

Assignment 12

We keep talking about all these NP-Complete problems but I'm not sure if it's really sunk in why these problems are even vaguely interesting. Most of the questions below (aside from 4) are directly from lecture.

Let $\mathbf{P}$ denote the class of all problems that have a polynomial time algorithm that gives the correct output on all possible inputs.

Let $\mathbf{NP}$ denote the class of all problems that for which we can verify a “solution” to the problem in a polynomial amount of time.

1. Convince me that all problems in $\mathbf{P}$ are also in $\mathbf{NP}$.

Solution: Short version: Determining whether a solution exists is certainly harder than finding a solution.

Lengthy version: A problem is $\mathbf{NP}$ if given access to a solution we can verify an input in polynomial time. If the problem is in $\mathbf{P}$ already, then we can just run the algorithm that solves it in polynomial time and ignore the given solution to determine the correct output.

2. What would it mean about the “difficulty” of problems in $\mathbf{NP}$ if $\mathbf{P} = \mathbf{NP}$?

Solution: It means that all problems in $\mathbf{NP}$ can be solved in polynomial time, including those that are **NP-Complete**. So in order to solve a problem in polynomial time, it would be sufficient to be able to verify a solution in polynomial time.

3. Check out https://simple.wikipedia.org/wiki/RSA_algorithm and https://en.wikipedia.org/wiki/Integer_factorization#Difficulty_and_complexity. RSA algorithms are involved in a wide variety things, including many password protection models. These encryption schemes only ‘work’ so long as nobody knows how to factor integers ‘quickly’, say in a polynomial amount of time. Currently, the state of the art algorithms to factor integers are relatively slow.

4. Convince me that the problem of determining the prime factors of an integer is in $\mathbf{NP}$. And hence, if $\mathbf{P} = \mathbf{NP}$, then you could crack RSA encryption.

Solution: Given some integer to determine the prime factors of, if someone told us the prime factors themselves we could multiply them together to see if their product is the given number. Since multiplying integers can be done in polynomial time, this suffices.

5. Let A be any **NP-COMPLETE** problem. Convince me that if A is in $\mathbf{P}$, then $\mathbf{P} = \mathbf{NP}$.

Solution: Short version: If A is **NP-Complete**, then it's in $\mathbf{NP}$ and it's harder than all problems in $\mathbf{NP}$ up to a polynomial factor in runtime. So if A can be solved in polytime, then all problems that are easier than it (namely, all those problems in $\mathbf{NP}$) can also be solve in polytime, so $\mathbf{P} = \mathbf{NP}$.

Longer Version: We know that if $B \leq A$ and A is in $\mathbf{P}$, then B is in $\mathbf{P}$ (if you don't know how to convince me of this, you should review the lecture notes!). We know A is **NP-Complete**, so $B \leq A$ for all problems B in $\mathbf{NP}$. So if A is in $\mathbf{P}$, then B is in $\mathbf{P}$ for all problems B in $\mathbf{NP}$. Hence $\mathbf{P} = \mathbf{NP}$.

Assignment 12

6. Let A be **NP**-COMPLETE problem. Convince me that if A is not in **P**, then $\mathbf{P} \neq \mathbf{NP}$.

Solution: Since A is **NP**-Complete, we know that A is in NP . If A is not in **P**, then we have a problem that is in **NP** but not in **P**, so $\mathbf{P} \neq \mathbf{NP}$.

Make sure you understand those last two - I am certain that they will appear on the exam in some way.