Test de Primalité

Jean-Didier Pailleux - Robin Feron - Romain Robert - Damien Thenot - Maxence Joulin

UVSQ

10/01/2018

Introduction

- **Nombre Premier :** Entier divisible par 1 et lui-même.
  pause
- **Test Probabiliste :**Test avec marge d'erreur très faible mais rapide.
- **Test Deterministe :** Test fiable mais plus lent.
- 2^{64} **:**Taille maximale des nombre a tester $=>$ Unsigned long long int
- **Nombre Hautement Composé :**Entier qui possède strictement plus de diviseur que les nombres qui le précède

Introduction

Plan

Architecture

Qui ?

Plan

Qui ?

Plan

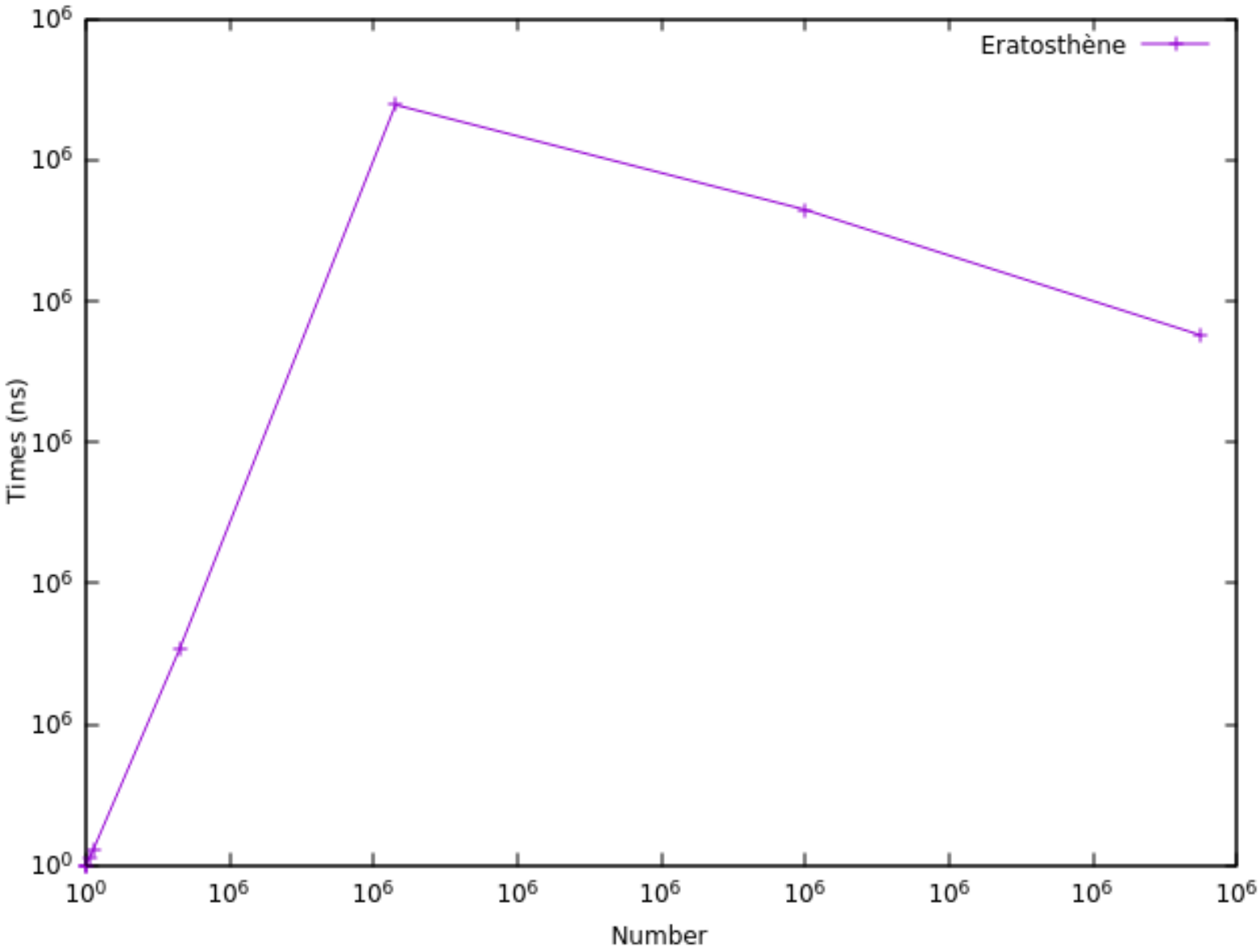
Fonctionnement du projet :

- Script appelé avec ./test.sh
  Options :

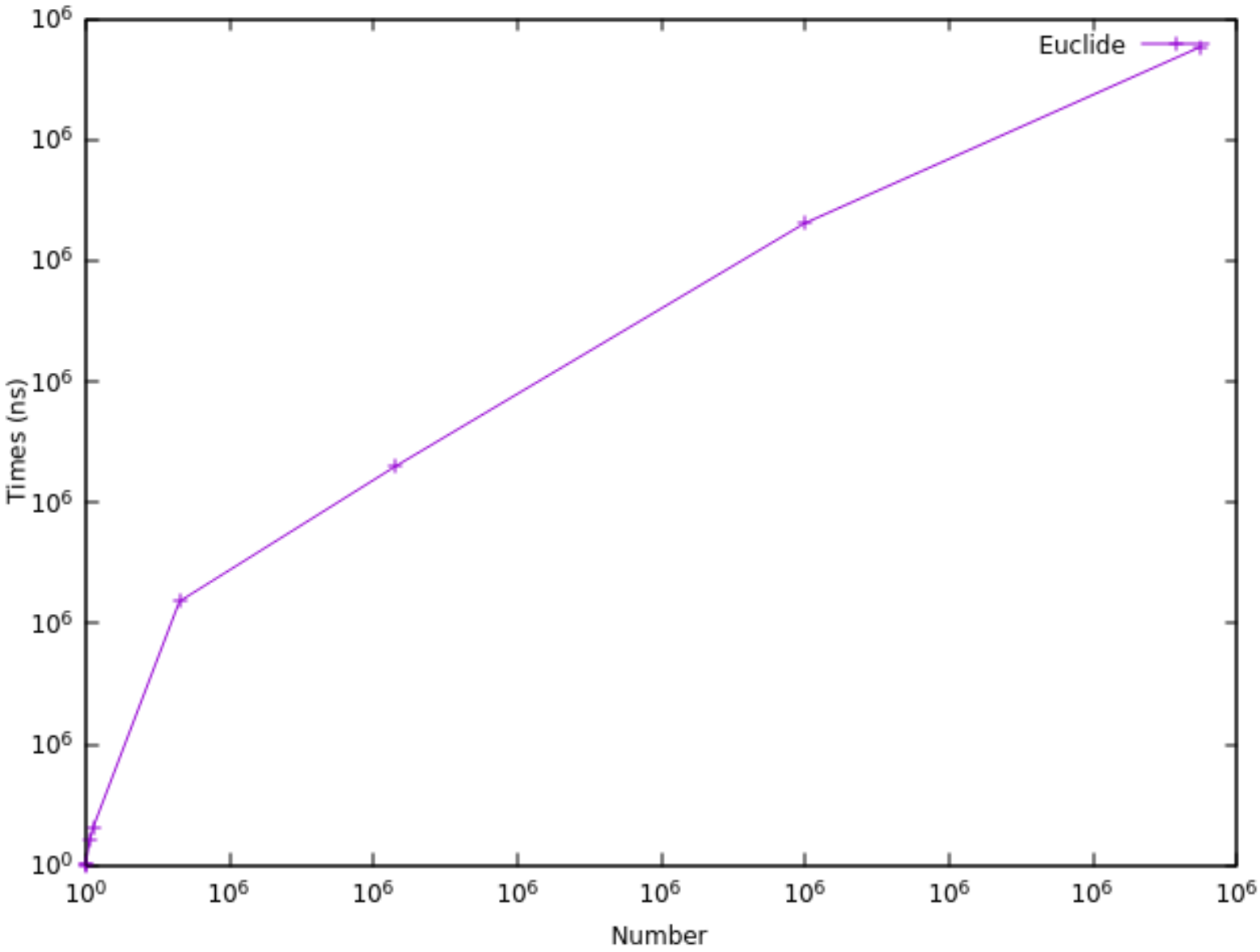
  a : Tous les algorithmes
  e : Euclide (computation bound)
  m : Crible d'eratosthene
  i : Miller-Rabin
  H : Nombre hautement composé def
  k : AKS
  o : Modulo (computation bound)
  p : Pocklington
  h : Nombre hautement composé naive

- Lequel utiliser ?
- Itération ?
- Combien de nombre ?
- Donner les nombres

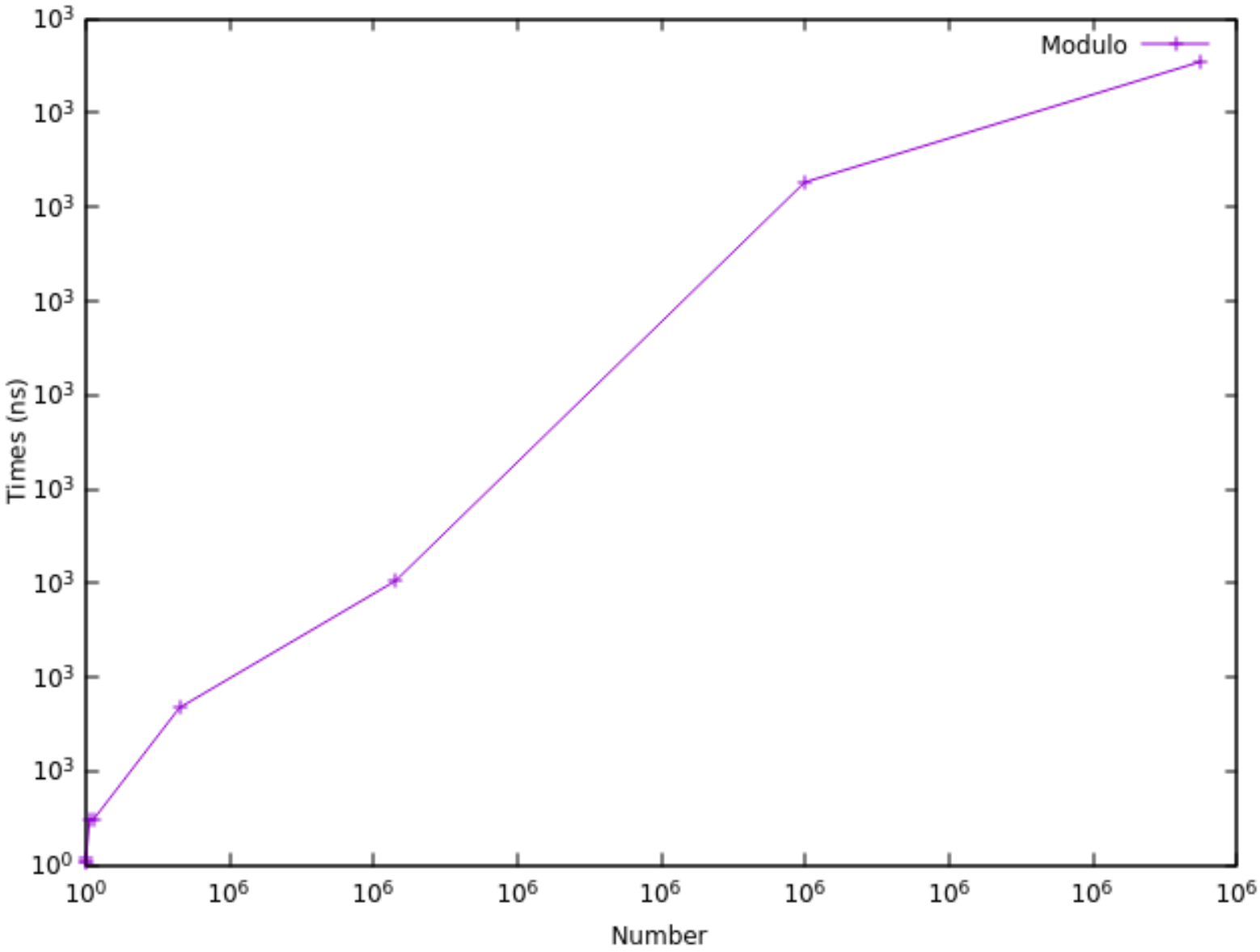
Evolution du temps d'exécution de Eratosthène/Memory Bound :



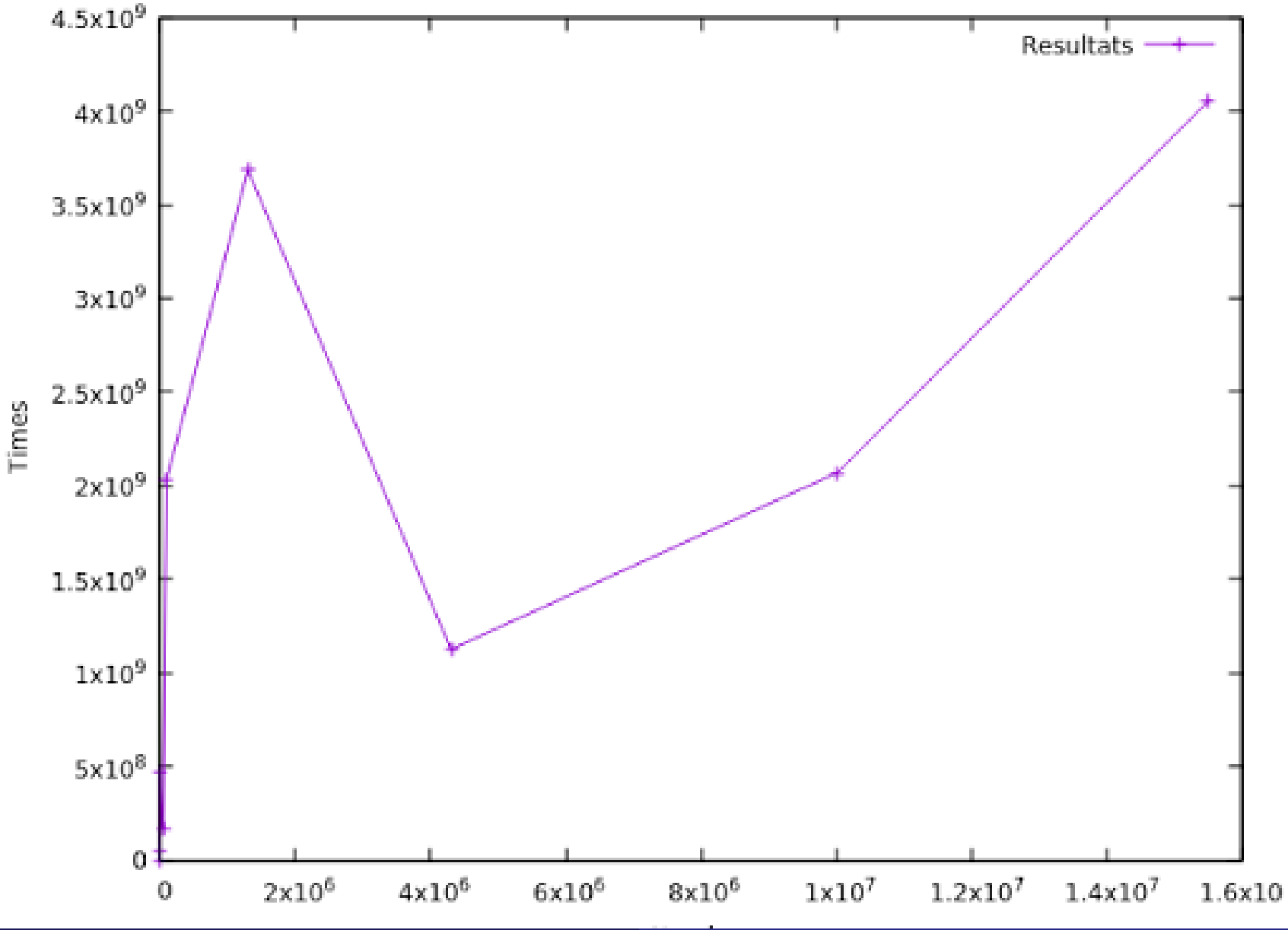
Evolution du temps d'exécution de Euclide/Computation Bound :



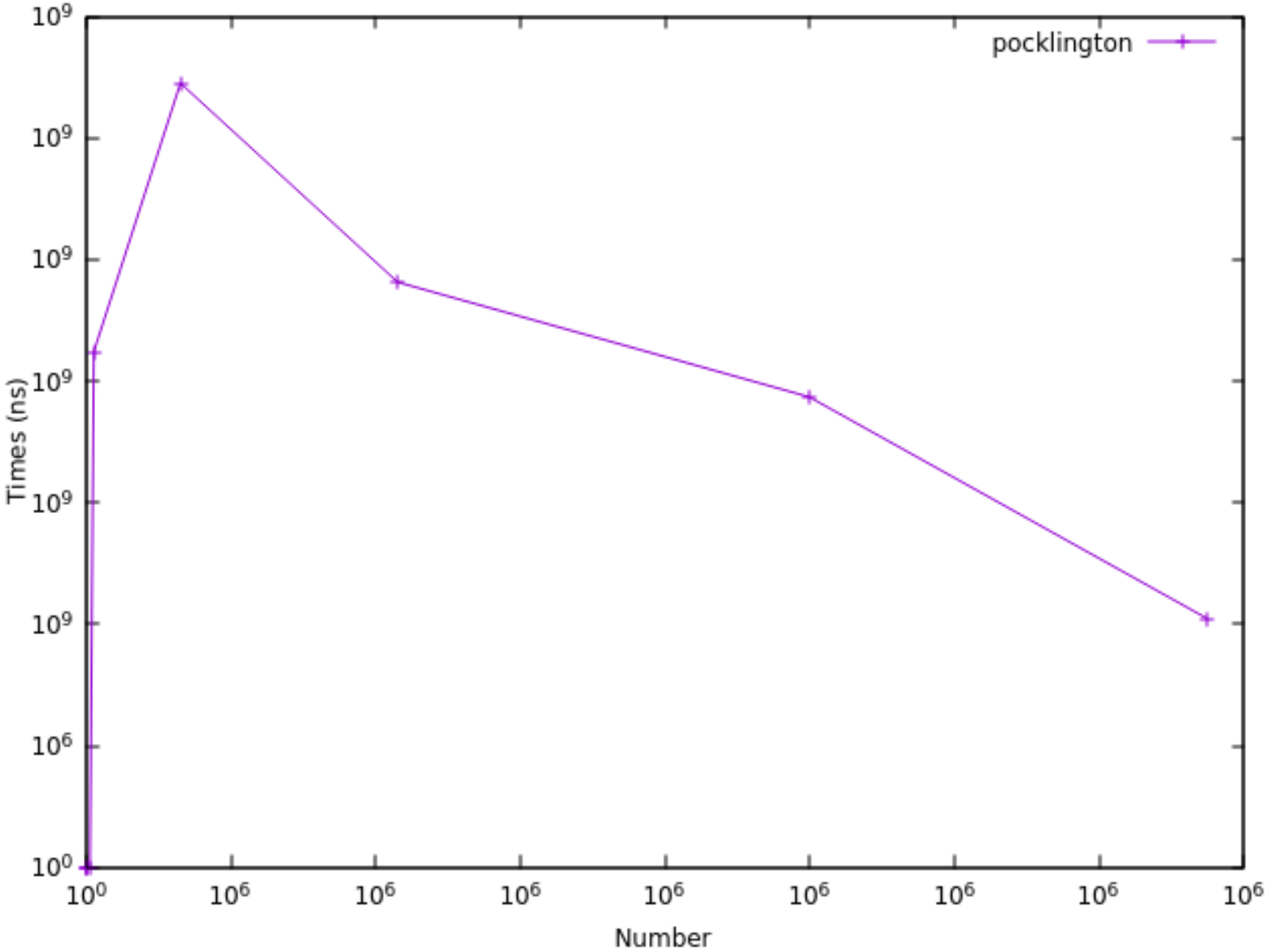
Evolution du temps d'exécution de Modulo/Computation Bound :



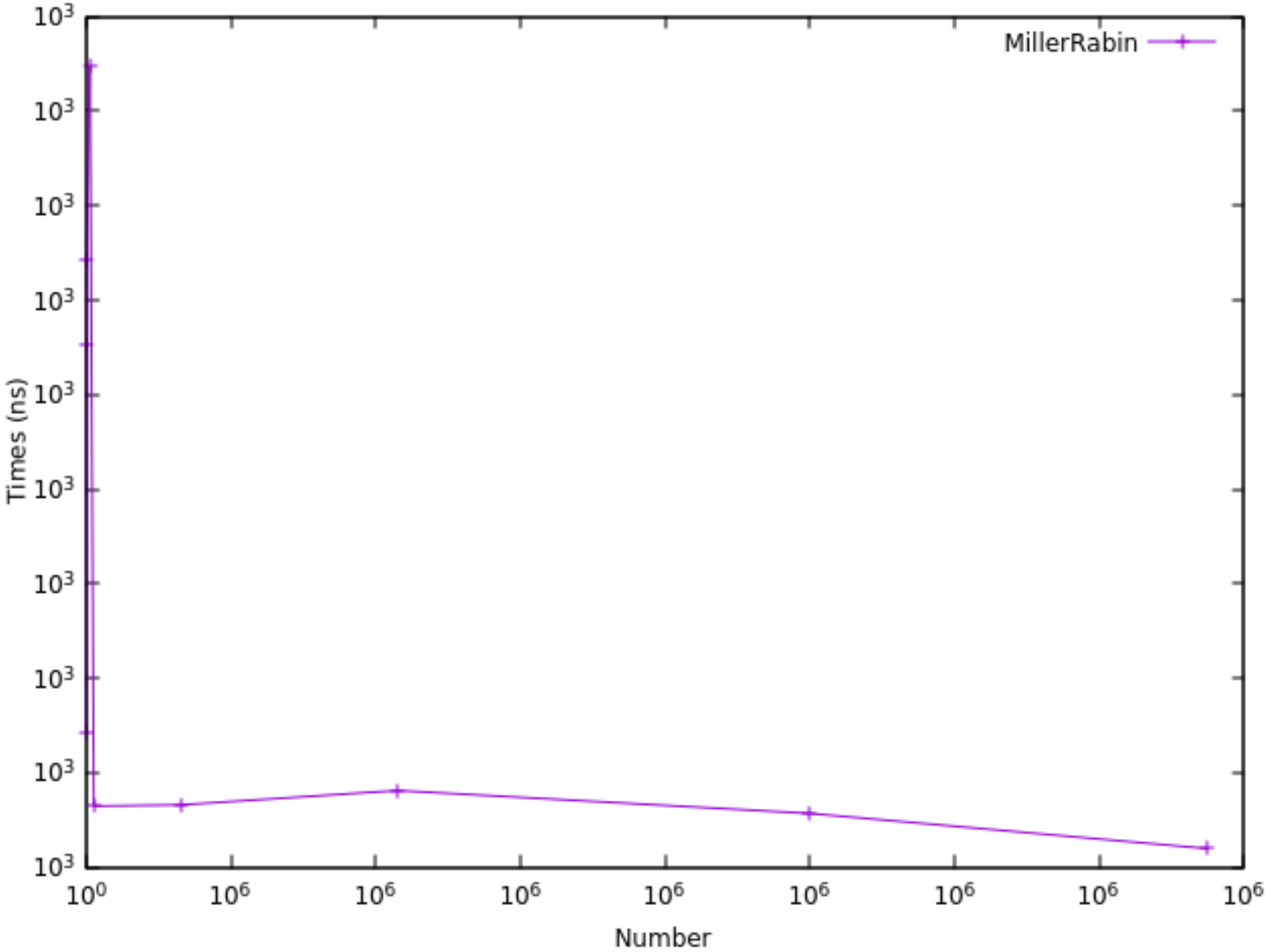
Evolution du temps d'exécution d'AKS :



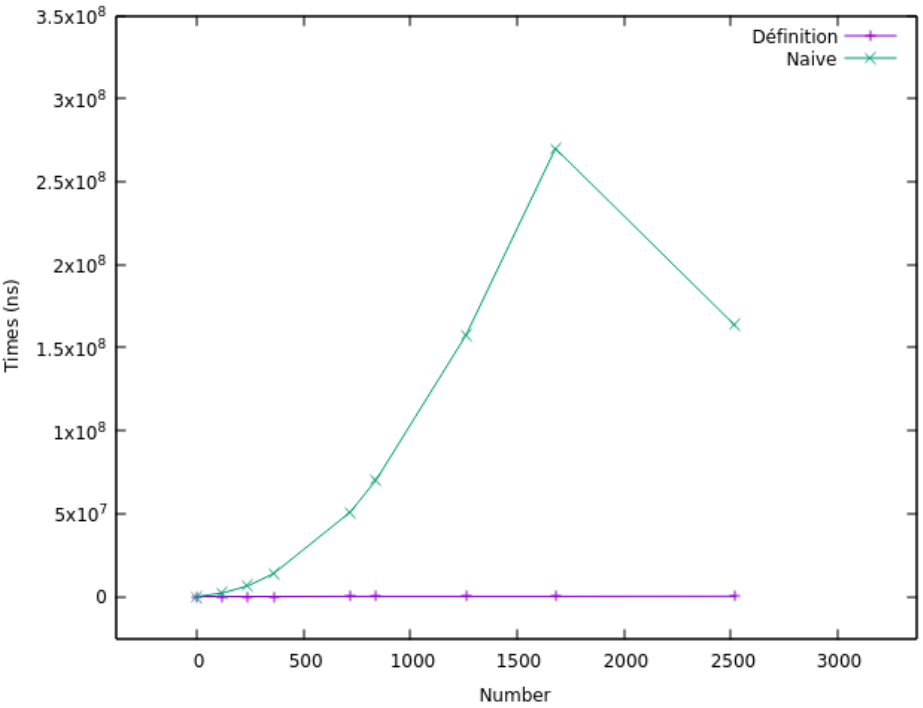
Evolution du temps d'exécution de Pocklington :

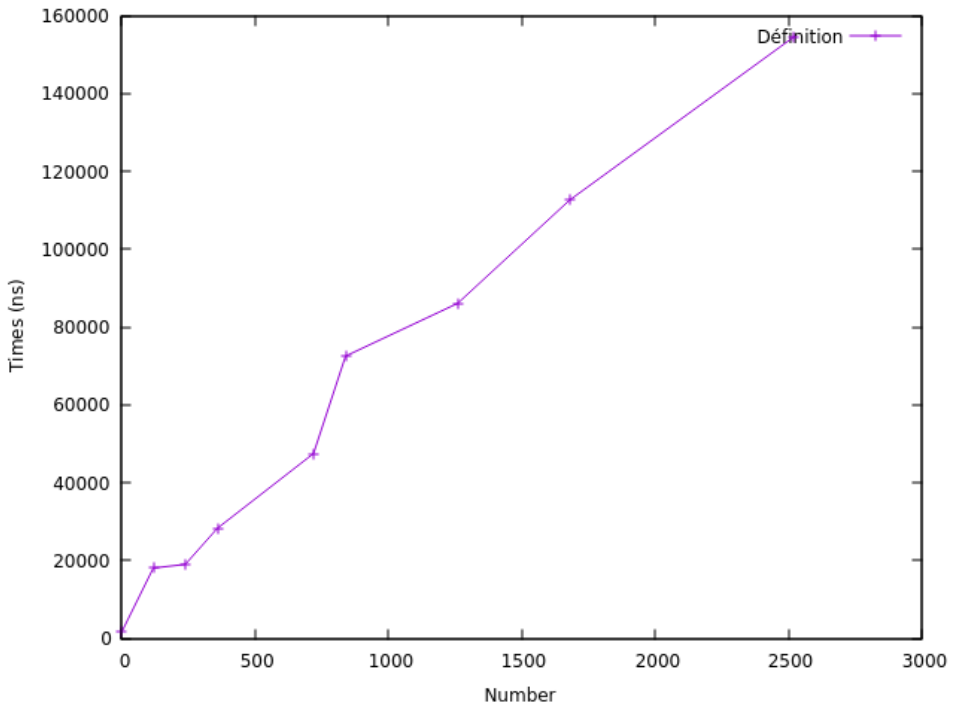


Evolution du temps d'exécution de Miller-Rabin :

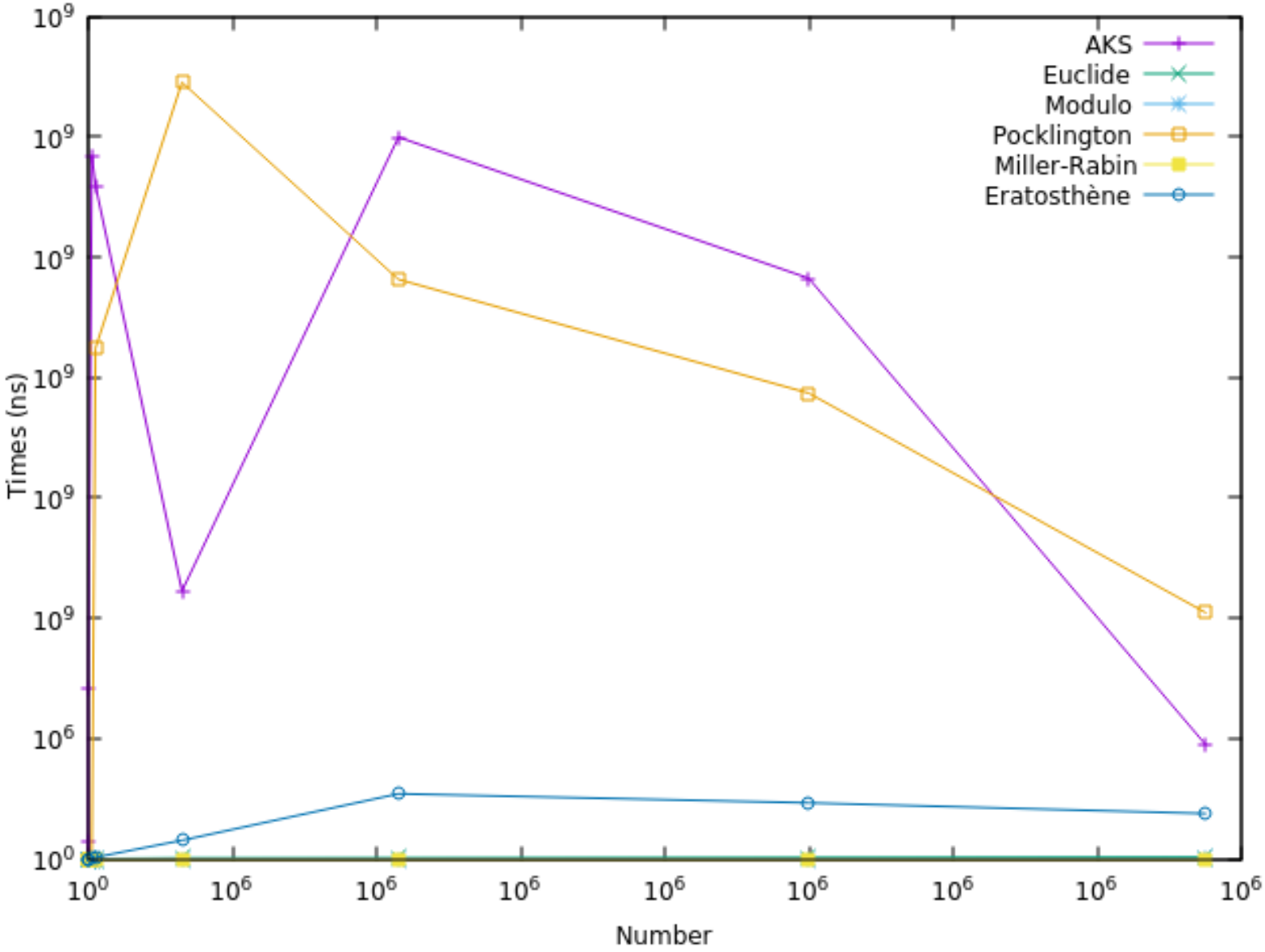


Evolution du temps d'exécution de Hautement composé (méthode naïve et définition) :





Analyse des Résultats



Plan

Bilan Technique

- **crible d'Eratosthène/ Mémory Bound :** Création d'un tableau de taille N+1 dans le crible => limité au niveau de la RAM pour N grand sur nos machines. Complexité de N pour le remplissage de la liste memory_bound.
- **Computation Bound :** Effectue $\sqrt{2^{log_2(n)}}$ divisions euclidiennes => Exécution en temps exponentiel
- **Pocklington :** Limite causé par la factorisation du nombre N-1 => facotisation très longues pour N très grand.

Bilan Technique

- **Miller-Rabin :** Résultats faux dans certains cas + Nombre d'itérations demandé élevé pour un meilleur résultat => augmentation du temps d'exécution.

- **AKS :** Avantage : Sa complexité en $log(n)^{12}$.
  Inconvénient : Utilisation de NTL qui effectue des vérifications superflue + Implémentation compliquée.

- **Hautement Composé :** N calcul du nombre de diviseurs d'un nombre => exécution très lente pour la méthode naïve.

Plan

Organisation interne du groupe

Tableau de répartition du travail :

Tâches	Jean-Didier	Maxence	Romain	Robin	Damien
Eratosthène/Memory Bound	x				
Euclide/Computation Bound		x			
AKS			x		
Pocklington					x
Miller-Rabin				x	
Highly Composite	x				
Cmake	x	x			
Script	x	x			

Plan

Conclusion

- De gros temps de calculs
- Difficultés d'implémentations
- Possibilités de parallélisation