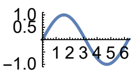
Pensieve header: MandelTime and Non-Commutative Gaussian Elimination, Day 3.

From http://www.math.toronto.edu/drorbn/classes/16-1750-ShamelessMathematica/About.html: **Possible Topics** (in no particular order). Whatever you may suggest, and the ~~Fibonacci numbers~~; ~~the Jones polynomial~~; ~~a more efficient Jones algorithm~~; ~~a riddle on spheres~~; ~~Khovanov homology~~; Γ-calculus; the Hopf fibration; Hilbert's 13th problem; ~~non-commutative Gaussian elimination~~; free Lie algebras; the Baker-Campbell-Hausdorff formula; wacky numbers; ~~an order 4 torus~~; ~~the Schwarz Lantern~~; knot colourings; the Temperley-Lieb pairing; the dodecahedral link; sound experiments; barycentric subdivisions; a Peano curve; braid closures and Vogel's algorithm; the insolubility of the quintic, phase portraits; ~~the Mamdelbrot set~~.

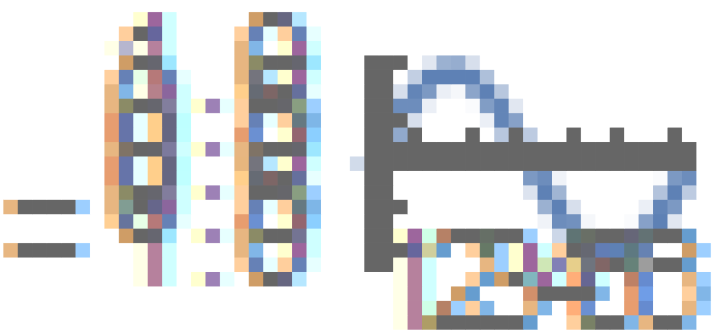
The Mandelbrot Set

```
$MandelTimeBound = 100;
MandelTime[c_] := Module[{z = 0, n = 0},
   While[n < $MandelTimeBound ∧ Abs[z] < 3, ++n; z = z^2 + c];
   n
  ];
```

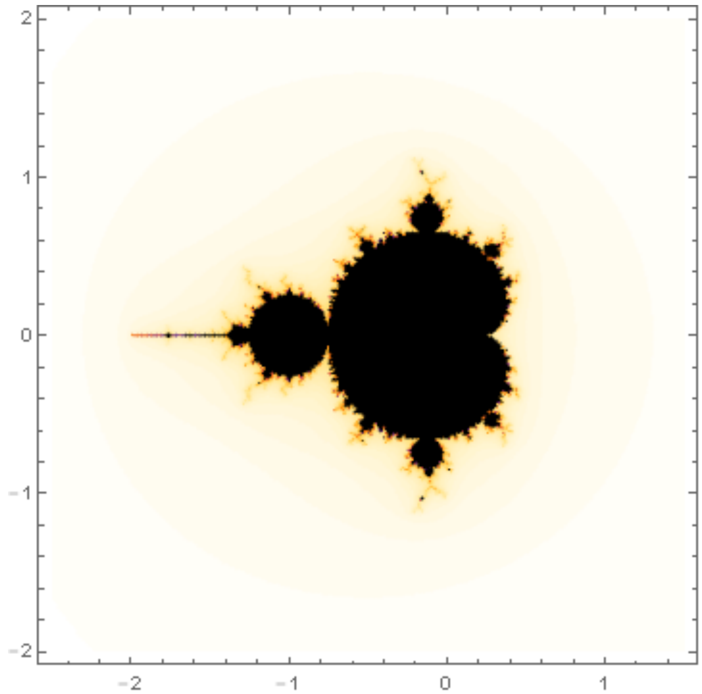
```
Plot[Sin[x], {x, 0, 2 π}, ImageSize → 50]
```




```
Plot[Sin[x], {x, 0, 2 π}, ImageSize → 50] // Rasterize
```



```
{x0, y0} = {-0.5, 0};
a = 2;
DensityPlot[-MandelTime[x + ⅈ y], {x, x0 - a, x0 + a}, {y, y0 - a, y0 + a},
  PlotPoints → 100, ColorFunction → "SunsetColors", PlotRange → All
 ] // Rasterize
```

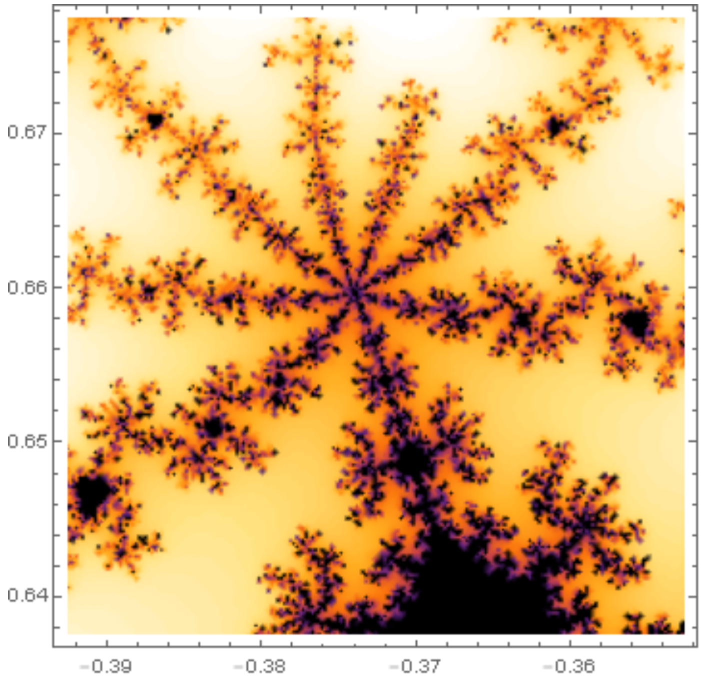



```
{{"-0.09909", "0.9826"}}
```

```
ToExpression["1+1"]
```

```
2
```

```
{x0, y0} = ToExpression /@ (Last@{{"-0.3726", "0.6575"}});
a = 0.02;
DensityPlot[-MandelTime[x + i y], {x, x0 - a, x0 + a}, {y, y0 - a, y0 + a},
  ColorFunction → "SunsetColors", PlotPoints → 50] // Rasterize
```



Non Commutative Gaussian Elimination

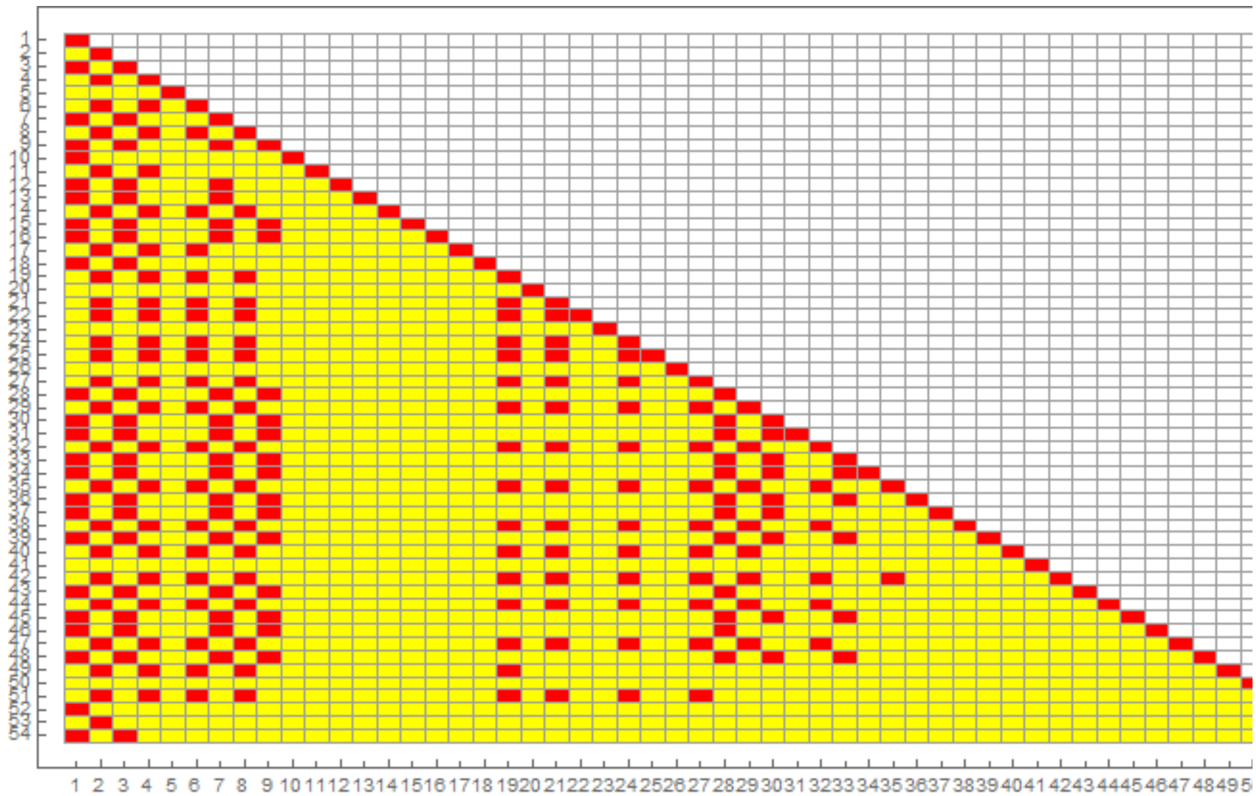
```
n = 54;
g1 = Cycles[{{1, 18, 45, 28}, {2, 27, 44, 19},
     {3, 36, 43, 10}, {46, 52, 54, 48}, {47, 49, 53, 51}}];
g2 = Cycles[{{7, 16, 39, 30}, {8, 25, 38, 21}, {9, 34, 37, 12},
     {13, 15, 33, 31}, {14, 24, 32, 22}}];
g3 = Cycles[{{28, 31, 34, 48}, {29, 32, 35, 47}, {30, 33, 36, 46},
     {37, 39, 45, 43}, {38, 42, 44, 40}}];
g4 = Cycles[{{1, 3, 9, 7}, {2, 6, 8, 4}, {10, 54, 16, 13},
     {11, 53, 17, 14}, {12, 52, 18, 15}}];
g5 = Cycles[{{1, 13, 37, 46}, {4, 22, 40, 49}, {7, 31, 43, 52},
     {10, 12, 30, 28}, {11, 21, 29, 19}}];
g6 = Cycles[{{3, 48, 39, 15}, {6, 51, 42, 24}, {9, 54, 45, 33},
     {16, 18, 36, 34}, {17, 27, 35, 25}}];
```

```
σ_ ∘ τ_ := PermutationProduct[τ, σ];
Feed[Cycles[{}]] := 1 + 1;
Feed[τ_] := Module[{i, j, k, l},
   i = Min[PermutationSupport[τ]];
   j = PermutationReplace[i, τ];
   If[Head[σi,j] === Cycles,
    Feed[InversePermutation[σi,j] ∘ τ],
    (*Else*) σi,j = τ; Pause[0.5];
    For[k = 1, k < n, ++k,
     For[l = k + 1, l ≤ n, ++l,
      If[Head[σk,l] === Cycles,
       Feed[σi,j ∘ σk,l]; Feed[σk,l ∘ σi,j]]
     ]]
   ]];
$RecursionLimit = ∞;
```

```
Table[Feed[gα];
  ∏_{i=1}^{n} (1 + Count[Range[n], j_ /; Head[σi,j] == Cycles]), {α, 6}] // Timing
```

```
{15.5938, {4, 16, 159 993 501 696 000, 21 119 142 223 872 000,
  43 252 003 274 489 856 000, 43 252 003 274 489 856 000}}
```

```
ArrayPlot[
  Table[
   Which[
    Head[σ_{i,j}] === Cycles ∨ i == j, Red, i < j, Yellow, i > j, White],
   {j, n}, {i, n}],
  Mesh → True, Frame → True,
  FrameTicks → Range@n, AspectRatio → 9/16, ImageSize → 720
 ] // Rasterize
```



Math Questions.

1. Why are there so many yellow cells in column 1?
2. Why are there so many yellow cells in column 2?
3. Why are there “new” yellow cells in column 3?
4. Why are there “new” yellow cells in column 4?
5. Why is column 5 entirely yellow?
6. Why’s the big yellow gap between colulmn 10 and 18?

Mathematica Challanges.

1. Can you turn the program into a “package”?
2. Can you make the porgram doubly efficient?
3. Can you keep track of the actual “tricks”?
4. Can you find the true lengths of the “tricks”?
5. Can you fix these sad results?
6. Can you actually solve cubes?

The Mathematica Package “Permutations”

Goal: Re-implement permutations, using “permutation lists” as the basic data type, rather than “cycle decompositions”.

```
? Cycles
```

Cycles[{cyc_1, cyc_2, …}] represents a permutation with disjoint cycles cyc_i. >>

```
? PermutationList
```

PermutationList[*perm*] returns a permutation list representation of permutation *perm*.

PermutationList[*perm*, *len*] returns a permutation list of length *len*. >>

```
? PermutationCycles
```

PermutationCycles[*perm*] gives a disjoint cycle representation of permutation *perm*. >>

```
? PermutationProduct
```

PermutationProduct[*a*, *b*, *c*] gives the product of permutations *a*, *b*, *c*. >>

```
? InversePermutation
```

InversePermutation[*perm*] returns the inverse of permutation *perm*. >>

```
? PermutationSupport
```

PermutationSupport[*perm*] returns the support of the permutation *perm*. >>

```
? PermutationReplace
```

PermutationReplace[*expr*, *perm*] replaces each part in *expr* by its image under the permutation *perm*.

PermutationReplace[*expr*, *gr*] returns the list of images of *expr* under all elements of the permutation group *gr*. >>

Finer Goal: Make a useful package implementing PL, PC, PP, IP, PS, and PR, to the point that we can re-write “Feed” so that it works on the following γ_α'*s*:

```
Table[γα = PermutationList[gα, n], {α, 6}] // MatrixForm
```

```
18 27 36  4  5  6  7  8  9  3 11 12 13 14 15 16 17 45  2 20 21 22 23 24 25
 1  2  3  4  5  6 16 25 34 10 11  9 15 24 33 39 17 18 19 20  8 14 23 32 38
 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
 3  6  9  2  5  8  1  4  7 54 53 52 10 11 12 13 14 15 19 20 21 22 23 24 25
13  2  3 22  5  6 31  8  9 12 21 30 37 14 15 16 17 18 11 20 29 40 23 24 25
 1  2 48  4  5 51  7  8 54 10 11 12 13 14  3 18 27 36 19 20 21 22 23  6 17
```

```
? Permutation
```

Information::notfound : Symbol Permutation not found. >>