

Table edges_test:

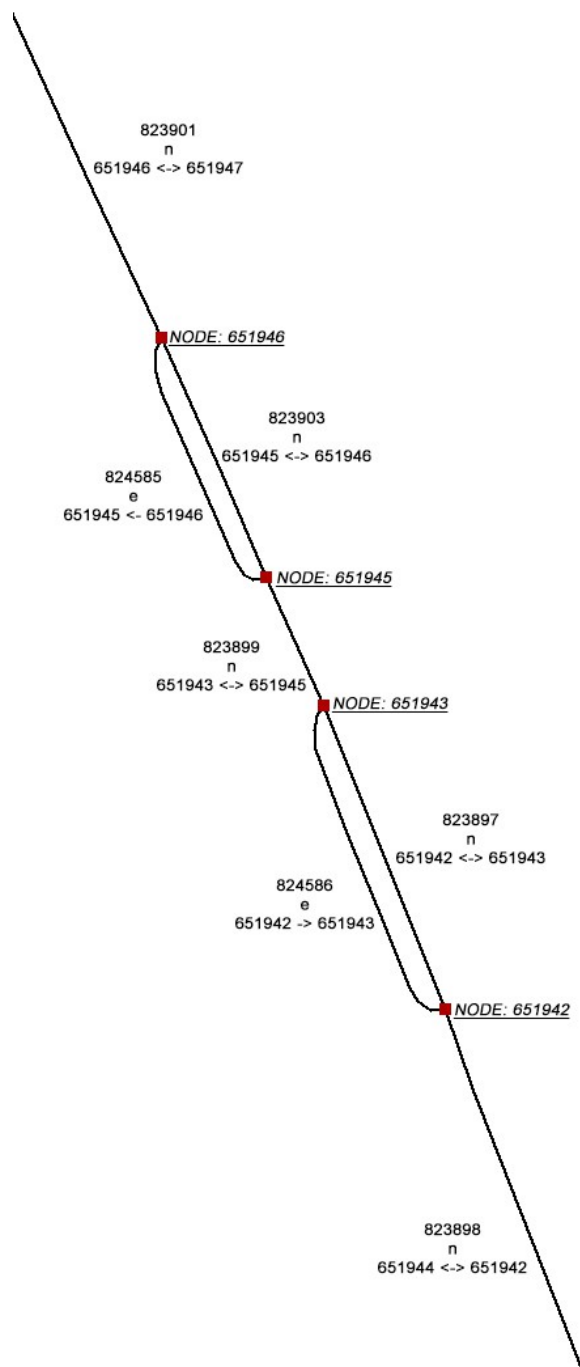
edge_id	from_node	to_node	length	speed	one_way	cost	rev_cost	geom
823897	651942	651943	158	70	n	6.5	6.5	
823898	651944	651942	406	70	n	16.7	16.7	
823899	651943	651945	68	70	n	2.8	2.8	
823901	651946	651947	1099	40	n	79.13	79.13	
823903	651945	651946	127	70	n	5.22	5.22	
824585	651946	651945	140	40	e	12.6	1000000	
824586	651942	651943	171	40	e	15.39	1000000	

one_way = 'n' → two way street

one_way = 'e' → oneway from the source to the target node (reverse_cost = 1000000)

one_way = 'l' → oneway from the target to the source node (cost = 1000000)

Visually:



Vertices Table:

```
SELECT pgr_createVerticesTable('edges_test',
                                'geom',
                                'from_node',
                                'to_node',
                                'true');

NOTICE:  PROCESSING:
NOTICE:  pgr_createVerticesTable('edges_test','geom','from_node','to_node','true')
NOTICE:  Performing checks, pelase wait .....
NOTICE:  Populating public.edges_test_vertices_pgr, please wait...
NOTICE:  ----->  VERTICES TABLE CREATED WITH  6 VERTICES
NOTICE:                                     FOR  7  EDGES
NOTICE:  Edges with NULL geometry,source or target: 0
NOTICE:                                     Edges processed: 7
NOTICE:  Vertices table for table public.edges_test is: public.edges_test_vertices_pgr
NOTICE:  -----
```

Analyze One Way Streets:

```
SELECT pgr_analyzeOneway('edges_test',
                          ARRAY['','n','l'],
                          ARRAY['','n','e'],
                          ARRAY['','n','e'],
                          ARRAY['','n','l'],
                          oneway:='one_way',
                          source:='from_node',
                          target:='to_node');

NOTICE:  PROCESSING:
NOTICE:
pgr_analyzeOneway('edges_test',{'","n,l}','{"","n,e}','{"","n,e}','{"","n,l}','one_way','fro
m_node','to_node',t)
NOTICE:  Analyzing graph for one way street errors.
NOTICE:  Analysis 25% complete ...
NOTICE:  Analysis 50% complete ...
NOTICE:  Analysis 75% complete ...
NOTICE:  Analysis 100% complete ...
NOTICE:  Found 0 potential problems in directionality
```

Analyze a Graph:

```
SELECT pgr_analyzeGraph('edges_test',
                        0.0000001,
                        'geom',
                        'edge_id',
                        'from_node',
                        'to_node',
                        'true');

NOTICE:  PROCESSING:
NOTICE:  pgr_analyzeGraph('edges_test',1e-
07,'geom','edge_id','from_node','to_node','true')
NOTICE:  Performing checks, pelase wait...
NOTICE:  Analyzing for dead ends. Please wait...
NOTICE:  Analyzing for gaps. Please wait...
NOTICE:  Analyzing for isolated edges. Please wait...
NOTICE:  Analyzing for ring geometries. Please wait...
NOTICE:  Analyzing for intersections. Please wait...
NOTICE:  ANALYSIS RESULTS FOR SELECTED EDGES:
NOTICE:  Isolated segments: 0
NOTICE:  Dead ends: 2
NOTICE:  Potential gaps found near dead ends: 0
NOTICE:  Intersections detected: 0
NOTICE:  Ring geometries: 0
```

KSP algorithm from 651942 node to 651946 node:

```
SELECT seq,
       id1,
       id2,
       id3,
       cost
```

```

FROM pgr_ksp('SELECT edge_id AS id,
                    from_node AS source,
                    to_node AS target,
                    cost::float8,
                    rev_cost::float8
                FROM edges_test',
            651942,
            651946,
            1,
            true)
WHERE id3 > 0

```

seq	id1	id2	id3	cost
0	0	651942	824586	15.3900003433228
1	0	651943	823899	2.79999995231628
2	0	651945	824585	1000000

WHY???

KSP algorithm from 651942 node to 651946 node (if only two way street):

```

SELECT seq,
       id1,
       id2,
       id3,
       cost
FROM pgr_ksp('SELECT edge_id AS id,
                    from_node AS source,
                    to_node AS target,
                    cost::float8,
                    rev_cost::float8
                FROM edges_test
                WHERE one_way = 'n'',
            651942,
            651946,
            1,
            true)
WHERE id3 > 0

```

seq	id1	id2	id3	cost
0	0	651942	823897	6.5
1	0	651943	823899	2.79999995231628
2	0	651945	823903	5.21999979019165

CORRECT!

Server: PostgreSQL 9.3.2
 (2.0.0,pgrouting-2.0.0,0,f26831f,master,1.49.0)