the group work that follows.

1. Before you begin, please introduce yourselves to each other, and get to know each other a little.

2. To facilitate working together, please choose a scribe (the person who will write down your answers), a communicator (the person who will communicate the group's questions to me or to the communicators of other groups) and a facilitator, who will gently keep the group on task. Now it's one-for-all and all-for-one: make sure that everyone's opinions and questions (no matter how quiet) are heard.

3. Suppose that we want to visit three cities named a, b, c? How many tours are there? List them all.

4. Answer the same question if we want to visit four cities a, b, c, d.

5. How many tours are there if there are five cities?

6. Suppose that you want to visit n cities, where n is some positive number. How many tours are there? (Here your answer will have to refer to n, of course.)

7. How many additions are required to find the length of a single tour through n cities?

8. Suppose your computer can perform 10 billion additions per second. How long will it take to perform the additions of the exhaustive algorithm if you want to visit n cities?

9. Estimate the time needed when $n = 100$. Make sure to express your answer in a reasonable unit of time.