Heidi's Cryptographic Commerce
Heidi et son Commerce Cryptographique
Heidis kryptographisches Gewerbe

Robert R. Enderlein

Hard Task

Heidi still hasn't found her private key, and people are now starting to use bigger and bigger primes! The decryption algorithm you wrote in the last sub-problem is now too slow for the messages she receives...

Luckily, Heidi can usually guess what message is contained in a given ciphertext (for example the alternatives could be "YES" or "NO", or something similar). Of course, since she thinks her encryption scheme is reasonably secure, she doesn't expect you to perform perfectly in this task: she will be happy if your decryption algorithm is correct in distinguishing which of the two candidate messages is the right one with probability 75%.

Can you write a decryption function that given the ciphertext (E,F), Heidi's public key (p,g,y) and two possible messages (m_1) and (m_2), recovers the message that was sent to her (which is either m_1 or m_2) with a probability of 75% of being correct?

(The sample code, by always returning m_1, has a probability of 50% of being correct.)

Note: there will be around eight thousand testcases in this sub-problem, and the size of the primes will all be between 2^{63} and 2^{64}. You may assume that all random values are distributed uniformly at random within their domain. We engineered that problem so that a program that solves each testcase with 75% probability will successfully solve this sub-problem, while a program that solves only 71.875% of the testcases will not solve this sub-problem.

Tâche Difficile

Heidi n'a toujours pas trouvé sa clef privée, et les gens commencent à utiliser des nombres premiers de plus en plus grands! L'algorithme de décryptage que tu as écrit dans le dernier sous-problème est maintenant trop lent pour les messages qu'elle reçoit...

Heureusement, Heidi peut généralement deviner quel message est contenu dans un texte chiffré donné (par example, les alternatives pourraient être "OUI" ou "NON", ou quelque chose de semblable). Bien sûr, puisqu'elle sait que son mécanisme de chiffrement est relativement sûr, elle ne s'attend pas à ce que tu décryptes le texte à coup sûr dans cette tâche: elle sera ravie si ton algorithme réussit à distinguer correctement lequel de deux messages se cache dans le texte chiffré avec une probabilité de 75%.

Peux-tu écrire une fonction de décryptage qui, étant donné un texte chiffré (E,F), la clef publique de Heidi (p,g,y) et deux messages possibles, (m_1) et (m_2), réussisse à distinguer correctement quel message a été envoyé à Heidi (soit m_1, soit m_2) avec une probabilité de 75% d'être correct?

(Le code d'exemple, en retournant toujours m_1, a une probabilité de 50% d'être correct.)

Remarque: il y aura environ huit mille cas de tests dans ce sous-problème, et les nombres premiers seront tous entre 2^{63} et 2^{64}. Vous pouvez supposer que toutes les valeurs aléatoires sont distribuées de manière uniforme et aléatoire dans leur domaine. On a conçu le problème de sorte à ce qu'un programme qui résolve chaque cas de test avec une probabilité de 75% réussisse à résoudre ce sous-problème, alors qu'un programme qui résout uniquement 71.875% des cas de test ne puissent résoudre ce problème.

Schwierige Aufgabe

Heidi hat ihren privaten Schlüssel immer noch nicht wiedergefunden. Nur benutzen ihre Freunde mittlerweile immer grössere Primzahlen! Somit ist euer Entschlüsselungsalgorithmus der letzten Unteraufgabe zu langsam für die aktuellen Nachrichten...

Zum Glück hat Heidi generell ein gutes Gespür für die verschlüsselte Nachricht (Sie denkt, es sei entweder „JA“ oder „NEIN“, oder etwas ähnliches). Da Heidi davon überzeugt ist, dass ihr Verschlüsslungsverfahren ziemlich sicher ist, erwartet sie kein perfektes Resultat von euch: es genügt, wenn euer Entschlüsslungsverfahren die richtige Nachricht in 75% der Fälle identifizieren kann.

Bitte schreibt eine Entschlüsslungsfunktion, welche verschlüsselten Text (E,F) und öffentlichen Schlüssel (p,g,y) erhält, sowie zwei mögliche Nachrichten (m_1) und (m_2), und die ursprüngliche Nachricht (entweder m_1 oder m_2) mit einer Wahrscheinlichkeit von wenigstens 75% identifiziert.

(Der Beispielcode, welcher stets m_1 zurück gibt, findet in 50% aller Fälle das richtige Resultat)

Bemerkung: Diese Unteraufgabe enthält etwa 8000 Testfälle, und die Primzahlen ligen zwischen 2^{63} und 2^{64}. Ihr könnt davon ausgehen, dass sämtliche Zufallszahlen gleichmässig über ihren Zufallsbereich verteilt sind. Die Testfälle sind so gewählt, dass ein Programm, welches jeden Testfall mit 75% Wahrscheinlichkeit löst, punktet. Eine Trefferquote von nur 71.875% wird jedoch keine Punkte einbringen.