The Third Assignment

3190102721 Xu Shengze

Note: This homework is completed after discussing with classmates, below I have written my own ideas for solving problems and my understanding and explanation of the answers given in the textbook.

Problem Prove that the predicate "x is the binary representation of a prime integer" belongs to NP.

Ideas The following are the ideas to solve this problem.

In the prime number test question, if the answer is no, that is, the integer m to be determined is not a prime number, then integers other than 1 and m appear in the factorization of the whole integer. At this time, Merlin only needs to input factorization to Arthur, and Arthur can Calculate the product of all factors of factorization within polynomial time and verify whether the product is m. Integer multiplication can be done in polynomial time, so Arthur can confirm it himself.

Therefore, the key to this question lies in the case where the answer is yes, how does Merlin prove that the integer entered is indeed a prime number. At this point Merlin proved to Arthur that $\{1, 2, \cdots, n-1\}$ (and the multiplication operation) is a cyclic group of order $n-1$, but only one generator g is not enough, because we cannot ask Arthur to verify $g, g^2, g^3, \cdots$ one by one. In this case, the algorithm is not polynomial time. But Arthur can verify $g^{n-1} \equiv 1 \pmod{n}$ in polynomial time, because the exponential operation with modulus can be done in polynomial time (There is such an algorithm in the textbook, so I won't repeat it here).

We noticed that after verifying $g^{n-1} \equiv 1 \pmod{n}$, there is still no guarantee that the order of the cyclic group is $n-1$, because it may be in $g, g^2, g^3, \cdots$. In the sequence, 1 is also obtained before g^{n-1}, and g^{n-1} is not the first time that 1 is obtained.

In fact, the place where $g^a \equiv 1 \pmod{n}$ may appear in advance must satisfy that a can divide $n-1$. So Merlin also showed Arthur the prime factorization of $n-1$, at which time he could verify by himself that the "possible 1s" did not actually appear in the places. At the same time, since Merlin may "lie", we have to prove that every number in the prime factorization of $n-1$ is indeed a prime number. This has to recurse in the same way until each number to be verified is a very small prime number (such as 2).

Finally, we can verify that the length of the entire recursive certificate is polynomial level, and the proof is complete.

Answer The following is my understanding of the answers given in the book, and expounded in my own language.

We prove that PRIMALITY $\in$ NP by showing how Merlin constructs a polynomial size "certificate" of primality that Arthur can verify in polynomial time.

Arthur's construction idea is: assuming a prime number p with a value greater than 2, then Arthur can obtain a cyclic group of order $p-1$ from p, expressed as Z/pZ. At the same time, Arthur can prove that $g^{p-1} \equiv 1 \pmod{p}$ is true, and Merlin can give the generator g of this cyclic group.

If the order of g is a nontrivial factor for $p-1$, and if Arthur can know the factorization of $p-1$, including its quality factors $q_1, q_2, \cdots$, then Arthur can show that $g^{(p-1)/q_i} \not\equiv 1 \pmod{p}$ is true.

However, because decomposition is a difficult problem, Arthur cannot calculate the qualitative factor by himself. He could only tell it, by Merlin, the result of factoring, but Merlin had to tell Arthur the importance of these prime factors and recursively tell Arthur that these prime factors are all primes.

The complete proof of primeness procedure is like a tree. Each node of the tree represents whether a number q is a prime or not, and a child node of that node represents a prime factor of number $q-1$. All leaves represent $q = 2$, nodes except leaves represent $q > 2$, generator h with group (Z/qZ).

We can estimate the total size of the proof process by obtaining the number of leaves of the proof tree. Because in the proof tree there are at most $n = \lceil \log_2 p \rceil$ leaves with twice the number of nodes and each node has a pair of n-bit numbers q and h.Therefore the total number of bits in the certificate is $O(n^2)$.

Finally, we can estimate the complexity of the validation process. For each $O(n)$ node, Arthur needs to prove that $h^{p-1} \equiv 1 \pmod{q}$ is true, and this requires an $O(\log q) = O(n)$ multiplication, computed by $O(n^3)$; A similar demonstration is made for each parent-child pair, and the amount of computation required for each pair is also $O(n)$, so the entire vegetal proof is completed by a circuit with a comput $O(n^4)$ that can be performed in polynomial time.

Some thinking The following are some thoughts and summaries of my solution to this problem.

First of all, we have Arthur and Merlin.

If we compare Arthur to an ordinary computer, then Merlin is an excellent assistant to it, and Merlin will tell Arthur what he knows. Because Merlin may "lie", Arthur needs to personally check whether Merlin's words are true or not. At the same time, Merlin also understands this, so he will choose the appropriate certificate to tell Arthur, so that

Arthur can determine the answer to the question in polynomial time after seeing the prompt.

If for any type of input to the problem, Merlin has a way to find a suitable problem-solving hint and tell Arthur so that the latter can determine the answer to the problem in polynomial time, then the problem is said to be NP.