# Statement

**Note:** The first exercise is taken from the exercise series. Since this exercise is a proof of NP-Completeness as we will be doing throughout this Workshop, it is essential to have understood this exercise. If you have not managed to complete it independently, you can start with this exercise. Otherwise, skip to exercise 2.

1. Consider the two Hamiltonian Cycle and Hamiltonian Path problems.

Hamiltonian Cycle

*Data*: an undirected graph *G*.

*Question*: Does *G* contain a Hamiltonian cycle?

Hamiltonian Path

*Data*: an undirected graph *G*, two distinct vertices *u* and *v* of *G*.

*Question*: Does *G* contain a Hamiltonian path (or Hamiltonian trail) between *u* and *v*?

Assuming that the Hamiltonian Cycle problem is NP-Complete. So it is necessary to prove that the Hamiltonian Path is also NP-Complete. To do this, it is necessary to show that the Hamiltonian Path problem is in NP (meaning that the correctness of a solution can be verified in polynomial time), and that it is NP-Hard (at least as hard as any NP problem). Since it is among the hardest NP problems, it is NP-Complete.

Reminder: The difference between a chain and a cycle is that a cycle returns to its starting point.

1. Show that the Hamiltonian Path problem is in NP:

Consider:

* an instance ICH of the Hamiltonian Path problem, comprising graph *G*=(*V*, *E*)
* a sequence of vertices SCh = {*u*1,…, *un*} of *V*.

Propose an algorithm that uses ICh and SCh as input parameters, and checks whether SCh is a Hamiltonian path. Prove that the asymptotic complexity of this algorithm is a polynomial with the same length as ICh.

1. Show that the Hamiltonian Cycle problem is reducible in polynomial time to the Hamiltonian Path problem:

Consider an algorithm that uses, as input parameter, an instance ICy of the Hamiltonian Cycle problem, comprising graph *G*=(*V*, *E*), and that returns the instance ICh of the Hamiltonian Path problem comprising:

* graph *G’,* obtained by adding a vertex *v* to *G*, and by connecting said vertex to all the neighbours of a vertex *u* arbitrarily chosen in *G*
* both vertices *u* and *v*

In the following example, the vertex chosen to transform ICy into ICh is *u*:

ICy

ICh

* + 1. Show that the asymptotic complexity of this algorithm is polynomial.
    2. Show that if there is a Hamiltonian path from *u* to *v* in *G’*, then there is a Hamiltonian cycle in *G*.

In the previous example, a possible solution to the instance ICh of the directed Hamiltonian path (or trail) between *u* and *v* is (*u, a, b, c, d, v*), and a possible solution to the instance ICy of the Hamiltonian cycle is (*u, a, b, c, d, u*), which is obtained by replacing *v* with *u* in the solution of ICh.

**Please note:** This example is meant to illustrate the reasoning, given that the goal is not to demonstrate the existence of a cycle in this particular graph, but in every graph obtained by making the transformation shown above from any graph.

* + 1. Show that if there is no Hamiltonian path from *u* to *v* in *G’*, then there is no Hamiltonian cycle in *G*.

Determine that the problem is NP-Hard.

1. Consider the following Knights of the Round Table decision problem:

Knights of the Round Table

*Data*: *n* knights, the list with all pairs of enemy knights.

*Question*: Is there a way of placing the knights around the round table so that two enemy knights do not sit next to each other?

Show that the Knights Of The Round Table problem is NP-Complete.

Here’s a hint: Represent each knight by a vertex, and the fact that two knights are not enemies by an edge between the two corresponding vertices.

Here’s another hint: You already know the problem to be used for the polynomial-time reduction.

1. Consider the Travelling Salesman problem:

Travelling Salesman

*Data*: A complete edge-weighted graph *G*, and an integer *k*.

*Question*: Is there a route that passes at least once through each vertex of *G* and whose sum of edge weights is at most *k*?

Show that the Travelling Salesman is NP-Complete.

1. Consider the Incomplete Travelling Salesman problem:

Incomplete Travelling Salesman Problem

*Data*: An edge-weighted graph *G*, and an integer *k*.

*Question*: Is there a route that passes at least once through each vertex of *G* and whose sum of edge weights is at most *k*?

Show that the Incomplete Travelling Salesman Problem is NP-Complete:

1. Find a way to transform an instance of the Incomplete Travelling Salesman Problem into an instance of the Travelling Salesman.
2. What specific property does the resulting graph verify, more specifically the distances? Can we determine from the NP-Completeness of the Travelling Salesman that of the Incomplete Travelling Salesman Problem?
3. Adapt the demonstration in question 7. and draw a conclusion.