

Pensieve header: Non-Commutative Gaussian Elimination.

## The Commutative Case

```
vs = Table[(i + j)^5, {i, 12}, {j, 8}];
vs // Column
{32, 243, 1024, 3125, 7776, 16807, 32768, 59049}
{243, 1024, 3125, 7776, 16807, 32768, 59049, 100000}
{1024, 3125, 7776, 16807, 32768, 59049, 100000, 161051}
{3125, 7776, 16807, 32768, 59049, 100000, 161051, 248832}
{7776, 16807, 32768, 59049, 100000, 161051, 248832, 371293}
{16807, 32768, 59049, 100000, 161051, 248832, 371293, 537824}
{32768, 59049, 100000, 161051, 248832, 371293, 537824, 759375}
{59049, 100000, 161051, 248832, 371293, 537824, 759375, 1048576}
{100000, 161051, 248832, 371293, 537824, 759375, 1048576, 1419857}
{161051, 248832, 371293, 537824, 759375, 1048576, 1419857, 1889568}
{248832, 371293, 537824, 759375, 1048576, 1419857, 1889568, 2476099}
{371293, 537824, 759375, 1048576, 1419857, 1889568, 2476099, 3200000}
```

### ? Position

Position[*expr*, *pattern*] gives a list of the positions at which objects matching *pattern* appear in *expr*.

Position[*expr*, *pattern*, *levelspec*] finds only objects that appear on levels specified by *levelspec*.

Position[*expr*, *pattern*, *levelspec*, *n*] gives the positions of the first *n* objects found.

Position[*pattern*] represents an operator form of Position that can be applied to an expression. >>

### ? FirstPosition

FirstPosition[*expr*, *pattern*] gives the position of the first

element in *expr* that matches *pattern*, or Missing["NotFound"] if no such element is found.

FirstPosition[*expr*, *pattern*, *default*] gives *default* if no element matching *pattern* is found.

FirstPosition[*expr*, *pattern*, *default*, *levelspec*] finds only objects that appear on levels specified by *levelspec*.

FirstPosition[*pattern*] represents an operator form of FirstPosition that can be applied to an expression. >>

## On to Rubik's Cube

### ? Cycles

Cycles[{{*cyc*<sub>1</sub>, *cyc*<sub>2</sub>, ...}}] represents a permutation with disjoint cycles *cyc*<sub>*i*</sub>. >>

```

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}}];

```

#### ? PermutationProduct

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

```
a_ ° b_ := PermutationProduct[a, b]
```

#### ? 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*. >>