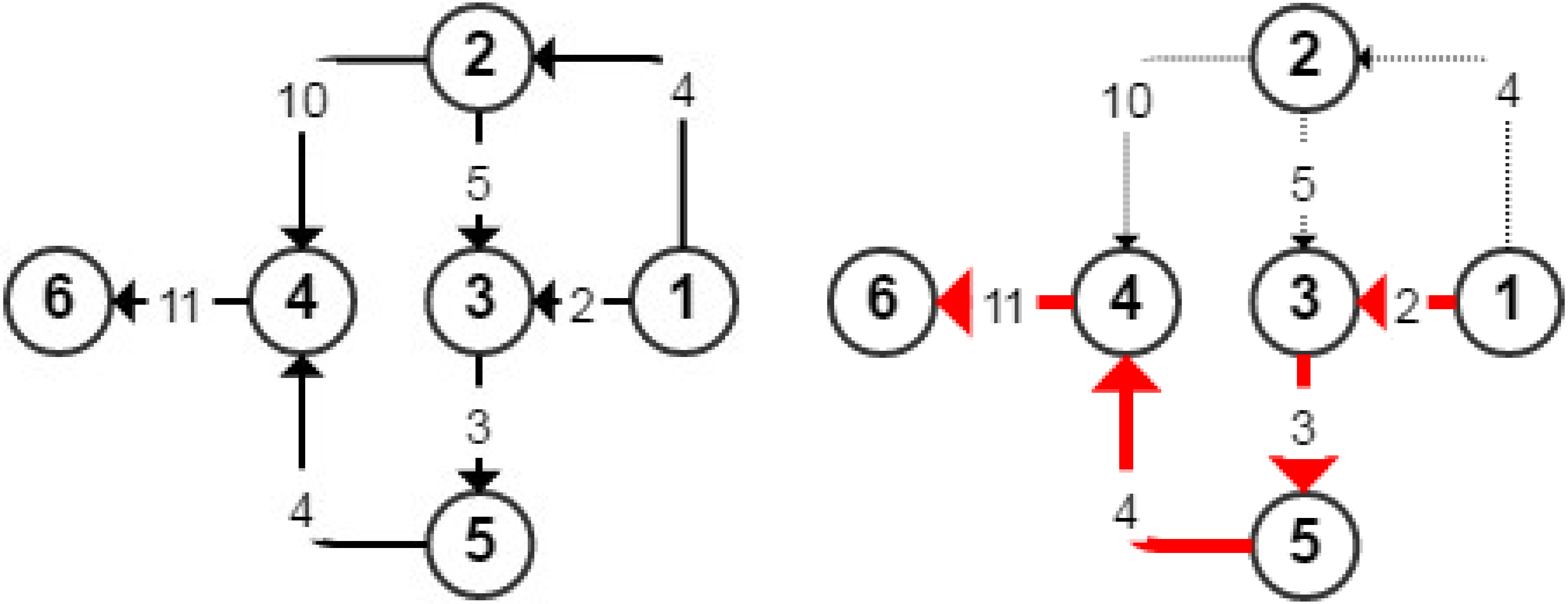
Pg****

https://github.com/raitraidma/tex

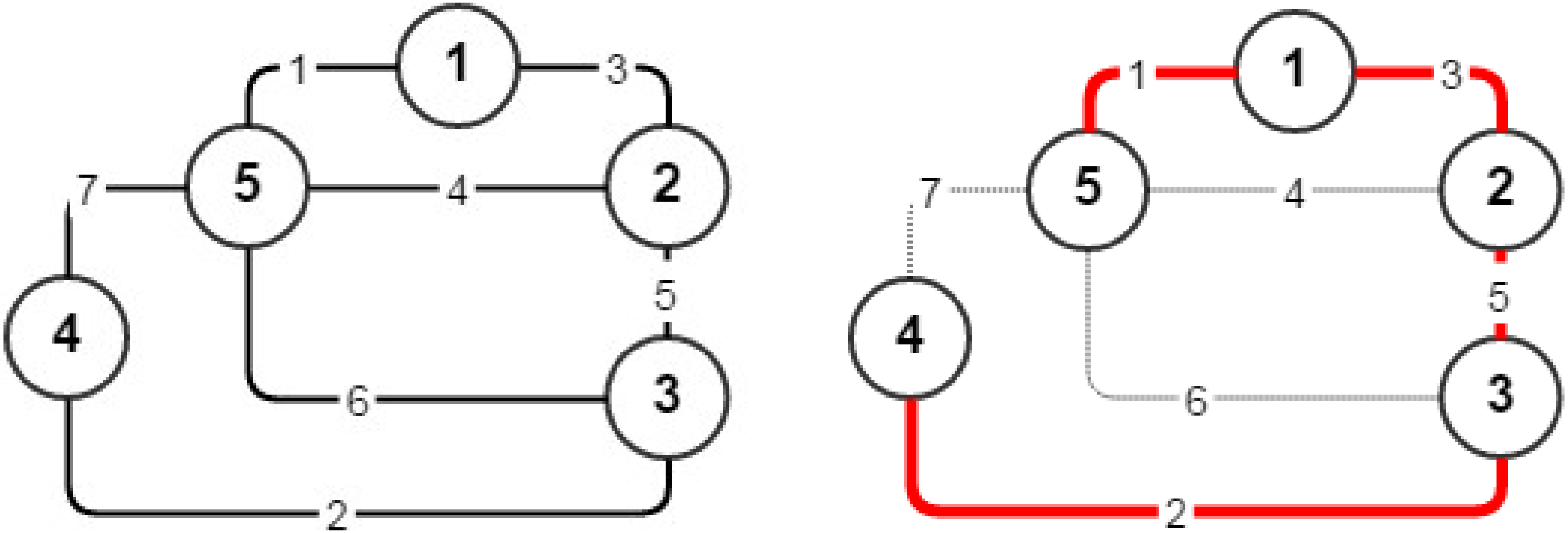
PgRouting

- http://pgrouting.org/
- pgr_dijkstra(edges_sql, start_vid, end_vid, directed);



PgGraph

- https://github.com/raitraidma/pggraph
- MST
- pggraph.kruskal(edges_sql);



PlPythonu

-
  ```
  CREATE OR REPLACE FUNCTION f_http_get(s_url TEXT)
    RETURNS text AS
  $$
  import urllib2
  try:
    result = urllib2.urlopen(s_url).read()
  except:
    result = ''
  return result
  $$
  LANGUAGE 'plpythonu'
    SECURITY DEFINER
    SET search_path=public, pg_temp;
  ```

PlPythonu

```
CREATE OR REPLACE FUNCTION f_geocode_lng_lat(
  IN s_address TEXT
, s_api_key VARCHAR
) RETURNS text ARRAY[2] AS
$$
from geopy.geocoders import GoogleV3
try:
  geolocator = GoogleV3(api_key=s_api_key)
  location = geolocator.geocode(s_address)
  lat, lng = location.latitude, location.longitude
except:
  lat, lng = 0, 0
return lng, lat
$$
LANGUAGE 'plpythonu'
  SECURITY DEFINER
  SET search_path=public, pg_temp;
```

PlPythonu

```
CREATE OR REPLACE FUNCTION f_geocode_address(
  IN d_latitude DOUBLE PRECISION
, IN d_longitude DOUBLE PRECISION
, s_api_key VARCHAR
) RETURNS text AS
$$
from geopy.geocoders import GoogleV3
try:
  geolocator = GoogleV3(api_key=s_api_key)
  location = geolocator.reverse(str(d_latitude) + ", " + str(d_longitude))
  address = location[0].address
except:
  address = "
return address
$$
LANGUAGE 'plpythonu'
  SECURITY DEFINER
  SET search_path=public, pg_temp;
```

PgEnd