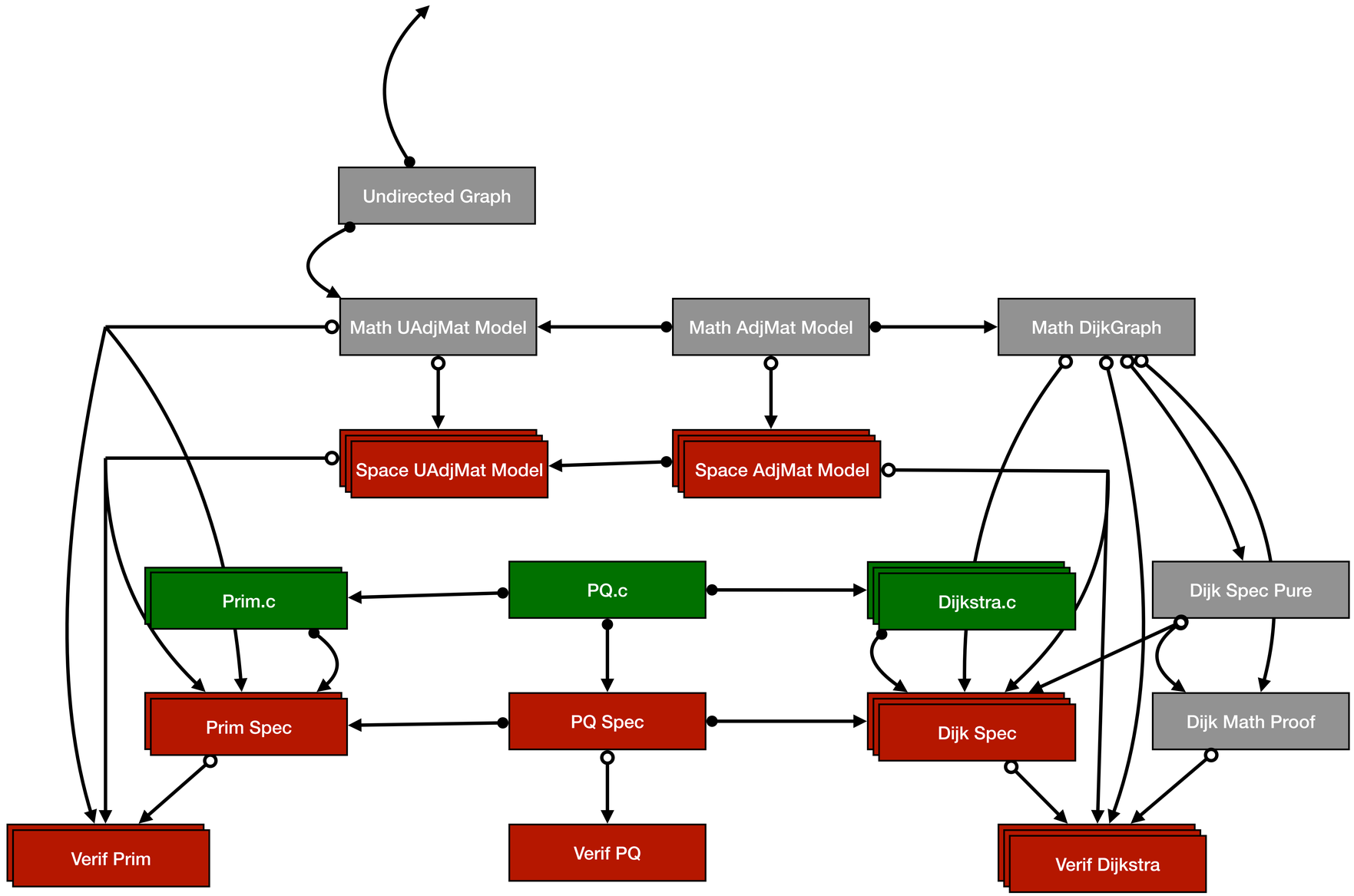
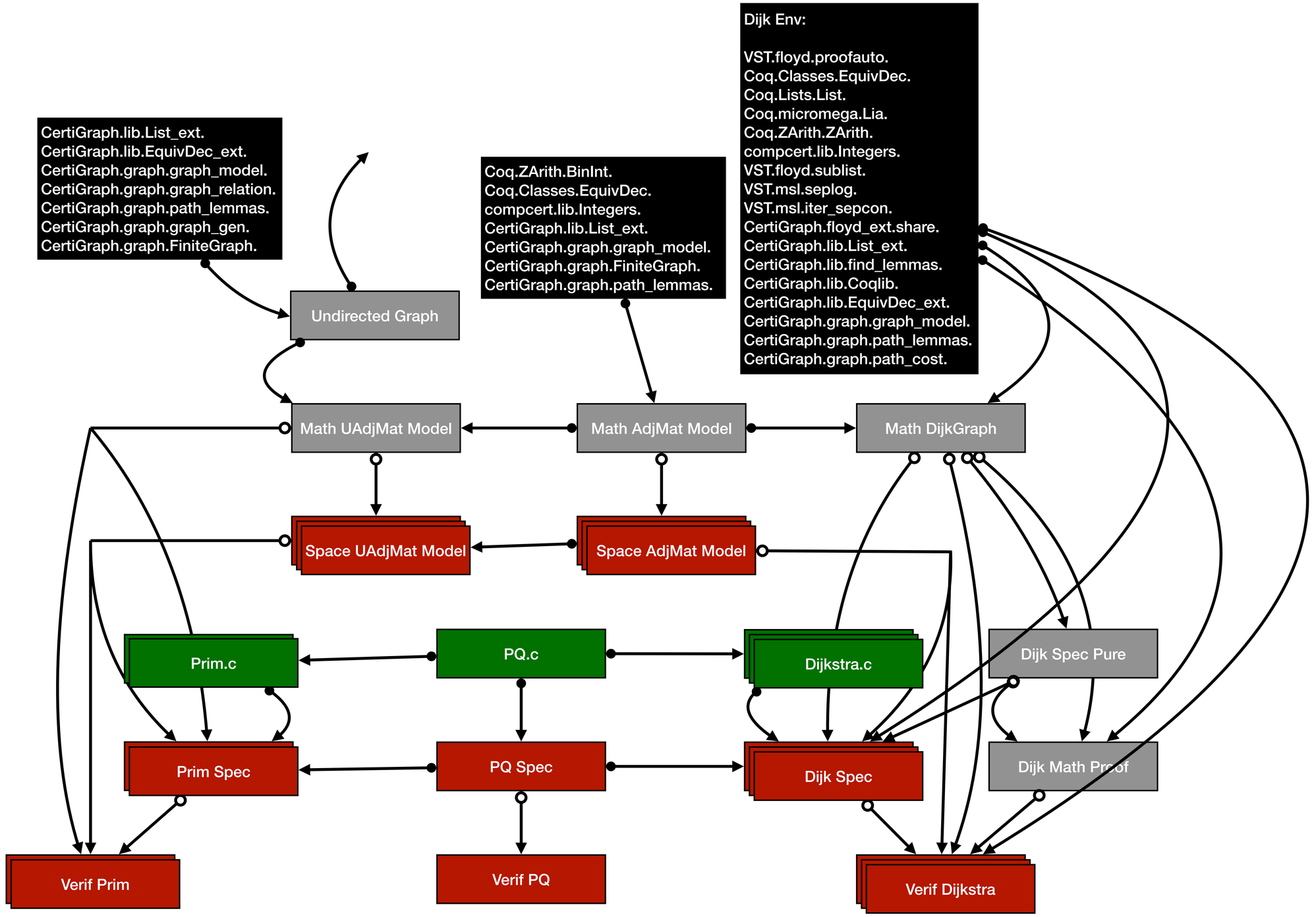
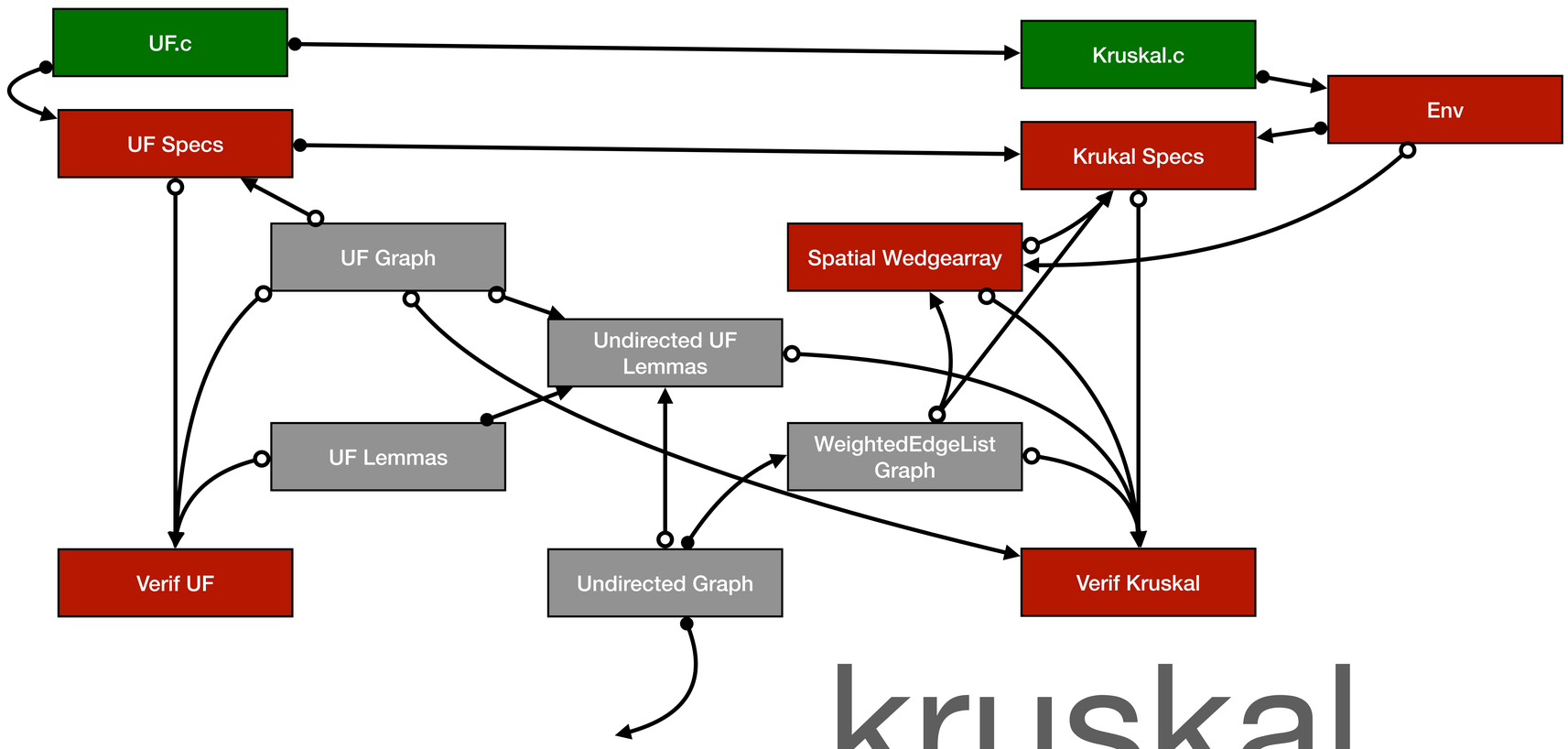


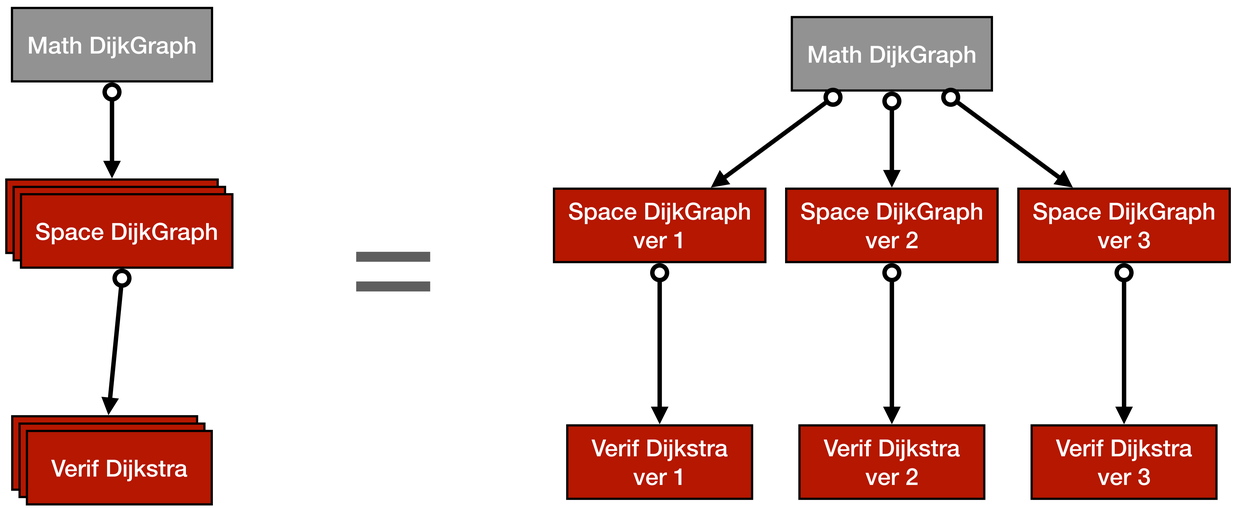
prim and dijk



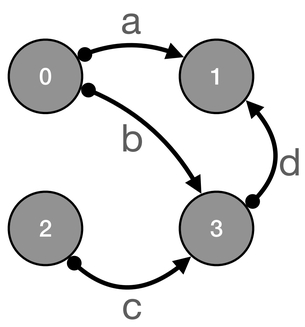
showing key imports


UF.c
Kruskal.c
Env
UF Specs
Krukal Specs
UF Graph
Spatial Wedgearray
Undirected UF
Lemmas
UF Lemmas
WeightedEdgeList
Graph
Verif UF
Undirected Graph
Verif Kruskal

kruskal


Math DijkGraph
Space DijkGraph
Verif Dijkstra
=
Math DijkGraph
Space DijkGraph ver 1
Space DijkGraph ver 2
Space DijkGraph ver 3
Verif Dijkstra ver 1
Verif Dijkstra ver 2
Verif Dijkstra ver 3

V : Z
E: V x V
LE: Z
LV, LG: null



	0	1	2	3
0	0	a	inf	b
1	inf	0	inf	inf
2	inf	inf	0	c
3	inf	d	inf	0

```
g2m g
:=
[
 v2l 0;
 v2l 1;
 v2l 2;
 v2l 3
]
```

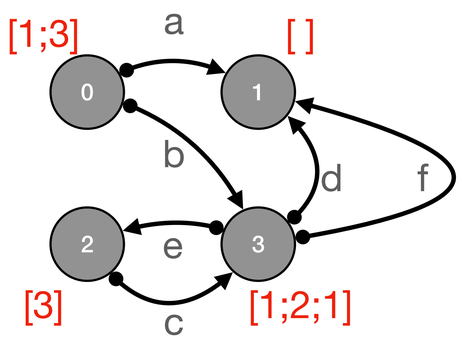
```
g2m g
:=
[
 map elabel [(0,0); (0,1); (0,2); (0,3)];
 map elabel [(1,0); (1,1); (1,2); (1,3)];
 map elabel [(2,0); (2,1); (2,2); (2,3)];
 map elabel [(3,0); (3,1); (3,2); (3,3)]
]
```

```
g2m g
:=
[
 [ 0;   a; inf;   b];
 [inf;   0; inf; inf];
 [inf; inf;   0;   c];
 [inf;   d; inf;   0]
]
```

I ask for a precon:
all lists in (g2m g) have length = size

Quietly, this is a fully-connected graph.
The edges with cost = inf are invalid due to the soundness condition placed on all AdjMats
Edjes to self are free, also due to a soundness condition placed on all AdjMats

```
Lemma elabel_g2m:
 forall i j,
 0 <= i, j < size ->
 elabel g (i, j) = (g2m g)[i][j]
```



	0	1	2
0	(1, a)	(3,b)	
1			
2	(3,c)		
3	(1,d)	(2,e)	(1,f)

```
g2m g
:=
[
 v2l 0;
 v2l 1;
 v2l 2;
 v2l 3
]
```

```
g2m g
:=
[
 map elabel [(0,1,0); (0,3,1)];
 map elabel [ ];
 map elabel [(2,3,0)];
 map elabel [(3,1,0); (3,1,1)]
]
```

```
g2m g
:=
[
 [a; b];
 [ ];
 [c];
 [d; e]
]
```

V : Z
E: V x V x Z
LE: Z
LV: list V check if Shengyi's Fin functions let us get away w/o
LG: null

```
Lemma elabel_g2m:
 forall i j id,
 0 <= i < size ->
 (vlabel i)[id] = j ->
 elabel g (i, j, id) = (g2m g)[i][id]
```