Se souvenir de la librairie `gb-tools`, version 0.7.0, développée en décembre 2010 par Daniel Diaz, Denise Vella-Chemla, mai 2026

En décembre 2010, j'avais raconté à Daniel Diaz [1] mes recherches autour de la conjecture de Goldbach et il avait programmé un logiciel très professionnel, qui s'appelle gb-tools. Il l'a modifié jusqu'à la version 0.7.0 et la version compressée est disponible sur la toile ici :

Cliquez ici pour télécharger le fichier .tar.

Tout était utile et nécessaire : le fichier d'accompagnement, un petit manuel d'utilisation miniature, la licence d'utilisation, et tous les programmes en `c`. Je reproduis ci-dessous, en souvenir.

1. ***Fichier*** `Readme.txt`

Goldbach Conjecture Tools
by Daniel Diaz
daniel.diaz@univ-paris1.fr
Dec 2010

Introduction

This set of tools has been initially developped for Denise Chemla <chemla.denise@orange.fr> who devote many years in the study of the conjecture. Most of her work can be found here : http://denise.vella.chemla.free.fr/.

Goldbach Conjecture

Goldbach Conjecture : "every even natural number n can be expressed as a sum of two primes $p+q$ (for $n \geq 4$)".

Until now we still don't have a complete proof of the conjecture.

NB : if we except the case $n = 4$ in any decomposition (p, q) of $n = p + q$ are both p and q odd primes (for $n = 4$ we have $4 = 2 + 2$, it is the unique case).

We are here interested in the number of decompositions (or partition) existing for an even number n ($n = p + q$ with $p \leq q$).

1. Daniel Diaz était en 1990 un chercheur de l'Inria avec qui j'avais travaillé lorsque j'étais ingénieure de recherches pour la société Syseca, alors que j'étais en poste au CENA (Centre d'Études de la Navigation Aérienne, basé à Athis-Mons) sur un projet de recherche concernant l'allocation de créneaux au décollage des avions. J'avais alors utilisé un logiciel qu'il avait développé et qui s'appelait `CLP(FD)` pour `Constraint Logic Programming (in Finite Domains)` et nous avions écrit un article pour le symposium d'Ithaca (New-York) 1994, ILPS'94 (International Logic Programming Symposium).

NB : there is no clear notation for this number. We follow [Oliveira e Silva] and denote $r(n)$ the number of Goldbach partitions of n.

The Goldbach conjecture is then equivalent to the statement that $r(n) > 0$ for every even integer $n > 3$.

Compiling the tools

This has only been tested under Linux with gcc 4.4.5 but should with few (or no) modifications on other architectures.

simply type :

```
make
```

Usage

The tools work with plain text files (this is not the most compact format but makes it possible to use standard text utilities like `grep`, `sed`, ...).

There are several tools :

gb-decomp :
computes the decompositions of even N. It can create a data file (preferably suffixed .gb) storing all this information.
The file is a plain text file formated as follows :
$start..stop$ = the first/last value for which the $r(n)$ is stored in the file
$r(start)$ = number of decomp of $start$
$r(start + 2)$ = number of decomp of $start + 2$
$\vdots$

$r(stop)$ = number of decomp of $stop$
NB : this can take very long time for big values !

gb-check : checks the consistency of a data file.

gb-showdata : this extracts information from a `.gb` file and shows it

gb-minor : this makes it possible to experiment various minoration function (to have a lower bound of $r(n)$).

gb-randata : extract randomly some values in a `.gb` file (e.g. for graph representation).

There are 2 utility files :

gb-utils.c : is a utility file used by all above programs.

gb-funcs.c : allows the user to define new functions (e.g. for a lower bound). Once compiled the functions give rise to a C function in gb-funcs.h. The file gbfuncs.c contain several function definitions which can be used as examples.

All this utilities accept a command-line as follows

[OPTIONS] [START][..STOP] [DATA_FILE_NAME]

OPTIONS are dependent from the utility (use `-h` to have a help)

START..STOP (one or both can be given) specifies the range of values

DATA_FILE_NAME name of the `.gb` file

Examples of Use

Create a data file `d.gb` for [0..1000000].

```
> gb-decomp 0..1000000 d.gb
***** End sieve in 0..1000000 there are 78498 primes (biggest is 999983)
```

Check some values of the `d.gb` file

```
> gb-check -n 30% d.gb
Current used seed: 1292109311
Read from file 4..1000000 = 499999 values
Checking 149999 values randomly (1 value per group of 6.0 values)
***** End sieve in 0..1000000 there are 78498 primes (biggest is 999983)
Verification OK
```

View the decompositions of 100

```
> gb-decomp -d 100
***** End sieve in 0..100 there are 25 primes (biggest is 97)
    3 + 97 = 100
   11 + 89 = 100
   17 + 83 = 100
   29 + 71 = 100
   41 + 59 = 100
   47 + 53 = 100
    100 has 6 decomp
```

View the decompositions of 80..100 :

```
> gb-showdata 90..110 d.gb
 90 has              9 decompositions
 92 has              4 decompositions
 94 has              5 decompositions
 96 has              7 decompositions
 98 has              3 decompositions
100 has              6 decompositions
102 has              8 decompositions
104 has              5 decompositions
106 has              6 decompositions
108 has              8 decompositions
110 has              6 decompositions
```

Test a minoration (modify gb-minor.c to put your own minoration function) :

```
> gb-minor d.gb -f Dan4
Minoration failures: 0

Not taking into account multiples of 6

    max distance r - f: 4542 for value 950950
    avg distance r - f: 598
    max ratio r / f: 6.00 for value 100
    avg ratio r / f: 1.36
```

Extract some values in an output file lines.dat

```
> gb-randata -n 1000 d.gb -o lines.dat
Current used seed: 1291990473
Read from file 4..1000000 = 499999 values
Extracting 1000 values randomly (1 value per group of 998.0 values)

> cat lines.dat
 882            39
1174            22
2610            98
3254            44
4414            49
5536            65   ···
```

Extract some values in 0..10000 + applying functions

```
> gb-randata -n 100 -o lines.dat -f Dan3 -f Dan4 -f Dan5 0..10000 d.gb
Current used seed: 1291990307
```

```
Read from file 4..10000 = 4999 values
Extracting 100 values randomly (1 value per group of 50.0 values)

> cat lines.dat
```

58	4	1	2	2	
120	12	2	3	9	
300	21	3	6	16	
394	11	4	8	7	
446	12	4	8	7	
534	22	5	9	18	
694	19	6	11	10	
716	14	6	12	11	
828	34	7	13	25	
976	19	8	15	13	···

The produced file can be used to plot a graph (e.g. with gnuplot). It is possible to ask gb-randata to produce a skeleton of gnuplot commands (with -g) :

```
> gb-randata -n 1000 d.gb -o lines.dat -g f.gplot

Current used seed: 1291990745
Read from file 4..1000000 = 499999 values
Extracting 1000 values randomly (1 value per group of 998.0 values)

> cat f.gplot

set terminal postscript eps
set output "graph.eps"
set title "Goldbach partitions"

set xlabel "n"
set ylabel "number of Goldbach partitions"

set xtics nomirror
set ytics nomirror

set key left

set xrange [XMIN:XMAX]
set yrange [YMIN:YMAX]

pow(x,n) = x ** n

plot "lines.dat" using 1:2 with dots title "r(n)"
```

It can be used as follows :

```
> gnuplot
gnuplot> load "f.gplot"
```

It is then possible to (re)plot some functions like

```
gnuplot> replot ceil(0.01647 * pow(x, 0.89)) title "0.01647n^0.89"
```

Using gb-randata to select a lot of points will produce a graph representing the well-known Goldbach Comet.

NB : experiment changing 'dots' by 'points' or 'lines' or 'linespoints'.

NB : we defined the pow function to cut/paste C code (see gb-minor.c)

Links

http://denise.vella.chemla.free.fr/ (aujourd'hui https://denisevellachemla.eu)
https://mathworld.wolfram.com/GoldbachPartition.html
https://oeis.org/A045917
https://sweet.ua.pt/tos/goldbach.html

2. *Fichier* `License.txt`

Neither the names of Denise Chemla, Daniel Diaz, nor the names of its contributors may be used to endorse or promote products derived from this Software without specific prior written permission.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE CONTRIBUTORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS WITH THE SOFTWARE.

Voici la comète jusqu'à 10^6 générée par ce programme.