CPSC 420 Lecture 19: Today's announcements:

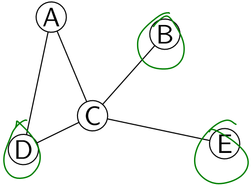
- HW3 is on Gradescope, due Mar 9, 23:59
- Examlet 3 on Mar 17 in class. Closed book & no notes
- Reading:
  NP-hardness [Erickson]
  NP-completeness proofs [Cormen, Leiserson, Rivest, Stein]

Today's Plan

- NP-hardness
  - CircuitSat ✓
  - 3SAT (and SAT) ✓
  - Independent Set
  - Vertex Cover
  - Clique
  - Hamiltonian cycle (and TSP)

Independent Set

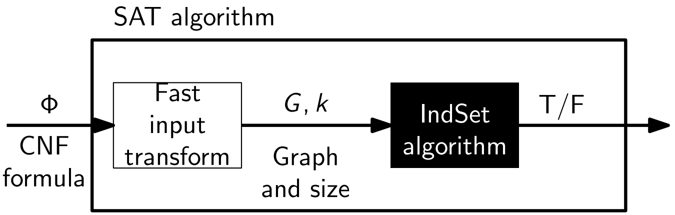
An **independent set** is a set of vertices in a graph G that share no common edge.



IndependentSet takes graph G and integer k and outputs "Yes" if G has an independent set of size k and "No" otherwise.

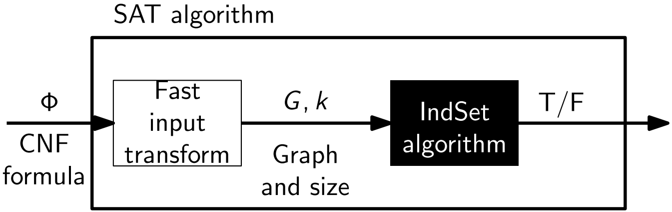
Claim: IndependentSet is NP-hard.

Reduce from SAT to IndependentSet



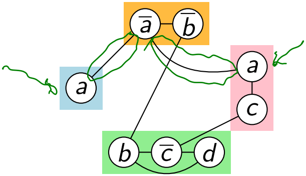
Transform a CNF formula Φ into a graph G and integer k so that Φ is satisfied if and only if G has an independent set of size k.

Reduce from SAT to IndependentSet



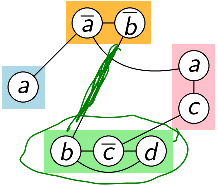
Transform a CNF formula Φ into a graph G and integer k so that Φ is satisfied if and only if G has an independent set of size k.

$$\Phi = (a) \wedge (\overline{a} \vee \overline{b}) \wedge (a \vee c) \wedge (b \vee \overline{c} \vee d)$$



Reduce from SAT to IndependentSet

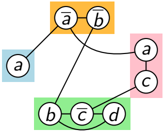
$$\Phi = (a) \wedge (\overline{a} \vee \overline{b}) \wedge (a \vee c) \wedge (b \vee \overline{c} \vee d)$$

$G =$  $k = 4$

1. Create a vertex for every occurrence of a literal in a clause.
2. Create edges between every literal occurrence and its negation.
3. For each clause, create edges between all literals in the clause.
4. Let the size of the desired independent set $k = \#$ clauses

Reduce from SAT to IndependentSet

$$\Phi = (a) \wedge (\overline{a} \vee \overline{b}) \wedge (a \vee c) \wedge (b \vee \overline{c} \vee d)$$

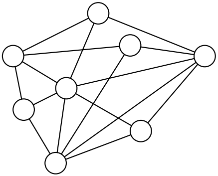
$G =$  $k = 4$

Claim: G contains an independent set of size k if and only if Φ is satisfiable.

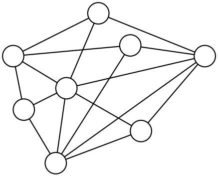
$\Rightarrow$ Let S be an independent set of size k in G. S cannot contain two literal nodes from the same clause, so every one of the k clauses contains one literal in S. S cannot contain a literal node and its negation. Set all literals in S to true. This satisfies Φ.

$\Leftarrow$ Let A be a truth assignment satisfying Φ. Every clause contains at least one True literal. Pick one for each of the k clauses and let S be the set of corresponding vertices. Since A doesn't assign True to a literal and its negation, S is an independent set of size k.

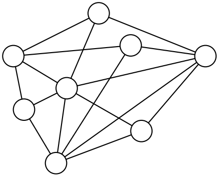
Clique and Vertex Cover are NP-complete



Independet Set



Clique



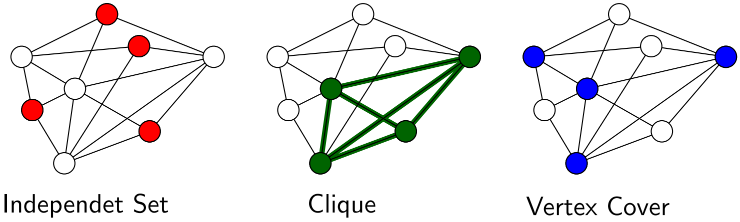
Vertex Cover

Independent Set: A set of vertices that share no common edge.

Clique: A set of vertices that form a complete subgraph of G.

Vertex Cover: A set of vertices that "cover" (contain at least one endpoint of) every edge of G.

Clique and Vertex Cover are NP-complete

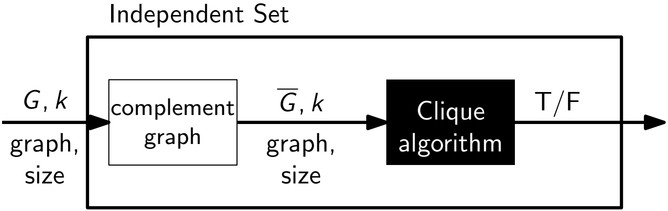


Independent Set: A set of vertices that share no common edge.

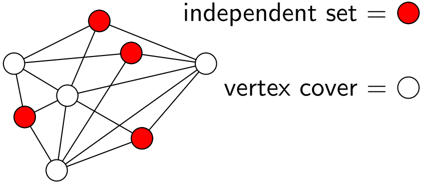
Clique: A set of vertices that form a complete subgraph of G.

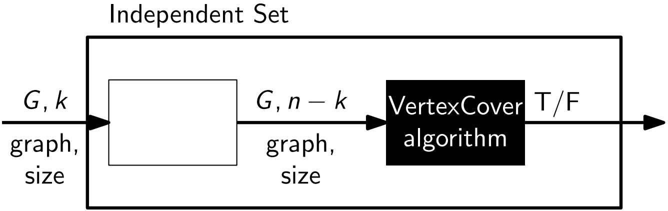
Vertex Cover: A set of vertices that "cover" (contain at least one endpoint of) every edge of G.

Clique is NP-complete



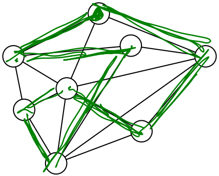
Vertex Cover is NP-complete





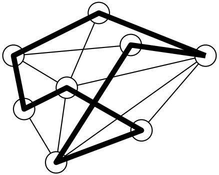
Hamiltonian Cycle is NP-complete

Hamiltonian Cycle: A cycle that contains every vertex exactly once (and returns to the start).



Hamiltonian Cycle is NP-complete

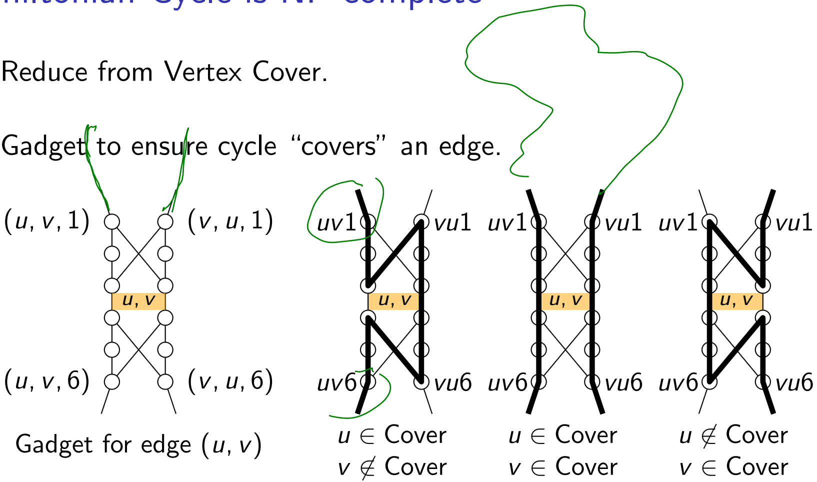
Hamiltonian Cycle: A cycle that contains every vertex exactly once (and returns to the start).



Hamiltonian Cycle is NP-complete

Reduce from Vertex Cover.

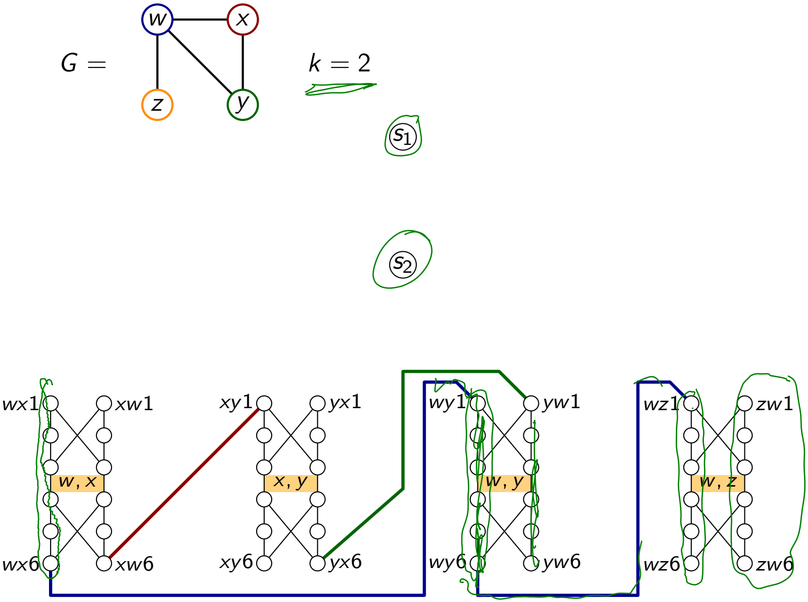
Gadget to ensure cycle "covers" an edge.



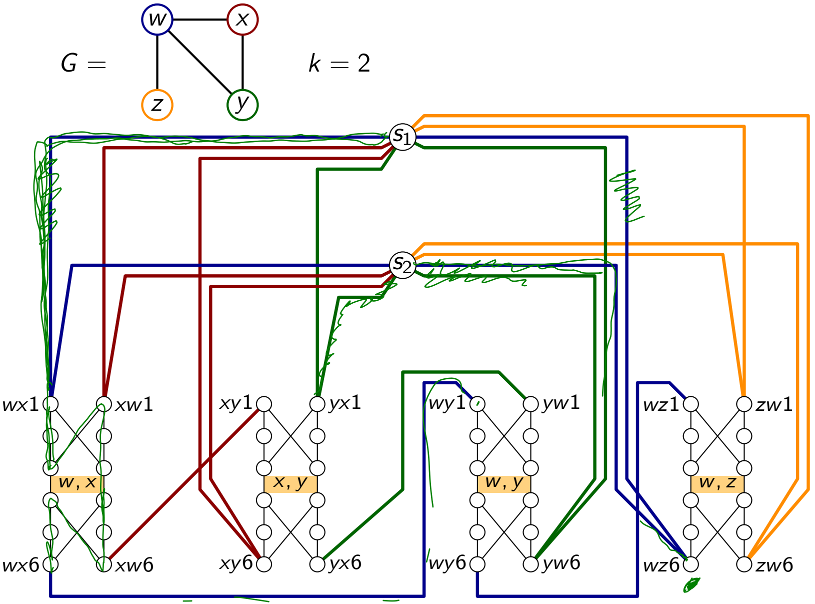
Gadget for edge (u, v)

Any Hamiltonian cycle must traverse the gadget in one of these three ways.

Hamiltonian Cycle is NP-complete



Hamiltonian Cycle is NP-complete



Hamiltonian Cycle is NP-complete