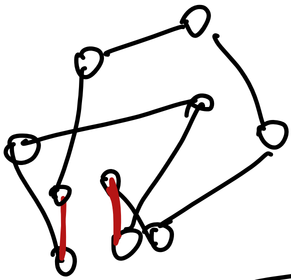
NP-hardness

Why bother?

If you prove a problem is NP-hard
- try a different problem/approach
- try to specialize
- approximate
- build heuristics that work in practice for your inputs



Training in transforming problems.

3SUM: Given a set X of integers are there elements $a, b, c \in X$ s.t. $a+b+c=0$?

$O(n^2)$ time

2016: $O(n^2/\log n)$ time

$O(n^{1.999})$ time?

What to reduce from?

Definition: X is NP-hard means
If we can solve X in poly time then $P=NP$

Cook-Levin:

3SAT is NP-hard

$\Downarrow$

If there is a poly-time reduction from 3SAT to X then X is NP-hard.

We can reduce from any NP-hard problem
to prove a new problem is hard

"It has a 3 in it"

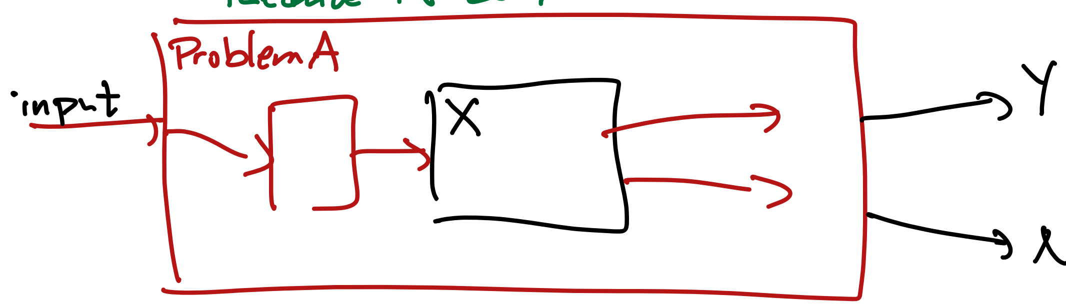
- 3SAT
- Circuit SAT } binary choices
- Max Clique
- Max Ind Set } large subset
- Min Vertex Cover } small subset
- 3 Color, 4 Color, etc.
- Min Colors } split into several subsets
- Hamiltonian Cycle/Path
- Longest Cycle/Path
- TSP } order
- Partition
- Subsum Sum
- 3 Partition } balance

3 Partition: Partition into sets of 3 elements each all with same sum.

known hard

trying to prove hard

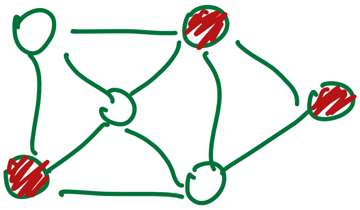
Reduce A to X



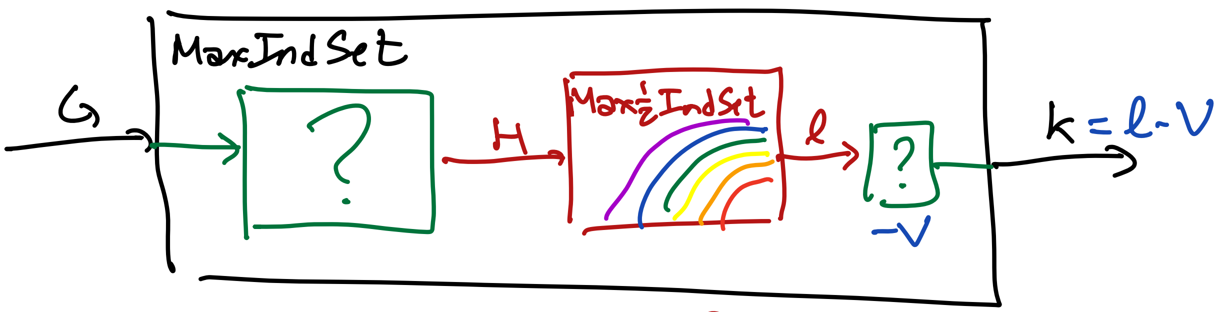
Choose tightly constrained problem to reduce from
consider arbitrary inputs

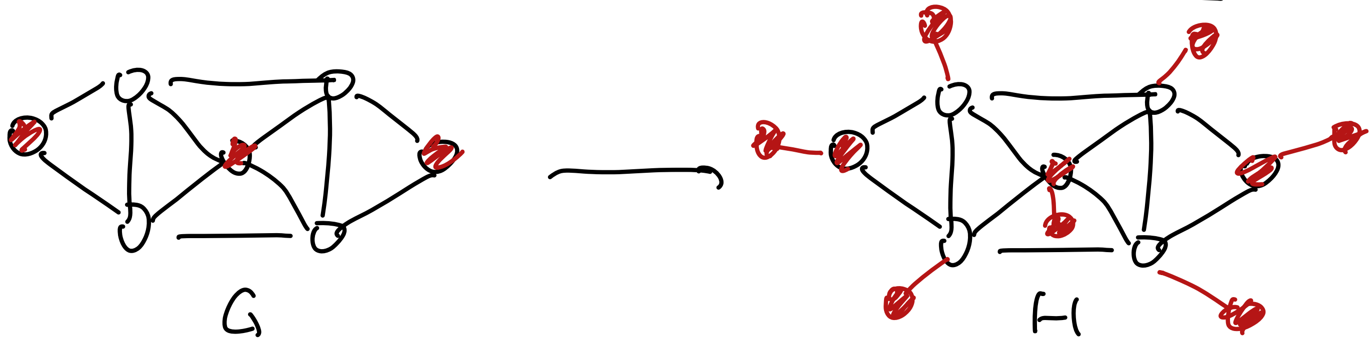
Build very special inputs to problem X we're trying to prove hard.

A subset S of vertices in an undirected graph G is *half-independent* if each vertex in S is adjacent to *at most one* other vertex in S. Prove that finding the size of the largest half-independent set of vertices in a given undirected graph is NP-hard.



Reduce From MaxIndSet





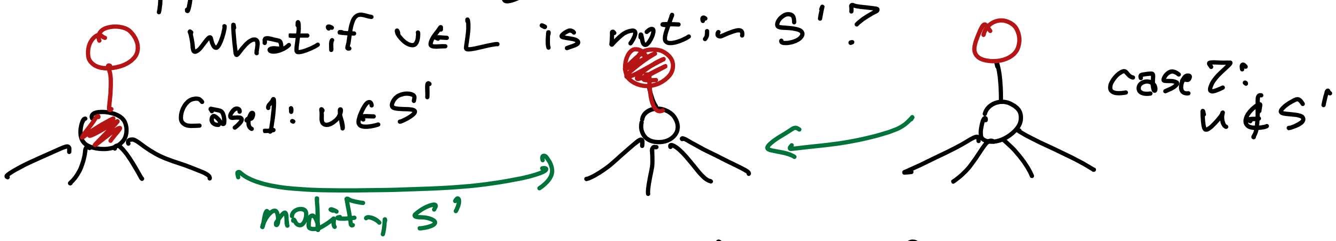
$\Rightarrow$ Suppose G has ind set S of size k

Then H has $\frac{1}{2}$ind set $S \cup L$ of size $k+V$

So $\boxed{\text{Max}\tfrac{1}{2}\text{IndSet}(H) \geq \text{MaxIndSet}(G) + V}$

$\Leftarrow$ Suppose H has $\frac{1}{2}$ind set S' of size $\ell \geq V$

What if $v \in L$ is not in S'?



So G has ind set of size $\ell - V$

$\boxed{\text{MaxIndSet}(G) \geq \text{Max}\tfrac{1}{2}\text{IndSet}(H) - V}$

A subset S of vertices in an undirected graph G is *sort-of-independent* if if each vertex in S is adjacent to *at most 374* other vertices in S. Prove that finding the size of the largest sort-of-independent set of vertices in a given undirected graph is NP-hard.

A subset S of vertices in an undirected graph G is *almost independent* if at most 374 edges in G have both endpoints in S. Prove that finding the size of the largest almost-independent set of vertices in a given undirected graph is NP-hard.

Charon needs to ferry n recently deceased people across the river Acheron into Hades. Certain pairs of these people are sworn enemies, who cannot be together on either side of the river unless Charon is also present. (If two enemies are left alone, one will steal the obol from the other's mouth, leaving them to wander the banks of the Acheron as a ghost for all eternity. Let's just say this is a Very Bad Thing.) The ferry can hold at most k passengers at a time, including Charon, and only Charon can pilot the ferry.[30]

Prove that it is NP-hard to decide whether Charon can ferry all n people across the Acheron unharmed (aside from being, you know, dead). The input for Charon's problem consists of the integers k and n and an n-vertex graph G describing the pairs of enemies. The output is either TRUE or FALSE.