If P = NP, which languages would be NP-complete?

If P were equal to NP, it would imply that every problem in NP can be solved in polynomial time. In such a scenario, the concept of NP-completeness, as we understand it today, would lose its significance because every problem in NP would also belong to P. NP-complete problems are those that are at least as hard as the hardest problems in NP. However, if P were equal to NP, then every problem in NP would be equally easy to solve, and the distinction between NP and NP-complete would disappear.

In other words, if P = NP, there would be no NP-complete problems in the traditional sense because all problems in NP would be solvable in polynomial time, and there would be no need to distinguish between them in terms of complexity.

It's important to note that if P were proven to be equal to NP, it would be a groundbreaking result with significant implications for the field of computer science and cryptography, as many cryptographic systems rely on the assumption that P is not equal to NP. However, as of my last knowledge update in September 2021, this problem remains unresolved, and it's one of the most important open questions in the field of theoretical computer science.