GAP in algebra class

Hideki SAWADA
Department of Mathematical Sciences, Faculty of Science
Yamagata University, Yamagata 990-8560, Japan
e-mail sawada@kszaoh3.kj.yamagata-u.ac.jp

Introduction

In autumn 1997 at the computer centre of the Yamagata University.
"Hey, Hanako! Is this number 1234567891 a prime or not?"
"Who knows, I bet it's a prime."

```
gap> IsPrime(1234567891);
true
```

"You win!"
This light-hearted but "*academic*" conversation occurred shortly after I explained how to use

GAP (Groups, Algorithms and Programming) 3.4.4

in the first class of Information Science for the 4th year students of the Department of Mathematical Sciences. I have never heard of such an enjoyable conversation in a math class since I began my career at university.

In this paper the author explains the effects of the usage of such a computer algebra package in class. This is a result of his two-year long project of improvement of teaching methods of algebra. In 1996 he gave the students a course of cryptography including its mathematical background (see [2]). Especially

Elementary number theory, Group theory and Finite fields.

As a result of recent dramatic expansion of graduate schools of mathematical sciences, mathematical departments of universities are not only a place for educating researchers of mathematics, but also for training engineers of mathematical sciences. Classes aided by computer are necessary for the both types of students. For students who want to be researchers it is very helpful to have a skill of calculating examples, and for students who are interested in applications of mathematics it is very useful to get used to such a math software like GAP, because they can learn mathematics through examples and also learn computer systems at the same time. Though most of my 46 students have already joined computer classes including C-programming for 3 semesters, they were all keen and gave positive answers in the questionnaires about the class.

It may be a common sense among the people who join ATCM'98 that a computer is a good partner in their classrooms. However the author stands at a place where he always has to explain why computer can be a partner in his class. It is a distress for him to spend most of his time not only preparing texts for the class but also defending this common sense. Therefore the author should like to ask the reader to listen to his motive and background, too.

In 1987 when the author first started a seminar on C-language at a department of mathematics somewhere in Tokyo, one of the professors of the department strongly asked him not to give such a course like computer language. It was a surprising and unexpected attitude towards computers for the author, because he had been thinking that a computer itself is a result of mathematical sciences, and further it was not realistic, because most of the graduates of that department go into industry and they also want to learn more about computers. The professor wanted to put the department on a traditional stage where students learn so called "pure mathematics" only.

What should be a tradition of university community? Is it to be conservative, or is it to ignore outside of university? Is it to refuse any new teaching method? No, certainly these should not be our tradition. The tradition of our community exists in its members'

Flexibility, Curiosity and Diversity

based upon the strong confidence of **Humanism** of each of them. Therefore a criticism like "Why do you use computers in your math class?" is apparently out of question.

The author now has strong confidence that a computer aided math class will soon be seen in any university, because

- there is a strong social demand for mathematical scientists and
- most of students want such a math class supported by computers and
- it certainly helps them to understand "pure mathematics" more deeply.

1 How and What

The class was given at the computer centre of Yamagata University in the autumn semester of 1997. The computers of the centre have now been replaced by more efficient ones, but the server for students at that time was S-4/1000E 3 CPU(384MB). It has 50 personal computers, FMV-466D2(i486DX2(66MHz), 24MB) as terminals. GAP (Groups, Algorithms and Programming) 3.4.4 itself was in the author's account of another server S-4/1000E 4 CPU(640MB). The NIS server was S-4/10m51H 1 CPU(96MB) and the NFS server was S-4/Cluster1 8 CPU(128MB) with 50GB hard disk.

50 students enrolled in the class, 4 left soon and after all 46 students passed. The course did not assume any knowledge of Unix. In the first class he explained them:

1. How to add a new path in there `.cshrc` and `source` it. He also explained about a role of shell;
2. how to start and how to use help, and how to end `gap`;
3. simple examples without explanations like:

```
gap> (11-6)*(5+2);
35
gap> 7^100;
3234476509624757991344647769100216810857203198904625400933895331391691459636928060001
gap> 27 mod 5;
2
gap> Factorial(100);
```

```
933262154439441526816992388562667004907159682643816214
685929638952175999932299156089414639761565182862536979
208272237582511852109168640000000000000000000000000
gap> Gcd(12345,6789);
3
```

in order to attract their interests (see [1]). Also just let them try calculations on symmetric groups like

```
gap> (1,2,3)*(1,2);
(2,3)
gap> (1,2,3)^-1;
(1,3,2)
gap> (1,2,3)*(1,3,2);
()
gap> 1^(1,2,3);
2
```

in order to remind of what they have learned so far (Notice that `1^(1,2,3)` gives the image of `1` under the permutation `(1,2,3)`.);

4. number theory for understanding RSA csryptosystem, including `gap` functions like: `QuoInt(a,b)`, `RemInt(a,b)`, `Gcd(a,b,c,...)`, `GcdRepresentation(a1,a2,..`
`Gcdex(a1,a2)`, `FactorsInt(a)`, `DivisorsInt(a)`, `IsPrime(a)`, `NextPrimeInt(a)`, `PowerMod(a,b,m)`, `ChineseRem([m_1,m_2,...],[a1,a2,...])` and `Phi(n)`;

5. finite group theory and `gap` functions, for example, first review the general proof of the proposition that S_n is generated by $(1,2)$ and $(1,2,3,...,n)$ then examine that proof when $n=5$ by `gap`. Then define the symmetric group S_5 by

```
gap> s5:=Group((1,2),(1,2,3,4,5));
Group( (1,2), (1,2,3,4,5) )
```

and its commutator subgroup A_5 by

```
gap> a5:=CommutatorSubgroup(s5,s5);
Subgroup( Group((1,2),(1,2,3,4,5)),[(1,3,2),(2,4,3),
(2,3)(4,5)])
```

and try functions: `IsAbelian(s5)`, `ConjugacyClasses(s5)`, `Size(a5)`, `IsSimple(a5)`, `Index(s5,a5)`, `IsNormal(s5,s3)`, `RightCosets(s5,s3)`, `LeftCosets(s5,s3)`, `Orbit(s5,2)` and `Stabilizer(s5,2)`;

6. gap has a few functions on finite fields, for example, `Size(F)`, `OrderFFE(z)` and `CharFFE(z)` for a small finite field like:

   ```
   gap> F:=GF(5^3);
   GF(5^3)
   ```

   but it helps to understand the discrete logarithm problem: to find an integer x such that

   $$a^x = b$$

   for finite field elements a and b by its function `LogFFE`

   ```
   gap> LogFFE(z^3);
   3
   ```

For the examinations the author prepared gap functions like this:

```
LogTo("rsakey.out");
Print("This programme is rsakey.g. Logfile is rsakey.out.\n");

RsaKey:=function(p,q,m)
  local L, i, keys, x;
  if IsPrime(p)=true and IsPrime(q)=true then
    L:=Phi(p*q);
    keys:=[];
    for i in [1..L] do
      if Gcd(i,L)=1 then
        Add(keys,i);
      fi;
    od;

    for i in [1..m] do
      Print(i); Print("\n");
      x:=Random(keys);
```

```
      Print(x); Print("\n");
      Print(GcdRepresentation(x,L)[1]); Print("\n");
    od;

  else
    Print("p or q is not prime. \n");
  fi;
end;
```

and let them answer its mathematical meaning and a weak point when p and q are big primes. Another interesting function whose meaning was asked is

```
LogTo("isgenprime.out");
Print("This programme is isgenprime.g. Logfile is isgenprime.out.\n");

IsGenPrime:=function(a,m)
  local divisors, L, i;
  if IsPrime(m)=true then
    divisors:=DivisorsInt(m-1);
    Print(divisors); Print("\n");
    L:=Length(divisors);
      Print(L);
      Print(":the number of divisors of m-1. \n");
    for i in [1..L] do
      if PowerMod(a,divisors[i],m)=(1 mod m) then
        Print(i); Print("\n");
        Print(divisors[i]); Print("\n");
        Print(PowerMod(a,divisors[i],m)); Print("\n");
        return;
      fi;
    od;

  else
    Print("m is not prime, \n");
  fi;
end;
```

This is a function to find a primitive element of a prime field. For example the computations

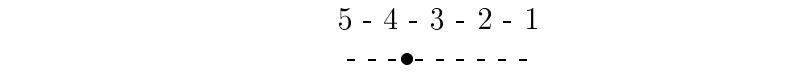
```
This programme is isgenprime.g. Logfile is isgenprime.out.
gap> IsGenPrime(2,3456789019);
[ 1, 2, 3, 6, 11677, 23354, 35031, 49339, 70062, 98678,
  148017, 296034, 576131503, 1152263006, 1728394509,
  3456789018 ]
16:the number of divisors of m-1.
16
3456789018
1
gap> IsGenPrime(3,3456789019);
[ 1, 2, 3, 6, 11677, 23354, 35031, 49339, 70062, 98678,
  148017, 296034, 576131503, 1152263006, 1728394509,
  3456789018 ]
16:the number of divisors of m-1.
16
3456789018
1
gap> IsGenPrime(4,3456789019);
[ 1, 2, 3, 6, 11677, 23354, 35031, 49339, 70062, 98678,
  148017, 296034, 576131503, 1152263006, 1728394509,
  3456789018 ]
16:the number of divisors of m-1.
15
1728394509
1
```

show that 2 and 3 are primitive elements of the prime field $F_{3456789019}$ but 4 is not.

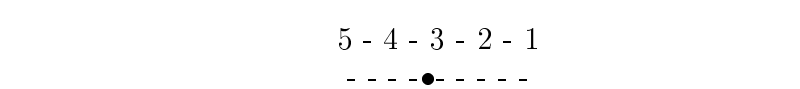
2 Students' Opinion

At the end of the semester 43 students answered the questionnaires on the course. They were asked to answer the following questions by choosing a number from 5 to 1 according to their satisfaction or affirmation. Therefore the numbers 5 and 4 mean "positive", 3 means "neutral" and 2 and 1 mean "negative". Each bullet • hits the average of their answers given in [ ].

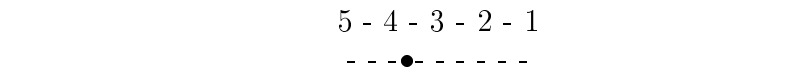
UNIX: Have you got used to Unix commands in this class? **[3.65]**



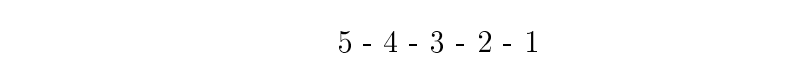
COMPUTER: Did the class help you to get more knowledge on computer system? **[3.25]**



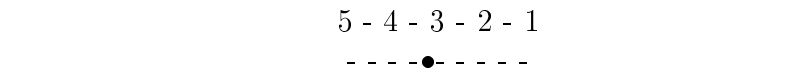
THEORY: Did the class help you to understand abstract algebra by calculating examples which are difficult to do by your pen and paper only? **[3.75]**



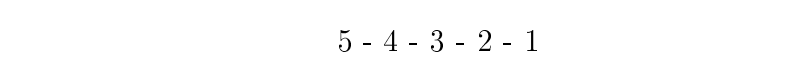
ByPen: Have your ever calculated examples of abstract algebra without computer? **[2.79]**



Euler: Did the class help you to understand Euler's Theorem: $a^{\varphi(n)} \equiv 1 \ (mod\ n)$ where $(a, n) = 1$? **[3.22]**



Ch.Rem: Did the class help you to understand Chinese Remainder Theorem? **[3.04]**



S_n: Did the class help you to understand how to compute cosets of a symmetric group S_n? **[3.29]**

5 - 4 - 3 - 2 - 1
- - - -●- - - - -

D.Log: Did the class help you to understand the discrete logarithm of finite fields? [**2.79**]

5 - 4 - 3 - 2 - 1
- - - - -●- - - -

DEMAND: Do you want a computer aided math class like this? [**4.22**]

5 - 4 - 3 - 2 - 1
- -●- - - - - -

GRADE: Could you tell me your grade of abstract algebra. [**1.69**]

A(3) - B(2) - C(1)
- - - - - - -●- - -

¿From the above table we can say that for many students abstract algebra was not easy to understand (9 students answered "A") or to calculate without computer, but the class helped them to understand the abstract theory especially number theory and group theory, and most of them want such a computer aided math class. More precisely the next table tells that
Gap in algebra class attracts not only grade A students but also B and C students, especially grade B students show more interests on computer and were less absent than others. Their grade of the class Grd(inf) is the highest among them.

Grd(alg)	Grd(inf)	Abs	Unix	Cmp	Theory	ByPen	Euler	Ch.Rem	S_n	D.Log	Dmnd
A	2.0	1.0	3.55	3.33	4.11	3.33	3.44	3.44	3.77	3.22	4.33
B	2.3	0.7	3.75	3.66	4.00	2.91	3.33	3.16	3.33	2.75	4.25
C	2.2	1.0	3.85	3.15	3.65	2.75	3.35	3.10	3.35	2.90	4.40

Grd=GRADE, Abs=ABSENCE, Cmp=COMPUTER, Dmnd=DEMAND

3 How to get GAP

The author should like to end this paper quoting some parts of Chapter 55 of [1]:

> GAP runs on a large number of different operating systems. It behaves slightly different on each of those. This chapter describes the behaviour of GAP, the installation, and the options on some of those operating systems.
> Currently it contains sections for *UNIX* (see "GAP for UNIX"), which runs on an ever increasing number of machines, for *MS-DOS* (see "GAP for MS-DOS"), which is one operating system on *IBM PC compatibles*, and *TOS* (see "GAP for TOS"), which is the operating system on *Atari ST* and *MacOS* (see "GAP for MacOS"), which is the operating system on Apple Macintosh computers.
> For other systems the section "Porting GAP" gives hints how to approach such a port.
> GAP is distributed *free of charge*. You can obtain it via `ftp` and give it away to your colleagues. GAP is *not* in the public domain, however. In particular you are not allowed to incorporate GAP or parts thereof into a commercial product.
> We distribute the *full source* for everything, the C code for the kernel, the GAP code for the library, and the LaTeX code for the manual, which has at present about 1600 pages. So it should be no problem to get GAP, even if you have a rather uncommon system. Of course, ports to non UNIX systems may require some work. We already have ports for IBM PC compatibles with an Intel processor under MS-DOS, Windows, or OS/2, for the Atari ST under TOS and Apple Macintosh using the CodeWarrior compiler. Note that about 8 MByte of main memory and about 20MB of disk space are required to run GAP. A full GAP installation, including all share packages and data libraries can use up to 100MB of disk space.
> The easiest way to get GAP 3.4 for most users is probably via the World Wide Web. The main GAP Web site is found at
> `http://www-gap.dcs.st-and.ac.uk/~gap`.

There are three *mirror sites* updated automatically each night, at

`http://www.math.rwth-aachen.de/LDFM/GAP`
`http://www.ccs.neu.edu/Cobwebs/GAP`
`http://wwwmaths.anu.edu.au/algebra/GAP/WWW.`

At these sites you can browse this manual, download the system and contributed extensions, read past postings to the GAP forum, and find out about authors of and contributors to GAP, publications that cited GAP and GAP related events.
GAP 3.4 can also be obtained by anonymous *ftp* from the following servers.

`ftp-gap.dcs.st-and.ac.uk`
School of Mathematical and Computational Sciences,
University of St Andrews, Scotland
directory `/pub/gap/gap/`.

`ftp.math.rwth-aachen.de`
Lehrstuhl D fur Mathematik, RWTH Aachen, Germany,
directory `/pub/gap/`.

`math.ucla.edu`
Math. Dept., Univ. of California at Los Angeles,
directory `/pub/gap/`.

`wuarchive.wustl.edu`
Math. Archives, Washington Univ. at St. Louis,
directory `/edu/math/source.code/group.theory/gap`.

`dehn.mth.pdx.edu`
PSU Mathematics Department, Portland State Univ.,
directory `/mirror/gap/`

`pell.anu.edu.au`
School of Mathematical Sciences, Australian National Univ., Canberra,
directory `/pub/algebra/gap/`.

`ftp` to the server *closest* to you, login as user `ftp` and give your full e-mail address as password. Remember when you transmit

the files to set the file transfer type to *binary image*, otherwise you will only receive unusable garbage. Those servers will always have the latest version of GAP available.

The `ftp` directory contains the following files. Please check first which files you need, to avoid transferring those that you don't need.

`README`
the file you are currently reading.

`gap3r4p4.zoo`
This file contains the *complete* distribution of GAP version 3 release 4 current patchlevel 4. It is a `zoo` archive approximately 18 MByte large.

`unzoo.c`
A simple `zoo` archive extractor, which should be used to unpack the distribution. The `utils` subdirectory contains ready compiled executables for common systems.
More files are in the following *subdirectories*:
`bin`
This directory contains *executables* for systems that dont come with a C compiler or where another C compiler produces a faster executable. The `KERNELS` file tells you which executables are here.

`split`
This directory contains the complete distribution of GAP 3r4p4 in several archives. This allows you to get only the parts that you are really interested in. The `SPLIT` file tells you which archive contains what.

`utils`
This directory contains several utilities that you may need to get or upgrade GAP, e.g., `unzoo` and `patch`. The `UTILS` file tells you which files are here.

Acknowledgments: The author is very grateful for his student Takahiro Abe and the staffs of the computer centre of the Yamagata University for preparing comfortable computing environments for him and his students.

References

[1] Gap, Groups, Algorithms and Programming 3.4.4 *Lehrstuhl D fuer Mathematik RWTH Aachen, 1997*

[2] H.Sawada, Cryptography and algebra (in Japanese) *Kaibun-do, 1997*