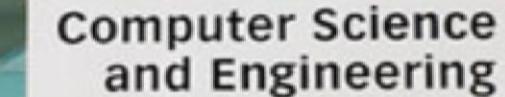


EECS 3101

Prof. Andy Mirzaian

A photograph of a red and white directional sign for York University's Computer Science and Engineering department. The sign is mounted on a metal pole and features the York University logo (a red 'U' with 'YORK UNIVERSITY' written above it) at the top. Below the logo, the text 'Computer Science and Engineering' is written in black, followed by '120 Campus Walk' at the bottom.

Computer Science
and Engineering

120 Campus Walk

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

NP-completeness

STUDY MATERIAL:

- [CLRS] chapter 34

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

TOPICS

- Preliminaries
- Complexity Classes P, NP, co-NP, NP-complete
- Polynomial Assignment Project & Exam Completeness
- NP-Completeness
<https://eduassistpro.github.io/>
Add WeChat edu_assist_pro

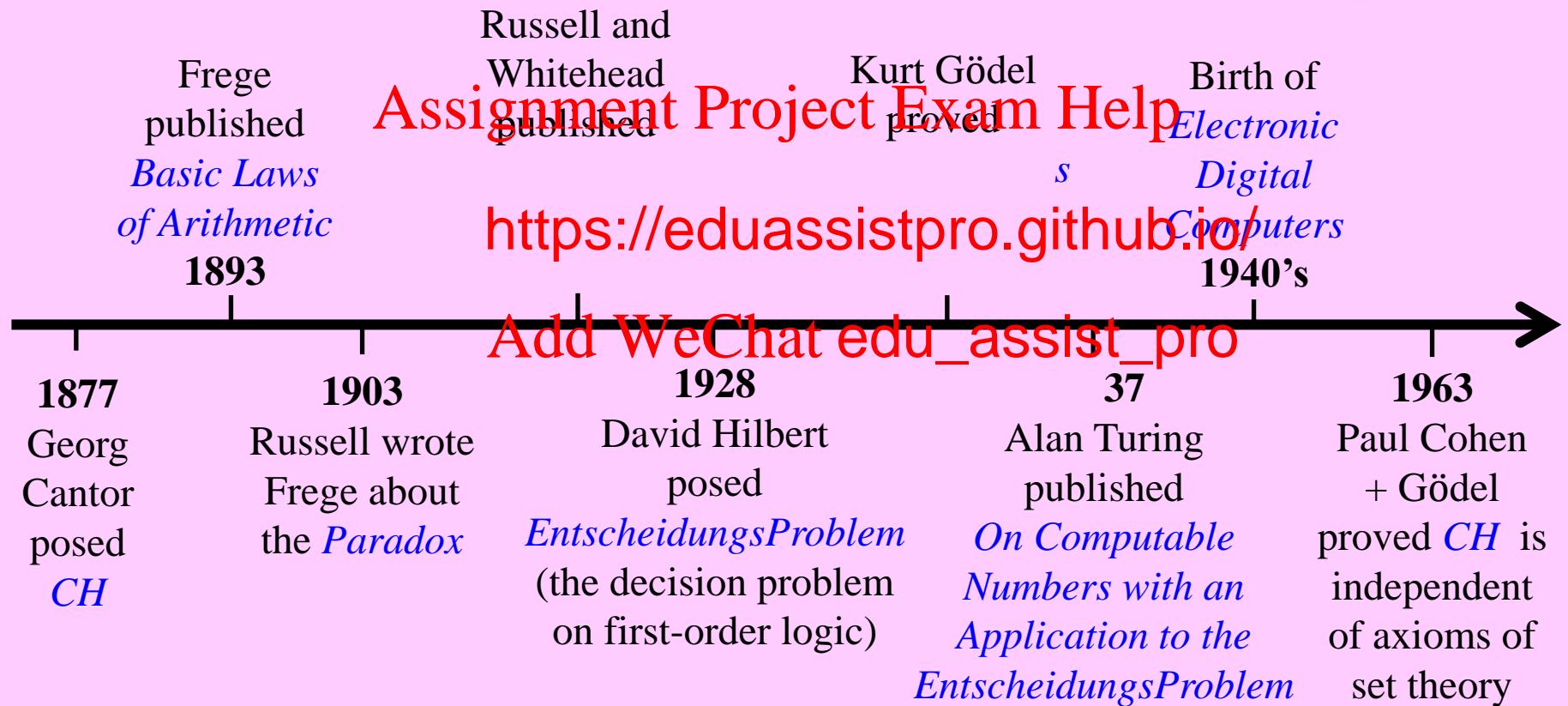
PAssignment Project Exam Help
es
<https://eduassistpro.github.io/>
Add WeChat **edu_assist_pro**

Foundations & Limitations

True
or
False ?

Cantor's Continuum Hypothesis (CH):

There is no set S such that $|\mathbb{N}| < |S| < |\mathbb{R}|$.



Un-Computable Problems

No algorithm exists for certain computational problems, such as:

- **Diophantine Equations** [Diophantus of Alexandria, 3rd century AD]
Does a multi-variable polynomial with integer coefficients, e.g.,

$$x^3y^2z^2 + 7xy^4z^3 - 3x^2y^5z = 8,$$

Assignment Project Exam Help

David Hilbert [1900]: $\vdash \phi \rightarrow \psi$ if and only if $\phi \vdash \psi$.
is problem solvable.

Matiyasevich [1970] p <https://eduassistpro.github.io/> all diophantine equations.

- The Halting Problem [Alan M. Turing, 1936] Add WeChat edu_assist_pro
 - he was then a student at Cambridge University, Cambridge, England.

There were no digital computers or programming languages!

Arguably, these things came about *exactly because* of Turing's brilliant thoughts.

Other contributors: Emil Post [1920], Kurt Gödel [1930], Alonzo Church [1935].
 - There are infinitely many uncomputable problems, e.g.,
Buffer Overflow, Post's Correspondence, ...

Computational Complexity Classes

- So, there are computationally unsolvable problems.
- Computable problems themselves have vastly different computational complexity.
- **Complexity Classes** classify problems by their computational complexity.

Uncomputable

Computable
Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro
 $P = \text{polyn}$

Linear Programming

Given linearly constrained (in-)equalities & a linear objective function on many variables with integer coefficients, find **real** values for the variables that satisfy the constraints and optimize the objective function.

$$\begin{array}{ll}\text{minimize} & 3x + 5y - 2z \\ \text{subject to:} & 4x - y + 7z \leq 8 \\ & 2x + 4y = 3\end{array}$$

<https://eduassistpro.github.io/>

Add WeChat [edu_assist_pro](#)

- Computational Complexity of LP = polyno [hachyian 1979]
- LP is a versatile computational model for problem formulation.
 - Virtually all problems we have studied so far can be modeled as LP problems (with the exception of 0/1 Knapsack) and have polynomial time complexity, e.g., Max Flow, Min Cut, Shortest Paths, ...

Integer Linear Programming

Given linearly constrained (in-)equalities & a linear objective function on many variables with integer coefficients, find **integer** values for the variables that satisfy the constraints and optimize the objective function.

$$\begin{array}{ll} \text{minimize} & 3x + 5y - 2z \\ \text{subject to:} & + \leq \end{array}$$

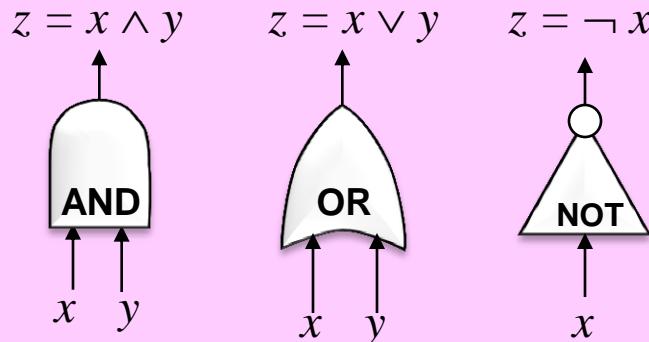
Assignment Project Exam Help

<https://eduassistpro.github.io/>
x, y, z are integers

Add WeChat edu_assist_pro

- Computational Complexity of ILP: polynomial ???
- ILP is at least as powerful and versatile as LP.
- **Search problems** can be cast as ILP, e.g.,
 - 0/1 Knapsack, Graph Coloring, Matching, Max Cut,
 - Traveling Salesman Problem, Boolean Formula Satisfiability,

Circuit SAT



Combinatorial Circ

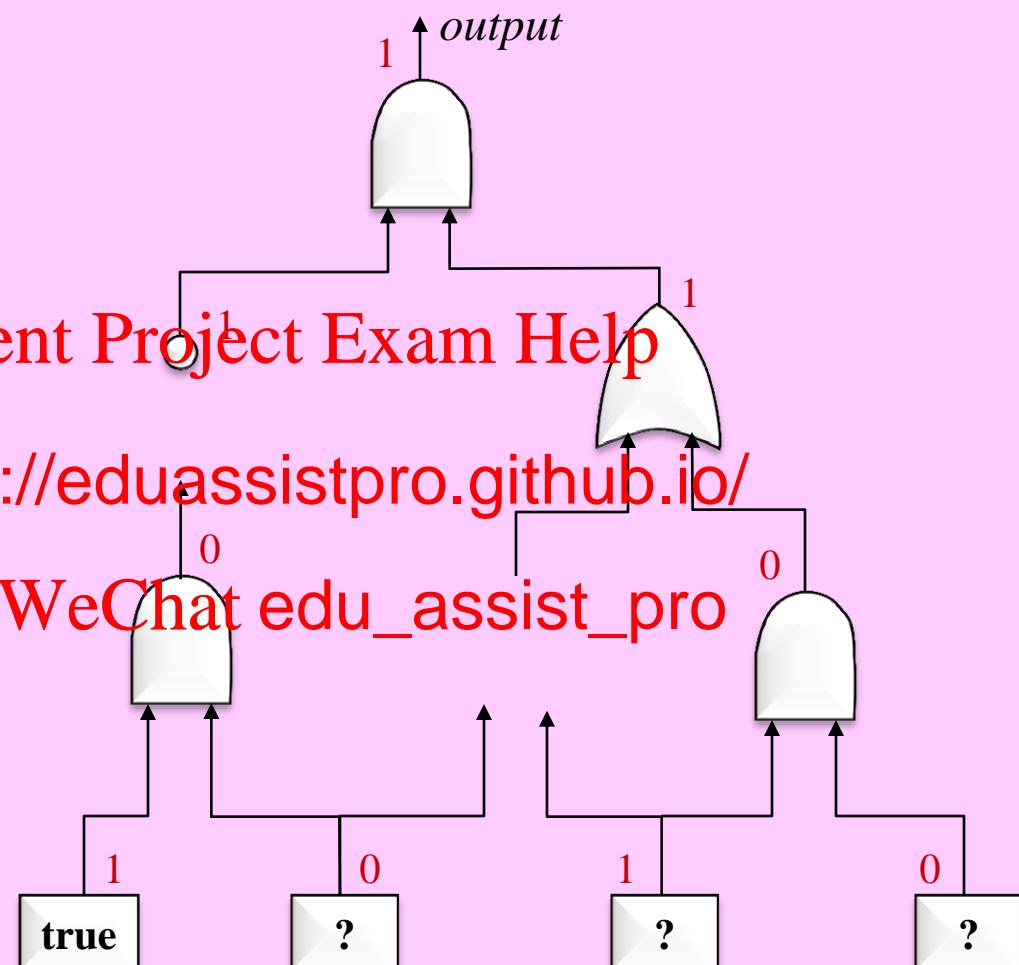
(a DAG of logical ga

<https://eduassistpro.github.io/>

is satisfiable

if and only if

there is a truth assignment
to its variable input gates
that makes the output true.



For a given truth assignment,
evaluate gate outputs in topological order.

Boolean Formula SAT

$$\varphi = (x \vee y \vee z) \wedge (x \vee \bar{y}) \wedge (y \vee \bar{z}) \wedge (z \vee \bar{x}) \wedge (\bar{x} \vee \bar{y} \vee \bar{z}) \wedge (w \vee \bar{x} \vee z)$$

is an instance of **CNF-SAT** or **SAT** for short:

a Boolean formula in conjunctive normal form (CNF),

i.e., a conjunction (\wedge) of a number of **clauses** (the parentheses);

each clause is disjunction (\vee) of some **literal**s

each literal is a Boolean variable.

- **k SAT:** instances are in CNF with k literals.
- **3SAT:** instances are in 3-CNF.
- A Boolean formula is **satisfiable**,
if there is a truth assignment to its variables that makes the formula true.
(Is φ satisfiable?)
- A Boolean formula is a **tautology**,
if every truth assignment of its variables makes the formula true.
(Negation of a tautology is unsatisfiable.)

<https://eduassistpro.github.io/>

Add WeChat `edu_assist_pro`

2SAT $\in \text{P}$

Input: a boolean 2-CNF formula Φ .

Question: Is Φ satisfiable?

Step 1:

Idea: $(x \vee y) \equiv (\bar{x} \Rightarrow y) \equiv (\bar{y} \Rightarrow x)$

Construct the directed graph $G = (V, E)$:

$V = \{x, \bar{x} \mid x \text{ is a variable in } \Phi\}$.

$E = \{ \bar{x} \rightarrow y, \bar{y} \rightarrow x \mid (x \vee y) \text{ is a clause in } \Phi \}$

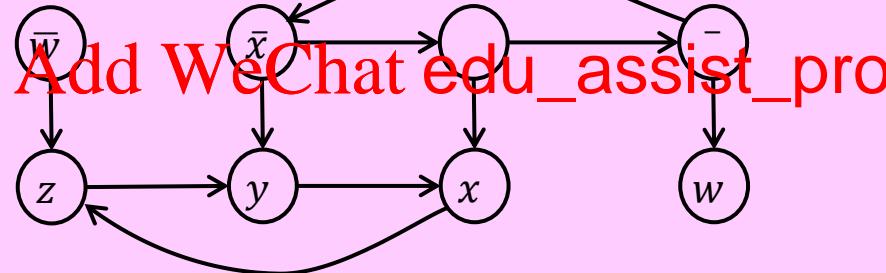
Assignment Project Exam Help

Example: $\Phi = (x$

$\neg z \vee \neg w) \wedge (\neg x \vee z) \wedge (\neg y \vee w) \wedge (\neg z \vee w) \wedge (w \vee z)$

<https://eduassistpro.github.io/>

$G:$



Add WeChat edu_assist_pro

Question: Can we assign T or F labels to vertices of G such that

- no opposite literal pair (i.e., x and \bar{x}) is labeled the same, and
- no edge has its incident vertices labeled as: $T \rightarrow F$?

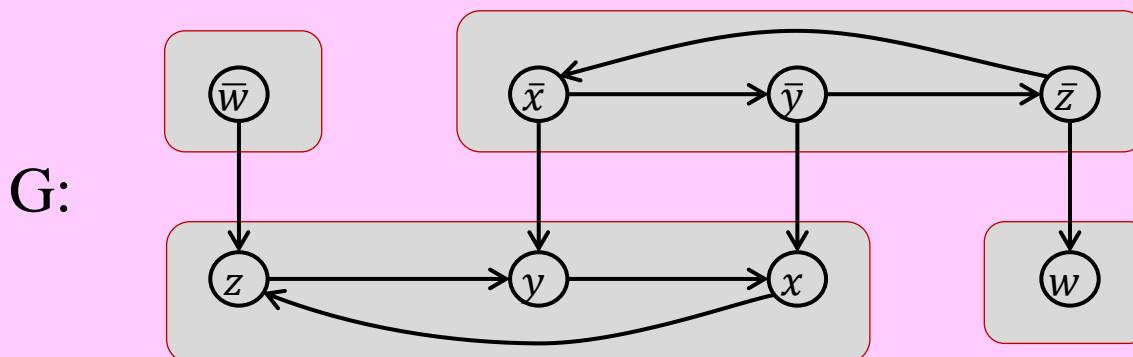
2SAT $\in \mathbf{P}$

CLAIM: Φ is satisfiable \Leftrightarrow No SCC of G contains a variable and its negation.

- **Proof sketch:** See Exercise 10.

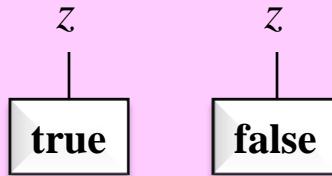
Proof of $[\Rightarrow]$: In a ~~Assignment Project Exam Help~~ node has in-degree 0, hence in the same SCC, \Rightarrow all nodes in the same SCC have the same truth value (all true or all false).

Proof of $[\Leftarrow]$: In the SC <https://eduassistpro.github.io/>, node has in-degree 0, its “negated” SCC node must have out-deg 0. What truth values would you assign to ~~Add WeChat~~ edu_assist_pro nodes? Adapt the SCC algorithm to get a full truth assignment in linear time.



Circuit SAT \longrightarrow SAT

Give the output of each gate a distinct name.
Convert each gate to an equivalent small CNF-SAT.
Then take their conjunction, including the circuit output.

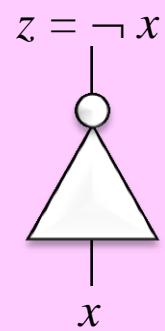


Assignment | Project | Exam | Help
? https://eduassistpro.github.io/
Add WeChat edu_assist_pro

(z) (\bar{z})

-
 $(-$
 $(- - -$

$(z \vee \bar{x})$
 $(z \vee \bar{y})$
 $(\bar{z} \vee x \vee y)$



$(z \vee x)$
 $(\bar{z} \vee \bar{x})$

Solution Construction vs Verification

Circuit-SAT, SAT, ILP, LP ... are (Combinatorial) Search Problems.

Search Problem. Instance I:

Construction: given I, output a valid solution S for instance I.

Verification: given a solution S for instance I.

<https://eduassistpro.github.io/>

Obvious: Construction is at least as hard as Verification.

Question: Is Construction strictly harder than Verification?

Example: Consider establishing a Mathematical Fact.

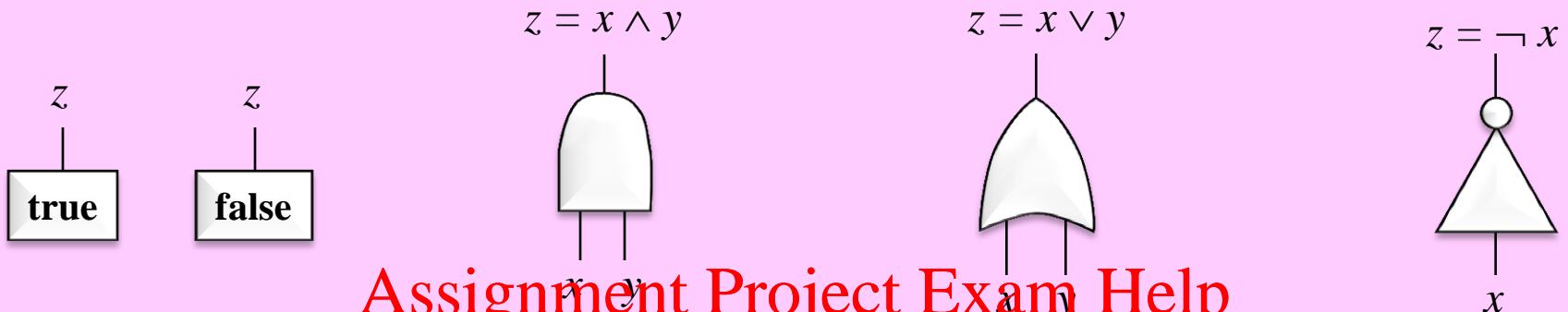
Construction: Provide a Proof.

Verification: Given a “proof”, verify its validity.

Circuit SAT Verification



LP



Assignment Project Exam Help

$$z = 1$$

$$z = 0$$

$$\frac{z \leq x}{z \leq y} \quad \frac{z \geq x}{z \geq y}$$

$$\frac{z \geq x + y - 1}{z \leq x + y}$$

$$z = 1 - x$$

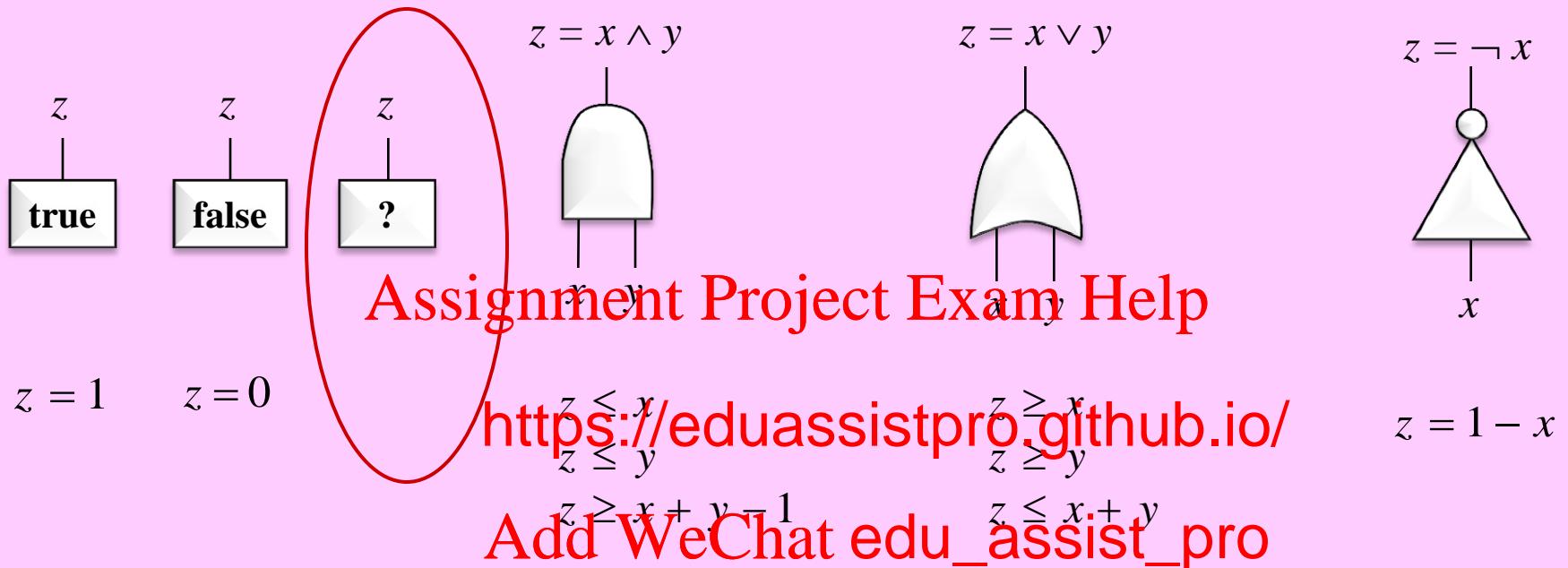
Plus, for all gate-output variables z , add the constraints:

$$0 \leq z \leq 1.$$

Objective: Maximize z_{OUT} (the circuit output).

Now we have an instance of LP

Circuit SAT Construction \longrightarrow ILP



Plus, for all gate-output variables z , add the constraints:

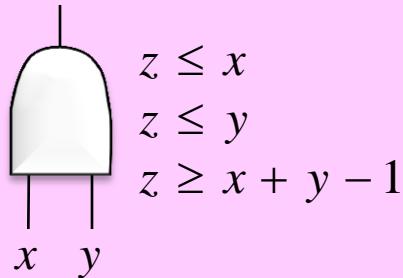
$0 \leq z \leq 1$, and z is an integer.

Objective: Maximize z_{OUT} (the circuit output).

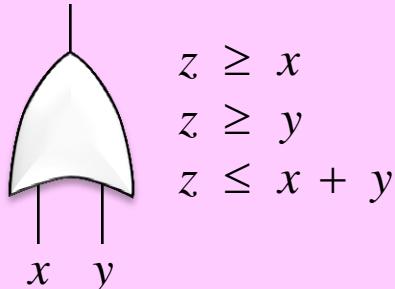
Now we have an instance of ILP.

Why ILP and not LP?

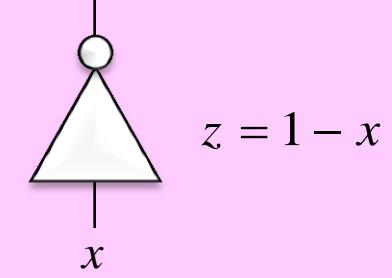
$$z = x \wedge y$$



$$z = x \vee y$$



$$z = \neg x$$



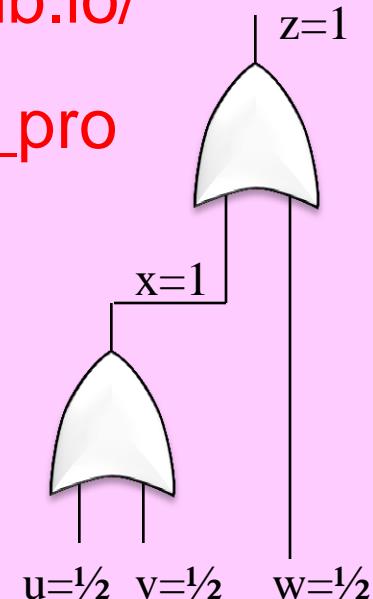
Plus the constraints $0 \leq z \leq 1$, for every variable z.

Integrality constrain <https://eduassistpro.github.io/>

Consider the circuit for an ~~And~~ satisfiable clause
 Add WeChat edu_assist_pro
 Ignore the integrality constraints and assign $t_{1/2} = 1 - t_{1/2}$ to each variable and its negation.

Part of the circuit for an arbitrary clause $(u \vee v \vee w)$:

Conjunction of all these “1” clause outputs will be a “1” !



Is the circuit satisfiable or not ?!

Construction HARD Verification EASY ???

EASY

Linear Programming

Search Problem Verification

Assignment Project Exam Help
3SAT Verification

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro
Integer Linear ing

HARD

Search Problem Construction

3SAT Construction

Complexity Classes:

P, NP, NP-Complete

Assignment Project Exam Help

Add WeChat edu_assist_pro

Optimization vs Decision Problems

Optimization Problem: output is an optimum solution to the input instance

Shortest Path:

Instance: $\langle G, s, t \rangle$

Output: Shortest path in graph G from vertex s to t .

Assignment Project Exam Help

Decision Problem: <https://eduassistpro.github.io/>
answer.

Add WeChat edu_assist_pro

Decision version of Shortest Path:

Instance: $\langle G, s, t, K \rangle$

Question: Does graph G have a path from vertex s to t with length at most K ?

For a maximization problem, “at most” is replaced by “at least”.

Why Decision Problems ?

1. Optimization version of a problem is at least as hard as its decision version.

Example: To answer whether G has a path of length at most K from s to t, we can find the shortest path from s to t and compare its length with K.

2. So, if we can establish the fact that the decision version is hard, that would imply that the optimization version must also be hard.

Assignment Project Exam Help

3. With uniform “yes/no” (or reduce) one decision

<https://eduassistpro.github.io/> intent to transform
all see.

4. The purpose of these reductions is to establish (as was done in the Sorting/Selection Slides), rather than obtaining an algorithmic upper-bound (as was done by reducing bipartite matching to max-flow).

5. By the way, the converse of (2) is usually true as well:

If the decision version is easy, so is the optimization version.

Example: with integer numeric input, do binary search on K in the decision version to find the shortest path.

The Complexity Class P

P = Deterministic **Polynomial** time:

the class of problems that admit a deterministic algorithm whose running time is $O(n^d)$ for some constant d, where n is the input size.

[Alan Cobham 1964] [Jack Edmonds 1965]

A problem belongs to P,

otherwise it is <https://eduassistpro.github.io/>

Justification (polynomial vs exponential) Add WeChat edu_assist_pro

- P is preserved under many important computational models, e.g., RAM, Turing Machine, etc.
- Polynomials are the smallest class that contain linear functions & are closed under: composition, addition, multiplication.
- P scales nicely (multiplicative rather than additive)
- Dilemma: $O(n^{1000})$ vs $O(1.1^n)$! Explain.

The Complexity Class NP

NP = Non-deterministic Polynomial time:

the class of decision problems that admit a non-deterministic polynomial time algorithm.

Decision problem A Assignment Project Exam Help

Decision language of A: <https://eduassistpro.github.io/> instance of problem A }

$A \in NP$

Add WeChat edu_assist_pro

- $\Leftrightarrow L_A = \{ \langle I \rangle \mid ALG(I) = \text{"yes"} \}$
for some non-deterministic polynomial time algorithm ALG
(ALG gives no termination guarantee on "no" instances!)
- $\Leftrightarrow L_A = \{ \langle I \rangle \mid \exists \text{ solution certificate } S, |S| \in \text{poly}(|I|), Verify(I, S) = \text{"yes"} \}$
for some deterministic algorithm $Verify$ with running time $\in \text{poly}(|I|)$.
(Note: the "guessed" solution certificate S cannot be too large.
It must satisfy: $|S| \in \text{poly}(|I|)$. See Exercise 9 at the end of these slides.)

P, NP, EXP

“P vs NP” ≡ “polynomial time Construction vs Verification”

EXP

Assignment Project Exam Help
NP

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

P

The \$1,000,000 Question

- $P \subseteq NP$
- [Steven Cook, 1971]: Is $P = NP$ or $P \neq NP$?
This has turned out to be one of the most challenging open problems in all of mathematics and computer science!
- At the turn of the 20th century, The Clay Mathematics Institute posted 7 “Millenium Problems” for the solution of any one of them. <https://eduassistpro.github.io/>
- $P \neq NP \Leftrightarrow$ there exist problems in $NP -$ **Add WeChat edu_assist_pro**
The quest in search of such problems led
“what are the hardest problems in NP?”
This gave rise to the class of **NP-complete** problems; the hardest in NP .
- [Steven Cook, 1971] discovered the first NP-complete problem: SAT
- [Richard Karp, 1972] published a long list of other NP-complete problems.
These were seemingly intractable (?) highly researched problems, with important applications in science technology and engineering.
- By now there are thousands of published NP-complete problems.

Search/OPT Problems: HARD vs EASY

Hard Problems (NP-complete) Easy Problems (in P)

Integer Linear Programming

3SAT

Assignment Project Exam Help

Minimum Spanning Tree

Longest

3D Ma

0-1 Knapsack

Independent Set

Hamiltonian Cycle

Max Cut

Linear Programming

2SAT

Help or

um Spanning Tree

Shortest Path

atching

ital Knapsack

ent Set on trees

Eulerian Cycle

Min Cut

Some (unresolved) exceptions:

- Graph Isomorphism \leftarrow quasi-poly time [L. Babai, STOC2016]
- Integer Factoring \leftarrow poly-time quantum alg. [P. Shor, FOCS1994]

Complements of P and NP

co-P = {A | (complement of A) = $\bar{A} \in P$ }

co-NP = {A | (complement of A) = $\bar{A} \in NP$ }

P is closed under complementation [$P = co\text{-}P$]:

$$\forall A: A \in P \Leftrightarrow \bar{A} \in P$$

Assignment Project Exam Help
 \bar{A}



NP is not known to be closed under complementation.

$P = co\text{-}P$, $P \subseteq NP$, $co\text{-}P \subseteq co\text{-}NP$.

$$P \subseteq NP \cap co\text{-}NP$$

NP-complete

Assignment Project Exam Help
NP co-NP

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

P = NP ?

Open Questions:

P = NP \cap co-NP ?

NP = co-NP ?

Polynomial Reducibility

Assignment Project Exam Help

NP <https://eduassistpro.github.io/>
ess

Add WeChat edu_assist_pro

Hardest Problems in NP

- NP-complete = the class of hardest problems in NP.
- How do we show that a problem $A \in \text{NP}$ is NP-complete, i.e., A is at least as hard as any other problem in NP ?
- That is, if A is tractable, then **every** problem in NP is also tractable.
In other words: A
Assignment Project Exam Help
<https://eduassistpro.github.io/>
- **Polynomial Reducibility**, denoted “ \leq_{pol} ” to use.
Add WeChat: edu_assist_pro

Polynomial Reducibility

$A \leq_P B$: Problem A is polynomially reducible to Problem B,

if there is a polynomial-time computable transformation f from instances of A to instances of B that preserves membership:

1. $I \in A \Leftrightarrow f(I) \in B$, and
2. $f(I)$ can be

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

NOTE:

- the direction of reduction is from A to B (not the reverse).
- $I \in A \Rightarrow f(I) \in B$, &
- $I \notin A \Rightarrow f(I) \notin B$, &
- \exists a reduction algorithm that computes $f(I)$ in $\text{poly}(|I|)$ time, &
- length of instance $f(I)$ is also $\text{poly}(|I|)$.

Polynomial Reducibility

$A \leq_P B$

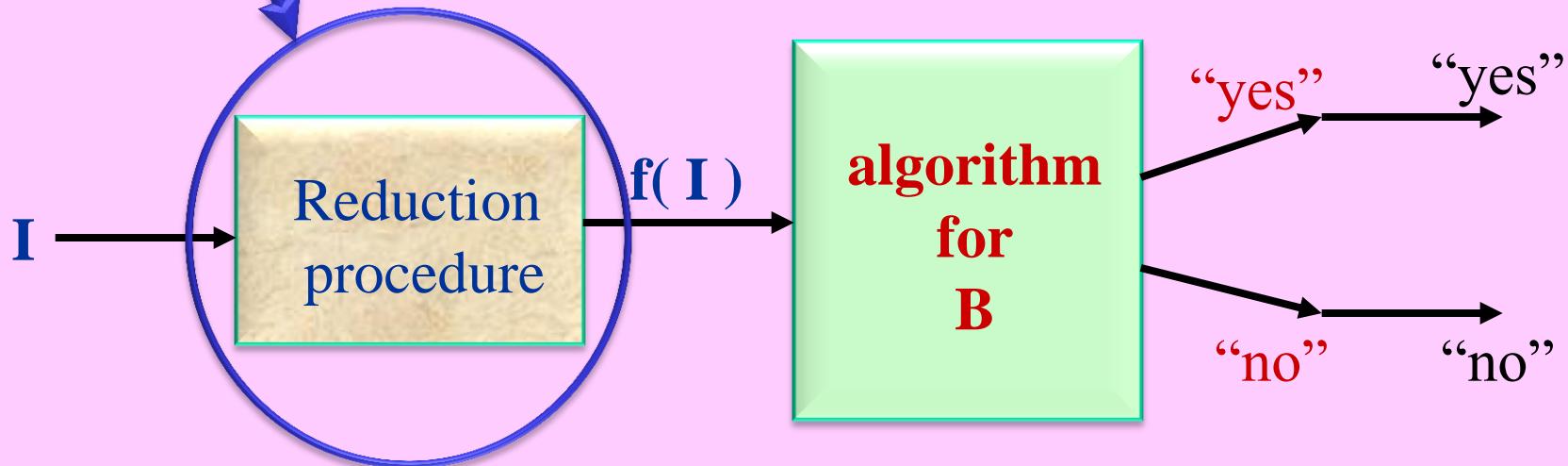
says:

A is not harder than B (to within a polynomial), &
B is at least as hard as A (the contra-positive)

- $B \in P \Rightarrow A \in P$
- $A \notin P$

<https://eduassistpro.github.io/>

Add WeChat for algorithm for
`edu_assist_pro`



NP-hard & NP-complete

B is NP-hard if every problem in NP polynomially reduces to B:

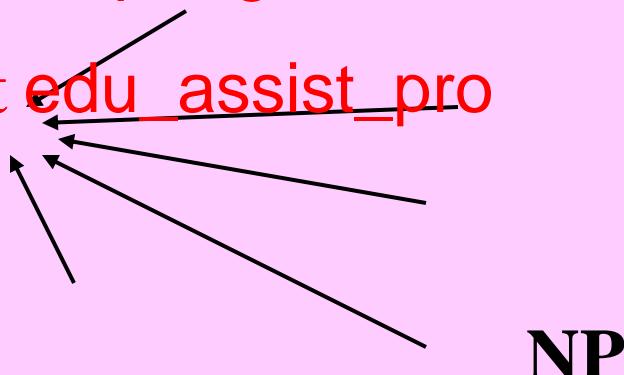
$$\forall A \in NP: A \leq_P B$$

Assignment Project Exam Help

<https://eduassistpro.github.io/>

B is NP-complete if
Add WeChat edu_assist_pro

1. $B \in NP$, and
2. B is NP-hard.



Circuit SAT is NP-complete

This is our first problem to be proved NP-complete.

We must show two things:

1. Circuit SAT \in NP,
2. Circuit SAT is NP-hard: $\forall A \in \text{NP}: A \leq_P \text{Circuit SAT}$.

Proof of 1. Circuit SAT \in NP: Assignment Project Exam Help
We have already seen this.

An instance of the problem <https://eduassistpro.github.io/>
of size n (# wires and gates)
Add WeChat edu_assist_pro

A certificate is a truth assignment to the input gates.

In time $\text{poly}(n)$ (in fact **in $O(n)$ time**) we can deterministically **verify**
whether the given certificate satisfies the circuit output gate:
evaluate gate outputs in topological order.

Circuit SAT is NP-complete

This is our first problem to be proved NP-complete.

We must show two things:

1. Circuit SAT \in NP,
2. Circuit SAT is NP-hard: $\forall A \in \text{NP}: A \leq_P \text{Circuit SAT}$.

Proof of 2 [sketch] Circuit SAT is NP-hard: Assignment Project Exam Help

Consider an arbitrary problem $A \in \text{NP}$.

\exists a deterministic algo
binary) & $|S| = \text{poly}(|I|)$ certificate S (encoded in
solution for instance I of problem A .
<https://eduassistpro.github.io/> files whether S is a valid

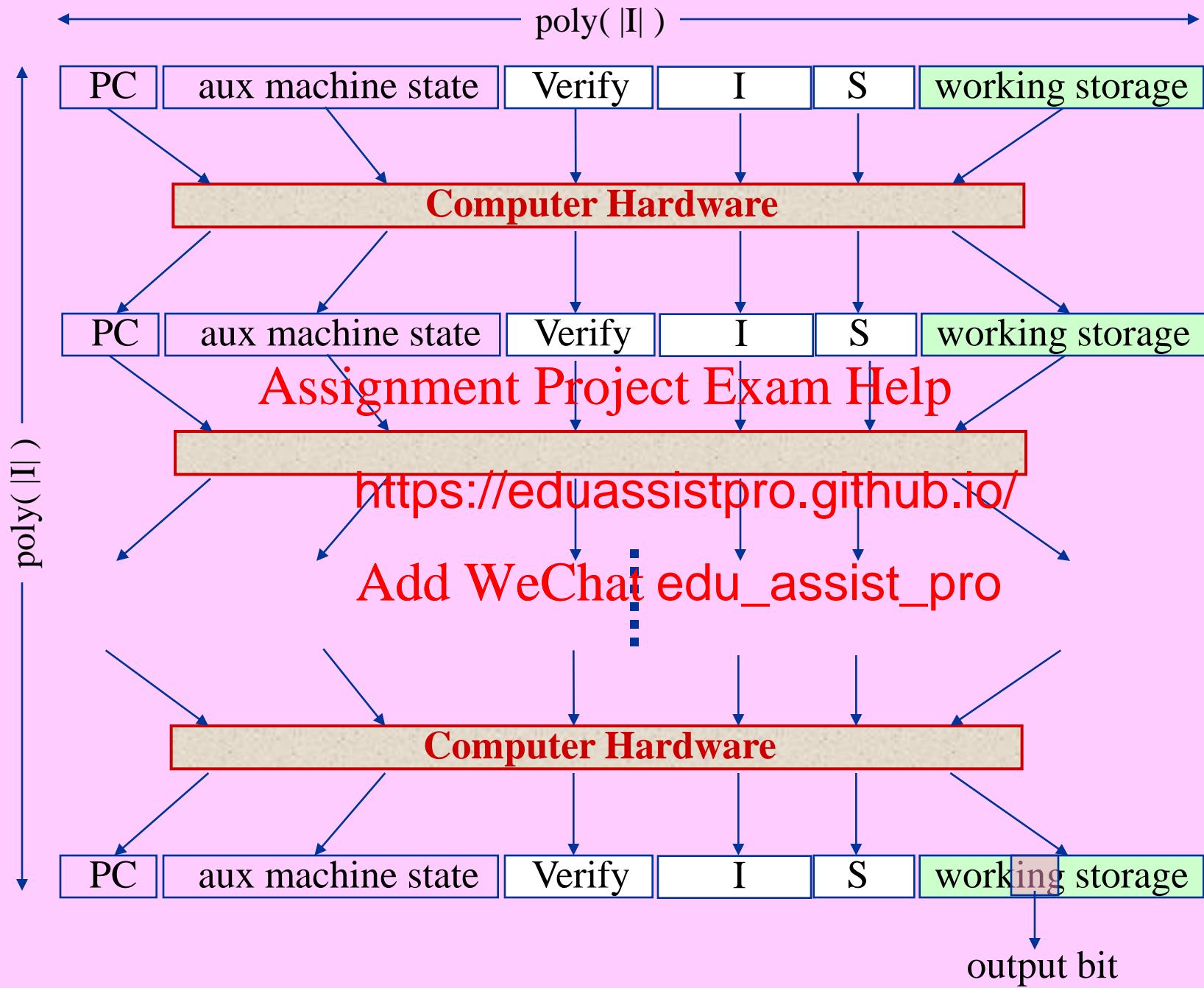
Add WeChat edu_assist_pro

Reduction Procedure: transform Verify(I

en instance I , to a circuit

C with the following properties:

- a) The reduction procedure constructs C in $\text{poly}(|I|)$ time,
- b) C has input bits I and S , where the I bits are given and fixed (0 or 1),
and the S bits are unknown variables (? bits),
- c) $\text{Verify}(I, S) = \text{"yes"}$
 \Leftrightarrow the truth assignment S to the variable input bits satisfies C . (P.T.O.)



Other NP-complete Problems

There is a shortcut on how to prove other problems are NP-complete.

FACT 1: \leq_P is transitive.

Proof: $A \leq_P B$ and $B \leq_P C \Rightarrow A \leq_P C$

(because polynomials are closed under composition)

Assignment Project Exam Help

$A \leq_P B : x \in A \Leftrightarrow f(x) \in B$
for some $f : f(n) = |x|^d$ & some constant d .

$B \leq_P C : x \in B \Leftrightarrow g(x) \in C$
for some $g : g(x)$ is computable in $O(n^c)$ time, $n = |x|$ & some constant c .

↓

$A \leq_P C : x \in A \Leftrightarrow h(x) \in C$
for $h(x) = g(f(x))$ computable in $O(n^{dc})$ time, $n = |x|$ & the constant dc .

Other NP-complete Problems

There is a shortcut on how to prove other problems are NP-complete.

FACT 1: \leq_P is transitive.

FACT 2: Suppose $A \leq_P B$. Then

Assignment Project Exam Help

A is NP-hard

<https://eduassistpro.github.io/>

Proof: $[\forall C \in NP: C \leq_P A]$ and $[A \leq_P B \Rightarrow C \leq_P B]$

How to prove a new problem B is NP-hard:

Select a known NP-complete problem A and show $A \leq_P B$.

If you have choices, pick a problem A that “resembles” B as much as possible to facilitate an easier reduction.

NP-completeness & P vs NP ?

FACT 3: The following statements are equivalent:

- (1) $P = NP$,
- (2) All NP-complete problems are in P,
- (3) Some NP-complete problem is in P.

Assignment Project Exam Help

Proof: (1) \Rightarrow (2) \Rightarrow

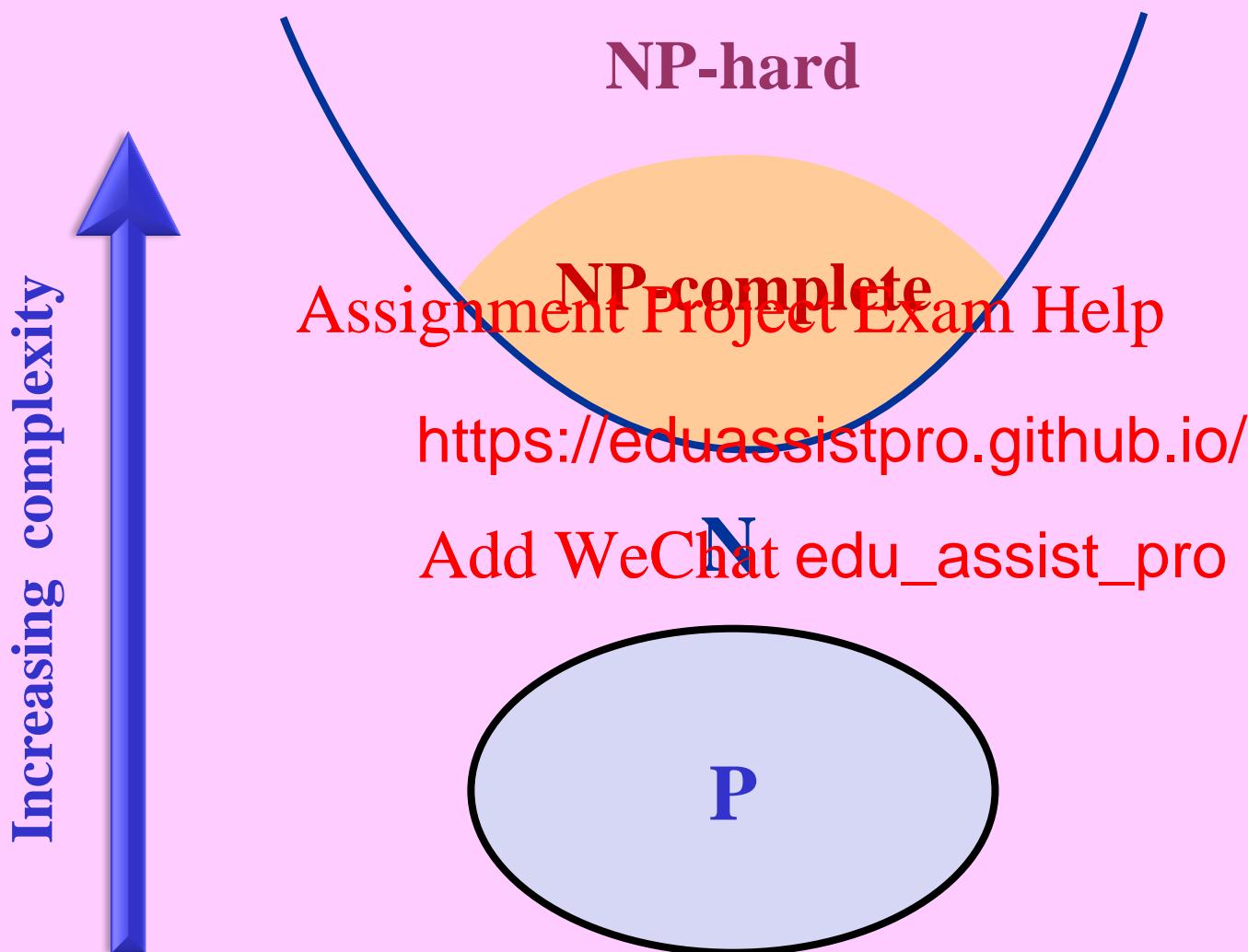
(3) \Rightarrow (1): $\text{https://eduassistpro.github.io/}$

$A \leq_P B$ and B
Add WeChat edu_assist_pro

If a single NP-complete problem is tractable, then all are!

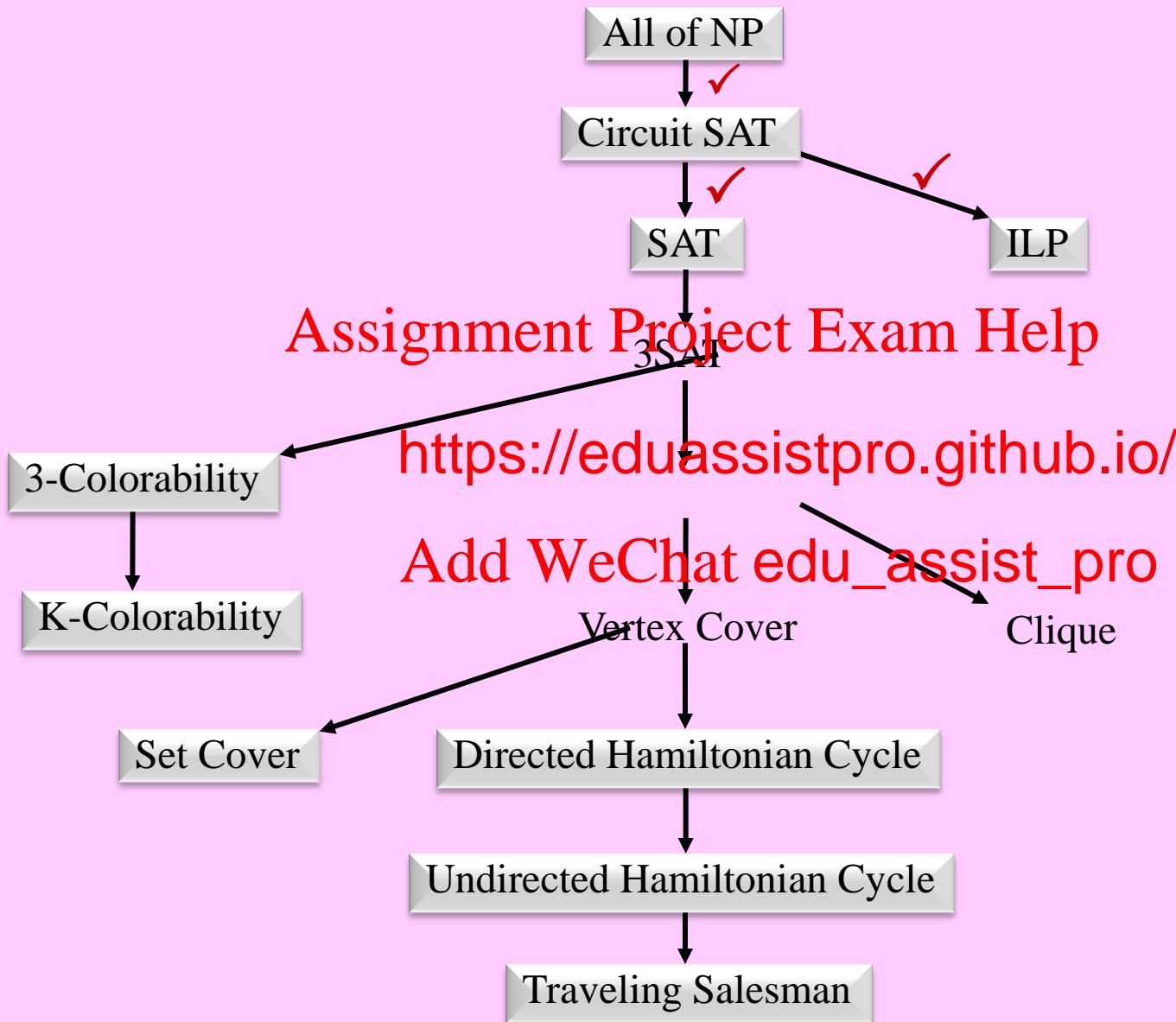
This is a **strong but inconclusive** evidence that $P \neq NP$,
since despite decades of intense research by experts,
no polynomial-time algorithm has been found for any one
of a multitude of well known NP-complete problems.

If $P \neq NP$



Additional
Assignment Project Exam Help
N **te**
<https://eduassistpro.github.io/>
Probl
Add WeChat edu_assist_pro

Some Reductions



These are all in NP

To prove a problem is NP-complete, we must show 2 things:

- (1) It is in NP (i.e., it has a polynomial-time verification algorithm)
- (2) It is NP-hard (the indicated reductions will show this)

(1) The listed problems are all in NP. (All except ILP are easy to show.)

Some examples:

Assignment Project Exam Help

3SAT ∈ NP:

Verify that a given 3SAT instance:
Done in linear time

3-Colorability ∈ NP: Add WeChat `edu_assist_pro`

Given a coloring of vertices of a graph, verify that at most 3 colors are used, and verify that for each edge, its two incident vertices have different colors.

Undirected Hamiltonian Cycle ∈ NP:

Given a permutation certificate of vertices of a graph, verify that it forms a Hamiltonian cycle, i.e., a cycle in the graph that visits each vertex exactly once: Check that the given certificate is indeed a permutation, and that there is an edge in the graph between each successive pair of vertices in the permutation (cyclically).

SAT \leq_p 3SAT

In **polynomial time** transform an instance of SAT to an instance of 3SAT:
Convert each clause with $k > 3$ literals to $k-2$ clauses, each with 3 literals,
using some new Boolean variables:

$$\varphi = (a_1 \vee a_2 \vee \dots \vee a_k)$$



$$\psi = (a_1 \vee a_2 \vee x_1)(\bar{x}_1 \vee a_3 \vee x_2)(\bar{x}_2 \vee a_4 \vee x_3) \dots (\bar{x}_{k-4} \vee a_{k-2} \vee x_{k-3})(\bar{x}_{k-3} \vee a_{k-1} \vee a_k)$$

<https://eduassistpro.github.io/>

CLAIM: φ is satisfied $\Leftrightarrow \exists$ truth assign r which ψ is satisfied.

Add WeChat edu_assist_pro

Proof of [\Rightarrow]: If φ is satisfied, then some a_i

Set x_1, \dots, x_{i-2} to true and the rest false. That satisfies ψ .

Proof of [\Leftarrow]: If ψ is satisfied, then at least one a_i must be true.

Otherwise, the 1st clause of ψ forces x_1 to be true,

then the 2nd clause forces x_2 to be true, ...

eventually falsifying the last clause of ψ ; a contradiction.

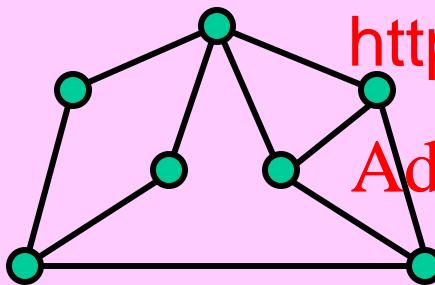
3-Colorability

The 3-Colorability Problem:

Given a graph G ,

is it possible to color each vertex of G by one of 3 colors (say red, blue, green) such that no edge of G has its both end vertices colored the same?

Assignment Project Exam Help



3-colorable

https://eduassistpro.github.io/
Add WeChat edu_assist_pro



Not 3-colorable

3SAT \leq_P **3-Colorability**

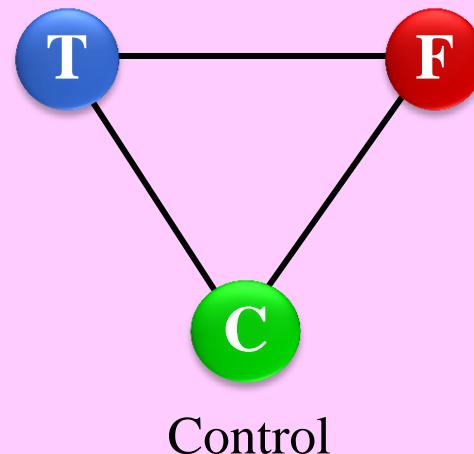
Reduction: Given any 3-CNF-SAT formula Φ , in time polynomial in $|\Phi|$, transforms Φ to a graph G such that

Φ is satisfiable $\Leftrightarrow G$ is 3-colorable.

G will be constructed from Φ
Assignment Project Exam Help
by assembling suitable pieces together

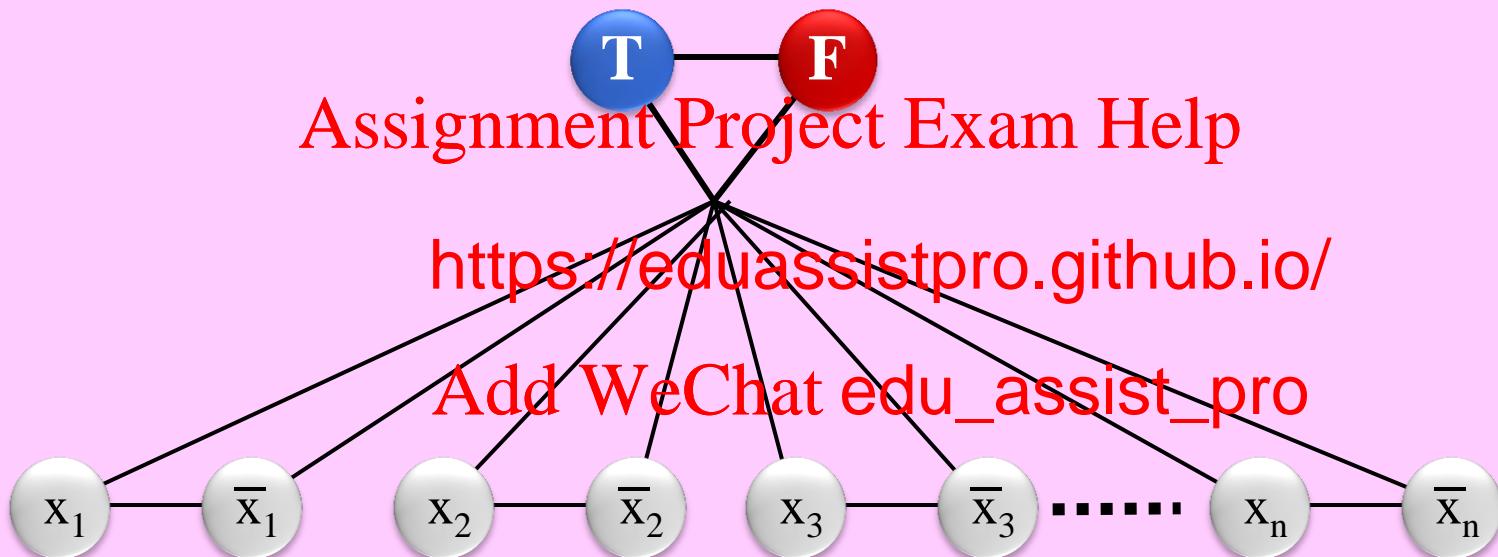
<https://eduassistpro.github.io/>

Add WeChat True **edu_assist_pro**



3SAT \leq_P 3-Colorability

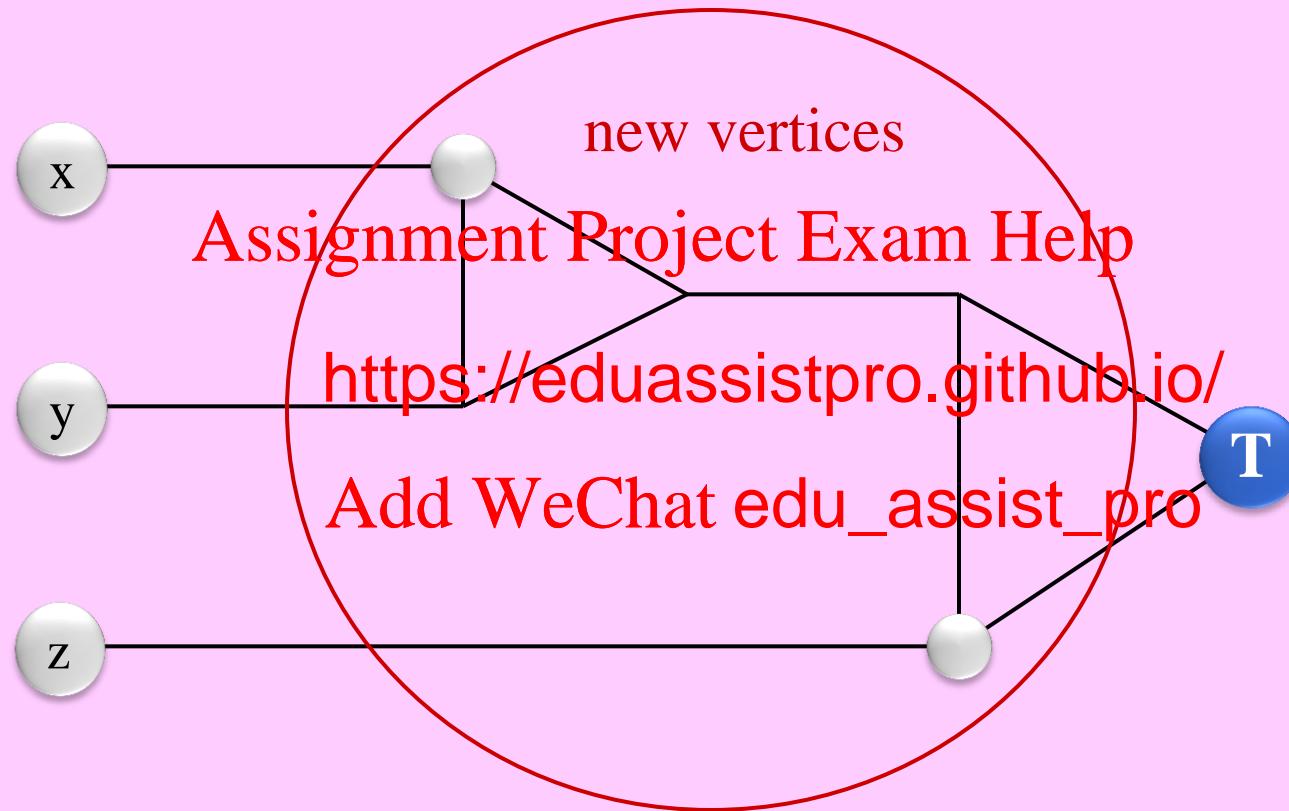
STEP 2: The assembly of Boolean literals



Each literal is forced a color T or F and its negation the opposite (F or T).
This forces a truth assignment.

$3\text{SAT} \leq_P 3\text{-Colorability}$

STEP 3: The assembly for each clause $(x \vee y \vee z)$

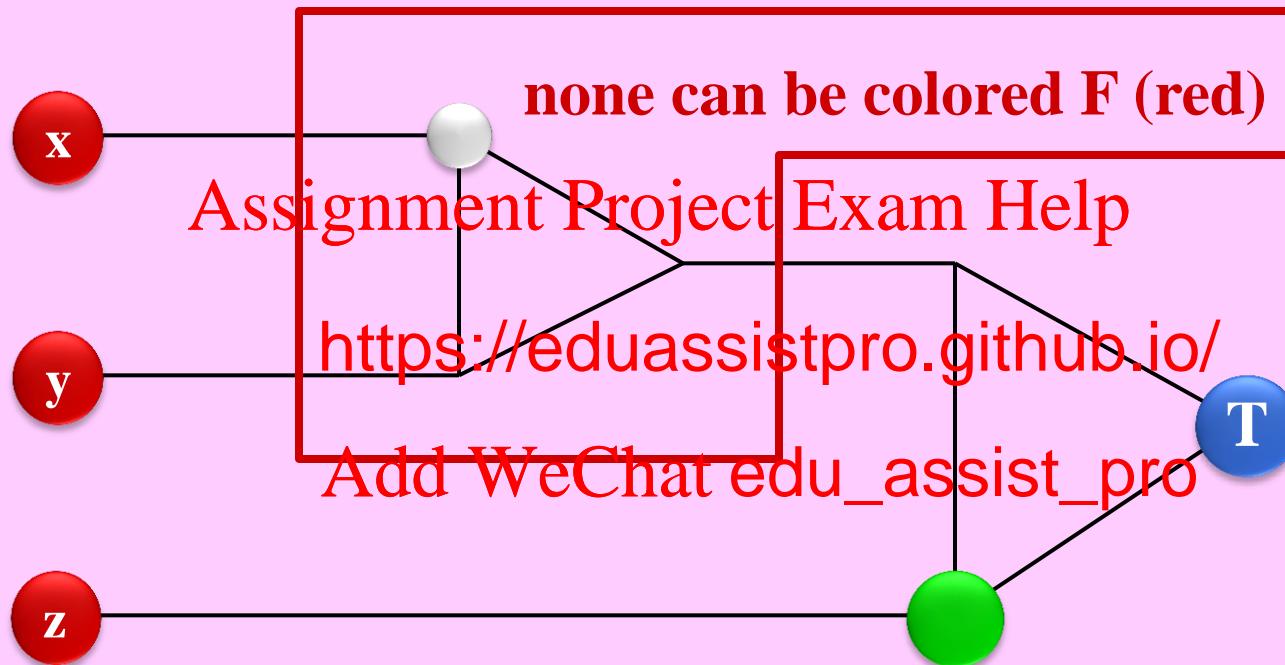


CLAIM: Suppose nodes x, y, z each get a color (T or F). Then, it is possible to complete a 3 coloring of the assembly \Leftrightarrow at least one of x, y, z got the color T.

3SAT \leq_P 3-Colorability

[\Rightarrow]: Suppose all x, y, z are colored F.

Then it's impossible to complete a 3 coloring of the assembly.



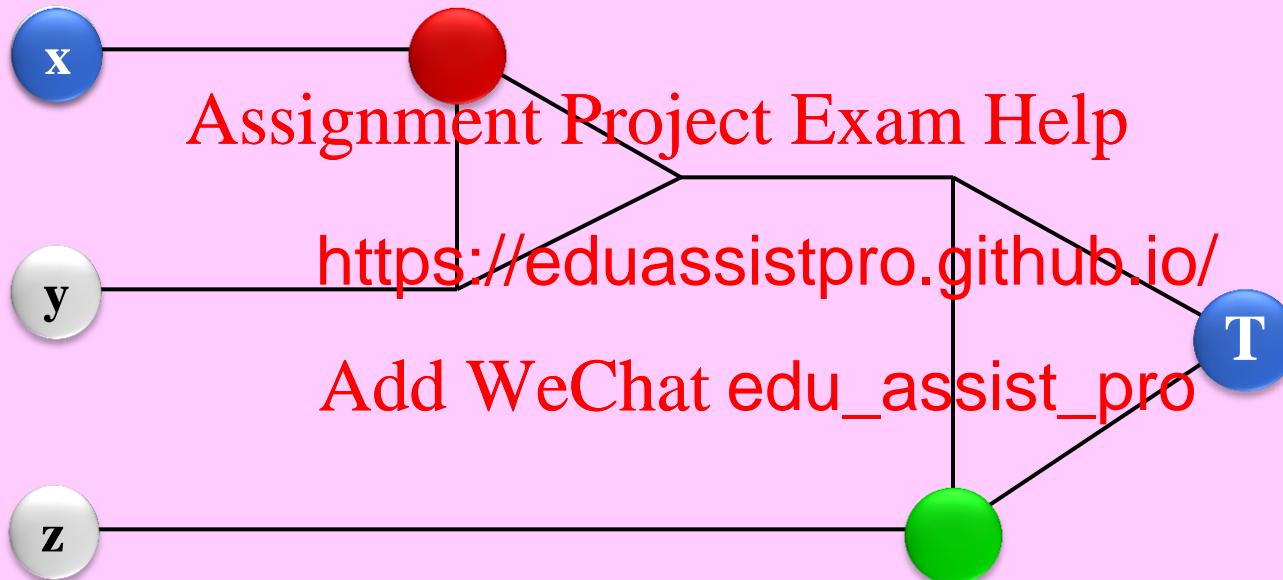
CLAIM: Suppose nodes x, y, z each get a color (T or F). Then, it is possible to complete a 3 coloring of the assembly
 \Leftrightarrow at least one of x, y, z got the color T.

3SAT \leq_P 3-Colorability

[\Leftarrow]: Suppose at least one of x, y, z is colored T.

Then it's possible to complete a 3 coloring of the assembly.

CASE 1: x is colored T: (the case that y is colored T is symmetric)



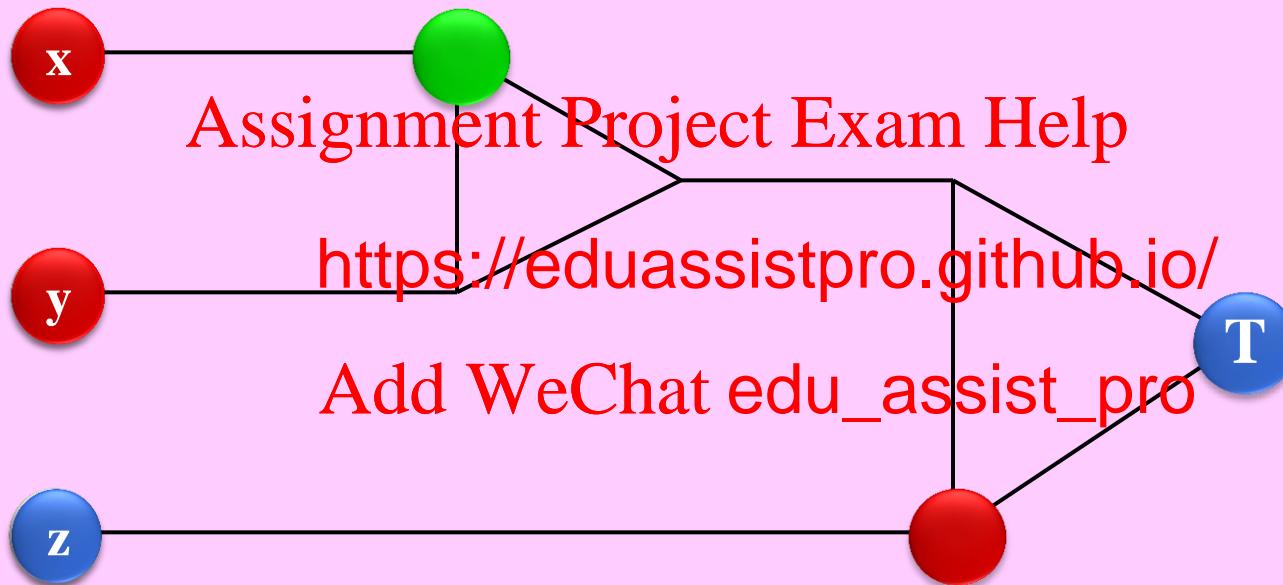
CLAIM: Suppose nodes x, y, z each get a color (T or F). Then, it is possible to complete a 3 coloring of the assembly
 \Leftrightarrow at least one of x, y, z got the color T.

3SAT \leq_P 3-Colorability

[\Leftarrow]: Suppose at least one of x, y, z is colored T.

Then it's possible to complete a 3 coloring of the assembly.

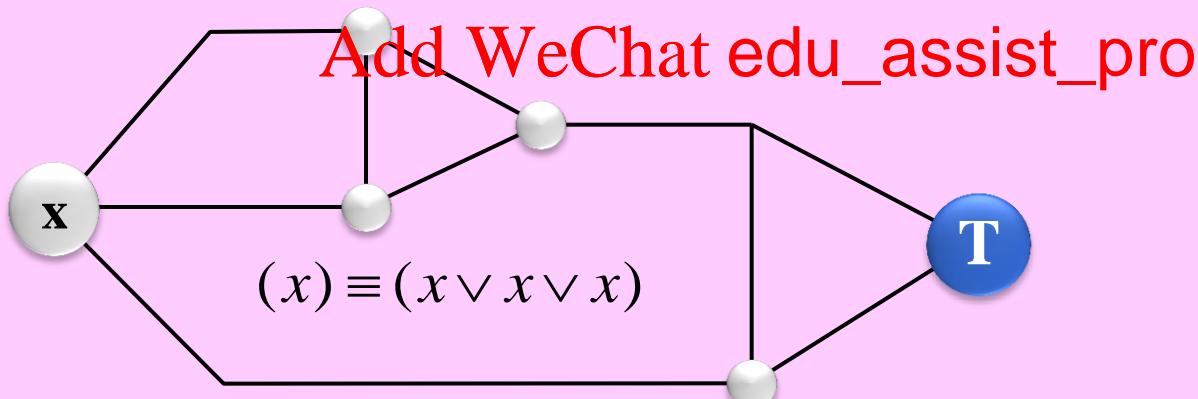
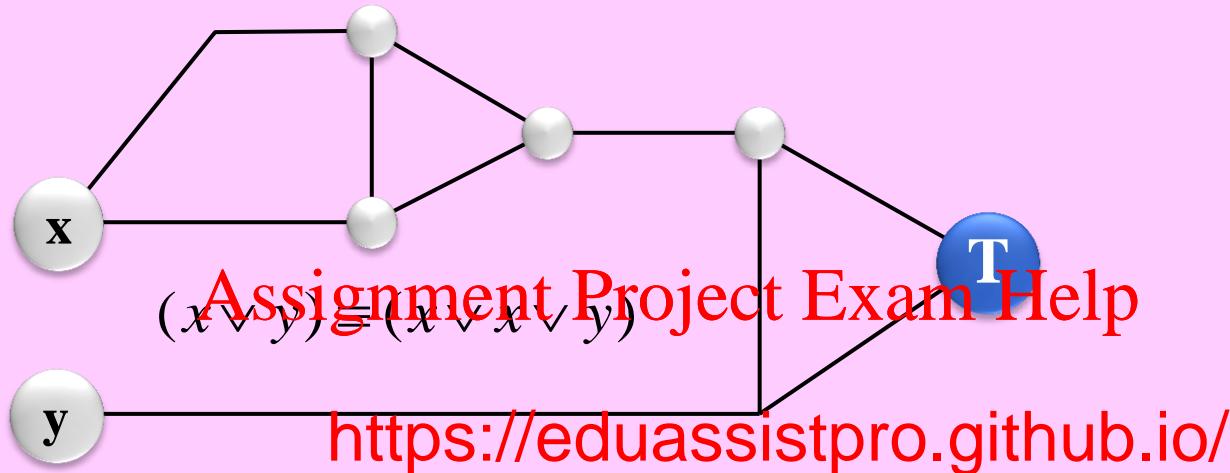
CASE 2: x and y are colored F and z is colored T:



CLAIM: Suppose nodes x, y, z each get a color (T or F). Then, it is possible to complete a 3 coloring of the assembly
 \Leftrightarrow at least one of x, y, z got the color T.

3SAT \leq_P 3-Colorability

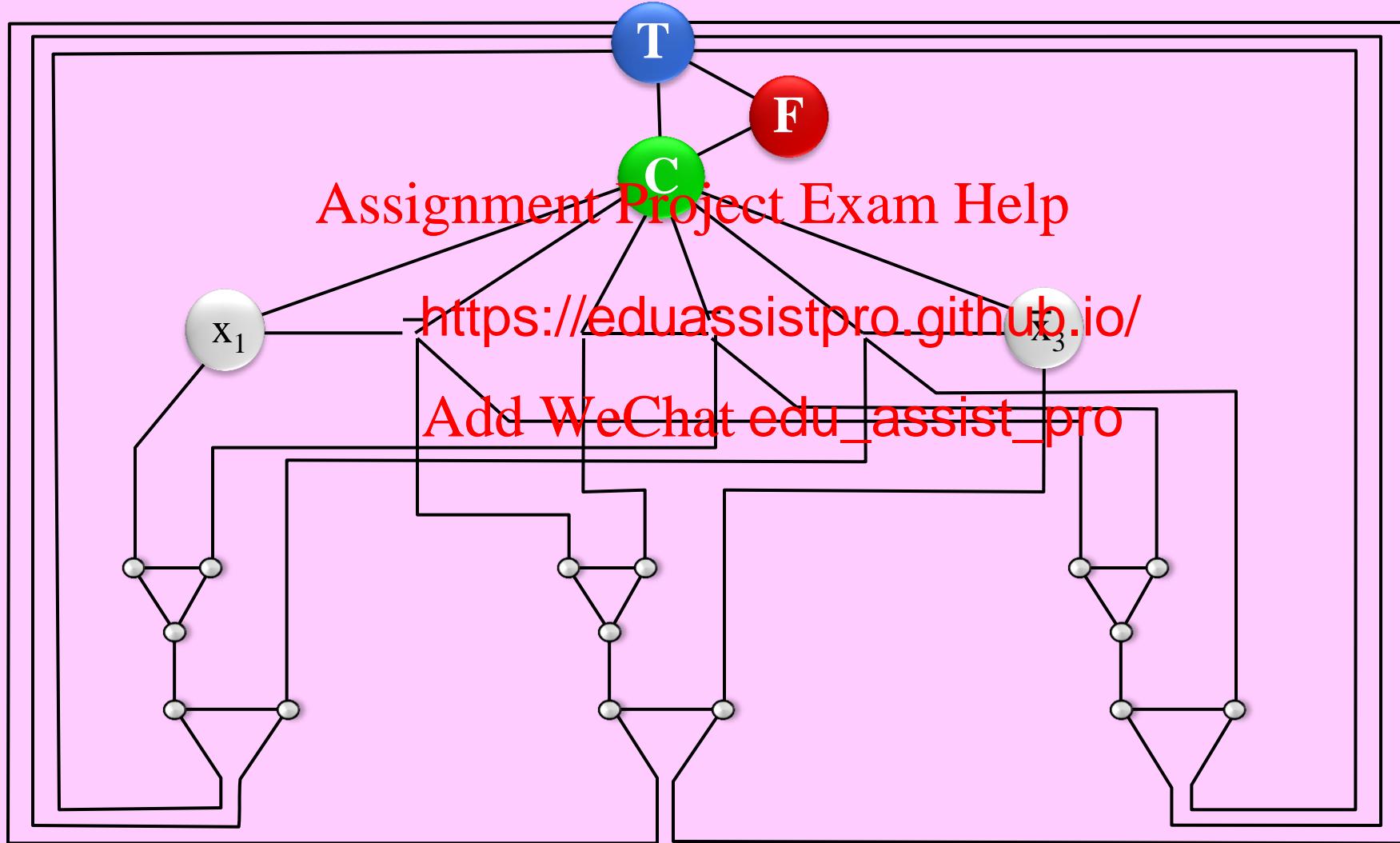
Clauses with less than 3 literals: “fuse” some literal “entry” points.



[These can be further simplified.]

3SAT \leq_P 3-Colorability

$$(x_1 \vee \bar{x}_2 \vee x_3) \wedge (\bar{x}_1 \vee x_2 \vee \bar{x}_3) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_3)$$



3SAT \leq_P Independent Set

The Independent Set Problem:

An independent set in a graph is a set of vertices, no pair of which are adjacent. Given a graph G and integer K, does G have an independent set of size $\geq K$?

Reduction: Convert each clause with L literals to a clique of L vertices.

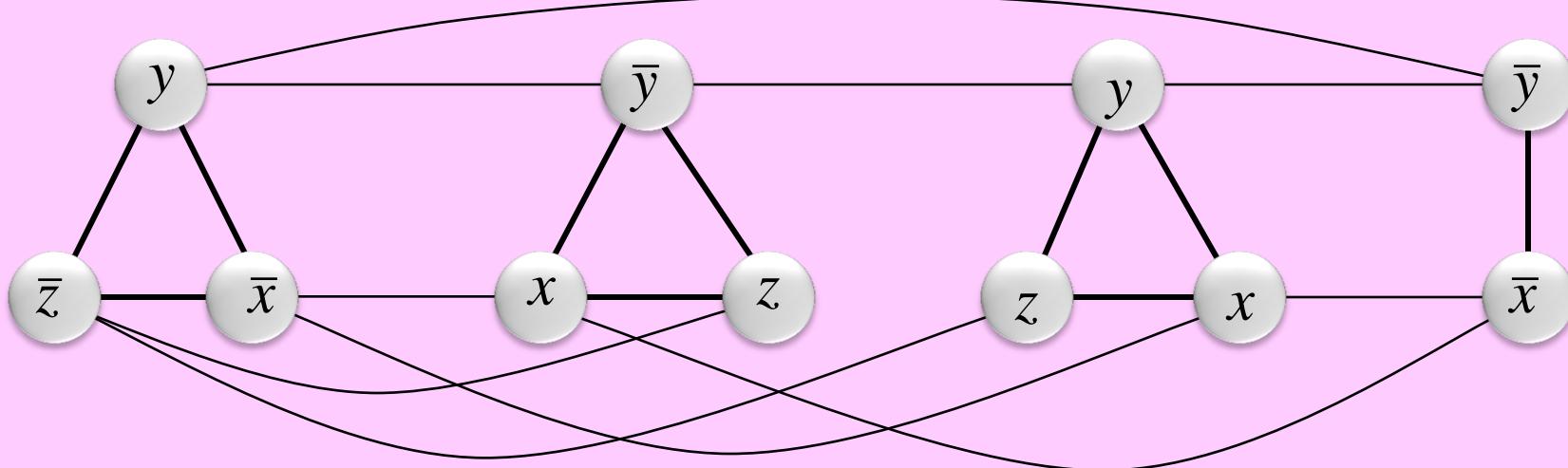
Add an edge between each literal and its negation. $K = \# \text{ clauses}$.

Assignment Project Exam Help

$$(\bar{x} \vee y \vee \bar{z}) \wedge (x \vee \bar{y} \vee z) \wedge (x \vee y \vee z) \wedge (\bar{x} \vee \bar{y})$$

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro



Independent Set \leq_P Clique

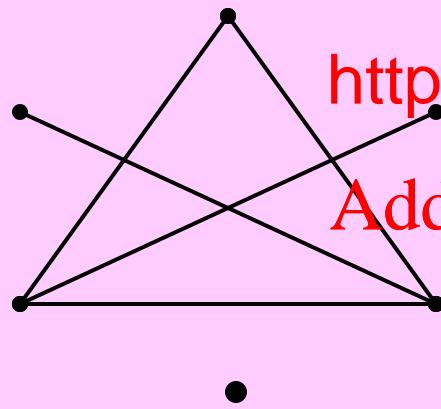
The Clique Problem:

A clique in a graph is a complete sub-graph,
i.e., a set of vertices, every pair of which are adjacent.

Given a graph G and integer K , does G have a clique of size $\geq K$?

Reduction: S is an independent set in $G \Leftrightarrow S$ is a clique in the complement \bar{G} .

Assignment Project Exam Help



An independent set
of size 4 in G .

<https://eduassistpro.github.io/>
Add WeChat edu_assist_pro



A 4-clique in \bar{G} .

Independent Set \leq_p Vertex Cover

The Vertex Cover Problem:

A vertex cover in a graph is a subset C of vertices that covers every edge, i.e., each edge of the graph has at least one end in C .

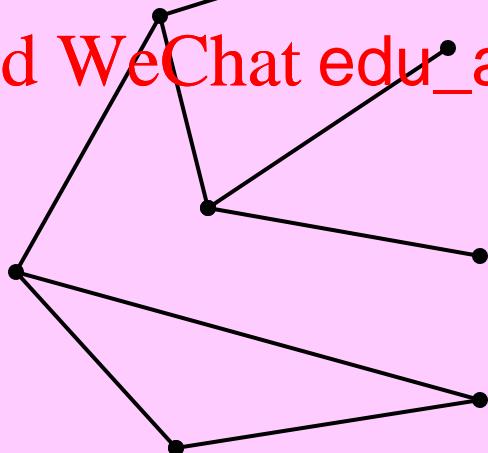
Given a graph $G = (V, E)$ and integer K , does G have a vertex cover of size $\leq K$?

Reduction: $C \subseteq V$ is a vertex cover in $G \Leftrightarrow V - C$ is an independent set in G .

G has an independent set of size $\geq |V| - K \Rightarrow G$ has a vertex cover of size $\leq |V| - K$.

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro



Vertex Cover \leq_P Set Cover

The Set Cover Problem:

Instance: A set $X = \{x_1, x_2, \dots, x_n\}$ and a collection $S = \{S_1, S_2, \dots, S_m\}$ of subsets of X such that S covers X , i.e., $S_1 \cup S_2 \cup \dots \cup S_m = X$.

A **cover set** is any sub-collection $C \subseteq S$ that also covers X .

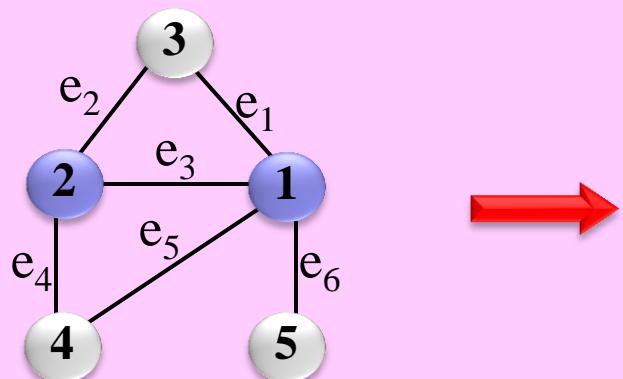
Question: Given $\langle X, S, K \rangle$, is there a cover set C of size $\leq K$?

Reduction: $\langle G, K \rangle \rightarrow \langle X, S, K \rangle$

<https://eduassistpro.github.io/> to $v \in G\}$, for $v \in V(G)$.

G has a vertex cover of size $\leq K \Leftrightarrow \langle X, S, K \rangle$ has a set cover of size $\leq K$.

Add WeChat **edu_assist_pro**



$X = \{e_1, e_2, e_3, e_4, e_5, e_6\}$
 $S = \{S_1, S_2, S_3, S_4, S_5\}$
 $S_1 = \{e_1, e_3, e_5, e_6\}, \quad S_4 = \{e_4, e_5\}$
 $S_2 = \{e_2, e_3, e_4\}, \quad S_5 = \{e_6\}$
 $S_3 = \{e_1, e_2\}$

$\{1, 2\}$ is a vertex cover in G

$\{S_1, S_2\}$ is a set cover of X

Hamiltonian Cycle

The Undirected (Directed) Hamiltonian Cycle Problem:

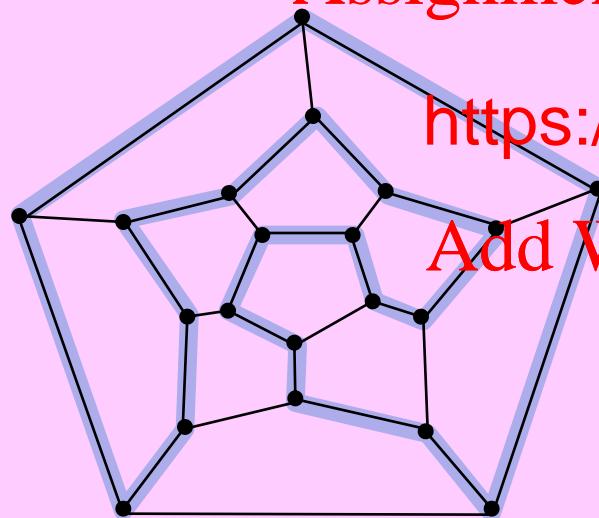
Instance: An undirected (directed) graph $G = (V, E)$.

Question: Is G Hamiltonian, i.e., does G have a Hamiltonian cycle; a cycle that goes through each vertex exactly once?

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat [edu_assist_pro](#)

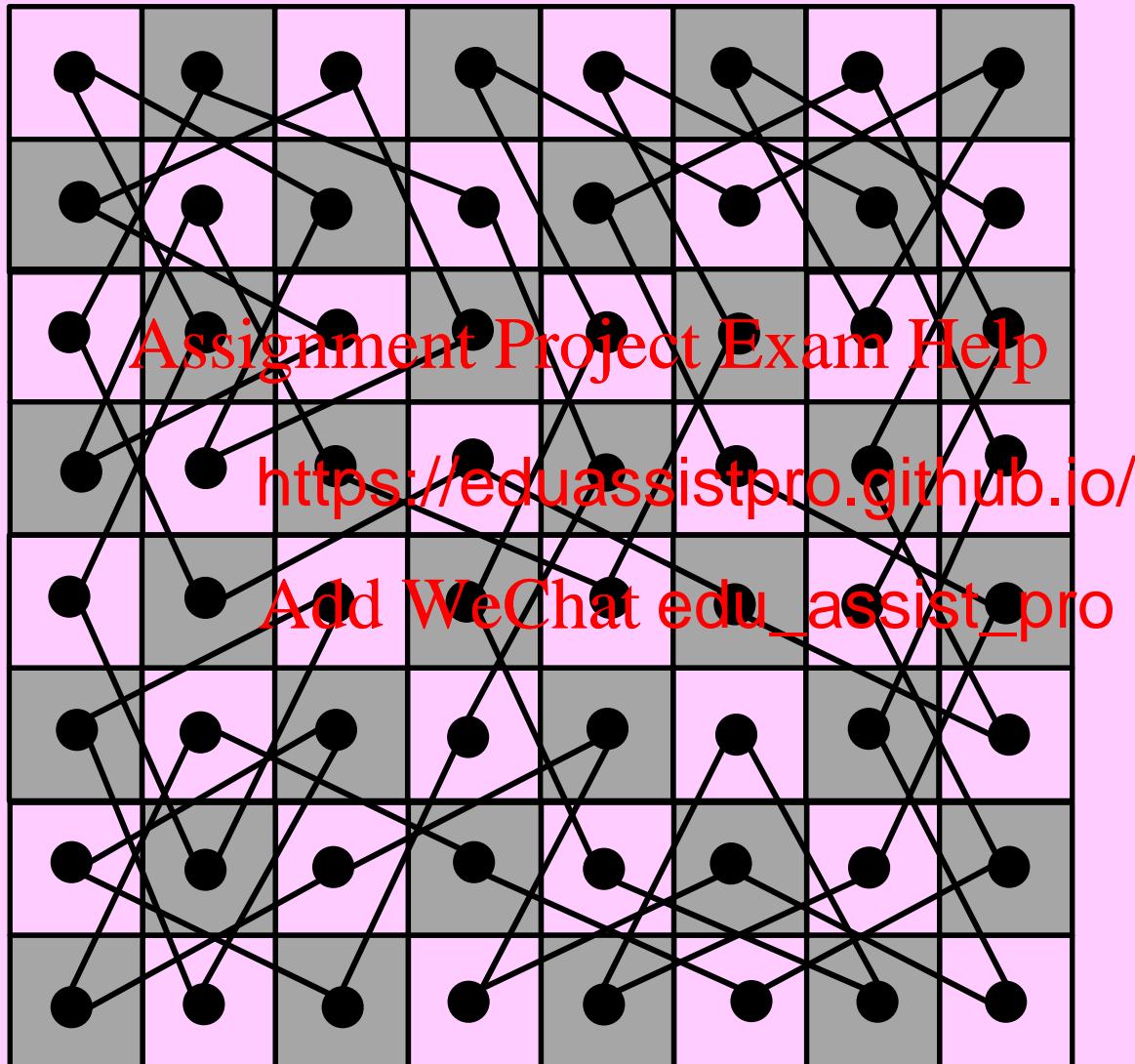


Hamiltonian
(skeleton of dodecahedron)



Non-Hamiltonian
(Peterson graph)

Knight's Tour on the chess board

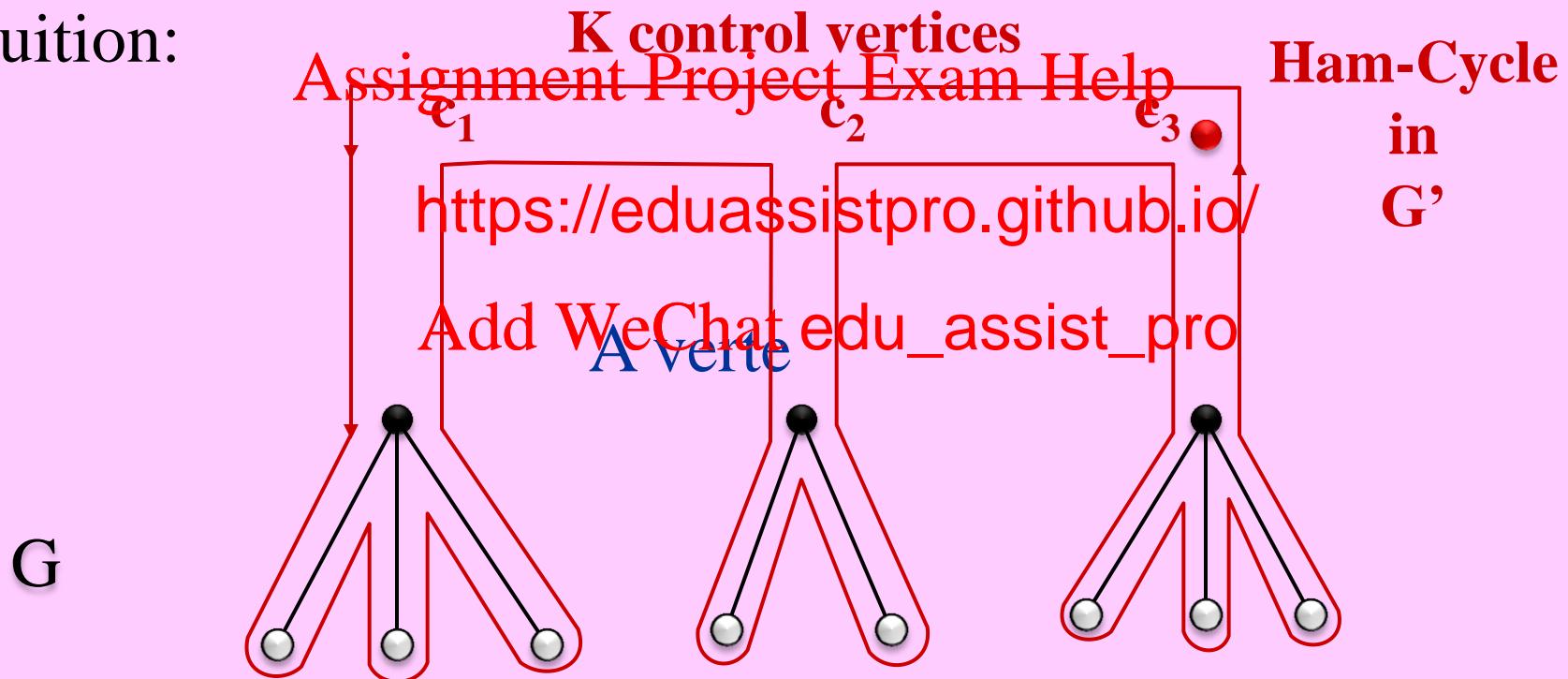


Vertex Cover \leq_P Directed Ham-Cycle

$\langle G, K \rangle$: does G have a vertex cover of size $\leq K$?

→ $\langle G' \rangle$: is digraph G' Hamiltonian?

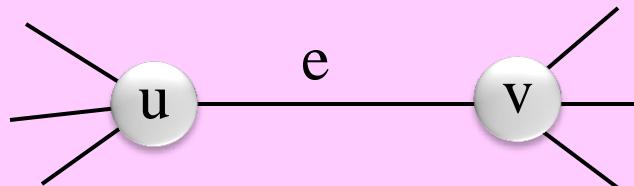
Intuition:



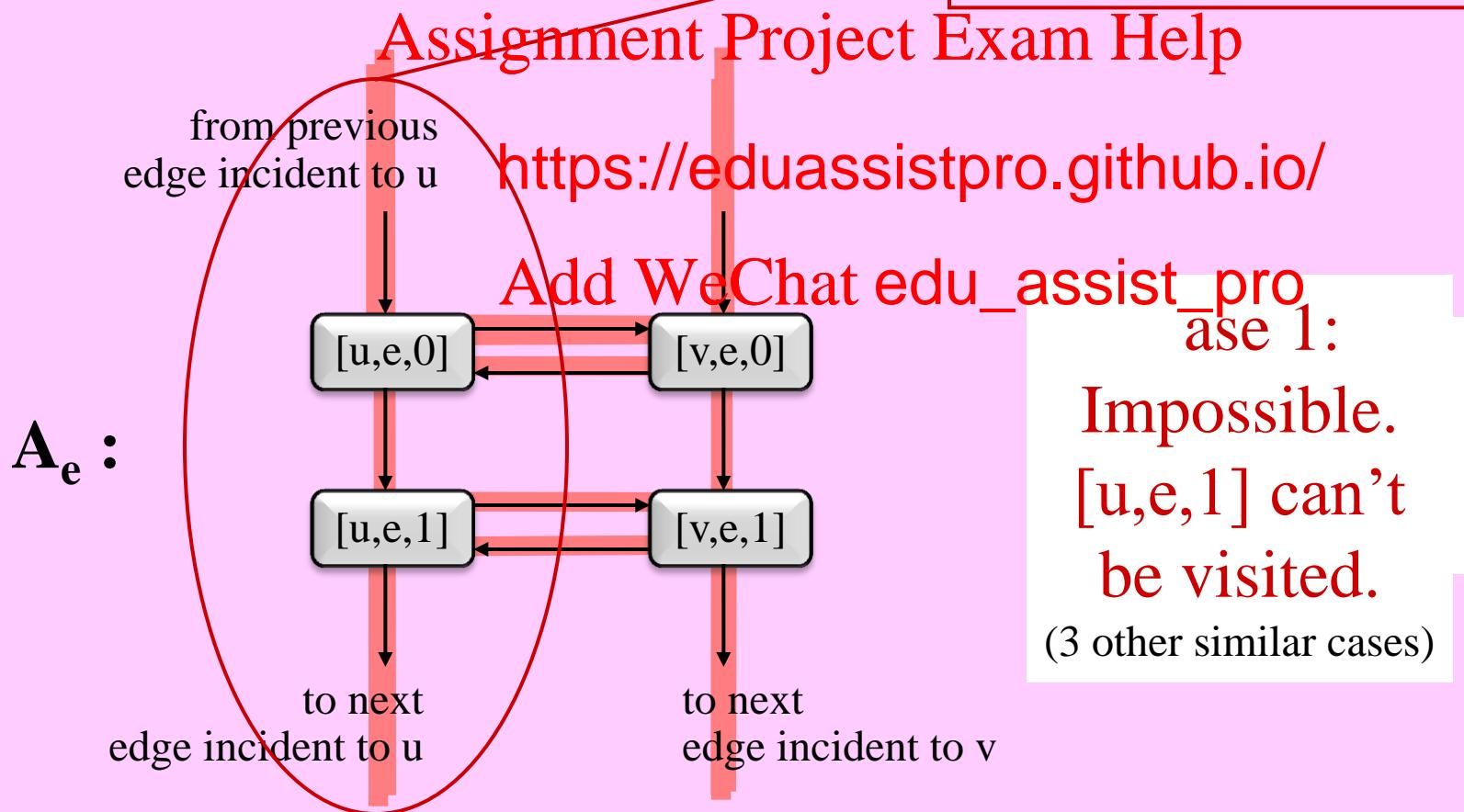
Vertex Cover \leq_P Directed Ham-Cycle

STEP 1: edge assembly A_e

G:

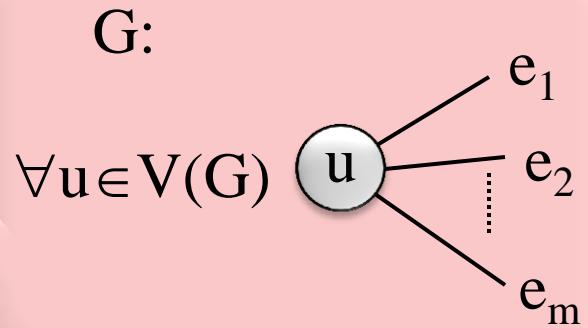
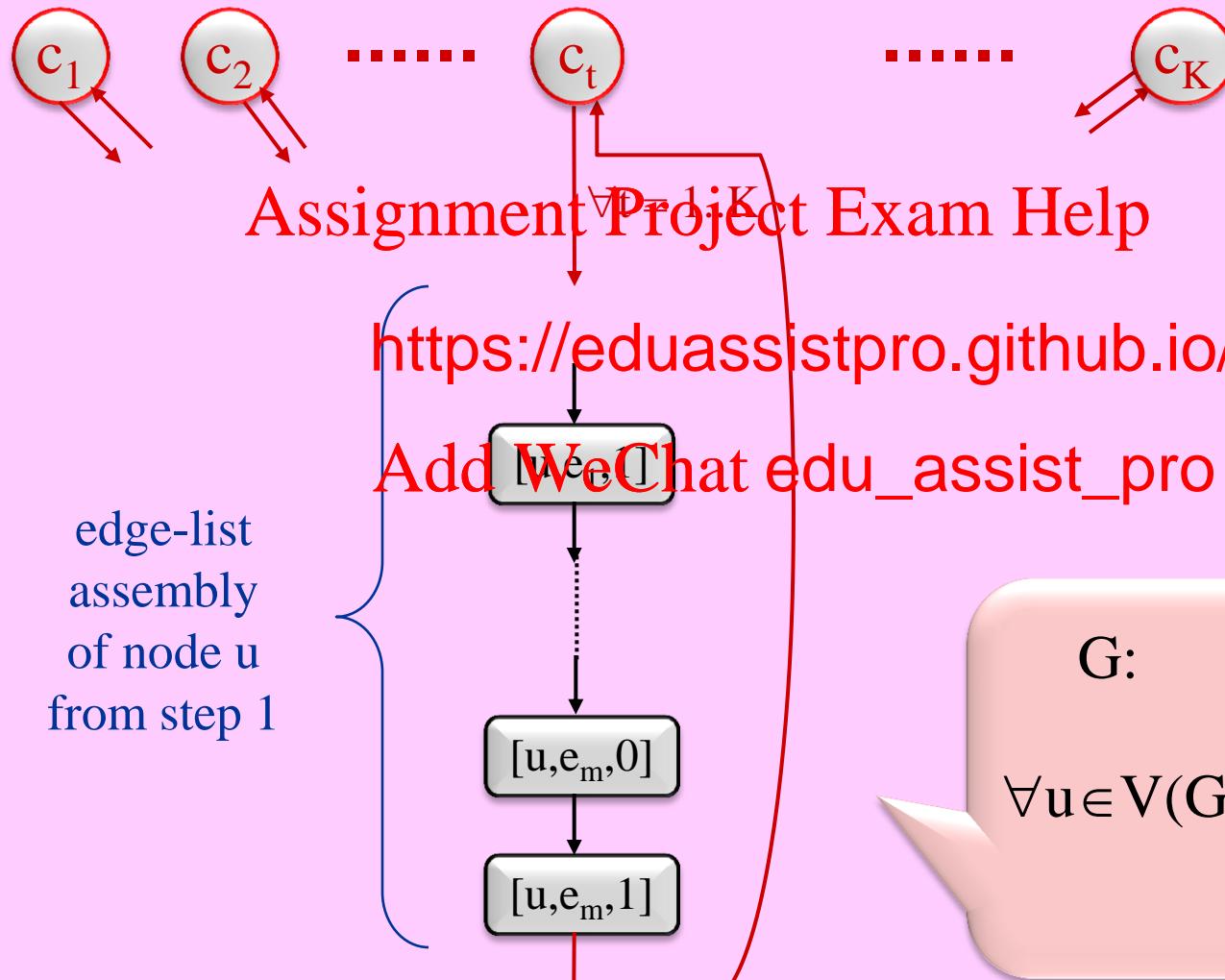


forms a sequential list of edges incident to node u



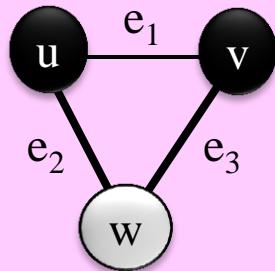
Vertex Cover \leq_p Directed Ham-Cycle

STEP 2: assembly for control nodes c_1, c_2, \dots, c_K

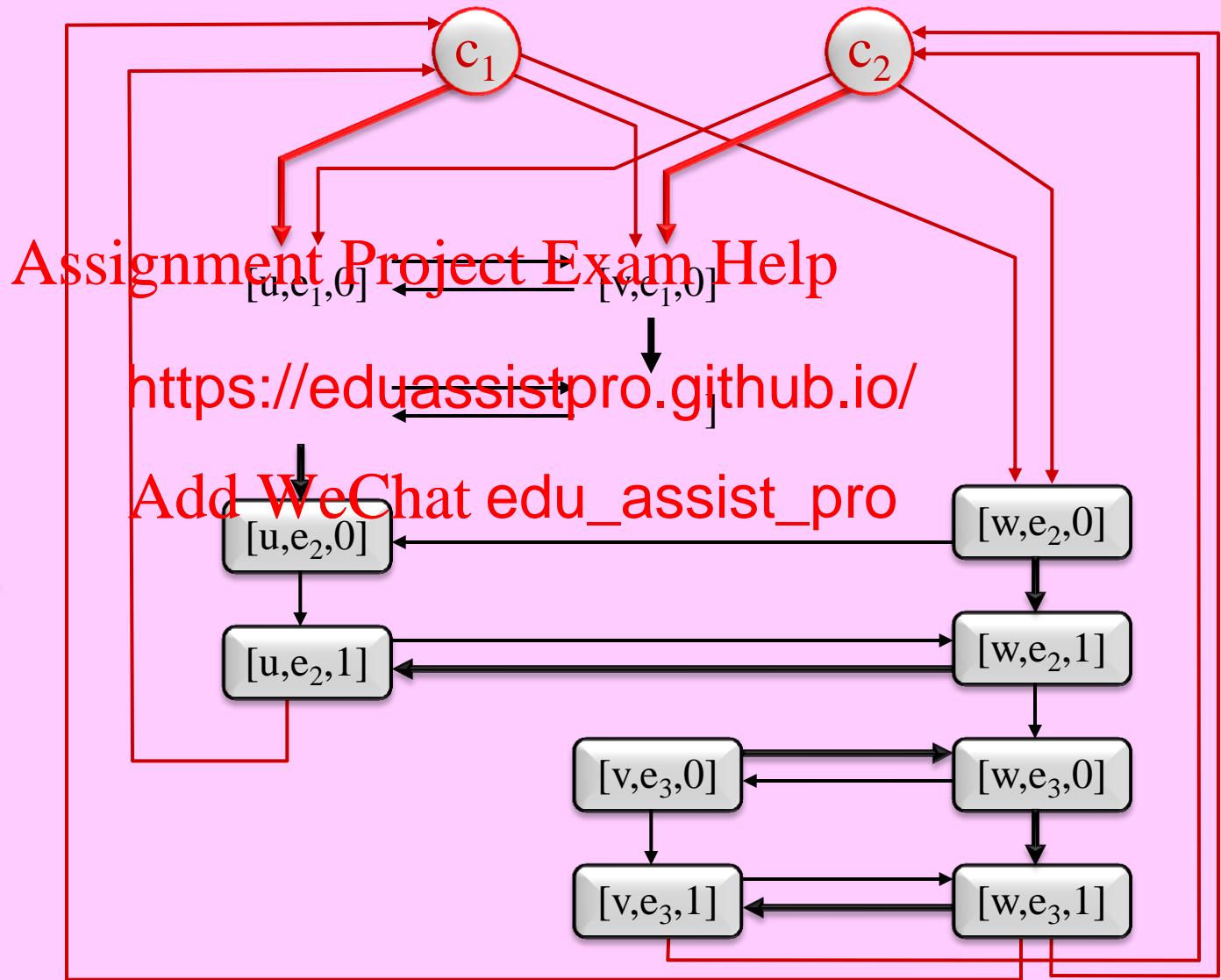


Vertex Cover \leq_P Directed Ham-Cycle

Vertex Cover
 $\{u, v\}$

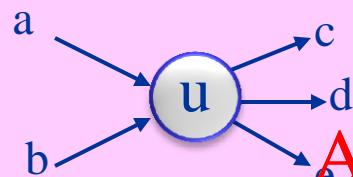


Thick edges
show
Ham-Cycle

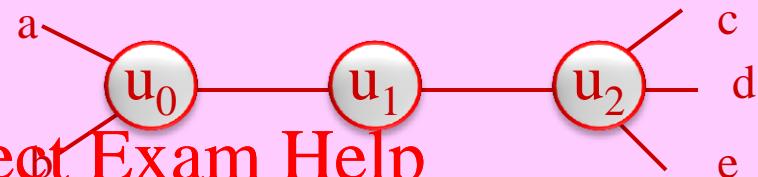


Directed Ham-Cycle \leq_P Undirected Ham-Cycle

Digraph G:



Undirected graph G':



Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro



CLAIM: G has a directed Ham-cycle \Leftrightarrow G' has an undirected Ham-cycle.

Hamiltonian Cycle \leq_P TSP

The Traveling Salesman Problem:

Instance: An $n \times n$ weighted adjacency matrix $D = (d_{ij})$ of a **complete** weighted graph on n vertices with integer edge lengths d_{ij} , and an integer K .

Feasible Sol: A TSP tour is any Hamiltonian cycle in this complete graph.

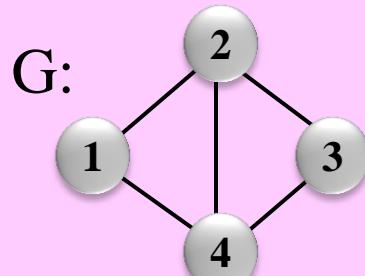
Question: Is there a TSP tour of length (or distance) $\leq K$?

Assignment Project Exam Help

$$G = (V, E) \rightarrow$$

$$d_{ij} = \begin{cases} 1 & \text{if } (i, j) \in E \\ 2 & \text{if } (i, j) \notin E \\ \text{for } i \neq j \end{cases}$$

https://eduassistpro.github.io/
distance matrix
Add WeChat edu_assist_pro



D =

0	1	2	1
1	0	1	1
2	1	0	1
1	1	1	0

CLAIM: G has a Hamiltonian cycle \Leftrightarrow D has a TSP tour of length $\leq n$.

Bibliography

- Steven Cook, “*The complexity of theorem proving procedures,*” Proceedings of STOC: pp. 151-158, 1971.
- Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullman, “*The Design and Analysis of Computer Algorithms,*” Addison-Wesley, 1974.
- Michael R. Garey, David S. Johnson, “*Computers and Intractability: A Guide to the Theory of NP-completeness,*” W. H. Freeman, 1979.
- John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman, “*Introduction to Automata Theory, Languages, and Computation,*” 3rd edition, 2001.
- Sanjoy Dasgupta, Christos Papadimitriou, Umesh Vazirani, “*Algorithms,*” McGraw Hill, 2008.
- Scott Aaronson, “ $P \stackrel{?}{=} NP$,” Electronic Colloquium on Computational Complexity (ECCC), Report No. 4, Jan. 8, 2017. [<https://eccc.weizmann.ac.il/report/2017/004/>]

Assignment Project Exam Help
S
<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

1. Merlin in King Arthur's Court:

In the court of King Arthur there dwelt 150 knights and 150 ladies-in-waiting. The king decided to marry them off, but the trouble was that some pairs hated each other so much that they would not even get married, let alone speak! King Arthur tried several times to pair them off but each time he ran into conflicts. So he summoned Merlin the Wizard and ordered him to find a pairing in which every pair was willing to marry. Now Merlin had supernatural powers and he saw immediately that none of the $150!$ possible pairings was feasible, and this he told the king. But Merlin was not only a great wizard, but a suspicious character as well, and King Arthur did not quite trust him. “*Find a pairing or I shall sentence you to be imprisoned in a cave forever!*” said Arthur.

Fortunately for Merlin, he could use his supernatural powers to browse forthcoming scientific literature, and he found several papers in the early 20th century that gave the reason why such a pairing could not exist. He went back to the King when all the knights and ladies were present, and asked a certain 56 ladies to stand on one side of the king and 95 knights on the other side, and asked: “*Is any one of you* *l*ights?”, and when all said “*No!*”, Merlin said: “*O King, how* <https://eduassistpro.github.io/> *among the remaining 55 kn*ciously dismissed him. cation did include the pigeonhole principle, saw that in this case Merlin had spoken t

Some time elapsed and the king noticed that famous round table, neighbors often quarreled and digestion and so once again he summoned Merlin and ordered him to find a way to seat the 150 knights around the table so that each of them should sit between two friends. Again, using his supernatural powers Merlin saw immediately that none of the $150!$ seatings would do, and this he reported to the king. Again, the king bade him find one or explain why it was impossible. “*Oh I wish there were some simple reason I could give to you! With some luck there could be a knight having only one friend, and so you too could see immediately that what you demand from me is impossible. But alas!, there is no such simple reason here, and I cannot explain to you mortals why no such seating exists, unless you are ready to spend the rest of your life listening to my arguments!*” The king was naturally unwilling to do that and so Merlin has lived imprisoned in a cave ever since.

[Explain the above paragraphs in terms of known facts about two graph problems.]

2. Optimization versus Decision:

TSP: given a distance matrix and a budget K , is there a TSP tour of length $\leq K$.

TSP-OPT: given a distance matrix, find the shortest TSP tour.

Show that if TSP can be solved in polynomial time, then so can TSP-OPT.

3. Search versus Decision:

Suppose you have an algorithm that runs in polynomial time and answers whether a given graph has a Hamiltonian cycle. Show that you can use such an algorithm as a subroutine to develop a polynomial time algorithm that returns the actual Hamiltonian cycle if there exists one

Assignment Project Exam Help

4. Hamiltonian Path Pro <https://eduassistpro.github.io/>

spanning path, i.e., a pat

does it contain a simple
ce? Such a path consists of
edges; it can start at any node and can
node of G .

| V | – 1 edges; it can start at any node and can

Add WeChat edu_assist_pro

Show that the Hamiltonian Path Problem is NP-hard.

[Hint: reduce from the Hamiltonian Cycle Problem.]

5. Hamiltonian Path in a DAG:

Describe a polynomial-time algorithm that finds whether a given DAG contains a (directed) Hamiltonian path.

6. K-Spanning Tree Problem:

Does a given undirected graph $G = (V, E)$ have a spanning tree in which each node has degree $\leq K$?

For any fixed $K \geq 2$, show that the K-Spanning Tree Problem is NP-complete.

[Hint: start with $K=2$ and consider the relationship with the Hamiltonian Path Problem.]

7. Proving NP-completeness by generalization:

For each of the problems below, prove that it is NP-complete by showing that it is a **generalization** of some NP-complete problem we have seen in these Slides.

- (a) **Subgraph Isomorphism:** Given two undirected graphs G and H, is G a subgraph of H? (that is, by deleting some vertices and edges of H we obtain a graph that is , up to renaming of vertices, identical to G).
- (b) **Longest Path:** Given a graph G and an integer K, does G have a simple path of length at least K^2 ? **Assignment Project Exam Help**
- (c) **Dense Subgraph:** Given a graph G and two integers α and β , does G have a set of α vertices with at least β edges between them?
- (d) **Sparse Subgraph:** <https://eduassistpro.github.io/> and β , does G have a set of α vertices with at most β edges between the them?
- (e) **Reliable Network:** We are given two $n \times n$ distance matrix $D = (d_{ij})$ and a connectivity requirement matrix $R = (r_{ij})$, **Add WeChat edu_assist_pro** get K. We must find a graph $G = (V, E)$, with $V = \{1, 2, \dots, n\}$ such that
(1) the total cost of all edges in G (sum of d_{ij} , over all edges $(i, j) \in E$) is K or less, and
(2) between any two distinct vertices i and j there are at least r_{ij} vertex-disjoint paths.
[Hint: Suppose that all d_{ij} 's are 1 or 2, $K = n$, and r_{ij} 's are 2. Which well known NP-complete problem is this?]

8. In P or NP-complete?

Determine which of the following problems are NP-complete and which are solvable in polynomial time. In each problem you are given an undirected graph $G = (V, E)$ and:

- (a) A set of nodes $L \subseteq V$, and you must find a spanning tree such that its set of leaves includes the set L .
- (b) A set of nodes $L \subseteq V$, and you must find a spanning tree such that its set of leaves is precisely the set L .
- (c) A set of nodes $L \subseteq V$, and you must find a spanning tree such that its set of leaves is included in the set L .
- (d) An integer K , and y or fewer leaves.
- (e) An integer K , and y or more leaves.
- (f) An integer K , and you must find a spanning tree with exactly K leaves.

[Hint: All the NP-completeness proofs are by exception for one.]

9. Integer Linear Programming:

We have already shown that ILP is NP-hard by a reduction from 3SAT. To complete the proof that ILP is NP-complete, show that ILP is in NP. [Hint: Be careful! Given an instance of ILP feasibility, you must show that if it has a feasible solution, then it must have one that is only polynomially long in the length of the input instance, and hence, it can be written down in polynomial time.]

- 10. 2SAT:** Complete the details of the proof that $2SAT \in P$. In fact, show that 2SAT can be solved in linear time.
- 11. Tautologies:** Determine whether a given Boolean formula is a tautology. Show that this problem is co-NP-complete (i.e, is in co-NP and is NP-hard).
- 12. Max2SAT:** Given a 2SAT formula Φ and an integer K , is there a truth assignment that satisfies at least K clauses of Φ ?

Assignment Project Exam Help

Consider the following 10 clauses:

$(x), (y), (z), (w), (\neg x \neg w), (y \vee \neg w), (z \vee \neg w)$.

<https://eduassistpro.github.io/>

- (a) Show that if $(x \vee y \vee z)$ is false, then at least 10 clauses can be satisfied.
(b) Show that if $(x \vee y \vee z)$ is false, then at least 10 clauses can be satisfied.
(c) Use (a) and (b) and a reduction from 3SAT to show that Max2SAT is NP-complete.

Add WeChat [edu_assist_pro](#)

13. Boolean-SAT, CNF-SAT & DNF-SAT:

A boolean formula is in CNF (conjunctive normal form) if it is the conjunction (\wedge) of a number of clauses, where each clause is the disjunction (\vee) of some literals.

A boolean formula is in DNF (disjunctive normal form) if it is the disjunction (\vee) of a number of clauses, where each clause is the conjunction (\wedge) of some literals.

- (a) Show a polynomial-time reduction from Boolean-SAT to CNF-SAT that takes any boolean-formula Φ and converts it to a CNF-formula Ψ (with possible additional variables) such that Φ is satisfiable if and only if Ψ is satisfiable. [Hint: start with the parse tree of Φ , and convert each node to an equivalent collection of CNF clauses.]
- (b) We have shown that CNF-SAT is NP-complete. Show that DNF-SAT $\in P$.
- (c) Show a reduction from CNF-SAT to DNF-SAT. [Hint: Use the Truth Table Method.]
- (d) Do parts (b) and (c) and in P, and hence P=NP?

<https://eduassistpro.github.io/>

14. Restricted 3SAT-1:

Show that 3SAT remains NP-complete even to formulas in which each literal appears at most twice.

[Hint: replace all m appearances of a variable a by $a_1, a_2, a_3, \dots, a_m$; then add the following clauses: $(\bar{a}_1 \vee a_2) \wedge (\bar{a}_2 \vee a_3) \wedge \dots \wedge (\bar{a}_{m-1} \vee a_m) \wedge (\bar{a}_m \vee a_1)$.]

15. Restricted 3SAT-2:

Consider a special case of 3SAT in which all clauses have exactly 3 literals, and each variable and its negation together appear exactly 3 times. Show that this problem can be solved in polynomial time. [Hint: Create a bipartite graph with clauses on the left, variables on the right, and edges whenever a variable appears in a clause. Then use Hall's Matching Theorem (see exercises at the end of our Graph Slides).]

16. 3D Matching: given a set of n girls n boys and n pets, and a set of m girl-boy-pet triples, is there a subset of n triples in which every girl and boy and pet appear exactly once?

Show that 3D Matching is NP-complete. [Hint: reduce from 3SAT.]

17. The Hitting Set Problem:

We are given a set $X = \{x_1, x_2, \dots, x_n\}$, a collection $S = \{S_1, S_2, \dots, S_m\}$ of subsets of X , and a budget K.

A **hitting set** is any subset $H \subseteq X$ that intersects every S_t , i.e., $H \cap S_t \neq \emptyset$ for all $t = 1..m$.

Question: Given (X, S, K) , is there a hitting set H of size $\leq K$?

Show that the Hitting S

<https://eduassistpro.github.io/>

18. Clique + Independent Set: Prove that the following problem is NP-complete: Given an undirected graph G and an integer K, return a clique of size K as well as an independent set of size K, provided both exist.

19. The Maximum Common Subgraph Problem:

We are given two undirected graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, and a budget K. Determine whether there is a graph H with K nodes that is a subgraph of both G_1 and G_2 (up to renaming of nodes). Show that this problem is NP-complete.

20. In task scheduling, it is common to use a graph representation with a node for each task and a directed edge from task i to task j if i is a precondition to j . This directed graph depicts the precedence constraints in the scheduling problem. Clearly, a schedule is possible if and only if the graph is acyclic; if it isn't, we'd like to identify the smallest number of constraints that must be dropped so as to make it acyclic. Given a digraph G , a subset $E' \subseteq E(G)$ is called a **feedback arc set** if removal of edges E' renders G acyclic.

Feedback Arc Set (FAS): Given a digraph G and a budget K , find a feedback arc set of $\leq K$ edges, if one exists.

(a) Show that FAS is in NP

Assignment Project Exam Help

FAS can be shown to be

instance (G, K) of Vertex cover of size $\leq K$, we can

cover $\dots, v_n\}$, then $V(G') = \{w_1, w_2, \dots, w_n\} \cup \{v'_1, v'_2, \dots, v'_n\}$, where w_i and v'_i are the corresponding vertices in G' .

Vertex Cover. Given an undirected graph and we want a vertex cover of size $\leq K$ as follows. If $V(G) = \{v_1, v_2, \dots, v_n\}$ and $E(G)$ consists of the

following $n + 2|E(G)|$ directed edges:

(i) (w_i, w'_i) for all $i = 1..n$,

(ii) (w'_i, w_j) and (w'_j, w_i) for every $(v_i, v_j) \in E(G)$.

(b) Show that if G contains a vertex cover of size K , then G' contains a feedback arc set of size K .

(c) Show that if G' contains a feedback arc set of size K , then G contains a vertex cover of size (at most) K . [Hint: Given a feedback arc set of size K in G' , you may need to first modify it slightly to obtain another one which is of a more convenient form, but is of same size or smaller. Then argue that G must contain a vertex cover of the same size as the modified feedback arc set.]

21. [CLRS, Problem 34-4, page 1104] Scheduling with Profits and Deadlines:

Suppose you have one machine and a set of n jobs J_1, J_2, \dots, J_n . Each job J_k has a processing time t_k , a profit p_k , and a deadline d_k . The machine can process only one job at a time, and job J_k must run uninterruptedly for t_k consecutive time units. If you complete job J_k by its deadline d_k , you receive a profit p_k , but if you complete it after its deadline, you receive no profit. As an optimization problem, you are given the processing times, profits, and deadlines for a set of n jobs, and you wish to find a schedule that completes all the jobs and returns the greatest amount of profit.

Assignment Project Exam Help

- (a) State this problem as a decision problem.
- (b) Show that the decision problem is NP-hard. <https://eduassistpro.github.io/>
- (c) Give a polynomial-time algorithm for the optimization problem, assuming that all processing times are integers in the range $[1..n]$.
[Hint: Use dynamic programming.]
- (d) Give a polynomial-time algorithm for the optimization problem, assuming that all processing times are integers in the range $[1..n]$.

22. Node-Disjoint Paths Problem: Given an undirected graph in which some vertices have been specially marked: a certain number of “sources” s_1, s_2, \dots, s_k and an equal number of “destinations” t_1, t_2, \dots, t_k . The goal is to find k node-disjoint paths (that is, paths that have no nodes in common) where the i^{th} path goes from s_i to t_i .

Show that this problem is NP-complete.

[Hints: Here is a sequence of progressively stronger hints.

- (i) Reduce from 3SAT.
- (ii) For a 3SAT formula with m clauses and n variables, use $k = m + n$ sources (and destinations). Introduce one source/destination pair (s_x, t_x) for each variable x , and one source/destination pair $(s_{\bar{x}}, t_{\bar{x}})$ for each literal occurring in that clause. [Hint: one for each clause, one for each literal occurring in that clause.]
- (iii) Note that if the path from s_x to t_x goes through an intermediate vertex representing, say, an occurrence of variable x , then no other path can go through that vertex. What vertex would you like the other path to be forced to go through instead?]

23. The Set Partitioning Problem: given a set S of n integers, can S be partitioned into two subsets A and B = S – A such that the sum of integers in A is equal to the sum of integers in B? Show that this problem is NP-complete.

24. The Knapsack Problem: Given a set of n pairs of positive integers $S = \{(v_i, w_i) \mid i = 1..n\}$, a target value V and a knapsack capacity W (also both positive integers), is there a subset C of S such that $\sum\{v_i \mid (v_i, w_i) \in C\} \geq V$ and $\sum\{w_i \mid (v_i, w_i) \in C\} \leq W$? Show that this problem is NP-complete.

25. DNA Sequencing [Assignment Project Exam Help](https://eduassistpro.github.io/)
 By Hybridization One experimental procedure for identifying a new DNA sequence repeated length k) it contains. Based on the above information, we can say that if a string x is a k-mer, then it must contain all the k-mers (contiguous substrings of length k) it contains. Based on the above information, we can say that if a string x is a k-mer, then it must contain all the k-mers (contiguous substrings of length k) it contains.

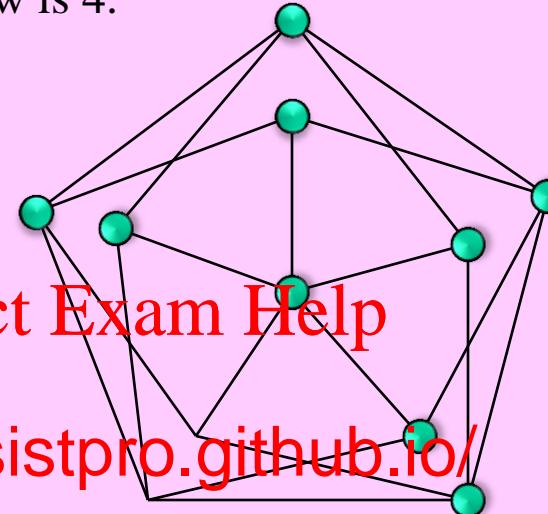
Let's now formulate the reconstruction problem. Let $\Gamma(x)$ denote the multiset of k-mers (contiguous substrings of length k) contained in a string x. For any string x (the DNA sequence), let $\Gamma(x)$ denote the multiset of k-mers (contiguous substrings of length k) contained in x. Then $\Gamma(x)$ contains exactly $|x| - k + 1$ elements.

The reconstruction problem is now easier. Given a multiset of k-length strings, find a string x such that $\Gamma(x)$ is exactly this multiset.

- (a) Show that the reconstruction problem *reduces to* the Hamiltonian Path Problem.
[Hint: Construct a digraph with one node for each k-mer, and with an edge from a to b if the last $k - 1$ characters of a match the first $k - 1$ characters of b.]
- (b) But in fact, there is much better news. Show that the same problem also *reduces to* the Eulerian Path Problem. [Hint: This time, use one directed edge for each k-mer.]

26. Chromatic Number of a graph:

The chromatic number of a graph is the minimum number of colors needed to color the vertices of the graph so that no pair of adjacent vertices are colored the same.
Show the chromatic number of the graph below is 4.



Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

27. Triangle-free graphs with high chromatic

Show that there are triangle-free graphs (i.e., with no clique subgraph of size 3) that have arbitrarily large chromatic number.

[Hint: Look at the graph above. We started with the pentagon, then doubled up the vertex set, then added the central vertex. Using that pattern, recursively build up larger triangle-free graphs with increasing chromatic numbers.]

28. Degree-Restricted 3-Colorability: Show that the 3-colorability problem remains NP-complete even if we restrict it to graphs of maximum vertex degree 4.

[Hint: Show that the “5-star” graph shown in Fig (a) below is 3 colorable, and in any valid 3 coloring of that graph all 5 vertices u_1, \dots, u_5 have the same color. Analogously define a “d-star” for any integer $d > 4$.]

29. Planar 3-Colorability: Show that the 3-colorability problem remains NP-complete even if we restrict it to planar graphs of maximum vertex degree 4.

[Hint: Show that the graph in Fig (b) below is 3 colorable. Furthermore, any coloring of the vertices $\{u, u', v, v'\}$ can be extended to a 3 coloring of that graph if and only if vertices u and u' have the same color.]

Combine this with the p <https://eduassistpro.github.io/>

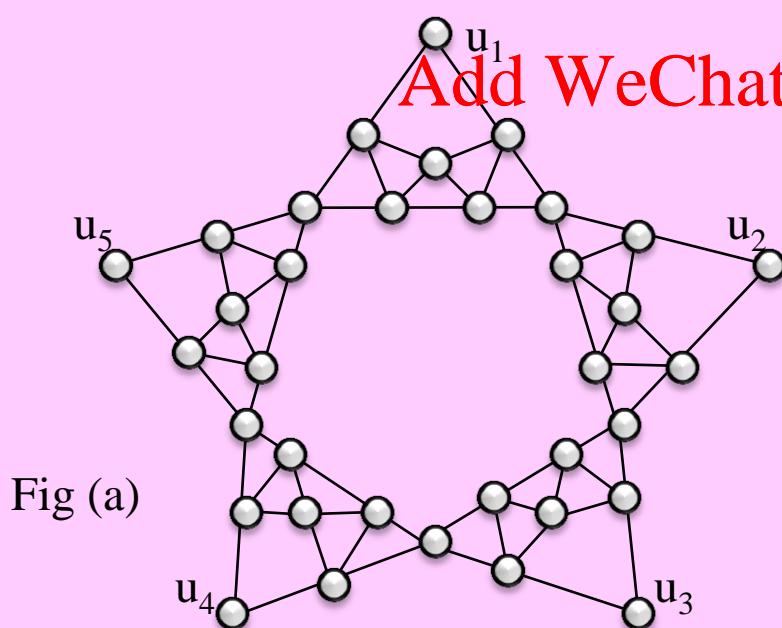


Fig (a)

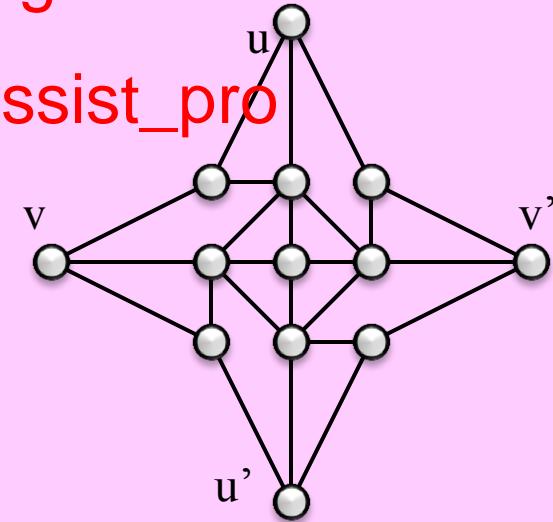


Fig (b)

30. 3-Colorability of circle intersections: Consider n circles in the plane such that no 2 of them are tangent and no 3 of them pass through the same point. Furthermore, assume each circle intersects at least one other circle.

Now consider the planar (multi-) graph whose vertices are the circle intersection points, and whose edges are the arcs of the circles whose both end points are vertices and do not contain any other circle intersection point in their relative interior.
(The figure below shows an example.)

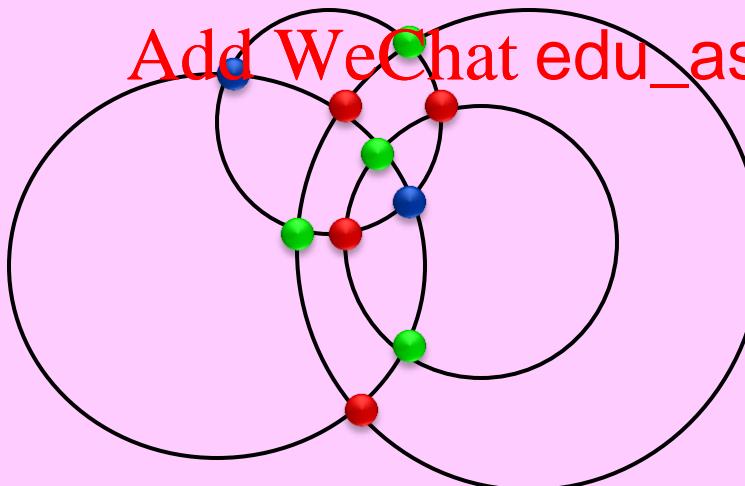
Assignment Project Exam Help

Is it true that all such graphs are 3 colorable?

If yes prove it, if not giv

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro



Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro