Problem 11. «Useful Proof-of-work for blockchains»

Special Prize from the Program Committee!

Proof-of-work system is one of the key parts of modern blockchain-based platforms implementations, like cryptocurrency **Bitcoin** or **Ethereum**. Proof-of-work means that user is required to perform some work in order to request some service from the system, e. g. to send an e-mail or to create a new block of transactions for the blockchain.

For example, in Bitcoin system, if some user wants to create a block of transactions and add it to the chain, hash value of his block must satisfy certain condition, which can be achieved by iterating special variable X inside the block many times and checking the resulting hash value on every iteration.

What is important about the problem in a proof-of-work system, is that

- It is known that the solution for the problem exists, and it is also known how many iterations (on average) are required to find it, using best known algorithm $\mathcal{A}$;
- There are no algorithms for solving the problem, that perform significantly better than $\mathcal{A}$; it is believed also that such algorithms will not be found soon;
- Problem depends on some input data I, so you can not find solutions for the problem in advance (before input I is known) and then use these solutions without performing any work;
- Given a problem and a solution to it, it is easy to verify that provided solution is correct.

Unfortunately, solving the problem of finding specific hash values (used in Bitcoin and Ethereum) does not yield any information that is useful outside the system, therefore tremendous amounts of calculations performed to solve the problem are wasted.

Some other implementations of proof-of-work system solve this issue. For example, solutions of proof-of-work problem used in cryptocurrency **Primecoin** give us special chains of prime numbers, useful for scientific research.

Turn to the next page.

Your task is to construct a problem **P** that can be used in a proof-of-work system, such that information obtained in the proccess of solving it can be useful outside the system. More formally:

- **P** is, in fact, a family of problems, parametrized by two variables: I (input data, you can assume that I is a 256 bit string, or introduce other sensible formats), and C (complexity, e. g. some positive integer). For fixed input and complexity, $\mathbf{P}(I, C)$ is a problem that can be solved by using some algorithm $\mathcal{A}$ (should be provided in your solution to this task). It should not be possible to find a provable solution for the problem $\mathbf{P}(I, C)$ if I is not known;
- Average time T (amount of computational steps or iterations), required to find a solution of $\mathbf{P}(I, C)$ using algorithm $\mathcal{A}$ is known (assuming input data I is chosen randomly and uniformly), and depends on C, so $T = T(C)$, and $T(C)$ can be made very small, infeasibly large, or something in-between by adjusting complexity variable C;
- It should be easy to verify whether any provided solution is correct or not;
- Any kind of proof that there are likely no significantly better algorithms for solving **P** than the given algorithm $\mathcal{A}$, is desirable. For example, proof that proposed problem is $\mathcal{NP}$-hard, or any other considerations;
- You should describe how information obtained in the process of solving **P** can be useful outside of the proof-of-work system.

For example, in Bitcoin system, $\mathbf{P}(I, C)$ is a problem of finding an integer X such that if we apply SHA-256 hash function to the pair (I, X) twice, resulting hash value, represented as an integer, will be not greater than C. Here C is a nonnegative integer, defining complexity of the problem, and I — a block header, containing information about all transactions included in it, along with some other information — is an input.