A Hamiltonian cycle is a cycle in a graph that visits every vertex exactly once.

The Traveling Salesman Problem (or TSP for short):

Given a list of “cities” and the “distances” between each pair of cities, find the shortest possible route that visits every city exactly once and then returns to the origin.

In other words, given a weighted graph, the TSP asks to find a Hamiltonian cycle of the smallest weight.

A decision problem is a yes/no question that you can ask a computer to figure out. Decision problems are organized into complexity classes, depending on how “hard” it is to solve them. Usually, “hardness” is measured in terms of the amount of time (or, more precisely, the number of steps) needed to solve the problem.

The complexity class \( P \) contains problems that can be, in some sense, solved efficiently (i.e., quickly).

The complexity class \( NP \) contains problems such that if the answer is “yes,” then there is an efficient way to check whether a proposed solution is correct.

One of the most important open problems in modern mathematics is to find out if \( P = NP \) or not.

A problem in \( NP \) is \( NP\)-complete if it is, in a certain sense, at least as hard as any other problem in \( NP \). Assuming \( P \neq NP \), it is hopeless to find an efficient general method for solving any \( NP\)-complete problem. Because of that, when faced with an \( NP\)-complete problem, it is often reasonable to use heuristics instead of trying to find the exact solution (for example, the Nearest Neighbor heuristic for the TSP).