BS PREMIERS

Calculs-Affichages de nombres premiers

Version initiale 21/09/2004
Maj. du 11/07/2023

Tables de Matières

1. Introduction

Vous trouverez dans cette documentation, des informations sur trois programmes de calculs et d'affichages de nombres premiers.

1.1 Les nombres premiers de Mersenne

Ce sont des nombres de la forme 2^p -1 où p est un nombre premier pour les nombres premiers de Mersenne. Il est à noter que tous les nombres de la forme 2^p -1 ne sont pas forcement premiers.

2. Les versions des programmes

Nom du programme	***Version***	***Date***	***Système / OS / Compilation***
bs_prime	1a	31/08/2002	Windows
liste_premier	1.0b		Linux
liste_premier	1.1a	15/09/2004	Windows
test_lm_bs	1.4k		Linux
test_lm_bs	1.4l	14/09/2004	Windows

Les programmes sont écrits en C. Ils ont été testés sous Linux avec gcc, Solaris avec gcc et cc ainsi que sous Windows avec Visual C++ et gcc.

3. Le programme liste_premier

3.1 Généralités

Ce programme permet de tester, calculer ou afficher des nombres premiers avec différentes méthodes. Selon la méthode utilisée, les capacités du programme sont plus ou moins limitées.

L'option pour tester les nombres de Mersenne via la méthode de Lucas-Lehmer est nettement la plus performante pour atteindre les grands nombres premiers.

3.2 Les options du programme

3.2.1 Usage

```
Usage : liste_premier -[option][param] [fichier]
-v : affichage de la version
-h : aide
-p1 n1 n2 [fichier] : liste des nombres premiers de n1 jusqu'au premier après n2
-p2 n1 n2 [fichier] : liste des nombres premiers de n1 a n2
-p3 n1 n2 max_exposant [fichier] : liste des nombres de Mersenne premiers (petit exposant)
-p4 n1 n2 max_exposant [fichier] : liste des nombres de Mersenne premiers (grand)
-p5 n1 [fichier] : test pour Mersenne premier (petit)
-p6 n1 max_exposant [fichier] : test pour nombre de Mersenne premier (avec indéterminé)
-p7 n1 [fichier] : test de Lucas-Lehmer pour Mersenne premiers (avec GNU MP)
-p72 n1 [fichier] : reprend un test arrete de Lucas-Lehmer pour Mersenne premiers (avec GNU
MP)
-p8 n1 [fichier] : affiche M(p) en base 10 (avec GNU MP)
```

Avec l'option -p7 vous pouvez tenter votre chance pour battre le record du monde!

Les options –p7 et –p8 ne sont disponibles que lorsque le programme est compilé avec la librairie GNUP MP.

[fichier] : est un fichier de résultat optionnel sinon la sortie se fera sur la sortie standard.

- Option –v

  Affiche des informations sur la version du programme. Si le programme est compilé avec la librairie GNU MP, il affiche la version utilisée.

  Exemple :

```
@(#)Prog.   : liste_premier
@(#)Auteur  : Bernard SAULME
@(#)Date    : 28/08/1999
@(#)Modif   : 15/09/2004
@(#)Version : 1.1a
@(#)Compil. : Sep 15 2004 20:08:01
@(#)GNU MP  : 4.1
```

- Option –p1 n1 n2 [fichier]

Affiche la liste des nombres premiers compris entre n1 et le premier nombre premier après n2.
La méthode de test pour savoir si un nombre est premier est le crible d'Erathostène (Eratosthene Sieve ***: anglais)***
.

- Option –p2 n1 n2 [fichier]

Idem a p1, mais avec jusqu'a n2 seulement !

- Option –p3 n1 n2 max_exposant [fichier]

Affiche une liste des nombres premiers de Mersenne pour les 'petits' exposants.
Test pour la liste des nombre premiers de n1 à n2, si M(n) est premier. n est premier et calculé par le programme par le crible d'Erathostène.
Le test est fait en divisant par tous les nombres premiers jusqu'à sqr(n). Ce test est très long pour n > 58. De plus il est limité par les modulos sur 32 bits (dépendant du système). Permet d'éliminer les nombres non premiers car divisibles avec des petits diviseurs.

Exemple :

```
liste_premier_10e -p3 3 60 20

M(p) pour p = 3 => P
M(p) pour p = 5 => P
M(p) pour p = 7 => P
M(p) pour p = 9 => N
M(p) pour p = 11 => N
M(p) pour p = 13 => P
M(p) pour p = 15 => N
.
.
.
M(p) pour p = 51 => N
M(p) pour p = 53 => N
M(p) pour p = 55 => N
M(p) pour p = 57 => N
M(59) Modulo 646927
M(p) pour p = 59 => ?
Il a 29 tests M(p) de 3 a 60
Il a 1 ? de 3 a 60 pour max exp modulo 20
Il a 7 P de 3 a 60 pour max exp modulo 20
Temps total : 3.000 seconds
```

Si le nombre est premier alors il est qualifié de Ps sinon N. Dans le cas ou le programme n'a pu définir (finir le calcul avec tous les diviseurs) le nombre est qualifié de ? (Indéfini). Lors de calcul longs le programme affiche les informations intermédiaires (`M(59) Modulo 646927)`.

- Option –p4 n1 n2 max_exposant [fichier]

Affiche une liste des nombres premiers de Mersenne pour les 'grands' exposants.
Test pour la liste des nombre premiers de n1 à n2, si M(n) est premier.
Si le programme est compilé avec la librairie GNU MP, le test est fait en utilisant le test de Lucas-Mehmer, éventuellement.

Si n'est pas compilé avec la librairie GNU MP, les calculs se font avec l'arithmétique du processeur => calcul sur les petits nombres de n uniquement.

En fonction de max_exposant.
max_exposant = 0 Lucas-Lehmer
max_exposant > 0 Normal puis si besoin Lucas-Lehmer
max_exposant < 0 Normal (sans Lucas-Lehmer)

L'option 'Normal' utilise l'algorithme d'exponentiation rapide pour éliminer les nombres non premiers.

Affichage du résultat :
Comme pour l'option –p3 : P = Premier, N = Non premier, ? Indéfini.

- Option –p5 n1 [fichier]

Test de Lucas-Mehmer avec l'arithmétique du processeur pour un nombre n1.

- Option –p6 n1 max_exposant [fichier]

Test si un nombre de Mersenne est premier en utilisant l'arithmétique du possesseur. C'est la méthode utilisée par les options –p3 et –p4 pour tester su un nombre est divisible par les 'petits' exposants.

Utilise l'algorithme d'exponentiation rapide pour éliminer les nombres non premiers.

Affichage du résultat :
P = Premier, N = Non premier, ? Indéfini.

- Option –p7 n1 [fichier]

Test si un nombre de Mersenne est premier en utilisant l'algorithme de Lucas-Lehmer. Cette option utilise la librairie GNU MP.
C'est option permet de faire les tests jusqu'au p = 2*31.

Le test sauvegarde les résultats intermédiaires tous les n seconds (n = 1800 en général, selon les compilations) pour permettre l'éventuelle reprise de calcul en cours. Les informations sont stockées dans les fichiers lm_bs_P.dat et lm_bs_P.dat_old où P est égal à p (exemple : lm_bs_31.dat si p 31). Le fichier _old est la sauvegarde 'n-1'.

- Option –p72 n1 [fichier]

Reprend un test d'un nombre de Mersenne est premier en utilisant l'algorithme de Lucas-Lehmer.
Si le fichier de sauvegarde (crée par l'option –p6 ou -p7) est absent, le calcul entier est effectué.
Les caractéristiques du test sont les mêmes que pour l'option –p7.

- Option –p8 n1 [fichier]

Affiche si un nombre de Mersenne en base 10. Cette option utilise la librairie GNU MP.

4. Le programme test_lm_bs

4.1 Généralités

Ce programme permet de vérifier si un nombre de Mersenne est premier via la méthode de Lucas-Lehmer. Il n'utilise aucune librairie externe.

4.2 Les options du programme

4.2.1 Usage

```
Usage : test_lm_bs -[v|p|r|t|t1|t2] [p] [fichier]
-v : version
-p p [fichier] : test L.M. pour p
-r p [fichier] : reprise d'un test L.M. pour p
-t p : affiche la taille mémoire allouée pour -p p
-t1 p : test d'une liste de M(p) avec le test L.M jusqu'a p
-t2 p : test L.M jusqu'a p pour tous les premiers compare avec une liste de M(p)
```

Nota : l'option –p est identique en fonctionnalité à l'option –p7 du programme *liste_premier*.

Les résultats des tests du programme du programme sont historisés dans le fichier « lm_bs_historique.txt »

- Option –v

  Affiche des informations sur la version du programme

  Exemple :

```
@(#)Prog.   : test_lm_bs
@(#)Auteur  : Bernard SAULME
@(#)Date    : 07/11/1999
@(#)Modif   : 08/09/2002
@(#)Version : 1.4k/32b/PCwin
@(#)Compil. : Sep  8 2002 18:24:52
```

- Option –p p [fichier]

Test si un nombre de Mersenne est premier. P doit être premier pour avoir le test de Lucas-Lehmer valide. Si p n'est pas premier le programme affiche un message le signalant. Cette vérification est faire en utilisant le crible d'Erathostène.

Le fichier de sortie est optionnel.

Le programme sauvegarde les tests intermédiaires à intervalles réguliers pour permettre l'éventuelle reprise de calcul en cours. Les informations sont stockées dans les fichiers lm_bs_P.dat et lm_bs_P.dat_old où P est égal à p (exemple : lm_bs_31.dat si p 31). Le fichier _old est la sauvegarde 'n-1'.

- Option –r p [fichier]

Idem que l'option –p p [fichier] a la différence que le calcul est repris à l'endroit arrêté précédemment.

- Option –t p

Affiche la taille de la mémoire dynamique allouée par le programme pour permettre le calcul de p.

```
Ex : test_lm_bs_14k -t 100
Tableaux (de 4 octets) de dimension 66
Memoire allouee pour le test de M(1000) : 792 octets
```

- Option –t1 p

Test d'une liste de M(p) avec le test de Lucas-Lehmer. Le programme commence a M(3) et fini au dernier de la liste des nombres premiers de Mersenne connus au moment de la compilation du programme ou au nombre p passé en paramètre (si p inférieurs au nombre connu). Cette option sert à vérifier le bon fonctionnement du programme avec les nombres premiers de Mersenne connus.

- Option –t2 p

Test les nombres de M(p) premiers, avec le test de Lucas-Lehmer, pour tous les p premiers (calculés par le programme via le crible d'Erathostène) et les comparent avec la liste de nombre de M(p) premiers connus (au moment de la compilation du programme) les uns après les autres.
Commence au nombre p=1 fini au dernier nombre p premier inférieur ou égal au nombre donné en paramètre.
Dans le cas ou un nombre M(p) n'est pas trouvé premier alors qu'il est dans la liste alors le programme affiche une erreur. De même si est dans la liste alors que le programme ne le trouve pas premier.

5. Le programme bs_prime

5.1 Généralités

Ce programme permet l'affichage en base 10 d'un nombre de Mersenne.
L'algorithme est crédité à Slowinski de CRI. Ce programme a été récupéré sur Internet et modifié par Bernard pour fonctionner sur PC et être compilé sous Visual C++.

5.2 Les options du programme

Pas d'option sous la version courante. L'appel du programme sans paramètre affiche $2^{859433} - 1$. Le programme ne prend qu'un paramètre, l'exposant du nombre de Mersenne à afficher.

Usage : bs_permier exposant.

Nota : l'option –p8 du programme *liste_premier* qui utilise la Librairie GNU MP est plus rapide que le programme bs_prime.

6. Méthodes et Algorithmes

6.1 Le test de Lucas-Lehmer

Ce test permet de définir si un nombre de Mersenne est premier ou non. Le test de Lucas pour les nombres premiers a été redéfini par Lehmer en 1930.

Le test de Lucas-Lehmer définie qu'un nombre de Mersenne $M_p = 2^p - 1$ (avec p premier et supérieur à 2) est premier si et seulement si M_p divise $S(p-1)$ avec $S(1) = 4$ et $S(n+1) = S(n)^2-2$ pour $n> 1$.

6.2 Exponentiation rapide

Pour effectuer le calcul de puissances a^e (**mod** n) pour de grandes valeurs de l'exposant e. Il existe en effet un algorithme très simple pour effectuer rapidement le calcul.

```
long puissance (long a, long e, long n) {
  long p;

  for (p = 1; e > 0; e = e / 2) {
    if (e % 2 != 0)
      p = (p * a) % n;
    a = (a * a) % n;
  }
  return p;
}
```

Cette fonction repose sur l'écriture de l'exposant e en numération binaire. Si par exemple e s'écrit 1011001, on a :
$e = 1 + 8 + 16 + 64 = 89$ et $a^e = a * a^8 * a^{16} * a^{64}$
La suite a, a^2, a^4, a^8, a^{16}, a^{32}, a^{64} s'obtient par une suite d'élévations au carré. Pour calculer $p = a^e$ il suffit donc, pour chacun des chiffres binaires de l'exposant e (lus de droite à gauche), d'effectuer les opérations suivantes :

1. Multiplier p par a si le chiffre binaire vaut 1 ;
2. Élever a au carré.

Remarque : cette fonction donne des résultats incorrects si l'un des produits déborde avant la réduction modulo n ; telle quelle, elle ne peut donc être utilisée que pour des valeurs de n inférieures à 2^{16} pour un calcul en 32 bits.

Complexité : le nombre total de multiplications est égal au nombre de chiffres de l'exposant e en numération binaire (pour les élévations au carré), plus le nombre de chiffres égaux à 1 ; soit 7 + 4 = 11 multiplications pour calculer a^{89}. Cette complexité est donc *linéaire* en fonction du *nombre de chiffres* de l'exposant, autrement dit proportionnelle au *logarithme* de l'exposant.

7. Références

- La présente documentation : https://softs.saulme.fr/download/download.php?h=application/pdf&d=bs_premiers.pdf
- GIMPS Great Internet Mersenne Prime Search : https://www.mersenne.org/
- Test de Lucas-Lehmer : http://www-groups.dcs.st-and.ac.uk/~history/Mathematicians/Lucas.html
- Transformée de Fourier rapide : http://brassens.upmf-grenoble.fr/IMSS/mamass/graphecomp/Fourier/Fourier.html
- Algorithmes de bases de calcul numériques : https://www.labri.fr/Perso/~betrema/deug/poly/exp-rapide.html
- GNU MP : https://gmplib.org/