How do you show a problem is NP-hard/-complete?

Let B be the new problem you are interested in.

To show B is NP-complete

Reduce from a known NP-hard problem A to B.

Show that B is in NP.

To show B is NP-hard

Reduce from a known NP-hard problem A to B.

Hamilton

On a directed graph G:

A Hamiltonian Path is a path that visits every vertex exactly once.

A Hamiltonian Cycle is a Hamiltonian Path with an extra edge connecting the first vertex to the last vertex.

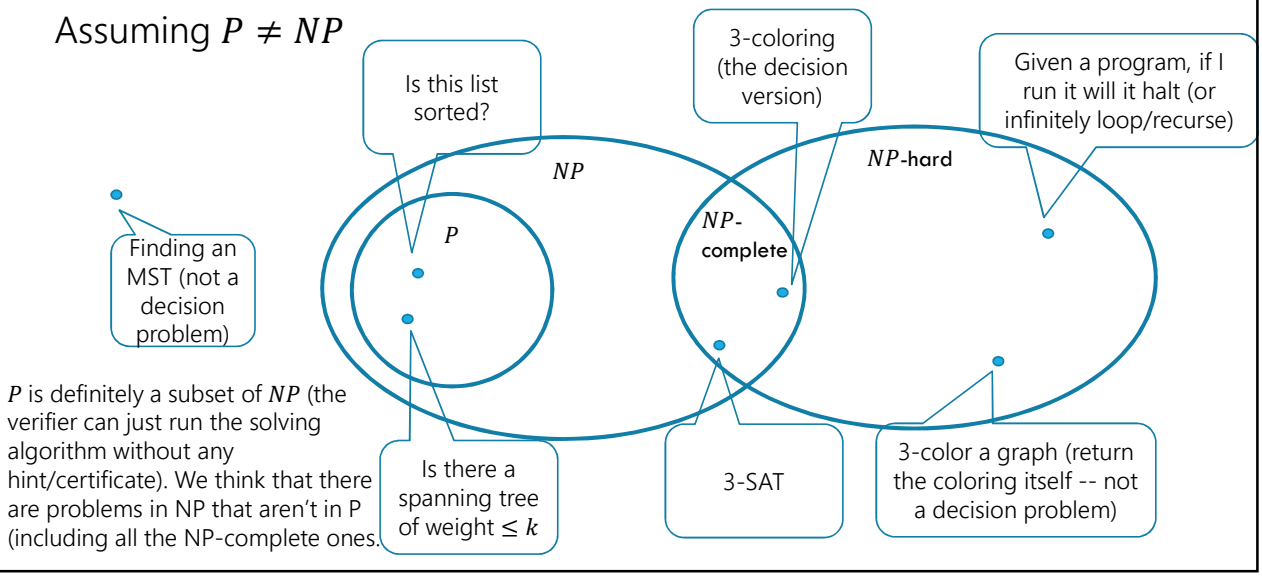
Assume that Hamiltonian Path is NP-hard (it is)

Use that to prove Hamiltonian Cycle is NP-hard.

Pollev.com/robbie

What (we think) the world looks like

Assuming $P \neq NP$



P is definitely a subset of NP (the verifier can just run the solving algorithm without any hint/certificate). We think that there are problems in NP that aren't in P (including all the NP-complete ones.

We can do better

When k is big, not much we can do. What about when it's small?

Our running time depends on k anyway, let's focus in on making our algorithm better when k is small.

Key idea: pick an edge (u, v)

There is a vertex cover of size k if and only if

There is a vertex cover of size $k - 1$ in $G - u$ or $G - v$.

i.e. at least one of u, v in the minimum vertex cover.