An Introduction to *Groups, Algorithms, and Programming* (GAP)

GAP is free (open source) mathematical software which is particularly great for computing with groups. A few helpful links...

- http://www.gap-system.org

  GAP's homepage. Here you can find the official GAP manuals and download the latest version of GAP.

- http://college.cengage.com/mathematics/gallian/abstract_algebra/5e/shared/gap/gap_manual.pdf

  A GAP manual written specifically to accompany Joe Gallian's *Contemporary Abstract Algebra* (5^{th} edition). This is the easiest gentle introduction to GAP that I've found. I also like that it's written to be used with Gallian's text (each section parallels a chapter of the textbook).

- http://www.math.hawaii.edu/∼williamdemeo/GAP/GapNotes.pdf

  These notes list a lot of different group related GAP commands for finding things such as the lattice of subgroups etc.

- http://www.math.colostate.edu/∼hulpke/lectures/m666/ctblex1.pdf

  What does displaying a character table tell you?

- http://ggap.sourceforge.net

  GGAP is a frontend for the GAP system (GAP is command line software). I've had a lot of trouble getting GGAP to run on my office Mac and my PCs at home (right now GGAP doesn't work on any of them, but it has in the past).

- http://www.math.colostate.edu/∼hulpke/CGT/education.html

  This person has two convenient installers (one for PC and one for Mac). They include GGAP. If they work for you, fantastic (I've had mixed results).

- https://cloud.sagemath.com

  The SAGE cloud. SAGE is a computer algebra system suite much like Maple. However, SAGE is *free* (open source) and is available online (through the SAGE cloud). GAP will run inside of SAGE. This is your quickest way to access GAP. You can login to the SAGE cloud, open a GAP console, and start right away (no need to install any software).

  Getting GAP to run...

  - On a PC? Download the latest version from GAP's homepage. Use the Windows (.zip) version. Uncompress the .zip file to where ever you want to keep GAP. No need to run an installer – I think. It should be ready to go. Go to the `/bin` folder and find `gap.bat`. If you are using Windows 7, you'll need to right-click on the file and "run as administrator" to get the adequate permissions from mommy Windows (yes, mommy I would like my software to actually run).
  - Online? Go to the SAGE cloud. Create an account. Create a new project. Create a command line terminal. At the terminal (the dollar sign) type `sage`. This will load up SAGE. Then in SAGE type `gap_console()`. That's it. You're running GAP online!

- A lot of GAP's syntax is like Maple. For example, to get help on a topic type "`? topic`". Or to assign a mapping to the variable name f use "`f := x -> x^2;`". Notice that commands end with a semicolon. If you'd like to suppress output use a double semicolon.

- Gallian's GAP introduction has some nice notes on how to load/save work in command line GAP. If you have GGAP this is much easier. For example, you can use "`LogTo("filename");`" to save or "`LogTo();`" to save again.

- `Gcd(a,b);` and `Lcm(a,b);` compute what you'd think. To get the extended Euclidean algorithm information use `Gcdex(a,b);`. Then we have gcd = a*coeff1 + b*coeff2 as well as 0 = a*coeff3 + b*coeff4.

- `10 mod 3`

- To create a procedure...

```
myFunction := function(x)
  local t;
  t := x^2;
  return t;
end;
```

- Create a dihedral group using `D4 := DihedralGroup(IsPermGroup,8);`. This creates the dihedral group D_4 or $D_{2\cdot 4}$ of order 8. The "`IsPermGroup`" creates it as a permutation group.

- `Size(D4);  Elements(D4);  Center(D4);  IsAbelian(D4);  IsCyclic(D4);`

- Multiplying permutations `(1,2)(3,4)*(1,2,3)` is done **left to right** (the opposite of our convention)!

- Let `G := DihedralGroup(IsPermGroup,16);` and `a := G.1;` so `a` the first generator of G – that is – $(123\ldots8)$. `G[3]*G[5]` multiplies the third and fifth elements of G.

  `Subgroup(G,[a]);` creates the cyclic subgroup generated by `a`. More generally, `Subgroup(G,[a,b,etc.]);` creates the subgroup of G which is generated by `a,b,` etc.

- `Centralizer(G,a);` (the centralizer of a in G). `Order(a);` `Inverse(a);`

- Define the unit groups (mod n)...

```
uGroup := function(n)
  local s,i,o;
  o := One(Integers mod n);   # the One object to signal we're in U(n).
  s :=  n -> Filtered([1..n-1],i->Gcd(i,n)=1);
  return s(n)*o;
end;
```

```
G := uGroup(10);   # G = U(10)
```

- How does GAP do what it does? Well, it's open source so we can see. For example:

```
G := DihedralGroup(IsPermGroup,16);
code := ApplicableMethod(IsCyclic,[G],1);
Print(code);
```

  will display the code used to determine if G is a cyclic group or not.

- `Collected(List(Elements(G),Order));` computes how many elements of each order G has.
- `SymmetricGroup(n);  AlternatingGroup(n);  CycleStructurePerm(a);`
- `Filtered(Elements(SymmetricGroup(5)),s->CycleStructurePerm(s)=[,,1]);` finds all of the 4-cycles in S_5. Replacing `[,,1]` with `[1,1]` would find all of the elements consisting of a transposition and a 3-cycle (disjoint).
- `x^f` computes the image of x under f. `x/f` computes the preimage of x under f. If both `x` and `y` are group elements `x^y` conjugates `x` by `y`. For example: $(1,2,3)^(1,2) = (1,3,2)$, $2^(1,2,3) = 3$, $2/(1,2,3) = 1$, $(1,2,3)^(-1) = (1,3,2)$ (exponents works as you'd think).
- `Orbit(H,a)  Stabilizer(H,a)`
- `DirectProduct(A,B)`
- Let `G := AlternatingGroup(4);` and `H := Subgroup(G,[(1,2)(3,4),(1,4)(2,3)]);`.
  `IsNormal(H,G);  RightCosets(G,H);  FactorGroup(G,H);` or `G/H;`
- Free groups? `F := FreeGroup(2);  a := F.1;  b := F.2;  G := F/[a^4,b^2,(a*b)^2];` creates a generator and relation version of D_4 where a is the rotation of order 4 and b is a reflection.
- `List(ConjugacyClasses(G),Size);` finds the sizes of all of the conjugacy classes.
- `List(ConjugacyClasses(G),x->[x,Size(x)]);` lists the sizes of each conjugacy class with the class itself.
- `Size(AllGroups(Size,8,IsAbelian,true));` finds out how many Abelian groups of order 8 there are (up to isomorphism). GAP has a "small groups" library which has all groups up to order 2000 (except order 1024) classified.
- `SylowSubgroup(SymmetricGroup(7),2);` find a Sylow 2-subgroup of S_7.
- `Display(CharacterTable(SymmetricGroup(5)));` displays the character table of S_5. See the notes linked to on the first page for a cypher for this somewhat cryptic display.
- `quit;` breaks out of loop (in case of an error) or just plain old quits GAP.