Université de Paris 7 | **TP11 $(\mathbb{Z}/p\mathbb{Z})^\times$, applications** | **agreg 2010 (F.Han[1])**

Exercice I: Primalité ; temps de calculs

1) Etudier les fonctions `nextprime` et `isprime`, `is_pseudoprime`. Comparer les temps d'executions pour des nombres à 20 chiffres. Etudier la documentation.

2) a) Etudier : `n:=nextprime(10^i)*nextprime(20^i);` Essayer de factoriser n pour différententes valeurs de i. A partir de quand est ce que ca ne répond plus ?

b) Prendre des nombres premiers de cette taille. Etudier si l'on arrive à factoriser $p-1$.

3) a) En utilisant pari, obtenir un certificat de primalité par la méthode de Lehmer (et al) pour un nombre de cette taille. Interpréter la réponse.

b) En créant au hasard un nombre n dont vous connaissez la factorisation, trouver un nombre pseudopremier du type $n+1$ assez grand, et tentez de le certifier vous même.

c) Comment trouver la matrice des (facteurs,multiplicités) d'un entier. Comment effectuer une grande puissance en modulaire ?

d) Déduire de votre certificat un génerateur de $(\mathbb{Z}/p\mathbb{Z})^\times$.

e) Vérifier son ordre en utilisant pari. Cf menu aide,manuel,PARI-GP. Rappel, pour créer un nombre modulaire sous xcas lorsque l'on est en mode maple, par exemple le 1 de $\mathbb{Z}/p\mathbb{Z}$.

i) Soit on bascule[2] en mode xcas et l'on fait : `unp:=1%p;` puis l'on revient en mode maple.

ii) Soit on utilise la fonction (non documentée) : `unp:=makemod(1,p);`

iii) Soit on utilise le logiciel pari. On charge donc pari avec `pari();`, puis on peut utiliser la syntaxe usuelle de pari : `unp:=Mod(1,p);` (Cf doc PARI-GP dans le menu aide)

4) **log discret :**

a) Trouver une dizaine de nombres pseudopremiers du type $4p+1$ où p est un pseudopremier à au moins 13 chiffres. Est ce difficile de trouver des nombres premiers tels que $p-1$ ait un grand facteur ?

b) Chercher dans pari les intructions pour trouver un générateur g de $(\mathbb{Z}/n\mathbb{Z})^\times$ lorsqu'il est cyclique. Trouver un générateur g de $(\mathbb{Z}/p\mathbb{Z})^\times$ où p vaut `nextprime(10^13+100)`. En utilisant `pari_znlog(3,Mod(g,p))` trouver m tel que $3 = g^m[p]$. Expliquer pourquoi il a réussi ?

c) Recommencer avec un exemple du type $p = 4q+1$ trouvé précedemment.

d) Calculer la probabilité pour un élément x de $\mathbb{Z}/p\mathbb{Z}$ d'être un générateur de $(\mathbb{Z}/p\mathbb{Z})^\times$ pour quelques nombres premiers grands, puis le nombre de tirages nécéssaires pour avoir au moins 95% de chances d'avoir trouvé un générateur. Qu'en pensez vous (si l'on dispose d'une façon efficace de trouver l'ordre d'un élément donné) ?

e) Remarquez que `pari_znprimroot` retourne souvent 2, et dans le cas contraire c'est que 2 n'est pas un générateur.

f) Trouver des nombres premiers p de taille similaire, où les temps de factorisation de $p-1$ sont tres différents. Etudier le temps d'exécution de `znprimroot(p)` pour ces valeurs de p, ainsi que celui de `znorder(Mod(2,p)`. Conjecturer comment pari trouve un générateur de $(\mathbb{Z}/p\mathbb{Z})^\times$

Exercice II: Ecrit 2007

1) Choisir un nombre premier p à 13 chiffres, tel que le problème du log discret soit difficile modulo p. (Cf exercice précédent)

2) Choisir un entier b et rendre publique $(b, b^e[p])$. Commentez votre choix de b.

3) Remarquer que les lettres $A \dots Z$ ont des codes ASCII consécutifs. Comment obtenir les codes de chaque lettre d'un mot ? (et si l'on veut que A soit représenté par 0 et Z par 25 ?)

4) Votre voisin choisit un message, tire des k au hasard et code chaque lettre x avec $(b^k[p], x.(b^e)^k)[p])$ (sans connaitre e, seulement avec b et b^e

5) a) Décoder le message grâce à e.

b) Préparer un exposé de 15 minutes expliquant la méthode et son intérêt. (temps pour crypter, décryper avec et sans e. Statitiques sur les lettres du message crypté...)

[1] http ://www.math.jussieu.fr/∼han/agreg

[2] soit via le menu soit via la fonction maple_mode(0)