Recall

Summary of Lecture 18

We will study which problems (seemingly) cannot be solved in polynomial time.

P = the class of decision problems that have polynomial time algorithms

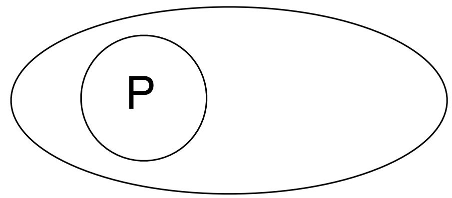
Assignment Project Exam Help

$X \leq_P Y$, for problems X, Y, "X reduces to Y in polynomial time", means: we can use a polynomial time algorithm for Y to make a polynomial time algorithm for X.

https://powcoder.com

Add WeChat powcoder

The class NP



A few decision problems in NP:

- Hamiltonian path/cycle
- Travelling Salesman Problem
- Independent Set

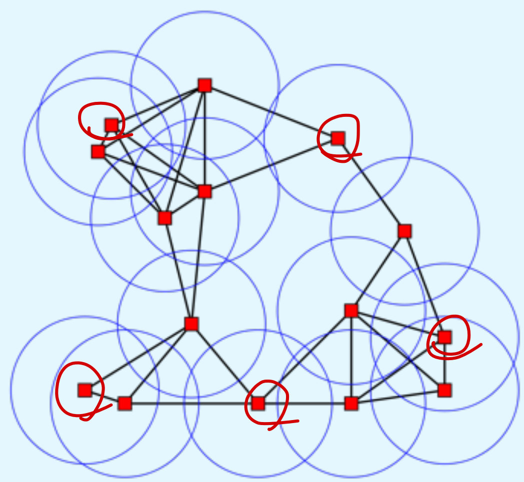
Assignment Project Exam Help

Common feature: if the answer is YES, then there is some succinct information (a ***certificate***) to ***verify*** that the answer is YES.

https://powcoder.com

Example: **Independent Set.** Given graph G, and number k, does G have an independent set of size $\geq k$

Add WeChat powcoder



How can I convince you that Yes, there is an independent set of size ≥ 5?

How can I convince you that No, there is no independent set of size ≥ 7?

A ***verification algorithm*** takes input + certificate and checks it. Formally:

Definition. Algorithm A is a ***verification algorithm*** for the decision problem X if

- A takes two inputs *x*, *y* and outputs YES or NO
- for every input *x* for problem X, *x* is a YES for X iff there exists a *y* (a ***certificate***) such that A(*x*,*y*) outputs YES

Assignment Project Exam Help

Furthermore, A is a ***polynomial time verification algorithm*** if

https://powcoder.com

- A runs in polynomial time
- there is a polynomial bound on the size of the certificate *y*

Add WeChat powcoder

We say X "can be verified in polynomial time" if there is a poly time verification algorithm for X.

Definition.
NP = the class of decision problems that can be verified in polynomial time

NP = Non-deterministic Polynomial time — because the certificate is like a non-deterministic guess

CS 360 covers non-deterministic Turing machines

Examples

Subset Sum ∈ NP

Given numbers $w_1, \ldots, w_n, W$ is there a subset $S \subseteq \{1, \ldots, n\}$

such that $\sum_{i \in S} w_i = W$

Certificate: the set S

Assignment Project Exam Help

Verification: check that $\sum_{i \in S} w_i = W$

https://powcoder.com

This takes poly. time.

TSP (decision version) ∈ NP

Add WeChat powcoder

Given a graph G, weights on edges, number k, does G have a TSP tour of length ≤ k

Certificate: a permutation of the vertices

Verification: check it's a permutation, check edges exist to make a cycle, check sum of weights of edges in cycle is $\leq k$. This takes poly. time.

Examples that don't seem to be in NP

Unique Subset Sum

Given numbers $w_1, \ldots, w_n, W$ is there a *unique* subset $S \subseteq \{1, \ldots, n\}$

such that $\sum_{i \in S} w_i = W$

Assignment Project Exam Help

You can verify that a given S is a solution.

But how can you verify that S is the only solution.

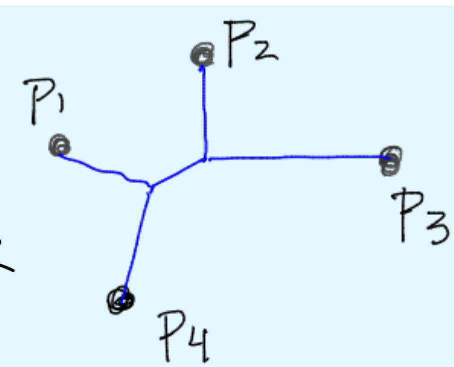
https://powcoder.com

Steiner tree in the plane Add WeChat powcoder

Given points in the plane, can you connect them (using extra points) with a tree of Euclidean length ≤ k

Two difficulties

- the coordinates of the extra points (are they rational?)
- checking sum of Euclidean lengths ≤ k is not known in poly. time because of √.



Claim. P $\subseteq$ NP, i.e. if X is in P then X is in NP.
Proof. The certificate is empty and the verification algorithm is just the poly time algorithm for X.

Definition.
coNP = the class of decision problems where the NO instances can be verified in polynomial time.

Assignment Project Exam Help

Example. **Primes**: Given a number n, is it prime?

https://powcoder.com

Primes $\in$ coNP

Add WeChat powcoder

to verify that n is NOT prime, the certificate is numbers $a, b \in \mathbb{N}$ $a, b \geq 2$ and verify $a \cdot b = n$

In fact, Primes $\in$ P. A poly time algorithm was found in 2002.
W https://en.wikipedia.org/wiki/AKS_primality_test

OPEN QUESTIONS

1. P =? NP worth $1 million (Millenium Prize) W https://en.wikipedia.org/wiki/P_versus_NP_problem

2. NP =? coNP

3. P =? NP ∩ coNP

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

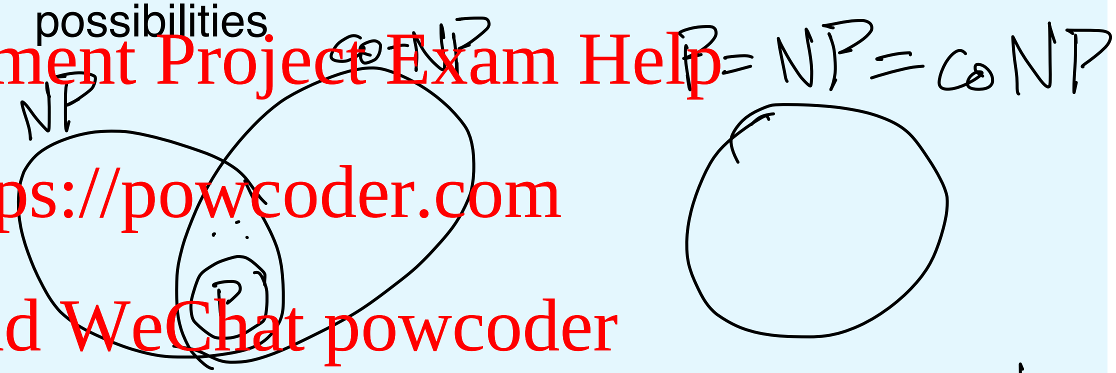
OPEN QUESTIONS

1. P =? NP worth \$1 million (Millenium Prize) W https://en.wikipedia.org/wiki/P_versus_NP_problem

2. NP =? coNP

3. P =? NP ∩ coNP

possibilities



we don't know which!

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Properties

1. P ⊆ NP, P ⊆ coNP

2. Any problem in NP can be solved in time $O(2^{n^t})$ by trying all certificates one by one

Summary of Lecture 19, Part 1

classes NP, coNP

What you should know from Lecture 19, Part 1:

- how to prove that a problem is in NP (certificate, verification)

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

Next:

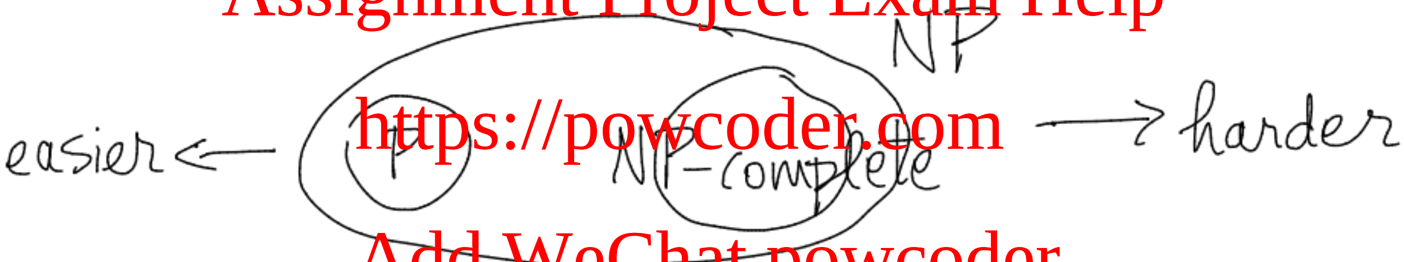
- NP-complete problems

Definition. A decision problem X is ***NP-complete*** if

- $X \in NP$
- for every Y in NP, $Y \leq_P X$

i.e. X is [one of] the hardest problem in NP.

Assignment Project Exam Help



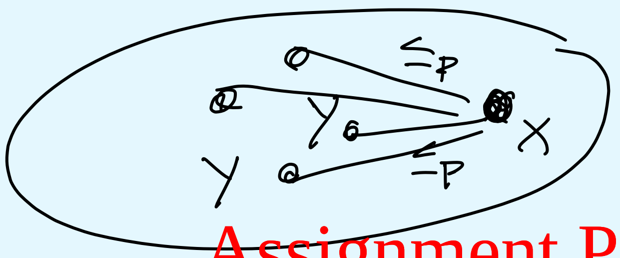
https://powcoder.com

Add WeChat powcoder

Two important implications of X being NP-complete

- if X can be solved in polynomial time then so can every problem in NP
  (if $X \in P$ then $P = NP$)
- if X cannot be solved in polynomial time then no NP-complete problem can be solved in polynomial time
- if $X \in$ co-NP then NP = coNP (this needs proof)

The first NP-completeness proof is difficult — must show that *every* problem $Y \in NP$ reduces to X



Assignment Project Exam Help

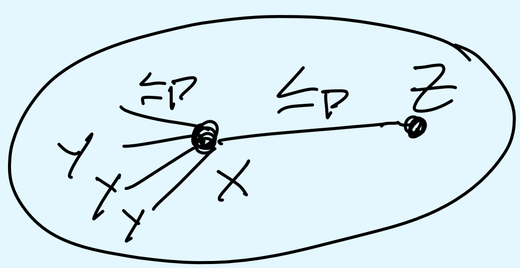
Subsequent NP-completeness proofs are easier because $\leq_P$ is *transitive*:

https://powcoder.com

Claim. If $Y \leq_P X$ and $X \leq_P Z$ then $Y \leq_P Z$

Add WeChat powcoder

So to prove Z is NP-complete, we just need to prove $X \leq_P Z$ where X is a known NP-complete problem.



Summary: to prove a decision problem Z is NP-complete

1. prove Z in NP
2. prove X $\leq_P$ Z for some known NP-complete problem X.

Our first NP-complete problem: Circuit Satisfiability
[definition and proof later]

Assignment Project Exam Help

second NP-complete problem: Satisfiability
[proof later, but definition now]

https://powcoder.com

Add WeChat powcoder

Satisfiability (SAT)
Input: a Boolean formula made of Boolean variables, and logical operands $\wedge$ "and", $\vee$ "or", $\neg$ "not"

e.g. $\neg(x_1 \wedge x_2) \vee (x_3 \wedge (x_5 \vee \neg x_4))$

Question: Is there an assignment of True/False to the variables to make the formula True?

e.g. $x_1 =$ False and others arbitrary makes the above formula True

Exercise. Prove that Satisfiability is in NP.

SAT is NP-complete, even the special case of "CNF" — Conjunctive Normal Form

Definition of CNF

formula is $\wedge$ of *clauses*; clause is $\vee$ of *literals*; literal is x or $\neg x$

$$\underbrace{(x_1 \vee \neg x_2 \vee x_3)}_{\text{clause}} \wedge (\neg x_1 \vee x_4) \wedge (x_3 \vee x_4 \vee \neg x_5)$$

literals

Assignment Project Exam Help

In fact, SAT is still NP-complete when all clauses have 3 literals — this is called 3-SAT

https://powcoder.com

Add WeChat powcoder

3-SAT
Input: A Boolean formula that is an $\wedge$ of clauses, each clause an $\vee$ of 3 literals, each literal a variable or negation of a variable.
Question: Is there an assignment of True/False to the variables to make the formula True?

Theorem. 3-SAT is NP-complete [proof later]

but 2-SAT is in P

There is a linear time algorithm for 2-SAT that uses strong connectivity of a directed graph.

W https://en.wikipedia.org/wiki/2-satisfiability

Summary of Lecture 19, Part 2

definition of NP-complete, the first NP-complete problems: SAT, 3-SAT

What you should know from Lecture 19, Part 2:

- what are the two steps to proving a problem is NP-complete

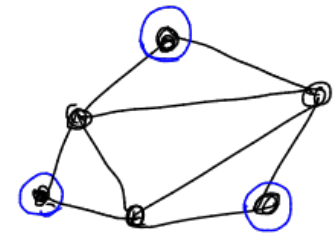
Assignment Project Exam Help

Next:

https://powcoder.com

- examples of NP-completeness proofs

Add WeChat powcoder



Independent Set
Input: Graph G = (V,E), number k.
Question: Does G have an independent set of size ≥ k?

Theorem. Independent Set is NP-complete.
Proof.

1. Independent Set is in NP — we already saw the idea of this in Part 1.

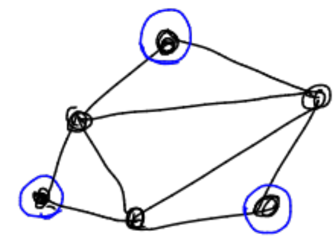
Assignment Project Exam Help

2. 3-SAT $\leq_P$ Independent Set

https://powcoder.com

Add WeChat powcoder

Independent Set
Input: Graph G = (V,E), number k.
Question: Does G have an independent set of size ≥ k?



Theorem. Independent Set is NP-complete.
Proof.

1. Independent Set is in NP — we already saw the idea of this in Part 1.

Assignment Project Exam Help

2. 3-SAT $\leq_P$ Independent Set

https://powcoder.com

Suppose we have a polynomial time algorithm for Independent Set.
Give a polynomial time algorithm for 3-SAT.

Add WeChat powcoder

Input: A 3-SAT formula F with clauses $C_1 \ldots C_m$ on variables $x_1 \ldots x_n$
Output: Is F satisfiable?

Idea: - construct a graph G and choose a number k such that
G has an independent set of size $\geq k$ iff F is satisfiable ★
- run the Independent Set algorithm on G, k
- return its answer

This is a ***many-one*** ("one-shot") reduction. To prove correctness, just prove ★

To prove poly time, just prove that G can be constructed in poly time (in size of F).

Proof. continued

Input: A 3-SAT formula F with clauses $C_1 \ldots C_m$ on variables $x_1 \ldots x_n$

Output: Is F satisfiable?

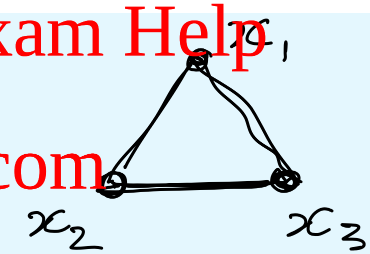
Idea: - construct a graph G and choose a number k such that

G has an independent set of size $\geq k$ iff F is satisfiable

- run the Independent Set algorithm on G, k
- return its answer

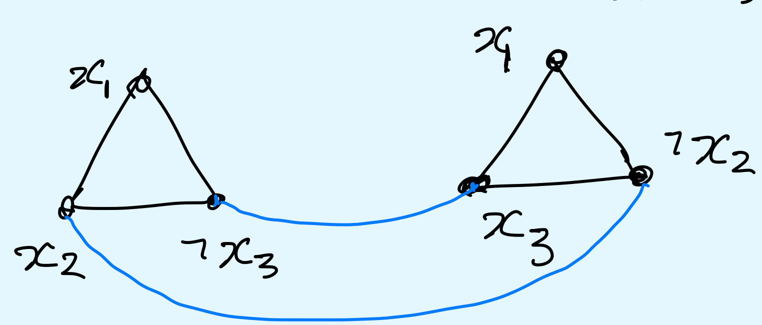
$(x_1 \vee x_2 \vee x_3) \implies$

choose at least one to be true



choose one vertex for the independent set.

$(x_1 \vee x_2 \vee \neg x_3) \wedge (x_1 \vee \neg x_2 \vee x_3)$



blue edges prevent choosing x_i and $\neg x_i$

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Proof. continued

Input: A 3-SAT formula F with clauses $C_1 \ldots C_m$ on variables $x_1 \ldots x_n$

Output: Is F satisfiable?

Idea: - construct a graph G and choose a number k such that

G has an independent set of size $\geq k$ iff F is satisfiable

- run the Independent Set algorithm on G, k
- return its answer

Construction:

Assignment Project Exam Help

- For each clause C_i with literals l_1, l_2, l_3, make 3 vertices joined by 3 edges
- if two literals are opposite, join them with an edge.
- $k := m$

https://powcoder.com

Add WeChat powcoder

Runtime: Prove that G can be constructed in poly time (in the size of F).
G has $3m$ vertices and can be constructed in time polynomial in m and n

Correctness: prove G has an independent set of size $\geq k$ iff F is satisfiable

- if F is satisfiable then each clause has (at least) one True literal. Choose the corresponding m vertices of G. They are independent.

- if G has an independent set of size $\geq m$ there must be one in each triangle. Set the corresponding literals True. This is valid, and satisfies F.

This completes the proof that Independent Set is NP-complete.

Recall

Definition. Problem X ***reduces to*** problem Y, written X ≤ Y, if an algorithm for Y can be used to make an algorithm for X.

Definition. A ***many one reduction*** X ≤ Y uses the algorithm for Y once and outputs its answer.

mnemonic: many-one = “one-shot” “many-one” is a standard name; one-shot is not

Assignment Project Exam Help

The form of a polynomial time many-one reduction $X \leq_P Y$:

https://powcoder.com

Assume we have an algorithm A for Y

Algorithm for X:

Add WeChat powcoder

- take input x and construct an input y for problem Y
- run A on y
- return the answer

For correctness we just need to prove:

the answer for x is YES iff the answer for y is YES

For poly time we just need to prove:

the construction of y takes polynomial time.

How to prove that a decision problem Z is NP-complete

1. prove Z in NP
2. prove X $\leq_P$ Z for some known NP-complete problem X.
   Use a *many-one* reduction.

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

Summary of Lecture 19

definition of NP-complete, first NP-completeness proofs

What you should know from Lecture 19 (and Lecture 20)

- how to prove a problem is NP-complete using a polynomial time many-one reduction

Assignment Project Exam Help

https://powcoder.com

Next:

- more examples of NP-completeness proofs

Add WeChat powcoder