Complexité

1. Premier ou pas ? O(Ln(n))^5 environ

1.1 Attention: la taille (en bits) de n est en Ln(n); ceci est donc une fonction polynomiale de la taille

On utilise des résultats inspirés du théorème de Fermat pour faire un premier tri « pseudo-premiers ».

1.2 De toutes façons si n est premier $\varphi(n)$ n'est pas un mystère

Il faut donc des entiers non premiers et difficiles à factoriser

2. Factorisation

2.1 Division par tous les entiers $<\sqrt{n}$? O($\sqrt{n}$); c'est à dire exponentielle en fonction de la taille

exemple:

factorisation(n):=block([x,k,L],x:n,L:[ ],for k thru sqrt(n) do (if mod(x,k)=0 then L:endcons(L,k)),return(L))dollar

on prend n et on teste l'un après l'autre jusquà $\sqrt{n}$ si k le divise

n=10^{10} environ 2.7 secondes

n=10^{20} environ 4 heures toujours « en cours »

2.2 Méthode de Fermat « n=a²-b² » ? O($n^{1/3}$) dans le cas « p et q proches »; exponentielle en fonction de la taille

exemple : on teste les entiers a à partir du premier entier supérieur à $\sqrt{n}$ et on cherche si a^2-n est un carré, si oui on pose $b^2 = a^2 - n$ et on a n=a^2-b^2=(a+b)(a-b)

si non on incrémente a->a+1 ...

2.3 Méthode ro -1 de Pollard « créer une suite périodique modulo p » O($n^{1/4}$); exponentielle en fonction de la taille

1221036714771372924O7

https://en.wikipedia.org/wiki/Pollard's_rho_algorithm#Example_factorization

Question 1.

Pourquoi préfère-t-on n=pq et pas plusieurs premiers

et pourquoi p et q proches ?

3 Que se passe-t-il s'il y a des facteurs premiers multiples ?

Exemple 2. $\mathbb{Z}/(3^2 5^3 7\mathbb{Z})$

3.1 Difficile de repérer les éléments du groupe des inversibles