Rapport sur l’algorithme Rho de Pollard

Introduction

L'algorithme Rho de Pollard est une méthode de factorisation d'entiers, développée par John Pollard en 1975.

Il appartient à la classe des algorithmes probabilistes et est efficace pour factoriser des nombres de grande taille en utilisant des techniques probabilistes et heuristiques.

Principes de base

L'algorithme Rho de Pollard repose sur deux principes fondamentaux :

Méthode du cycle de Floyd :

Cette méthode est utilisée pour détecter les cycles dans des suites générées par des fonctions itératives. Elle repose sur le principe selon lequel, dans une suite finie de valeurs générées par une fonction, s'il y a répétition d'une certaine valeur, cela indique qu'il existe un cycle.

Idée du Frobenius :

L'algorithme Rho de Pollard exploite l'idée du théorème de Frobenius qui stipule que si deux nombres a et b sont choisis au hasard, alors le PGCD(a - b, n) fournit des informations sur les facteurs de n.

Fonctionnement de l'algorithme

L'algorithme Rho de Pollard fonctionne comme suit :

Choix d'une fonction itérative : On commence par choisir une fonction itérative qui génère une séquence de valeurs en fonction d'une valeur initiale.

Génération de la suite : À partir d'une valeur initiale, la fonction itérative est répétée pour générer une séquence de valeurs.

Détection de cycles : En utilisant la méthode du cycle de Floyd, l'algorithme détecte la présence de cycles dans la séquence générée.

Utilisation du PGCD : Une fois qu'un cycle est détecté, l'algorithme utilise le théorème de Frobenius pour calculer le PGCD entre des paires de valeurs dans le cycle et déterminer ainsi un facteur de l'entier à factoriser.

Répétition : Si aucun facteur n'est trouvé, l'algorithme répète le processus en choisissant une autre fonction itérative ou en ajustant les paramètres de la fonction actuelle.

Efficacité et Limitations

L'algorithme Rho de Pollard est efficace pour factoriser des entiers de grande taille, mais sa performance dépend de la fonction itérative choisie et des paramètres utilisés. Cependant, il peut être inefficace pour des entiers avec des facteurs premiers très grands.

De plus, comme il s'agit d'un algorithme probabiliste, il ne garantit pas toujours de trouver une factorisation, bien que sa probabilité de succès soit élevée dans la pratique.

Conclusion

L'algorithme Rho de Pollard est une méthode efficace et largement utilisée pour factoriser des entiers de grande taille. En combinant des principes mathématiques astucieux avec des techniques probabilistes, il fournit une approche puissante pour résoudre le problème difficile de 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.

next_element: Calcule le prochain élément dans la séquence utilisée par l'algorithme rho de Pollard.

Algorithme rho de Pollard :

La fonction rho_pollard prend en entrée un nombre entiern à factoriser et trouve un facteur non trivial de

n en utilisant l'algorithme rho de Pollard.

Elle génère deux nombres aléatoires x et y initiaux.

Ensuite, elle itère à travers une séquence de nombres générée par la fonction next_element.

À chaque étape, elle calcule la différence entre x et y, trouve le PGCD de cette différence avec n, et vérifie si le PGCD est un facteur non trivial de n.

Si un facteur non trivial est trouvé, la boucle s'arrête et le facteur est retourné.

Fonction principale (main) :

La fonction principale demande à l'utilisateur d'entrer un nombre à factoriser.

Elle appelle ensuite la fonction rho_pollard pour factoriser ce nombre.

Si le facteur trouvé est égal au nombre donné en entrée, elle affiche que le nombre est premier.

Sinon, elle affiche le facteur trouvé.