Business Processes Modelling

MPB (6 cfu, 295AA)

Roberto Bruni
http://www.di.unipi.it/~bruni

* - P and NP problems

Computational Complexity Theory

Computability theory studies the existence of algorithms that can solve a class of problems

For example, no algorithm exists that can be used to decide in a finite amount of time if any C (or Java) program terminates or diverges (on a given input)

Computational complexity theory deals with the resources needed to solve a solvable problem

For example, how many steps (time) or memory (space) it takes to solve a problem

Decision problem

A **problem** defines a set of related questions,
each of finite length

A **problem instance** is one such question

For example, the factorization problem is:
"given an integer n, return all its prime factors"
An instance of the factorization problem is:
"return all prime factors of 18"

A **decision problem** requires just a **boolean answer**
For example: *"given a number n, is n prime?"*
And an instance: *"is 18 prime?"*

P

The complexity class **P** is the set of decision problems that can be solved by a deterministic (Turing) machine in a **P**olynomial number of steps (time) w.r.t. input size

Problems in **P** can be (checked and) **solved effectively**

NP

The complexity class **NP** is the set of decision problems that can be **solved** by a **N**on-deterministic (Turing) machine in a **P**olynomial number of steps (time)

Equivalently **NP** is the set of decision problems whose solutions can be **checked** by a deterministic (Turing) machine in a polynomial number of steps (time)

Solutions of problems in **NP** can be **checked effectively**

P vs NP

The question of whether **P** is the same set as **NP** is the most important open question in computer science

Intuitively, it is much harder to solve a problem than to check the correctness of a solution

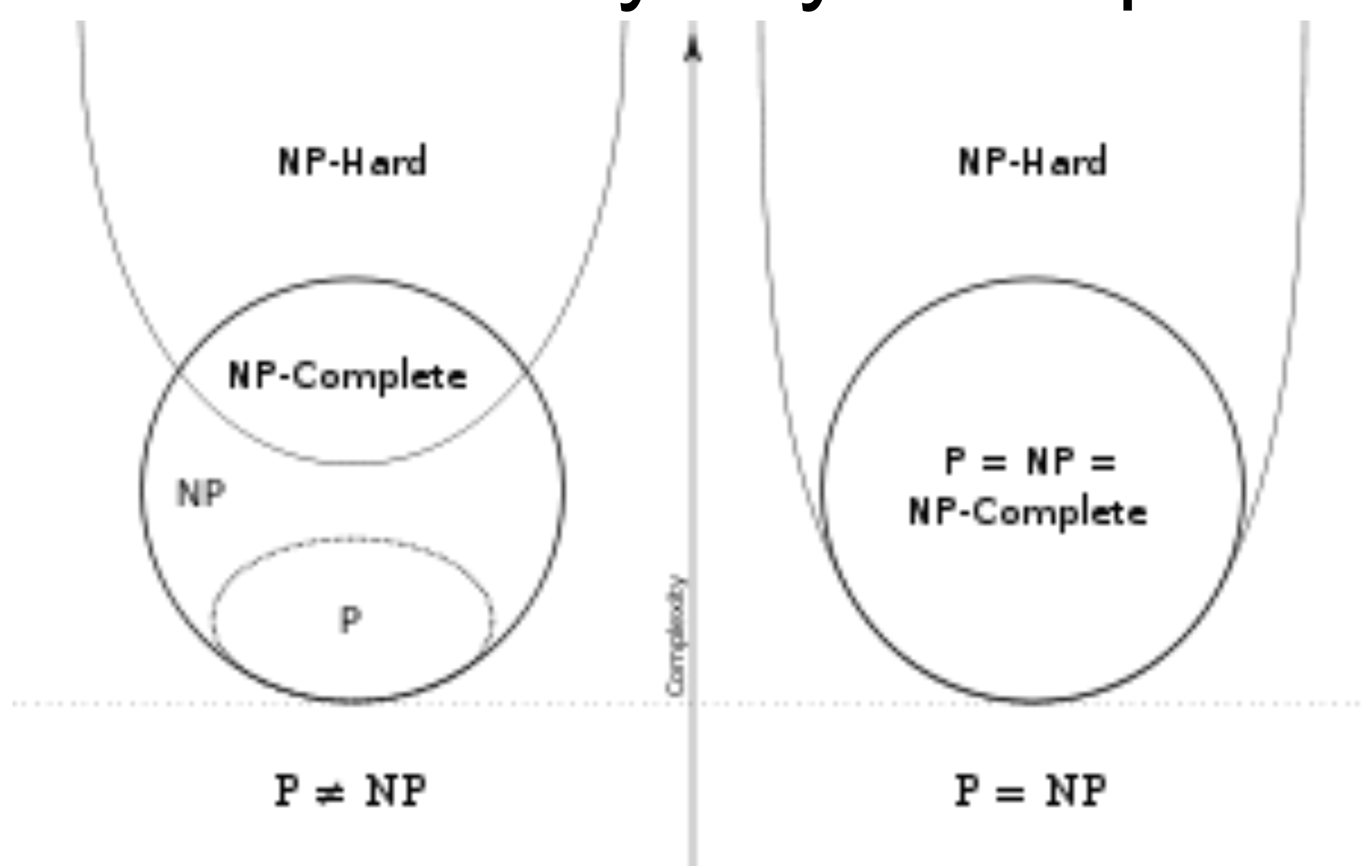
A fact supported by our daily experience, which leads us to conjecture **P** ≠ **NP**

What if “solving” is not really harder than “checking”? what if **P** = **NP**?

NP-completeness

A problem Q in **NP** is **NP-complete** if every other problem in **NP** can be reduced to Q (in polynomial time)

(finding an effective way to solve such a problem Q would allow to solve effectively any other problem in **NP**)

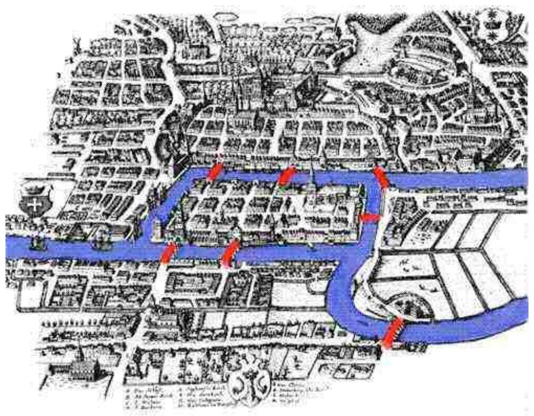


Eulerian circuit problem (P)

Given a graph G,
is it possible to draw an Eulerian circuit over it?
(i.e. a circuit that traverses each edge exactly once)

We have seen that it is the same problem as:

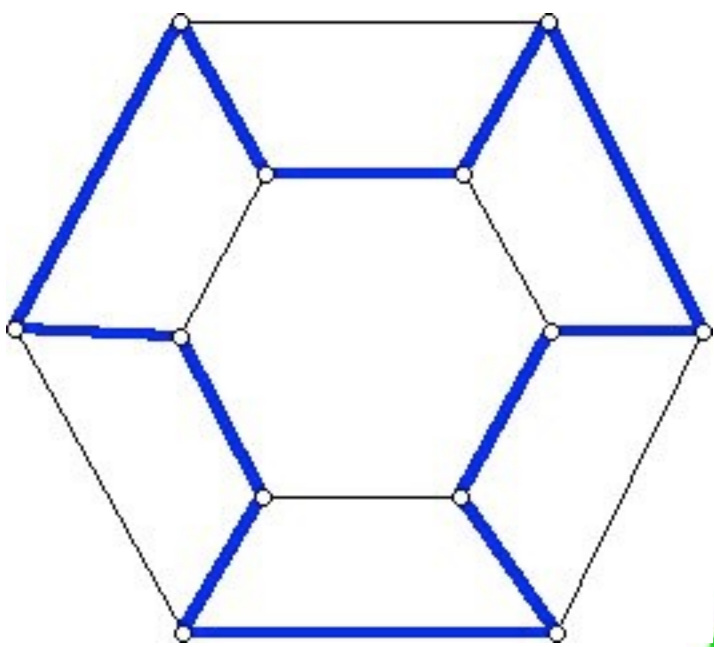
Given a graph G,
is the degree of each vertex even?



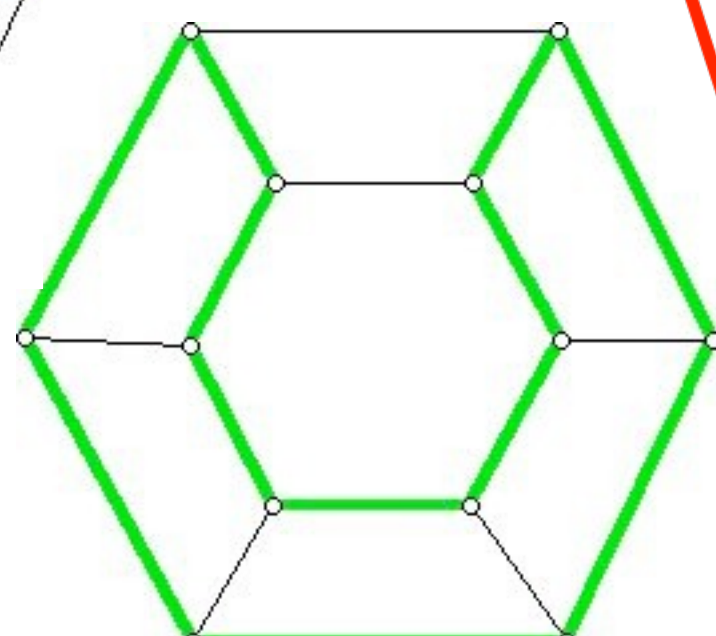
The problem can be solved effectively!

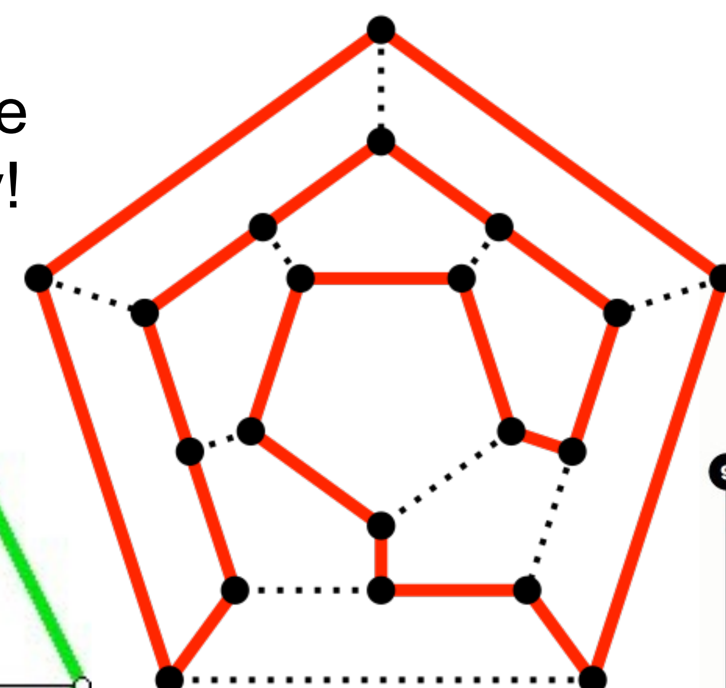
Hamiltonian circuit problem (NP-complete)

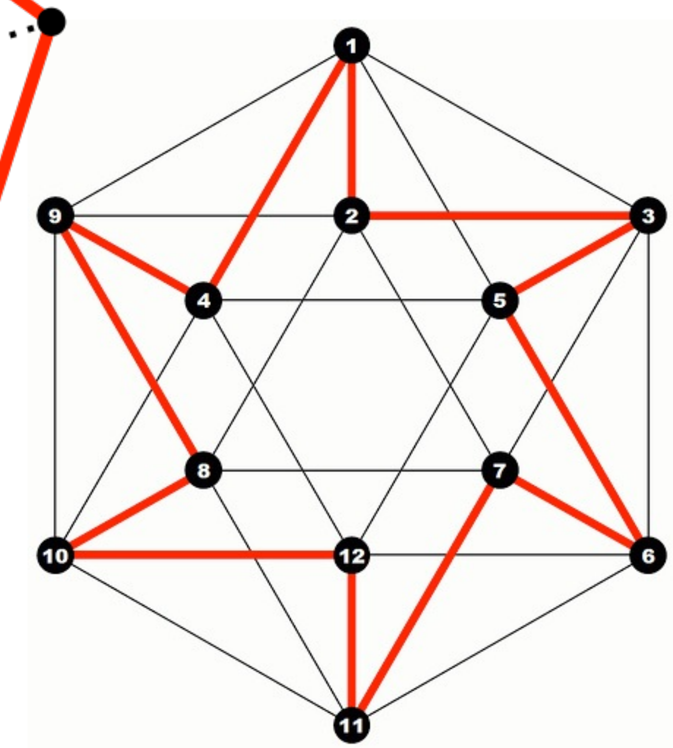
Given a graph G,
is it possible to draw an Hamiltonian circuit over it?
(i.e. a circuit that visits each vertex exactly once)



The problem can be checked effectively!



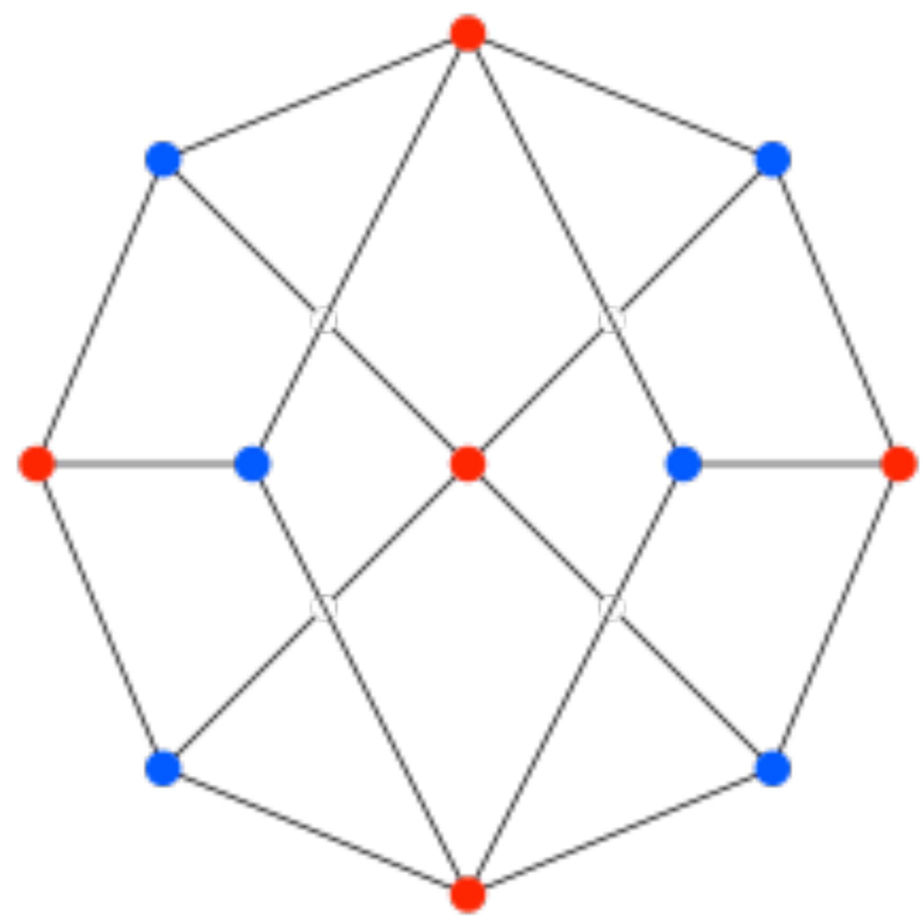




Hamiltonian circuit problem (NP-complete)

Given a graph G,
is it possible to draw an Hamiltonian circuit over it?
(i.e. a circuit that visits each vertex exactly once)

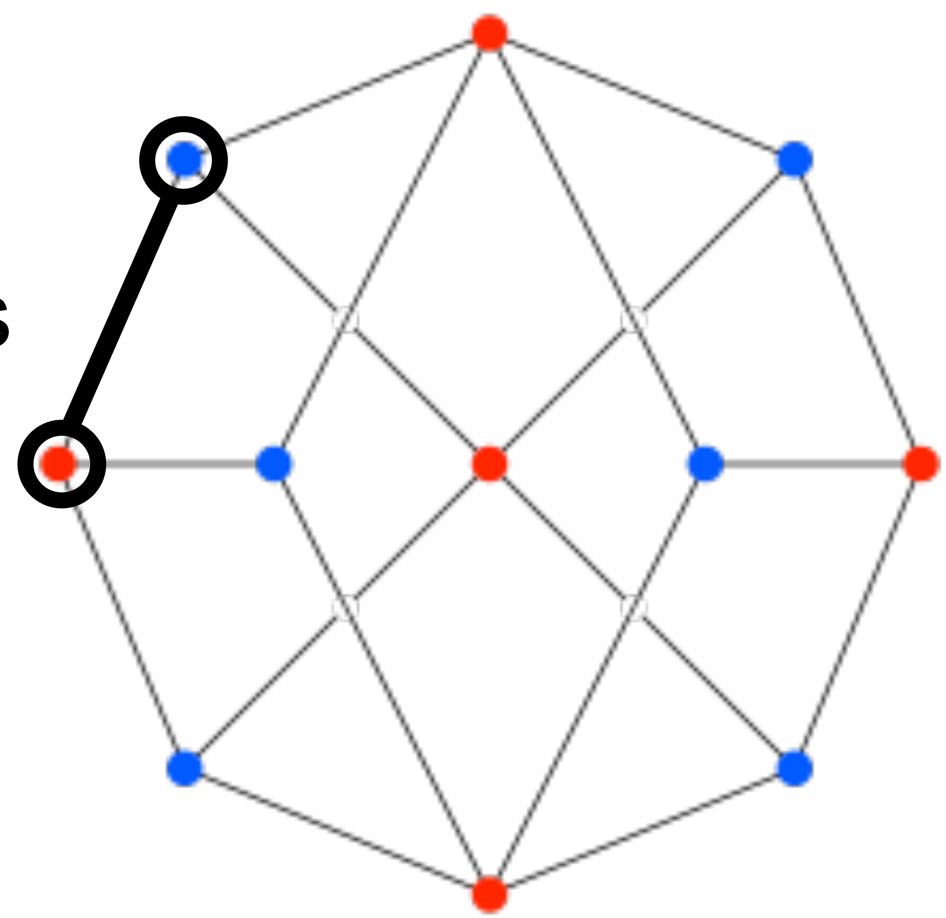
The problem looks difficult to solve



Hamiltonian circuit problem (NP-complete)

Given a graph G,
is it possible to draw an Hamiltonian circuit over it?
(i.e. a circuit that visits each vertex exactly once)

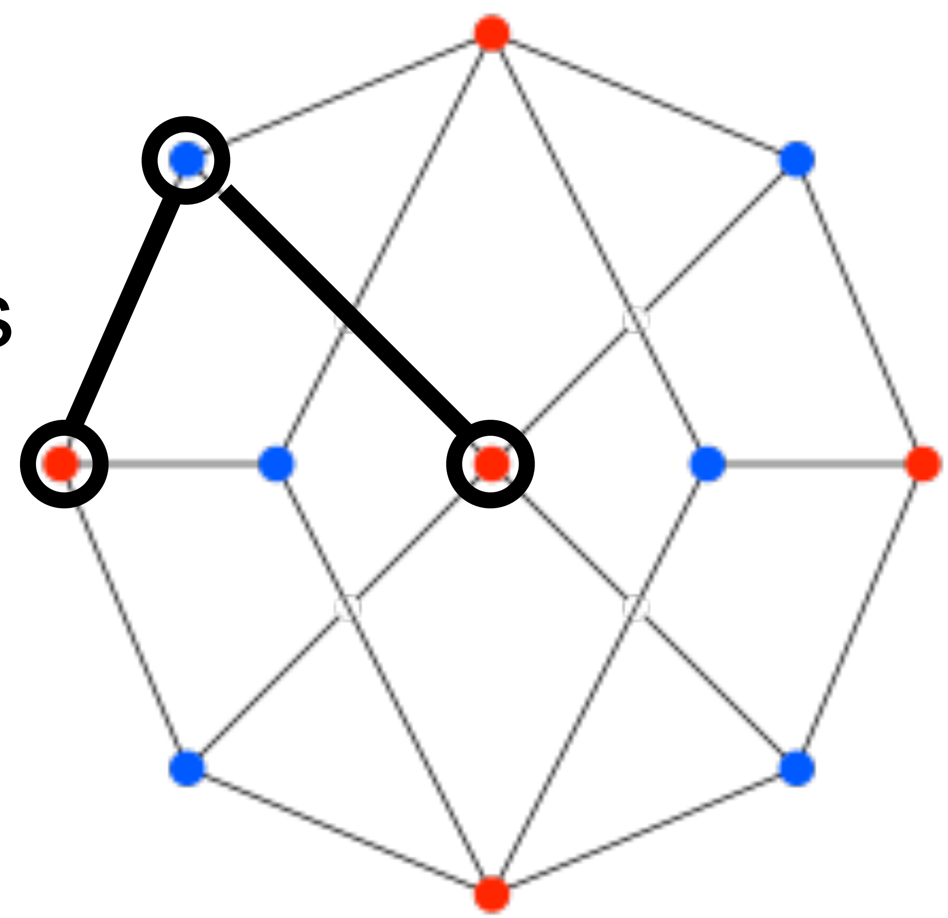
The problem looks difficult to solve



Hamiltonian circuit problem (NP-complete)

Given a graph G,
is it possible to draw an Hamiltonian circuit over it?
(i.e. a circuit that visits each vertex exactly once)

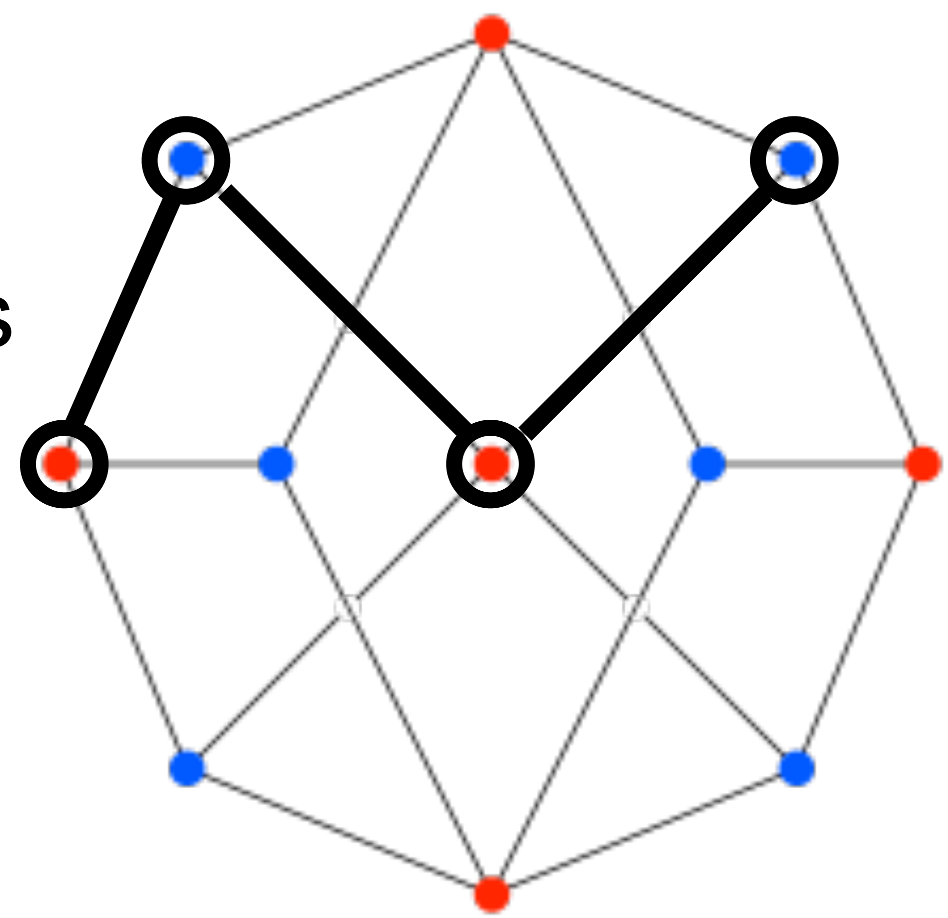
The problem looks difficult to solve



Hamiltonian circuit problem (NP-complete)

Given a graph G,
is it possible to draw an Hamiltonian circuit over it?
(i.e. a circuit that visits each vertex exactly once)

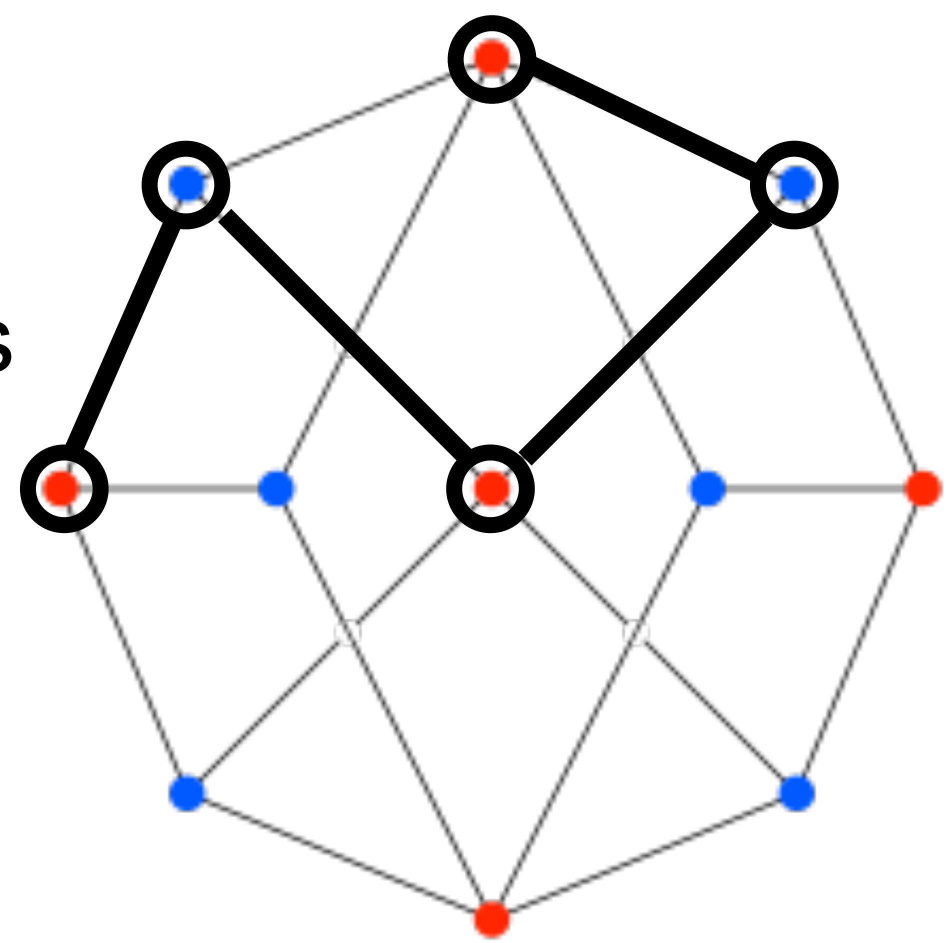
The problem looks difficult to solve



Hamiltonian circuit problem (NP-complete)

Given a graph G,
is it possible to draw an Hamiltonian circuit over it?
(i.e. a circuit that visits each vertex exactly once)

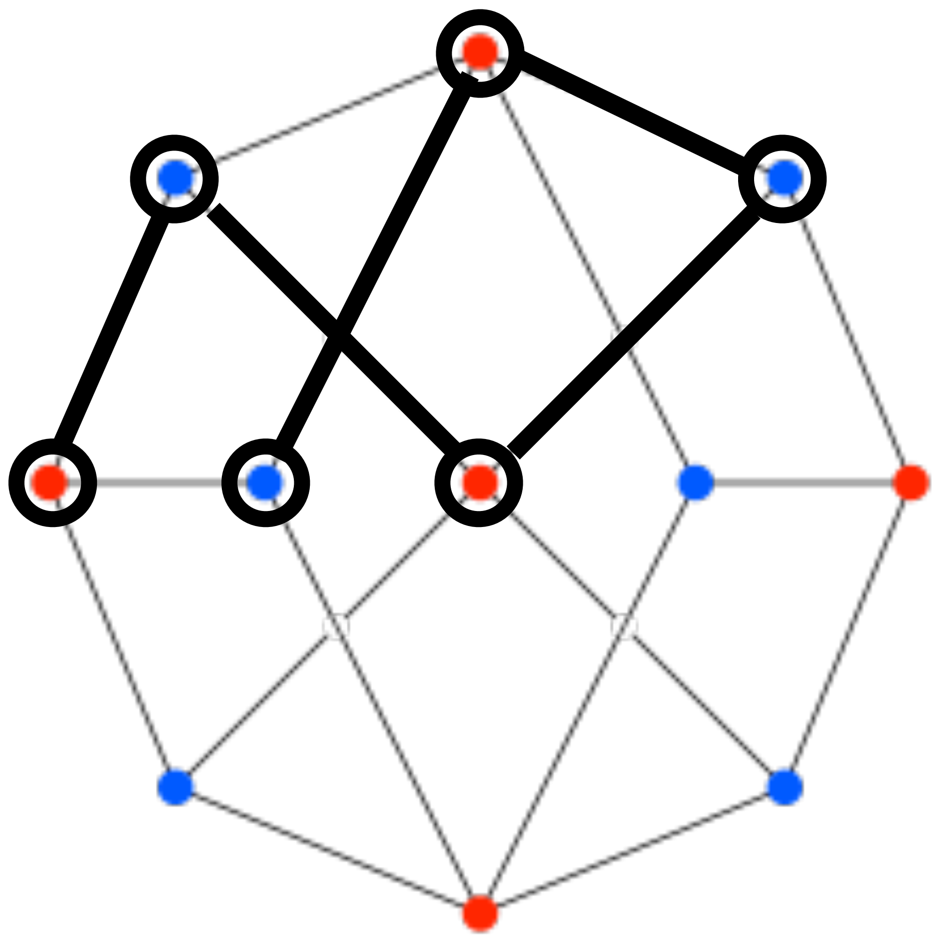
The problem looks difficult to solve



Hamiltonian circuit problem (NP-complete)

Given a graph G,
is it possible to draw an Hamiltonian circuit over it?
(i.e. a circuit that visits each vertex exactly once)

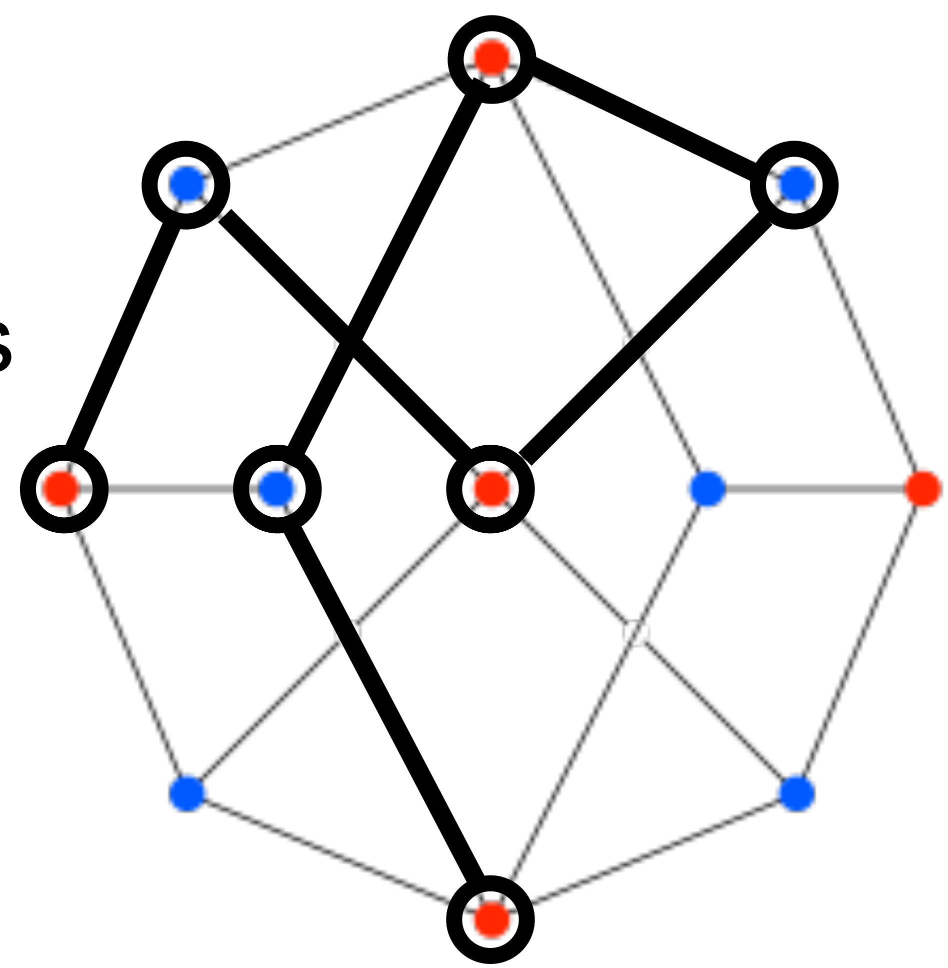
The problem looks difficult to solve



Hamiltonian circuit problem (NP-complete)

Given a graph G,
is it possible to draw an Hamiltonian circuit over it?
(i.e. a circuit that visits each vertex exactly once)

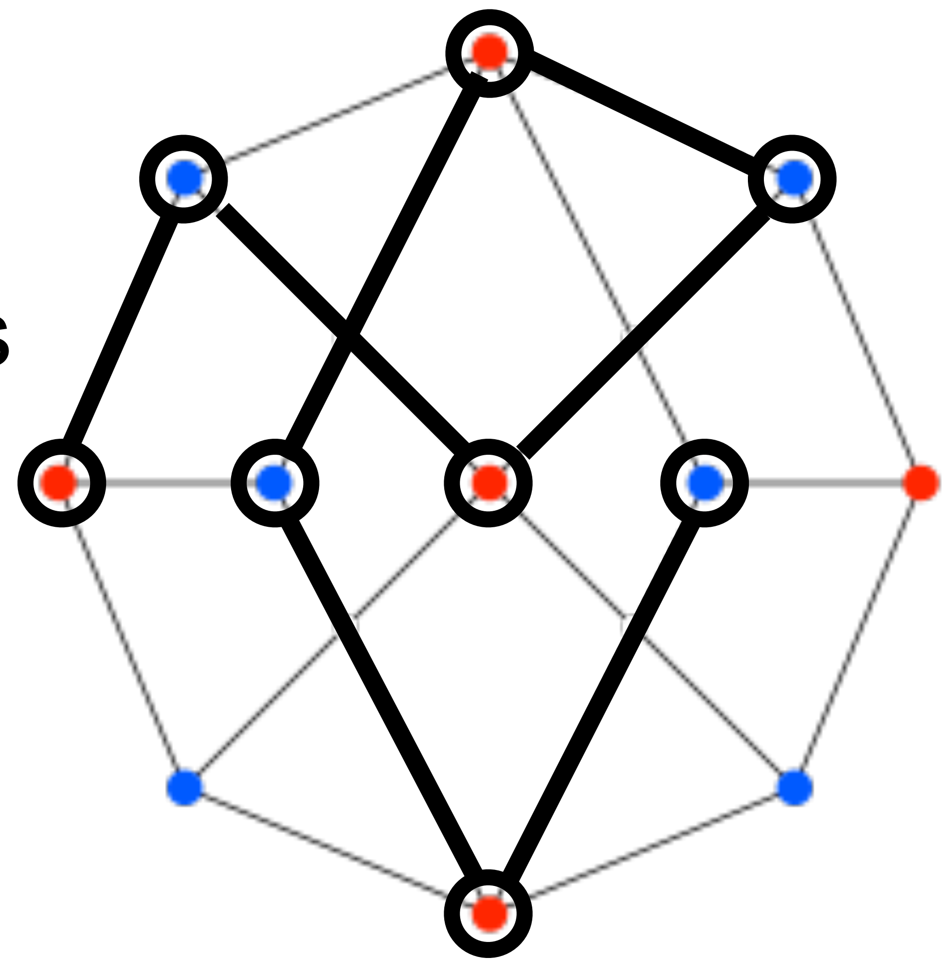
The problem looks difficult to solve



Hamiltonian circuit problem (NP-complete)

Given a graph G,
is it possible to draw an Hamiltonian circuit over it?
(i.e. a circuit that visits each vertex exactly once)

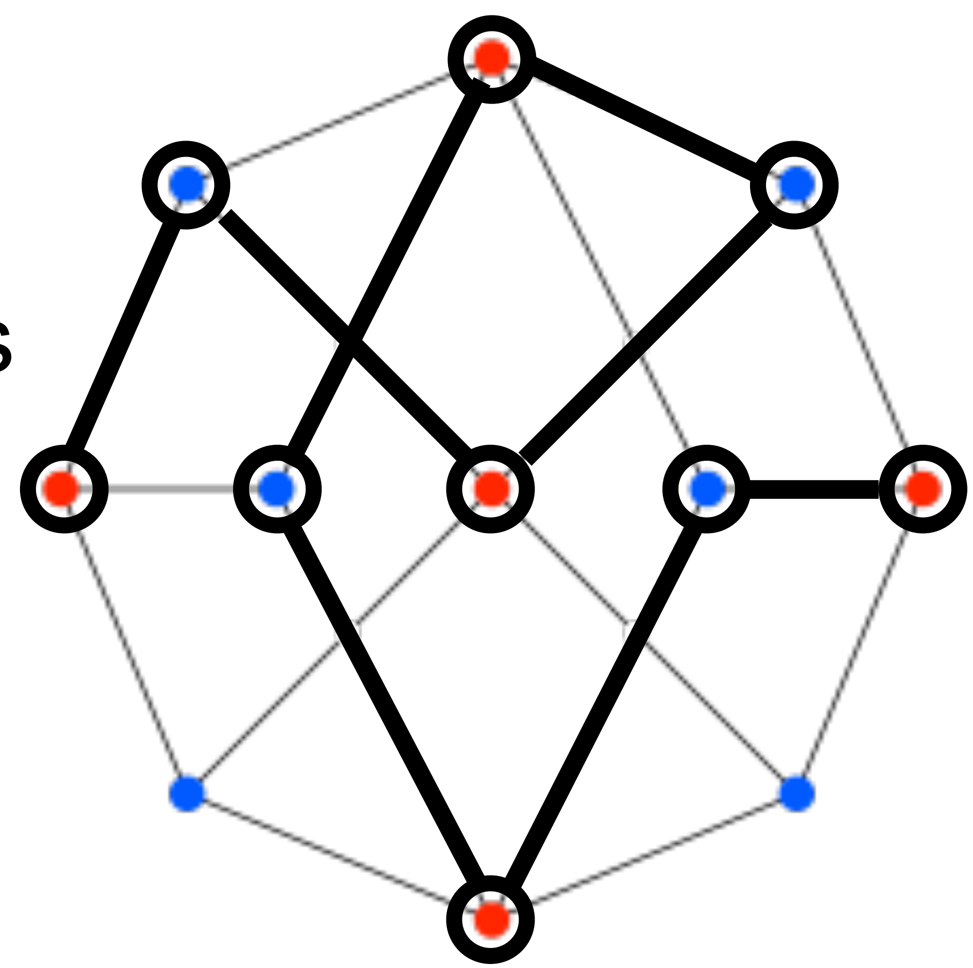
The problem looks difficult to solve



Hamiltonian circuit problem (NP-complete)

Given a graph G,
is it possible to draw an Hamiltonian circuit over it?
(i.e. a circuit that visits each vertex exactly once)

The problem looks difficult to solve



Hamiltonian circuit problem (NP-complete)

Given a graph G,
is it possible to draw an Hamiltonian circuit over it?
(i.e. a circuit that visits each vertex exactly once)

The problem looks difficult to solve