Rapport sur l’algorithme p-1 de Pollard

Introduction

L'algorithme p-1 de Pollard est une méthode de factorisation d'entiers, développée par John Pollard en 1974.

Il s'agit d'une variation de l'algorithme de factorisation de Pollard basé sur l'idée que pour certains entiers n, p-1 est fortement composé pour certains nombres premiers p. L'algorithme est particulièrement efficace pour factoriser des nombres qui sont produits de deux nombres premiers relativement proches.

Principes de base

L'algorithme p-1 de Pollard repose sur le théorème de Fermat et sur la propriété des nombres fortement composés. Les principes de base sont les suivants :

Théorème de Fermat :

Le théorème de Fermat stipule que si p est un nombre premier et a est un entier non divisible par p, alors a^(p-1) est congruent à 1 modulo p. Cela signifie que pour tout entier a, a^(p-1) est un multiple de p.

Nombres fortement composés :

Un entier n est considéré comme fortement composé s'il a de nombreux facteurs premiers petits. Les entiers de la forme p-1 sont souvent fortement composés pour les nombres premiers p.

Fonctionnement de l'algorithme

L'algorithme p-1 de Pollard fonctionne comme suit :

Choix d'un entier B : On choisit un entier B qui est une borne supérieure pour les facteurs premiers de l'entier à factoriser.

Calcul de a^B modulo n : Pour chaque entier a aléatoire, on calcule a^B modulo n en utilisant l'algorithme de l'exponentiation modulaire.

Calcul du PGCD : On calcule le PGCD entre a^B modulo n et n en utilisant l'algorithme d'Euclide. Si le PGCD est différent de 1 et de n, alors on a trouvé un facteur non trivial de n.

Répétition avec différentes valeurs de a : Si aucun facteur non trivial n'est trouvé, on répète le processus avec différentes valeurs de a jusqu'à ce qu'un facteur soit trouvé ou jusqu'à ce qu'on atteigne un nombre maximal d'itérations.

Efficacité et Limitations

L'algorithme p-1 de Pollard est efficace pour factoriser des nombres qui sont produits de deux nombres premiers relativement proches, mais il peut être inefficace pour d'autres types de nombres.

Sa performance dépend de la valeur choisie pour B et du choix des entiers a. Il peut également échouer à factoriser des nombres qui ne répondent pas aux conditions nécessaires pour que l'algorithme réussisse.

Conclusion

L'algorithme p-1 de Pollard est une méthode utile et largement utilisée pour factoriser des entiers de grande taille, en particulier ceux qui sont produits de deux nombres premiers proches.

Bien qu'il puisse être inefficace pour certains nombres, il reste un outil important dans la boîte à outils des mathématiciens pour la factorisation d'entiers.

Implémentation de l'algorithme rho de Pollard

Bibliothèques utilisées :

- stdio.h : Pour les opérations d'entrée/sortie standard.
- stdlib.h : Pour les fonctions de manipulation de la mémoire.
- gmp.h : Pour les fonctions de manipulation des grands entiers à précision arbitraire.

Fonctions auxiliaires :

- gcd : Calcule le PGCD de deux nombres.
- p_minus_1_pollard : Implémente l'algorithme p-1 de Pollard pour la factorisation d'un nombre.

Algorithme p-1 de Pollard :

- La fonction p_minus_1_pollard prend en entrée un nombre entier n à factoriser et une borne B.

- Elle choisit un nombre aléatoire a et itère à travers les puissances de a modulo n jusqu'à atteindre la borne B.
- À chaque étape, elle calcule le PGCD de a^b - 1 avec n pour trouver un facteur non trivial.
- Si un facteur non trivial est trouvé, la boucle s'arrête et le facteur est retourné.

Fonction pour tester si un nombre est probablement premier :

- is_probably_prime : Teste si un nombre est probablement premier en utilisant la fonction mpz_probab_prime_p de la bibliothèque GMP.

Fonction principale (main) :

- La fonction principale demande à l'utilisateur d'entrer un nombre à factoriser.
- Si le nombre est probablement premier, elle affiche qu'il est probablement premier.
- Sinon, elle exécute l'algorithme p-1 de Pollard jusqu'à ce que le nombre à factoriser soit un nombre premier lui-même.
- À chaque itération, elle affiche les facteurs trouvés et divise n par ces facteurs pour répéter l'algorithme avec le quotient.
- Enfin, elle affiche le dernier facteur trouvé, qui est un nombre premier.