Quiz Game!

True, False, or Open?

Your answer will be “true”, “false”, or “open”. If you get it wrong, the other team has a change to answer, but it’ll be correspondingly easier – so answer carefully!

Open

- P = NP

True

- If P≠NP, then there is no polynomial-time solution to the Traveling Salesman Problem.

Open

- Quantum computers can solve NP-complete problems in polynomial time.

True

- Every **regular** language is **decidable**.

True

- Every **sound** axiomatic mathematical theory is **consistent**.

Open

- If P≠NP, then there is no polynomial-time solution to factoring numbers.

True

- Either P≠NP or NP≠EXP, or both.

False

- Every **decidable** language is **context-free**.

False

- Every **consistent** axiomatic mathematical theory is **sound**.

False

- If P≠NP, then there is no polynomial-time algorithm for checking that a number is prime.

True

- CircuitSAT (with a circuit made of **AND**, **OR**, and **NOT** gates) is NP-complete.

Open

- All problems in NP are NP-complete.

False

- Every **context-free** language is **regular**.

True

- If there exists a **one-way function**, then there exists a **pseudorandom generator**.

True

- There is an algorithm for solving the Traveling Salesman Problem.

True

- All NP-complete problems are NP-hard.

False

- The halting problem is not recognizable by any Turing machine.
  - L is recognizable: TM accepts if x is in L
    TM rejects **or loops** if x is not in L

Open

- If there is a pseudo-random generator, then public-key cryptography works.

Open

- CircuitSAT (with a circuit made of **AND** gates) is NP-complete.

Open

- Quantum computers can break any public-key cryptosystem that you can think of in polynomial time.

False

- With an oracle for the Halting Problem, every language is decidable and every mathematical problem is solvable.

Open

- If P≠NP, then there exists a one-way function.

True

- The Halting Problem is NP-hard, but is not in NP.

True

- Everything you could do with a **deterministic finite automaton (DFA)** with N states, you can do with a **nondeterministic finite automaton (NDFA)** with N states.

Open

- There is an algorithm for solving the Traveling Salesman Problem in polynomial time.

Open

- P = PSPACE

True

- CircuitSAT (with a circuit made of **AND** and **NOT** gates) is NP-complete.

Open

- There exists a language *L* in NP, that is neither in P nor NP-complete.

True

- Every prime number P passes the Fermat's Little Theorem test. That is, $x^P - 1$ is divisible by P, for any x.

False

- There are uncountably-infinite regular languages.

True

- If there is a trapdoor one-way function, then public key cryptography works.

False

- No language that is **recognizable** by a Turing machine can be **decided** by a Turing machine with an oracle for the Halting Problem.

True

- If the RSA cryptosystem is secure against all polynomial-time attacks, then P≠NP.

False

- There are uncountably-infinite context-free languages.

Open

- Unlike classical computers or Turing machines, quantum computers can factor integers in polynomial time.

True

- If there exists a one-way function, then P≠NP.

False

- Everything you could do with a **nondeterministic finite automaton (NDFA)** with N states, you can do with a **deterministic finite automaton (DFA)** with N states.

False

- There are uncountably-infinite decidable languages.

True

- IP = PSPACE

False

- Every composite number C passes the Fermat's Little Theorem test. That is, $x^C - 1$ is not divisible by C, for any x.

True

- { <M> | there exists an x such that M(x) halts } is recognizable

False

- { <M> | for all possible inputs x, M(x) halts } is recognizable.

-- end --