Proof that P ≠ NP

Author
Robert DiGregorio
0x51B4908DdCD986A41e2f8522BB5B68E563A358De

Abstract
A problem exists that’s hard to solve but easy to verify a solution for.
Part 1: proof M runs in superpolynomial time

$\exists H \forall A \ [ah \in H(A) \Leftrightarrow ah \subseteq A \land |ah| = |A| / 2]$

- note: $H(A)$ is every possible half of $A$
- note: $|H(A)| = O(|A|! / (|A| / 2)!^3)$
- note: $O(|A|! / (|A| / 2)!^3)$ is superpolynomial

$\exists F \forall A \forall ah \in H(A) \forall x \in ah \ [x = x \& F(ah)]$

- note: $F(ah)$ is $ah$ folded over the bitwise and operation

$\exists \text{deterministic polynomial time Turing machine } V \forall A \forall ah \in H(A) \forall B \forall bh \in H(B) \ [V(ah, bh) = (F(ah) = F(bh))]$

$\exists \text{deterministic Turing machine } M \forall A \forall B \ [M(A, B) = \exists ah \in H(A) \exists bh \in H(B) \ [V(ah, bh)]]$

- note: $V$ verifies $M$

$\exists A \ [M \text{ iterates over } H(A)]$

Ordering $A$ does not order $H(A)$ by $F(ah)$

- note: $F(ah)$ could fold $ah$ over the bitwise or operation or the bitwise exclusive or operation to the same effect

By definition, it's impossible for a deterministic Turing machine to search an unordered set without iteration

$\exists A \ [M \text{ iterates over } H(A)] \Rightarrow M \text{ runs in superpolynomial time}$
Part 2: proof $P \neq NP$

$\exists L \subseteq \{0, 1\} \ [\forall w \in L \ [M \text{ accepts } w]]$

$M$ runs in superpolynomial time $\land \forall w \in L \ [M \text{ accepts } w] \Rightarrow L \notin P$

$V$ runs in polynomial time $\land V \text{ verifies } M \land \forall w \in L \ [M \text{ accepts } w] \Rightarrow L \in NP$

$L \notin P \land L \in NP \Rightarrow P \neq NP$