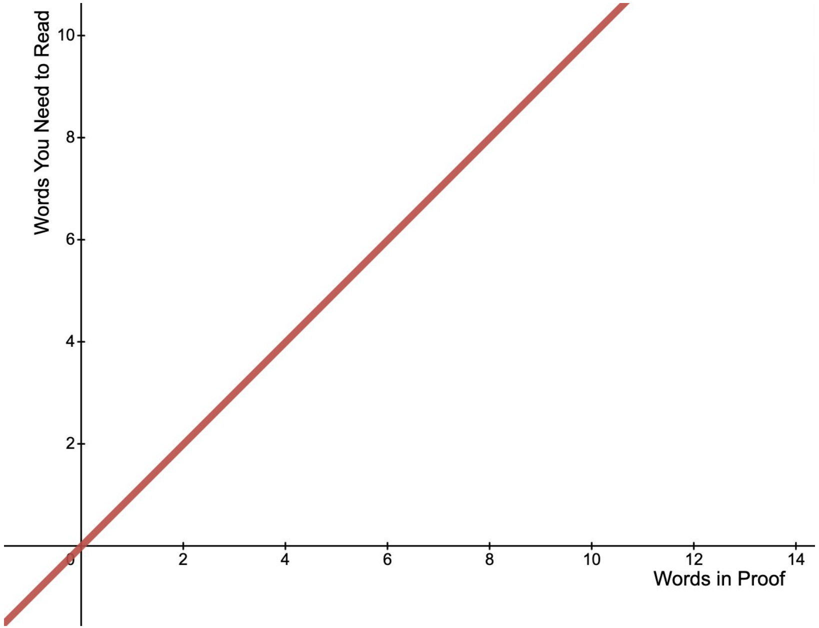
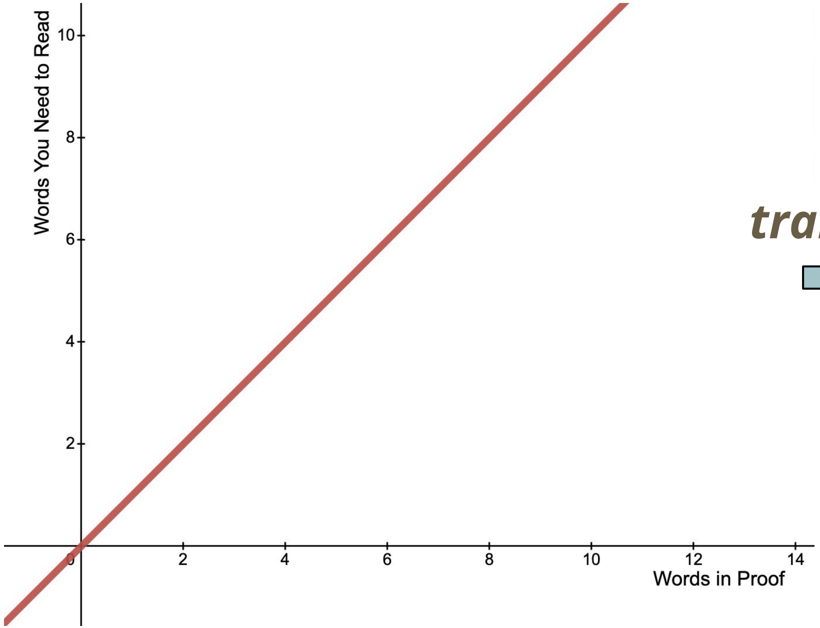
# Classical and Quantum Probabilistically Checkable Proofs

Jon Rosario and Laker Newhouse
Mentored by David Cui

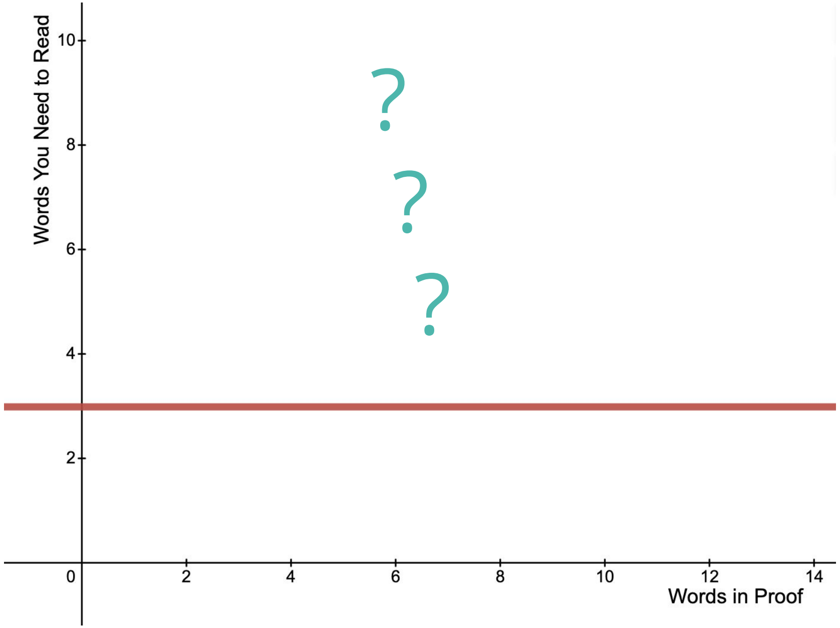
# *Breaking News:* Researchers Discover Proofs That Are Faster To Read



# *Breaking News:* Researchers Discover Proofs That Are Faster To Read




***transform***



# Quick Preliminaries: The Class P

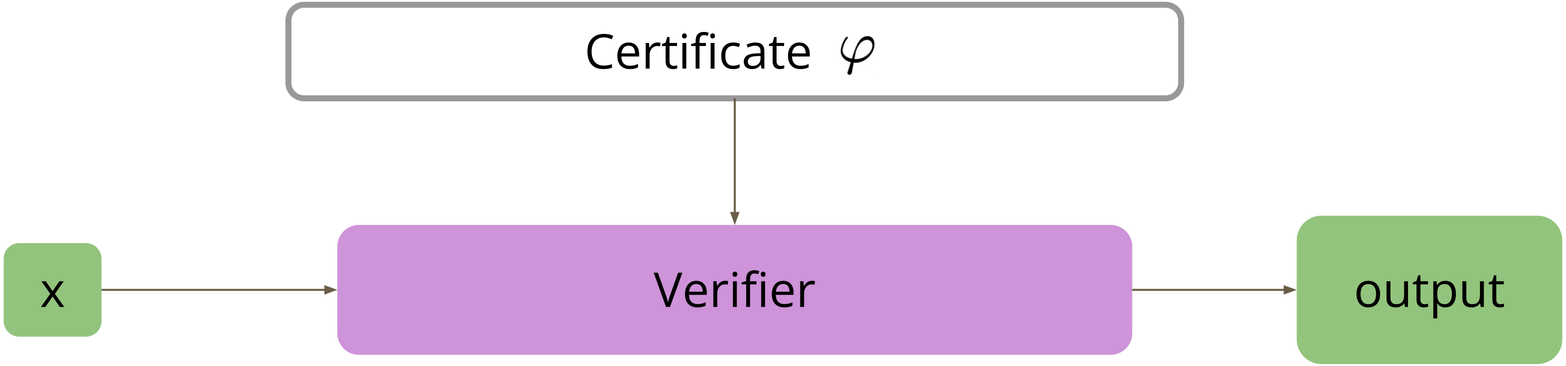
Verifier

# Quick Preliminaries: The Class P

# Quick Preliminaries: The Class P

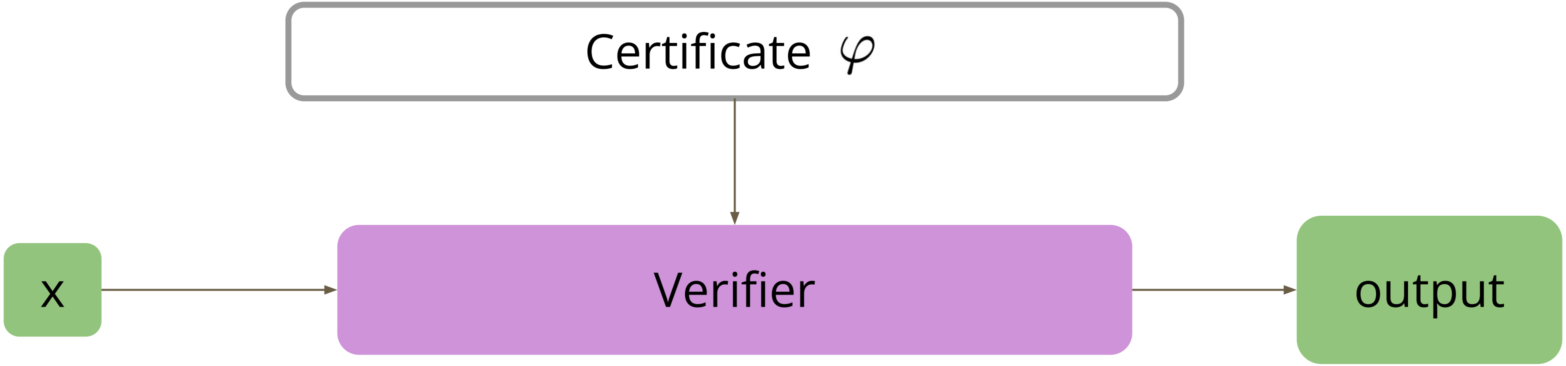

- $x \in L \implies V(x) = 1$
- $x \notin L \implies V(x) = 0$

# Quick Preliminaries: The Class NP (Nifty Proofs)



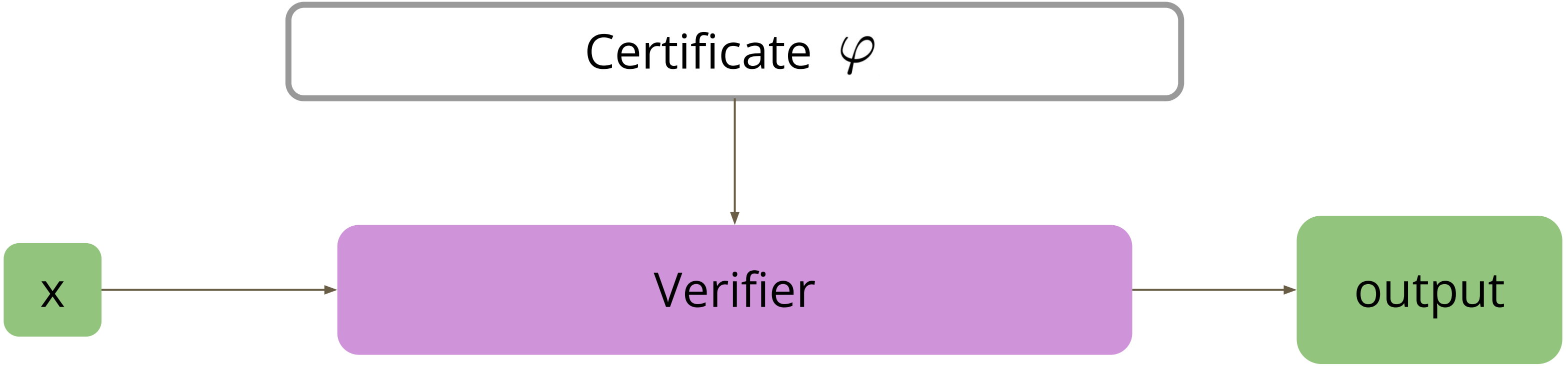

- $x \in L \implies V(x) = 1$
- $x \notin L \implies V(x) = 0$

# Quick Preliminaries: The Class NP (Nifty Proofs)




- $x \in L \implies \exists \varphi$ such that $V(x, \varphi) = 1$
- $x \notin L \implies \forall \varphi$, we have $V(x, \varphi) = 0$

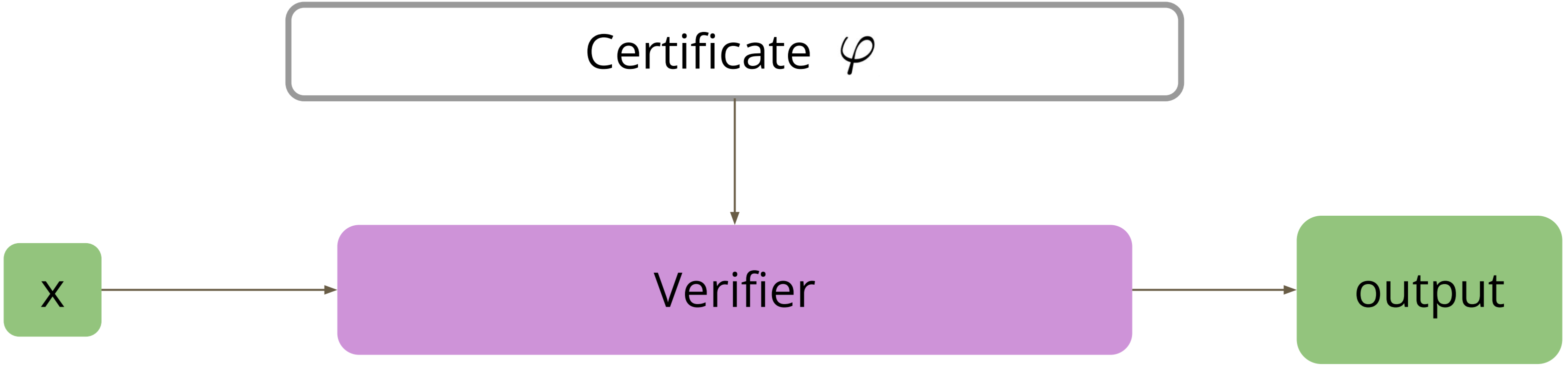
# Quick Preliminaries: The Class NP (Nifty Proofs)




- $x \in L \implies \exists \varphi$ such that $V(x, \varphi) = 1$
- $x \notin L \implies \forall \varphi$, we have $V(x, \varphi) = 0$

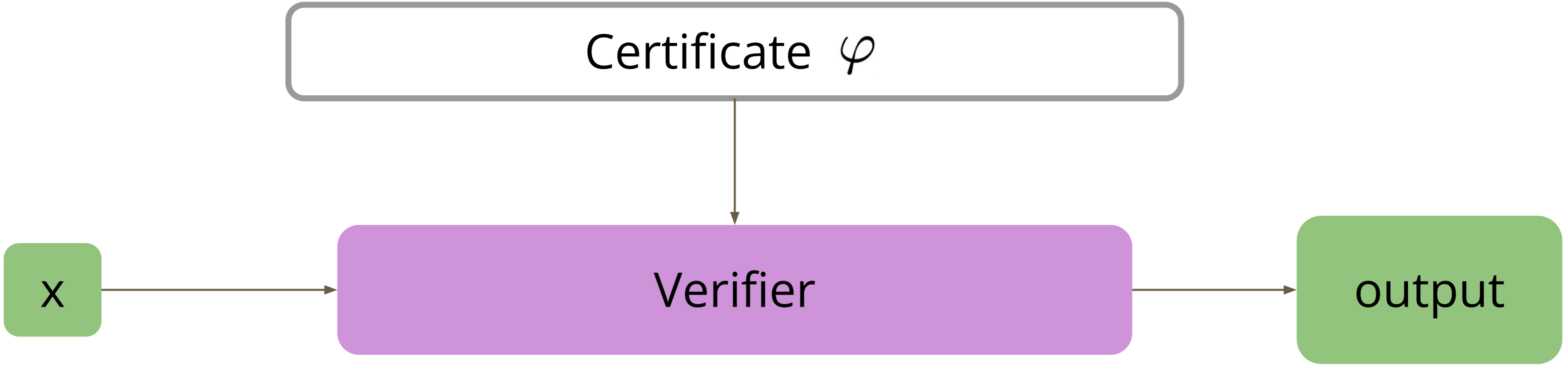
(At least one good certificate works)

# Quick Preliminaries: The Class NP (Nifty Proofs)



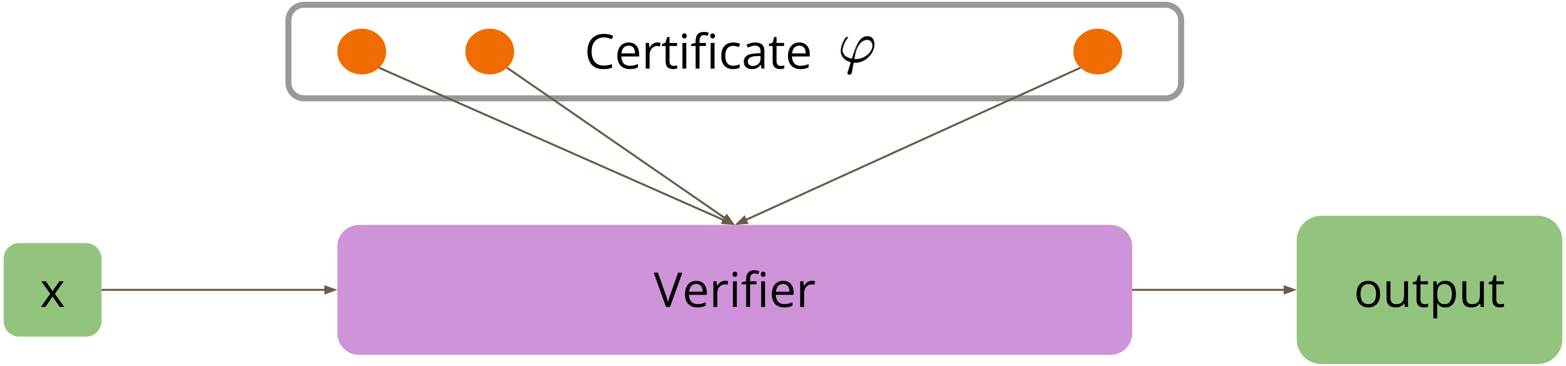

- $x \in L \implies \exists \varphi$ such that $V(x, \varphi) = 1$ (At least one good certificate works)
- $x \notin L \implies \forall \varphi$, we have $V(x, \varphi) = 0$ (All false certificates fail)

# Natural Extension #1: Probabilistic Verifier



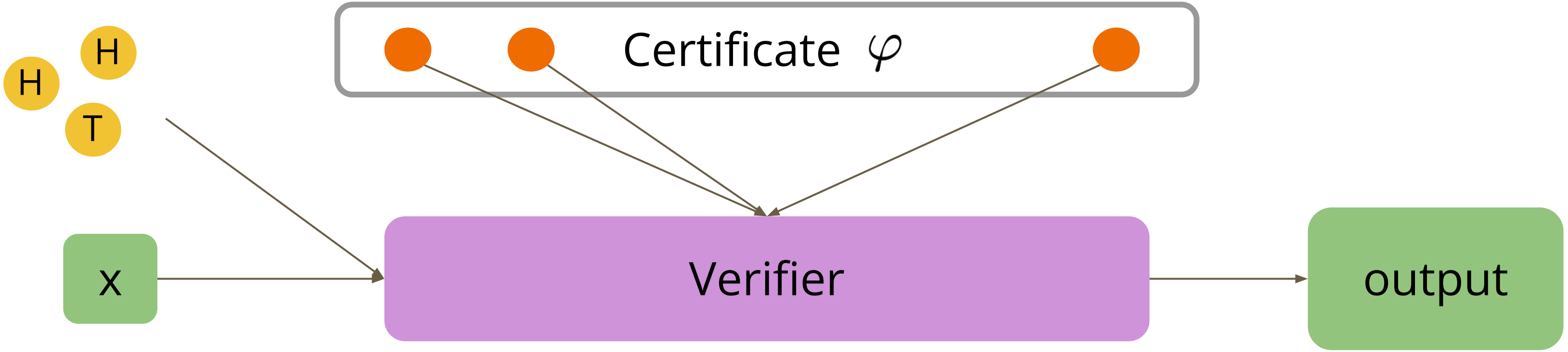

- $x \in L \implies \exists \varphi$ such that $P[V(x,\varphi) = 1] = 1$ (At least one good certificate works)
- $x \notin L \implies \forall \varphi$, we have $P[V(x,\varphi) = 1] \leq 1/2$ (All false certificates probably fail)

# Natural Extension #2: Bounded Queries




- $x \in L \implies \exists \varphi$ such that $P[V(x, \varphi) = 1] = 1$ (At least one good certificate works)
- $x \notin L \implies \forall \varphi$, we have $P[V(x, \varphi) = 1] \leq 1/2$ (All false certificates probably fail)

# Natural Extension #3: Bounded Randomness



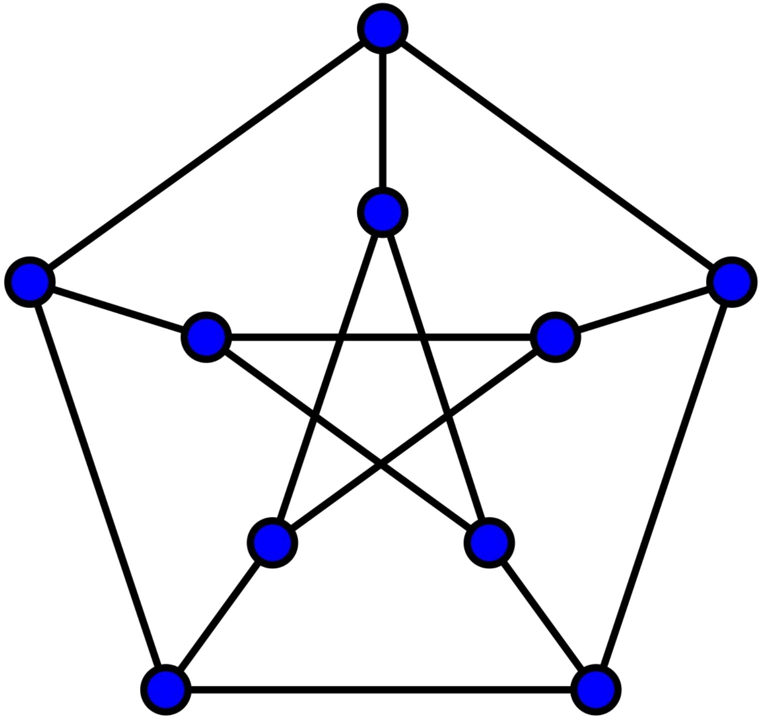

- $x \in L \implies \exists \varphi$ such that $P[V(x, \varphi) = 1] = 1$ (At least one good certificate works)
- $x \notin L \implies \forall \varphi$, we have $P[V(x, \varphi) = 1] \leq 1/2$ (All false certificates probably fail)

# The Theorem That Rocked The ‘90s

$$\mathsf{NP} = \mathsf{PCP}(O(\log n), O(1))$$

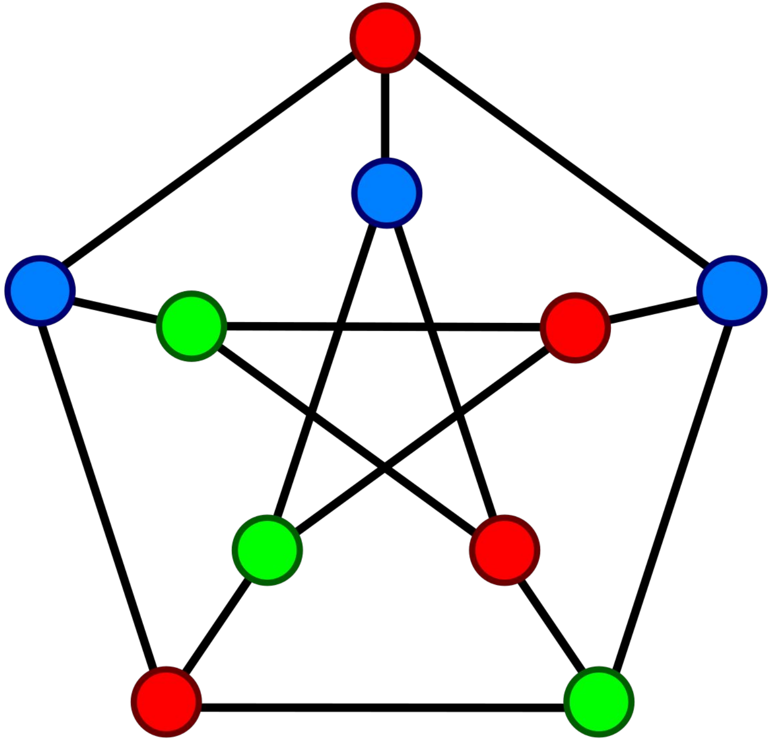
“randomness” “queries”

# Example: Graph Coloring



# Example: Graph Coloring

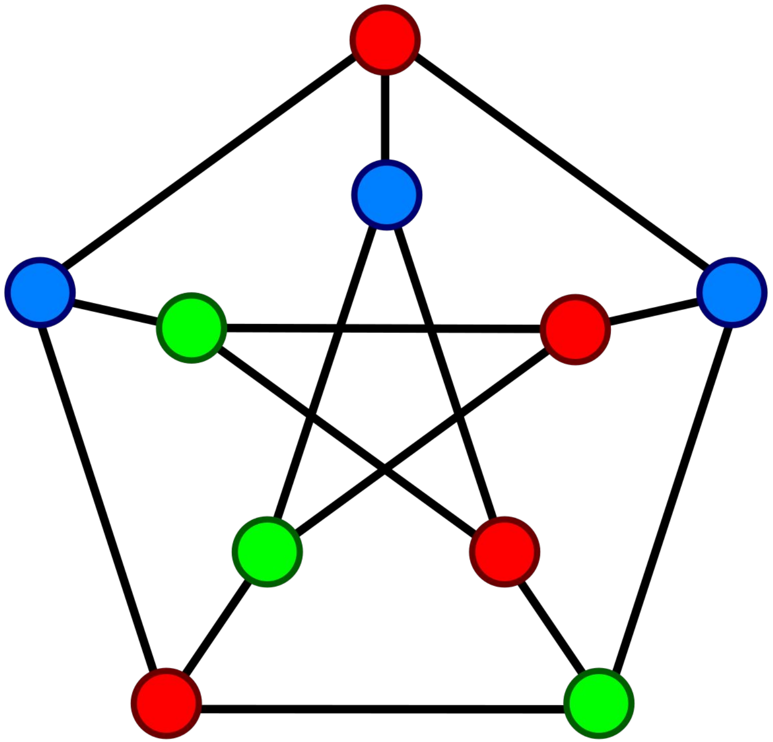
In NP: certificate is a color for each vertex



# Example: Graph Coloring

In NP: certificate is a color for each vertex

Probabilistic verifier: check just one edge

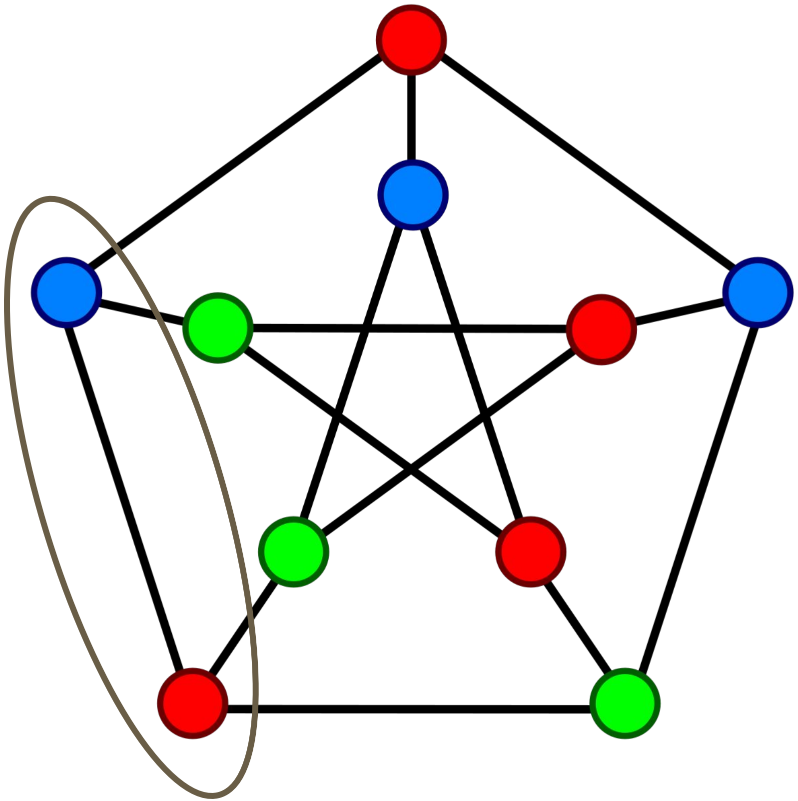


# Example: Graph Coloring

In NP: certificate is a color for each vertex

Probabilistic verifier: check just one edge

This edge



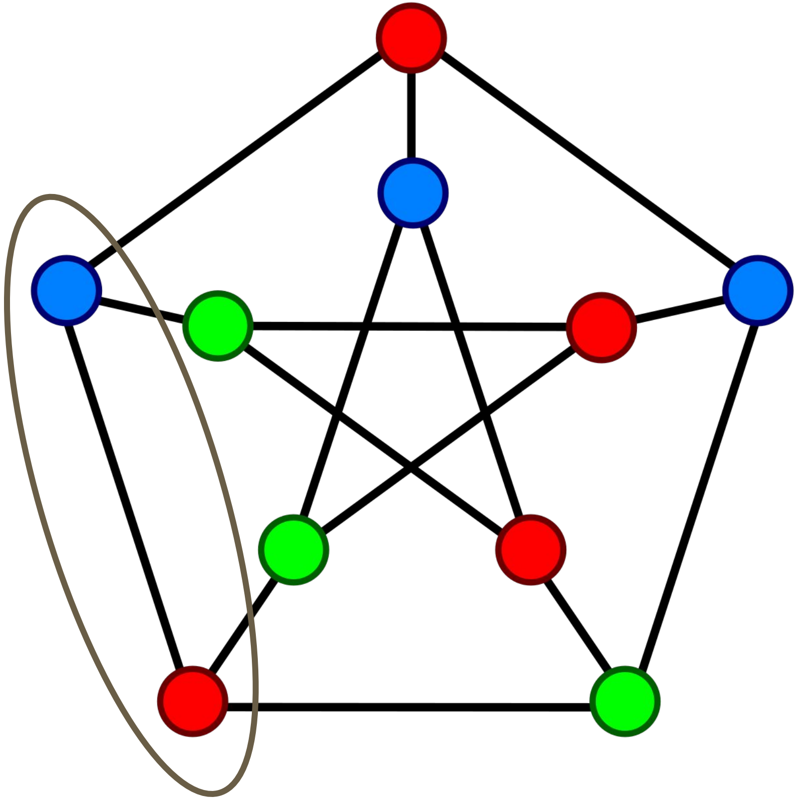
# Example: Graph Coloring

- $x \in L \implies \exists \varphi$ such that $P[V(x,\varphi) = 1] = 1$
- $x \notin L \implies \forall \varphi$, we have $P[V(x,\varphi) = 1] \leq 1/2$

In NP: certificate is a color for each vertex

Probabilistic verifier: check just one edge

This edge



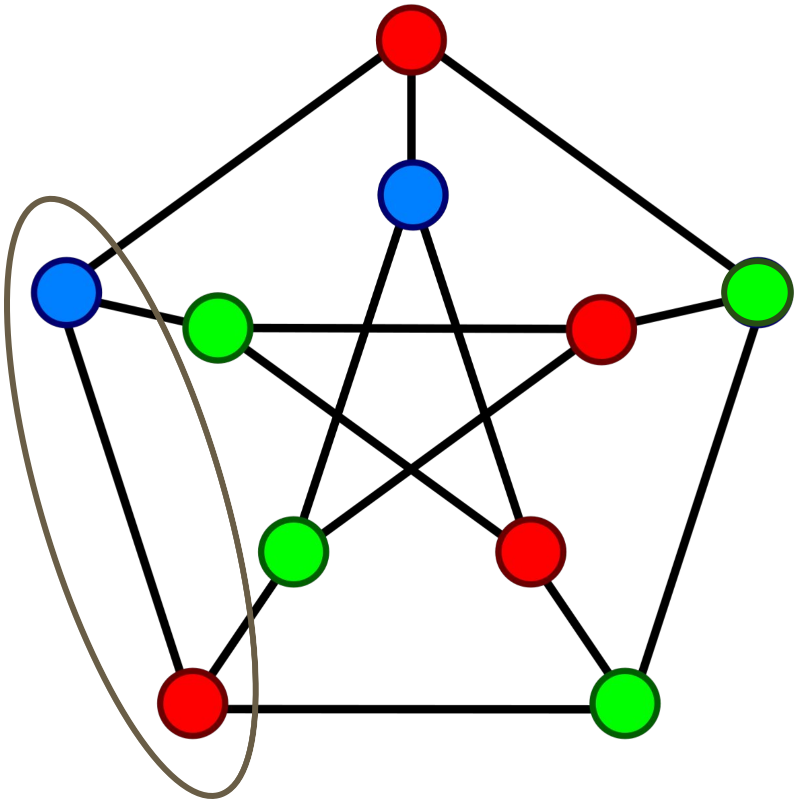
# Example: Graph Coloring

- $x \in L \implies \exists \varphi$ such that $P[V(x, \varphi) = 1] = 1$
- $x \notin L \implies \forall \varphi$, we have $P[V(x, \varphi) = 1] \leq 1/2$

In NP: certificate is a color for each vertex

Probabilistic verifier: check just one edge

This edge



What if just one rogue vertex wrong?

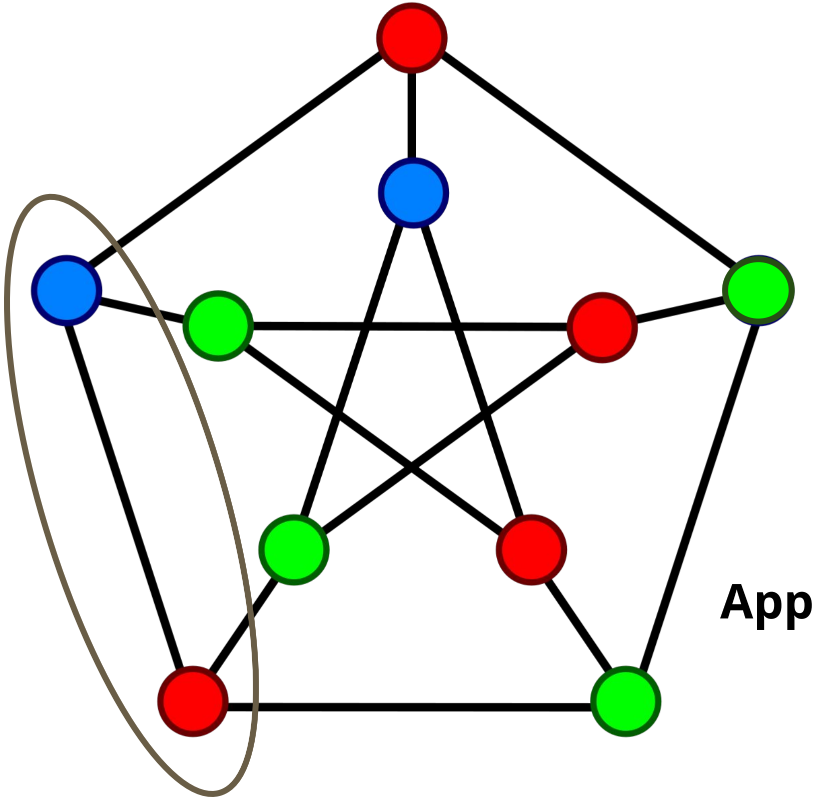
# Example: Graph Coloring

- $x \in L \implies \exists \varphi$ such that $P[V(x,\varphi)=1]=1$
- $x \notin L \implies \forall \varphi$, we have $P[V(x,\varphi)=1] \leq 1/2$

In NP: certificate is a color for each vertex

Probabilistic verifier: check just one edge

This edge



What if just one rogue vertex wrong?

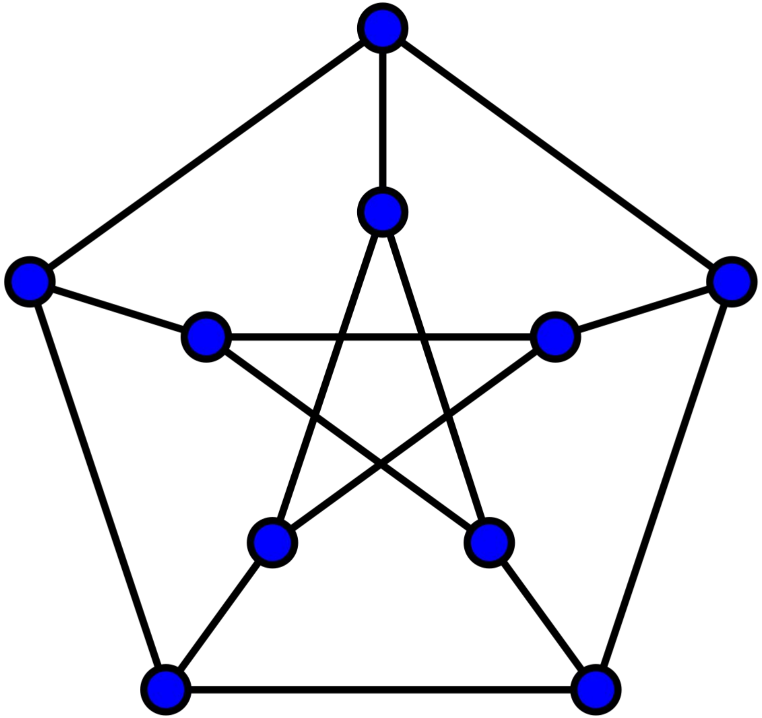
**Apply PCP Transformation**

# Example: Graph Coloring - CSP Formulation

CSP:

- One constraint per edge
- Constraints "query" constant number of vertices
- Goal: distinguish between cases in the promise

"constraint satisfaction problem"

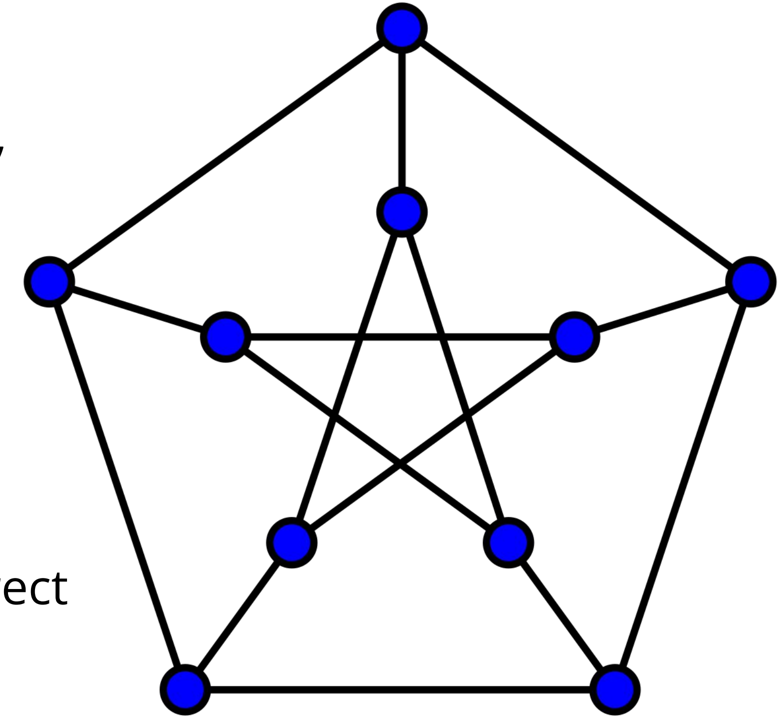


# Example: Graph Coloring - CSP Formulation

“constraint satisfaction problem”

CSP:

- One constraint per edge
- Constraints “query” constant number of vertices
- Goal: distinguish between cases in the promise



Promise:

- Either all colors correct
- Or > 10% incorrect

# Example: Graph Coloring - CSP Formulation

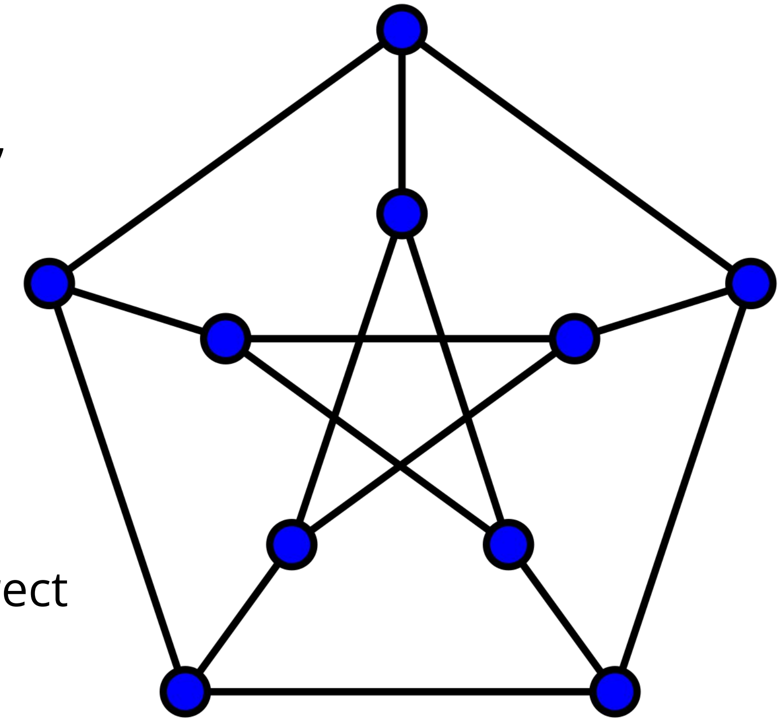
CSP:

- One constraint per edge
- Constraints “query” constant number of vertices
- Goal: distinguish between cases in the promise

Promise:

- Either all colors correct
- Or > 10% incorrect

“constraint satisfaction problem”



CSP Formulation

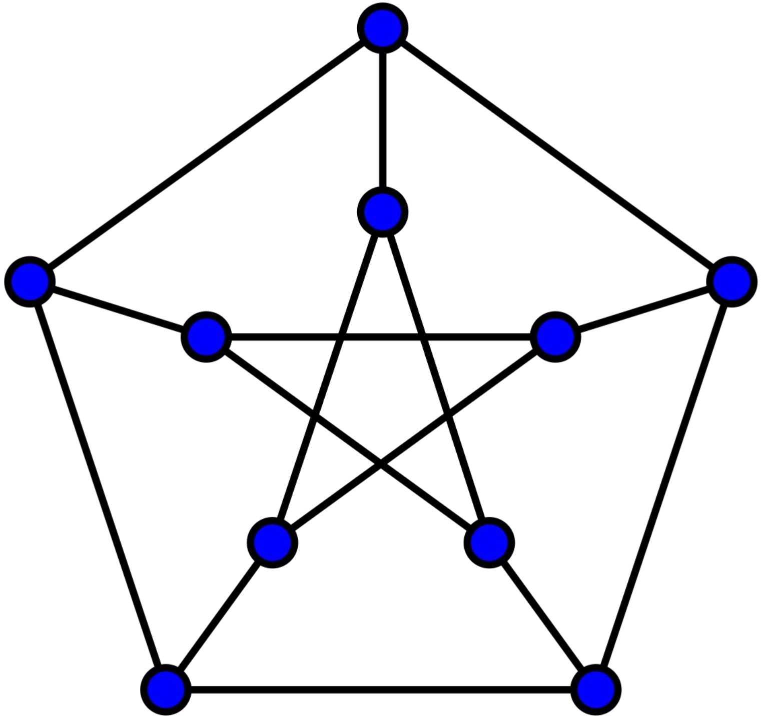
$\Longleftrightarrow$

PCP Formulation

# Example: Graph Coloring - Games Formulation

Two-player game:

- Edge player “E”
- Vertex player “V”
- Decision function



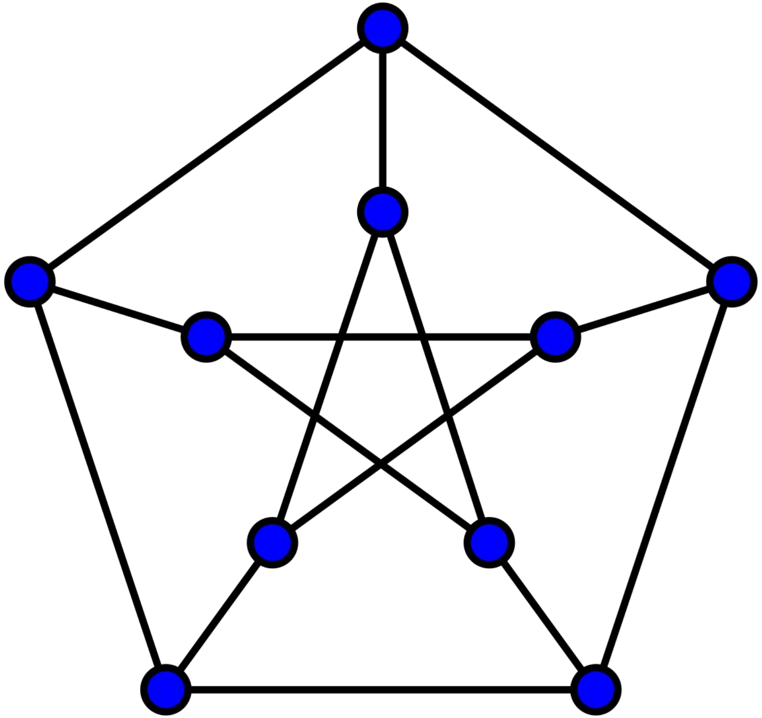
# Example: Graph Coloring - Games Formulation

Two-player game:

- Edge player “E”
- Vertex player “V”
- Decision function

Goal:

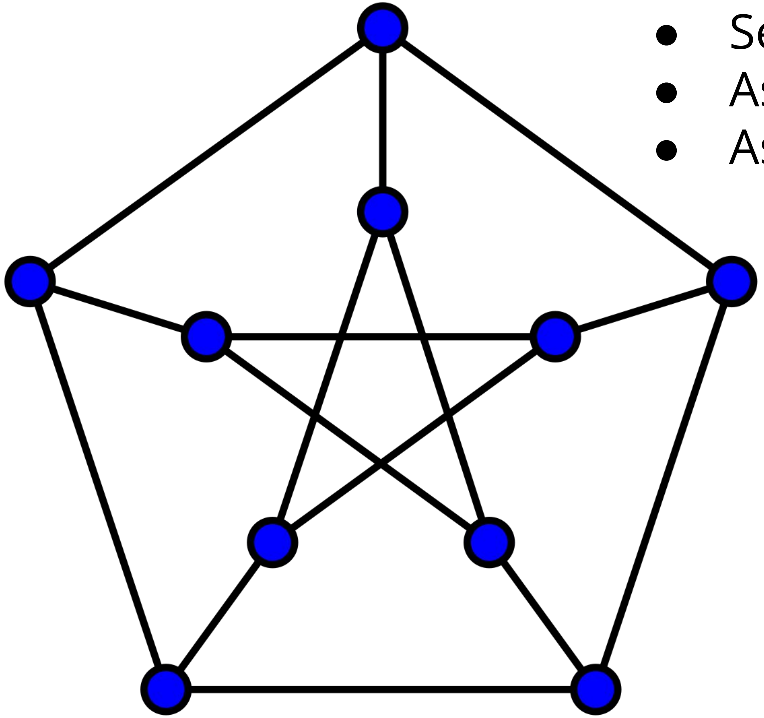
- Distinguish 100% win probability from < 50%.



# Example: Graph Coloring - Games Formulation

Two-player game:
- Edge player "E"
- Vertex player "V"
- Decision function

Goal:
- Distinguish 100% win probability from < 50%.



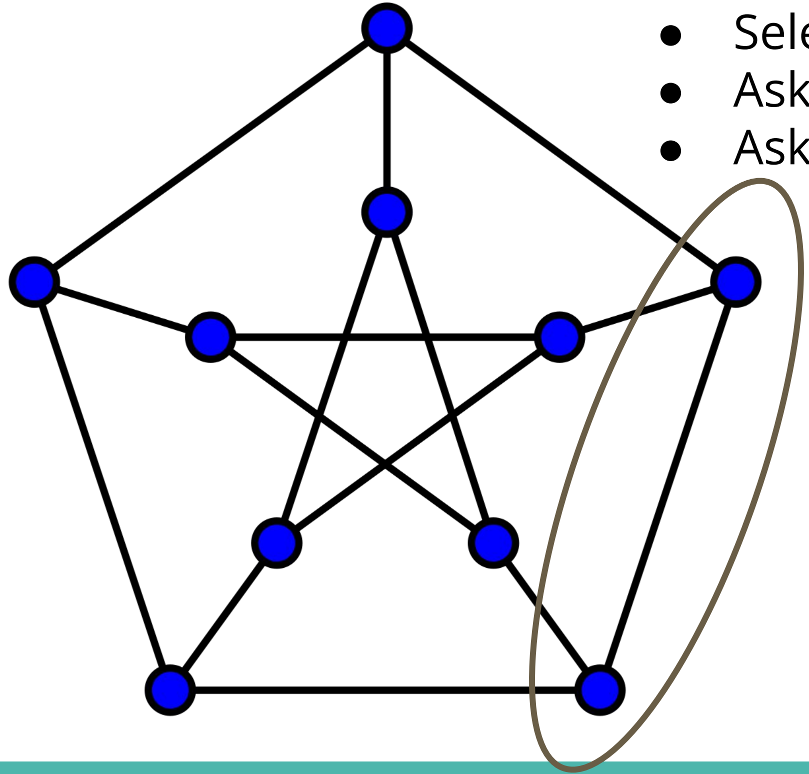
Question:
- Select random pair ($e$, $v$)
- Ask "E" for colors of $e$
- Ask "V" for color of $v$

Decision function:
- "E" answers two different colors
- "V" matches "E"

# Example: Graph Coloring - Games Formulation

Two-player game:
- Edge player “E”
- Vertex player “V”
- Decision function

Goal:
- Distinguish 100% win probability from < 50%.



Question:
- Select random pair (*e*, *v*)
- Ask “E” for colors of *e*
- Ask “V” for color of *v*

Decision function:
- “E” answers two different colors
- “V” matches “E”

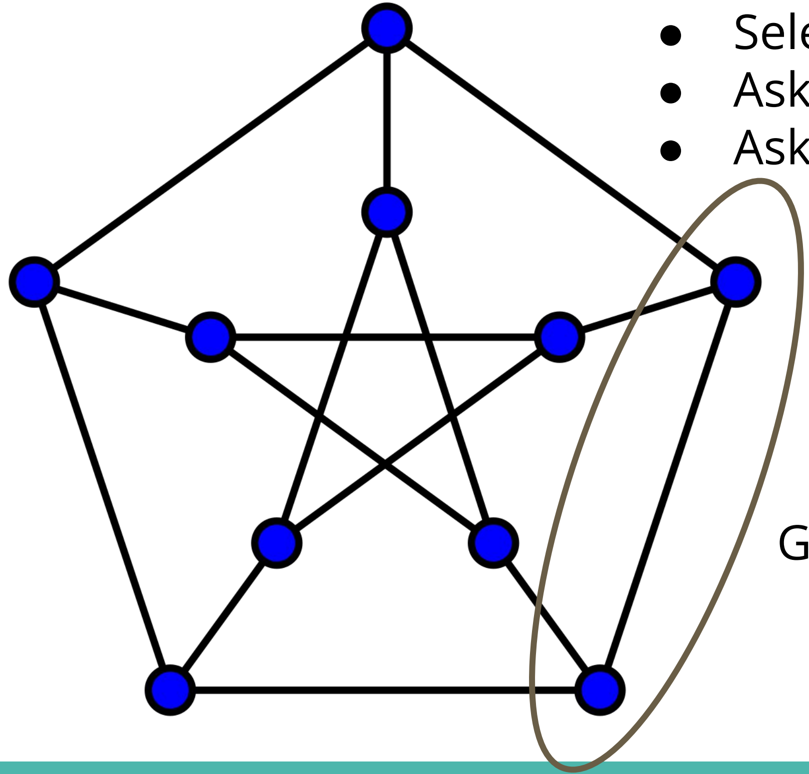
# Example: Graph Coloring - Games Formulation

Two-player game:

- Edge player "E"
- Vertex player "V"
- Decision function

Goal:

- Distinguish 100% win probability from < 50%.



Question:

- Select random pair (*e*, *v*)
- Ask "E" for colors of *e*
- Ask "V" for color of *v*

Decision function:

- "E" answers two different colors
- "V" matches "E"

Games Formulation

$\Longleftrightarrow$

PCP Formulation

# Key Takeaways

Games Formulation

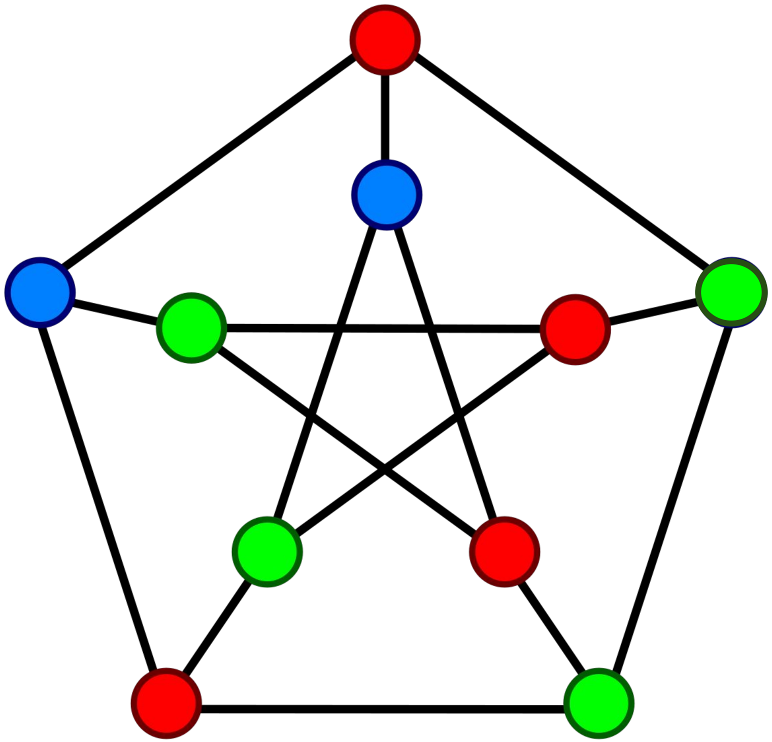
$\Longleftrightarrow$

CSP Formulation

$\Longleftrightarrow$

Proofs Formulation

(the easy part)



# Key Takeaways

Games Formulation

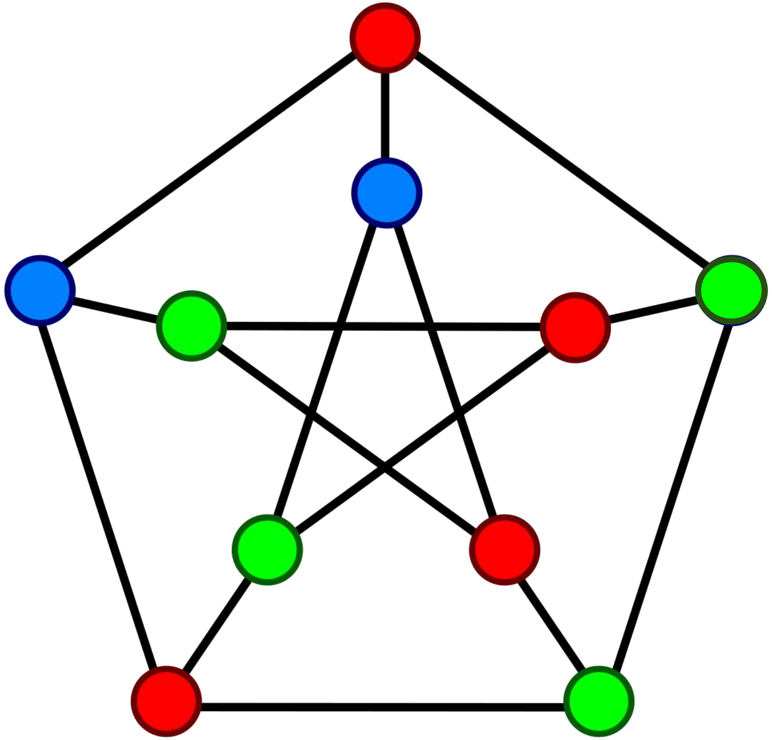
$$\Longleftrightarrow$$

CSP Formulation

$$\Longleftrightarrow$$

Proofs Formulation

(the easy part)



$$\mathrm{NP} = \mathrm{PCP}(O(\log n), O(1))$$

(the hard part)

# Key Takeaways

Games Formulation

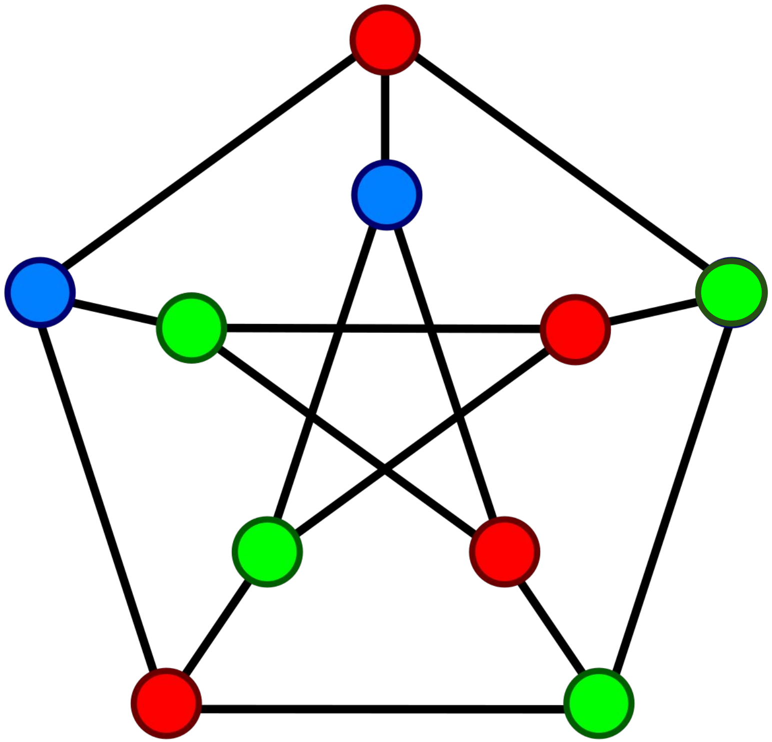
$\Longleftrightarrow$

CSP Formulation

$\Longleftrightarrow$

Proofs Formulation

(the easy part)



$$\text{NP} = \text{PCP}(O(\log n), O(1))$$

(the hard part)

CSP equivalence: it is even hard to approximate NP problems

# Into the Quantum Realm...

# q-Local Hamiltonian Problem

- Given *m* Hermitian matrices acting on q < n qubits $\mathbf{H_i} \in \mathbf{C}^{(2^n)^2}$
  - the total energy is $\boldsymbol{H = \Sigma H_i}$

# q-Local Hamiltonian Problem

- Given *m* Hermitian matrices acting on q < n qubits $\mathbf{H_i} \in \mathbf{C}^{(2^n)^2}$
  - the total energy is $\boldsymbol{H = \Sigma H_i}$
- Promise problem:

(YES instance) $\lambda_0(H) \leq a$

(NO instance) $\lambda_0(H) \geq b$

# q-Local Hamiltonian Problem

- Given *m* Hermitian matrices acting on q < n qubits $\mathbf{H_i} \in \mathbf{C}^{(2^n)^2}$
  - the total energy is $H = \Sigma H_i$
- Promise problem:

(YES instance) $\lambda_0(H) \leq a$

(NO instance) $\lambda_0(H) \geq b$

- Quantum **Cook-Levin Theorem** says the above is **QMA-hard** when
  - $b - a = 1/poly(n)$

# Dictionary: CSP vs. q-LH (Local Hamiltonian)

Constraints <---> Hamiltonians

# Dictionary: CSP vs. q-LH (Local Hamiltonian)

Constraints <---> Hamiltonians

**Locality means the same thing!**

# Dictionary: CSP vs. q-LH (Local Hamiltonian)

Constraints <---> Hamiltonians

**Locality means the same thing!**

Satisfy all constraints <---> Smallest eigenvalue less than $a$

# Dictionary: CSP vs. q-LH (Local Hamiltonian)

Constraints <---> Hamiltonians

**Locality means the same thing!**

Satisfy all constraints <---> Smallest eigenvalue less than $a$

Not all constraints satisfied <---> Smallest eigenvalue greater than $b$

# CSP $\leqslant$ Local Hamiltonian

- Local Hamiltonian problem naturally includes CSP as a subcase

# CSP ≤ Local Hamiltonian

- Local Hamiltonian problem naturally includes CSP as a subcase
- Force $\mathbf{H_i} \in \mathbf{C}^{(2^n)^2}$ to be a diagonal matrix with entries of 0 or 1
  - Forces eigenvalues to be exactly 0 or 1
  - Standard basis $\{e_1, \ldots, e_{2^n}\}$ are eigenvectors

# CSP ≤ Local Hamiltonian

- Local Hamiltonian problem naturally includes CSP as a subcase
- Force $\mathbf{H_i} \in \mathbf{C}^{\mathbf{(2\wedge n)\wedge 2}}$ to be a diagonal matrix with entries of 0 or 1
    - Forces eigenvalues to be exactly 0 or 1
    - Standard basis $\{e_1, \ldots, e_{2\wedge n}\}$ are eigenvectors
- Interpret each basis vector as encoding a classical n-bit string

# CSP ≤ Local Hamiltonian

- Local Hamiltonian problem naturally includes CSP as a subcase
- Force $\mathbf{H_i} \in \mathbf{C}^{(2^n)^2}$ to be a diagonal matrix with entries of 0 or 1
  - Forces eigenvalues to be exactly 0 or 1
  - Standard basis $\{e_1, \ldots, e_{2^n}\}$ are eigenvectors
- Interpret each basis vector as encoding a classical n-bit string
- Applying $\mathbf{H_i}$ to a basis vector $\mathbf{e_j}$ either incurs a cost of 0 or 1
  - We can now interpret the $H_i$'s as constraints on assignments to n bits

# CSP ≤ Local Hamiltonian

- Local Hamiltonian problem naturally includes CSP as a subcase
- Force $\mathbf{H_i} \in \mathbf{C}^{(2^n)^2}$ to be a diagonal matrix with entries of 0 or 1
  - Forces eigenvalues to be exactly 0 or 1
  - Standard basis $\{e_1, \ldots, e_{2^n}\}$ are eigenvectors
- Interpret each basis vector as encoding a classical n-bit string
- Applying $\mathbf{H_i}$ to a basis vector $\mathbf{e_j}$ either incurs a cost of 0 or 1
  - We can now interpret the $H_i$'s as constraints on assignments to n bits
- CSP problem of finding a satisfying assignment
  - ...same to LH problem to find if it's possible to get a cost of 0
  - But this is just a special case of LH!

# Quantum PCP Conjecture (qLH Formulation)

- QPCP conjecture says ∃γ>0 and q such that it is QMA-hard to distinguish YES and NO instances of q-LH on m Hermitian matrices with **b-a = γm**

(YES instance) $\lambda_0(H) \leq a$

(NO instance) $\lambda_0(H) \geq b$

# Quantum PCP Conjecture (qLH Formulation)

- QPCP conjecture says ∃y>0 and q such that it is QMA-hard to distinguish YES and NO instances of q-LH on m Hermitian matrices with **b-a = ym**

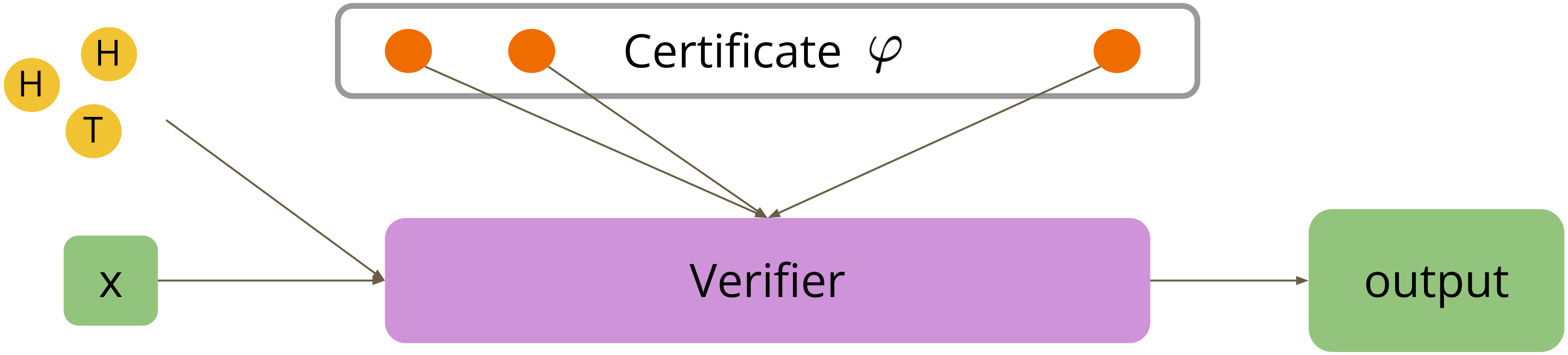
  (YES instance) $\lambda_0(H) \leq a$

  (NO instance) $\lambda_0(H) \geq b$

- How is this different from Quantum Cook-Levin?
  - QCL says **b-a = 1/poly(n)**
  - This is a statement the hardness of approximating the eigenvalue, similar to the classical promise problem of CSP

# Quantum PCP Conjecture (Proofs & Games)

- Proofs are easily adaptable to a quantum setting
  - Just allow for quantum certificates on a quantum computer
  - Allow our proofs to measure exactly **q qubits** before making a decision
  - Few more details...
  - Conjectured that there exists a **constant q** s.t. all languages in QMA can be solved in this setting

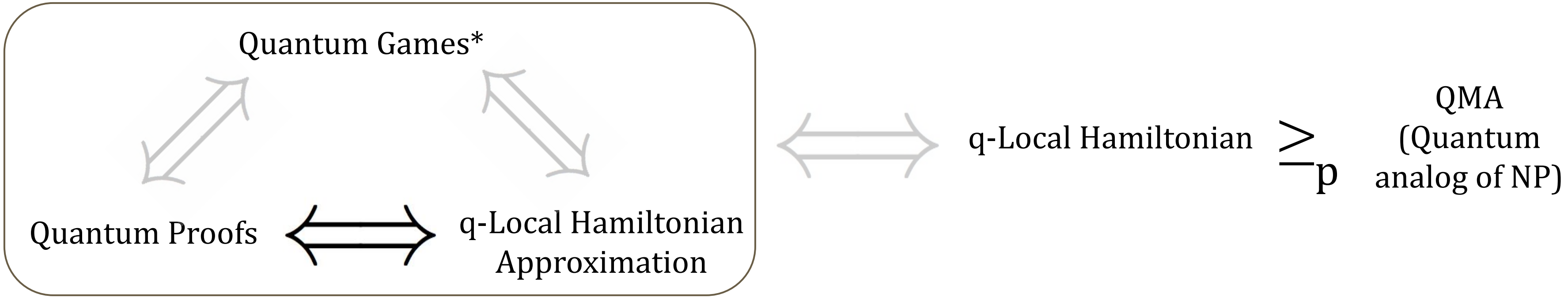


# Quantum PCP Conjecture (Proofs & Games)

- Proofs are easily adaptable to a quantum setting
  - Just allow for quantum certificates on a quantum computer
  - Allow our proofs to measure exactly **q qubits** before making a decision
  - Few more details...
  - Conjectured that there exists a **constant q** s.t. all languages in QMA can be solved in this setting
- Games also extend nicely to a quantum setting
  - Allow our players to compute their answers to any questions on a quantum computer
  - For anything interesting to happen, our players have to share entanglement
  - Few more details...

# Summary of Quantum Knowledge

- The quantum Local Hamiltonian and Proofs Formulations are proven to be equivalent
- Quantum games are complicated but conjectured to also be equivalent
- What is the full picture so far?



# Concluding Thoughts

- Beautiful theory, powerful applications
- Hot area, not well understood
- **MIP* = RE**

**Classical PCP**

Games Formulation

$\Longleftrightarrow$

CSP Formulation

$\Longleftrightarrow$

Proofs Formulation

Quantum Games*

Quantum Proofs $\Longleftrightarrow$ q-Local Hamiltonian Approximation

$\Longleftrightarrow$ q-Local Hamiltonian $\geq_p$ QMA (Quantum analog of NP)

# Concluding Thoughts

- Beautiful theory, powerful applications
- Hot area, not well understood
- **MIP* = RE**

And now **YOU** are equipped to go forth and learn more!

**Classical PCP**

Games Formulation

$\iff$

CSP Formulation

$\iff$

Proofs Formulation

Quantum Games*

Quantum Proofs $\iff$ q-Local Hamiltonian Approximation

$\iff$ q-Local Hamiltonian $\geq_p$ QMA (Quantum analog of NP)