



The Museum of HP Calculators

Search Member List Calendar Help

Welcome back, **Valentin Albillo**. You last visited: Yesterday, 10:28 PM ([User CP](#) — [Log Out](#))

[View New Posts](#) | [View Today's Posts](#) | [Private Messages](#) (Unread 0, Total 145)

Current time: 05-01-2019, 02:31 AM

[Open Buddy List](#)

[HP Forums](#) / [HP Calculators \(and very old HP Computers\)](#) / [General Forum](#) ▼ / [\[VA\] SRC#005- April, 1st Mean Minichallenge](#)

Pages (2): [« Previous](#) [1](#) [2](#)



[VA] SRC#005- April, 1st Mean Minichallenge

[Threaded Mode](#) | [Linear Mode](#)

04-13-2019, 01:33 PM

Post: #21

Bernd Grubert

Member

Posts: 69

Joined: Dec 2013

RE: [VA] SRC#005- April, 1st Mean Minichallenge

Hello Valentin,

Thanks for your kind and encouraging words. I will give Tier 2 of S&SMC#24 a try.

Valentin Albillo Wrote: →

(04-13-2019 02:08 AM)

The trick of computing the *Geometric Mean* by adding logarithms and then a final exponential is perfectly *Ok* for strictly positive datasets, as in this case, and the precision lost amounts to but a few *ulps* in the worst case, so it's perfectly acceptable as well.

You're right it works only for positive data sets. I didn't notice this.

Best regards

Bernd



04-15-2019, 11:29 PM (This post was last modified: 04-16-2019 08:54 AM by Mike (Austria).)

Post: #22

Mike (Austria)

Junior Member

Posts: 5

Joined: Sep 2018

RE: [VA] SRC#005- April, 1st Mean Minichallenge

Thanks Valentin for this inspiring minichallenge. I particularly enjoy math riddles when they involve little-known facts about well-known mathematics. This one definitely fits this description particularly well - the fact that the various means are discrete members of a continuous family had escaped my attention so far (for decades - embarrassingly enough :-)).

Since there have been no solutions for RPL, let me share my take on this. It should work for any version of RPL starting with the HP48G(X).

1. RPL function **GM** for the [Generalized Mean](#) function $G_p(x_1, \dots, x_n) = \left(\frac{1}{n} \sum_{i=1}^n x_i^p\right)^{1/p}$.

Code:

```
«
@ {list} real → real
@ {x_i} p → gm(x,p)

OVER SIZE → x p n @ n is size(x)
«
x @ prepare x on stack
IF p 0 == THEN @ geometric mean
  ΠLIST @ prod(x_i)
  n @ prepare exponent on stack
ELSE @ general case
```

Size: 106.5 bytes.

Exponent $p = k - 2$

Usage: put list $\{x_i\}$ and exponent p on stack, invoke GM.

I like the terseness of this function.

2. Test: four standard means

Code:

```
{ 4 1 20.19 } 'X' STO @ save list
X -1 GM -> 2.30852787042 @ harmonic
X 0 GM -> 4.32247115133 @ geometric
X 1 GM -> 8.39666666667 @ arithmetic
X 2 GM -> 11.8972840038 @ quadratic
```

3. Solving for $k = p + 2$ such that $G_p(x_i) = \pi$

Auxiliary function **G**: for list $\{x_i\}$ and exponent p in global variables X and P, return $GM(x,p)-\pi$

Rationale: the HP48 solver expects all parameters and the unknown to be solved for (p in this case) in global variables.

Code:

```
« X P GM π - »
'G' STO
```

Find zero of G

Code:

```
'G' RCL @ recall G to stack
'P' @ variable to solve for
0 @ initial guess
ROOT @ solve
2 + @ p -> k
-> 1.55425243274
```

4. Bonus: Code for [Generalized f-Mean](#) $G_f(x) = f^{-1} \left(\frac{1}{n} \sum_{i=1}^n f(x_i) \right)$

A further step in generalizing the mean is by replacing the p -th power of x_i by applying an arbitrary continuous strictly monotonic function and taking its inverse after computing the sum and dividing by n . Not very surprising, the geometric mean results from using $f(x) = \ln(x)$ the natural logarithm. Btw, the requirement that all x_i be positive is actually not a restriction because the [geometric mean](#) is defined for positive arguments only in the first place.

Since in general the inverse of f cannot be assumed to be known analytically, we first define an *auxiliary function* **FINV**, which, given f, y , and an estimate e , will return an approximation of $x = f^{-1}(y)$ by numerically solving $f(x) = y$ for the unknown x .

Code:

```
«
@ function real real → real
@ f y estimate → x such that f(x)=y

→ f y e
«
« tX f EVAL y - » @ find zero of this function
'tX' @ temporary global variable to solve for
```

```
e          @ initial estimate
ROOT
{ tX } PURGE @ eliminate tX
```

This is essentially the inverse function solver on page 2-54 of the HP48G Advanced User's Manual (AUM). I had initially hoped to use a compiled local variable (this is a local variable whose name starts with a backarrow ←) because the AUM promises that this can be used whenever a called function expects a global variable - but ROOT does not let me do this, thus the clumsy { tX } PURGE at the end.

With this complexity hidden away, it becomes straightforward to implement the generalized f -Mean function **GFM**:

Code:

```
«
@ {list} function → real
@      {x_i} f → gfm(x,f)

OVER SIZE → x f n    @ n is size(x)
«
f          @ prepare f: will be used by FINV
x f EVAL   @ f(x)
ΣLIST n /   @ sum(f(x_i)) / n
x 1 GET     @ first element as initial guess
```

Why pick one of the original list elements to estimate the solution: we make no assumptions about the domain of f . As long as the user supplies a list of valid elements, any of them is suitable as the starting point for the ROOT solver of the HP48, which is quite robust. This is cheaper than e.g. taking the arithmetic mean as an estimate or bracketing the solution by searching the minimum and maximum of $\{x_i\}$.

Examples:

Code:

```
X « INV » GFM -> 2.30852787043
X « LN » GFM -> 4.32247115132
X « » GFM -> 8.39666666667 @ empty program = identity function
X « SQ » GFM -> 11.8972840038
X « EXP » GFM -> 19.091387809 @ smooth maximum approximation
```

Up to the occasional 1ULP numerical error, above results are reproduced. The last example is closely related to the [LogSumExp function](#), which is one of the [Smooth Maximum Approximations](#) used in optimization problems.

I hope all this makes sense. Setting up these functions was fun - thanks for giving me an excuse to learn and play with stuff i'd otherwise never have looked at again. Despite RPL's deficiencies (such as handling many trivial cases incorrectly), I like the expressive power and abstraction / object orientation of this language.

P.S. This is my first post in the new forum. While I stumbled upon a nice [description](#) how to include math, I could not find a way to use a [pre] tag. So I had to reluctantly adorn my programs with **code** tags lest all formatting be lost. I'd gladly replace these by a better solution.



04-16-2019, 09:23 PM

Post: #23

Mike (Austria)

Junior Member

Posts: 5

Joined: Sep 2018

RE: [VA] SRC#005- April, 1st Mean Minichallenge

Addendum: for stack-only purists, here comes a 63.5 bytes two-liner implementation **GMS** which does the same as above GM in slightly less time:

Code:

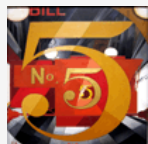
```
« DUP { DUP2 ^ ΣLIST ROT SIZE / SWAP }
  { DROP DUP ΠLIST SWAP SIZE } IFTE XROOT »
```

Now this looks like a candidate for Wlodek's [one minute marvels](#) collection, doesn't it?



04-21-2019, 10:38 PM

Post: #24


Valentin Albillo

Senior Member

Posts: 347

Joined: Feb 2015

Warning Level: 0%

RE: [VA] SRC#005- April, 1st Mean Minichallenge
Hi, Mike:
Mike (Austria) Wrote: →

(04-15-2019 11:29 PM)

Thanks Valentin for this inspiring minichallenge. I particularly enjoy math riddles when they involve little-known facts about well-known mathematics. This one definitely fits this description particularly well - the fact that the various means are discrete members of a

continuous family had escaped my attention so far (for decades - embarrassingly enough :-)).

You're welcome, thanks to you for your interest and awesome posts on this minichallenge. And I agree with you, I'm also enthralled when I find an interesting piece of mathematics that happens to be new to me, so my goal has always been to post the kind of articles and challenges that I myself would like to read and solve.

Quote:

Since there have been no solutions for RPL, let me share my take on this. It should work for any version of RPL starting with the HP48G(X).

Great ! I always welcome *RPL* solutions, they tend to be a little on the cryptic side but yours qualifies as an exception, pretty clear code made even clearer by your generous assortment of comments.

Quote:

I like the terseness of this function.

Terse it is, and the results are of course correct.

Quote:

A further step in generalizing the mean is by replacing the p -th power of x_i by applying an arbitrary continuous strictly monotonic function and taking its inverse after computing the sum and dividing by n .

Yes, it's an interesting way of generalizing it further. Nice addition to the subject matter.

Quote:

Why pick one of the original list elements to estimate the solution: we make no assumptions about the domain of f . [...] This is cheaper than e.g. taking the arithmetic mean as an estimate or bracketing the solution by searching the minimum and maximum of $\{x_i\}$.

Cheaper or not, I would've used the minimum/maximum as they're immediately available without search using the available matrix functions and they certainly bracket the solution perfectly. Not to say that your method isn't perfectly adequate, of course.

Quote:

I hope all this makes sense. Setting up these functions was fun - thanks for giving me an excuse to learn and play with stuff i'd otherwise never have looked at again. Despite RPL's deficiencies (such as handling many trivial cases incorrectly), I like the expressive power and abstraction / object orientation of this language.

I'm glad you enjoyed it all and thanks again for your extensive, very interesting post (from which I also learned interesting things) and for your time, much appreciated. Oh, and welcome to the new forum, for a very first post you did really great ! ... 😊

Best regards.

V.

.



04-22-2019, 04:02 PM

Post: #25



RMollov
Member

Posts: 224
Joined: Dec 2013

RE: [VA] SRC#005- April, 1st Mean Minichallenge

Mike (Austria) Wrote: →

(04-16-2019 09:23 PM)

Addendum: for stack-only purists, here comes a 63.5 bytes two-liner implementation **GMS** which does the same as above GM in slightly less time:

Code:

```
« DUP { DUP2 ^ ΣLIST ROT SIZE / SWAP }
  { DROP DUP ΠLIST SWAP SIZE } IFTE XROOT »
```

Now this looks like a candidate for Wlodek's [one minute marvels](#) collection, doesn't it?

I kinda miss to understand the "two-liner" thing.



04-24-2019, 08:54 PM

Post: #26

Mike (Austria)
Junior Member

Posts: 5
Joined: Sep 2018

RE: [VA] SRC#005- April, 1st Mean Minichallenge

RMollov Wrote: →

(04-22-2019 04:02 PM)

Mike (Austria) Wrote: →

(04-16-2019 09:23 PM)

Addendum: for stack-only purists, here comes a 63.5 bytes two-liner implementation **GMS** which does the same as above GM in slightly less time:

Code:

```
« DUP { DUP2 ^ ΣLIST ROT SIZE / SWAP }
  { DROP DUP ΠLIST SWAP SIZE } IFTE XROOT »
```

Now this looks like a candidate for Wlodek's [one minute marvels](#) collection, doesn't it?

I kinda miss to understand the "two-liner" thing.

I was just trying to be silly by applying a [meaningless metric](#) to a piece of free-from code. No harm meant! 😊



04-24-2019, 10:54 PM

Post: #27

Mike (Austria)

Junior Member

Posts: 5

Joined: Sep 2018

RE: [VA] SRC#005- April, 1st Mean Minichallenge

Hi Valentin,

Valentin Albillo Wrote: →

(04-21-2019 10:38 PM)

You're welcome, thanks to you for your interest and awesome posts on this minichallenge. And I agree with you, I'm also enthralled when I find an interesting piece of mathematics that happens to be new to me, so my goal has always been to post the kind of articles and challenges that I myself would like to read and solve.

[...]

I'm glad you enjoyed it all and thanks again for your extensive, very interesting post (from which I also learned interesting things) and for your time, much appreciated. Oh, and welcome to the new forum, for a very first post you did really great!

Wow, thanks for your appreciation!

Quote:**Quote:**

Why pick one of the original list elements to estimate the solution: we make no assumptions about the domain of f . [...] This is cheaper than e.g. taking the arithmetic mean as an estimate or bracketing the solution by searching the minimum and maximum of $\{x_i\}$.

Cheaper or not, I would've used the minimum/maximum as they're immediately available without search using the available matrix functions and they certainly bracket the solution perfectly. [...]

OK. Replacing in GFM the

Code:

```
x 1 GET
```

section by

Code:

```
x « MIN » STREAM
x « MAX » STREAM
2 →LIST
```

supplies the required brackets to FINV, no other changes needed. This is all I came up with for the HP48GX. It seems, however, you have something more advanced in mind.

BTW, I unsystematically benchmarked both versions (GFM = original version, GFBM = bracketing version), using Emu48 on Android (authentic speed) to compute the generalized *exp* mean of a 100 element uniform random list 'V100':

Code:

```
1 RDZ          @ seed RAND
1 100 START
RAND
NEXT           @ 100 RANDs on stack
100 →LIST      @ convert to list
'V100' STO
```

Using an instrumented function counting the number of calls

Code:

```
0 'C' STO
V100
« 1 'C' STO+ EXP »
GFM @ or GFBM respectively
```

shows indeed that bracketing reduces the number of function calls from 11 (12 with a different seed) to 8.

On the other hand, timing the original EXP function using an [anonymous timing routine from hpcalc](#),

Code:

V100
« EXP »
« GFM » @ or GFBM
XTGS

yields 3.055_s for GFM and 4.439_s for GFBM. Well, probably not too bad a trade of speed for reliability.

Best wishes, Mike



04-25-2019, 02:05 PM

Post: #28



RMollov
Member

Posts: 224
Joined: Dec 2013

RE: [VA] SRC#005- April, 1st Mean Minichallenge

Mike (Austria) Wrote: →

(04-24-2019 08:54 PM)

I was just trying to be silly by applying a [meaningless metric](#) to a piece of free-from code. No harm meant! 😊

Thanks Mike, I really got puzzled by the thing, interesting link - now it makes sense (in a way) 😊

Cheers,



<< Next Oldest | Next Newest >>

Enter Keywords Search Thread

Pages (2): << Previous 1 2



- View a Printable Version
- Send this Thread to a Friend
- Subscribe to this thread

User(s) browsing this thread: [Valentin Albillo*](#)