In [1]:
```
import os
os.environ['USE_PYGEOS'] = '0'
import geopandas as gpd
import osmnx as ox
import networkx as nx

graph = ox.load_graphml('graph.graphml')
```

In [2]:
```
type(graph)
```

Out[2]:
```
networkx.classes.multidigraph.MultiDiGraph
```

In [3]:
```
graph
```

Out[3]:
```
<networkx.classes.multidigraph.MultiDiGraph at 0x7fb38980da50>
```

In [4]:
```
ox.graph_to_gdfs(graph, nodes=True, edges=False)
```

Out[4]:

osmid	nodeID	x	y	geometry
0	0	474471.651693	4.813048e+06	POINT (474471.652 4813047.859)
1	1	474446.659714	4.813218e+06	POINT (474446.660 4813218.049)
4	4	474465.724070	4.817570e+06	POINT (474465.724 4817570.071)
2	2	473978.436524	4.814030e+06	POINT (473978.437 4814030.307)
3	3	474305.533925	4.813547e+06	POINT (474305.534 4813546.860)
...	...	...	...	...
19787	19787	985086.134288	4.900362e+06	POINT (985086.134 4900362.375)
19788	19788	486748.068451	5.005300e+06	POINT (486748.068 5005299.982)
19789	19789	486103.495526	5.005312e+06	POINT (486103.496 5005312.082)
19791	19791	475494.478782	4.697925e+06	POINT (475494.479 4697925.169)
19792	19792	475856.821534	4.698377e+06	POINT (475856.822 4698376.962)

19793 rows × 4 columns

In [5]:
```
nodes = ox.graph_to_gdfs(graph, nodes=True, edges=False)
edges = ox.graph_to_gdfs(graph, nodes=False, edges=True)
```

In [6]:
```
nodes.shape[0]
```

Out[6]:
```
19793
```

In [7]:
```
edges
```

Out[7]:

u	v	key	name	length	geometry
0	**1**	**0**	CABIN CREEK S.R.A. CONN #1	172.406720	LINESTRING (474471.652 4813047.859, 474471.427...
	96	**0**	CABIN CREEK S.R.A. CONN #1	199.893815	LINESTRING (474434.459 4812856.736, 474459.957...
		1	CABIN CREEK S.R.A. CONN #2	195.040370	LINESTRING (474434.459 4812856.736, 474451.459...
1	**4**	**0**	PACIFIC	4982.726110	LINESTRING (474446.660 4813218.049, 474441.478...
	0	**0**	CABIN CREEK S.R.A. CONN #1	172.406720	LINESTRING (474471.652 4813047.859, 474471.427...
...	**...**	**...**	...	...	...
19788	**19789**	**0**	E. MCMINNVILLE FRONTAGE RD	644.958312	LINESTRING (486103.496 5005312.082, 486130.563...
19789	**19788**	**0**	E. MCMINNVILLE FRONTAGE RD	644.958312	LINESTRING (486103.496 5005312.082, 486130.563...
	19790	**0**	E. MCMINNVILLE FRONTAGE RD	237.927444	LINESTRING (485932.091 5005457.706, 485939.145...
19791	**19792**	**0**	FOOTHILL BLVD. FRONTAGE RD	719.563780	LINESTRING (475494.479 4697925.169, 475608.321...
19792	**19791**	**0**	FOOTHILL BLVD. FRONTAGE RD	719.563780	LINESTRING (475494.479 4697925.169, 475608.321...

42508 rows × 3 columns

In [8]:
```
edges.shape
```

Out[8]:
```
(42508, 3)
```

In [9]:
```
edges.shape[0]
```

Out[9]:
```
42508
```

In [10]:
```
import os
import geopandas as gpd
import osmnx as ox
import networkx as nx

graph = ox.load_graphml('graph.graphml')
```

In [11]:
```
type(graph)
```

Out[11]:
```
networkx.classes.multidigraph.MultiDiGraph
```

In [12]:
```
nodes.crs
```

Out[12]:
```
<Derived Projected CRS: EPSG:32610>
Name: WGS 84 / UTM zone 10N
Axis Info [cartesian]:
- E[east]: Easting (metre)
- N[north]: Northing (metre)
Area of Use:
- name: Between 126°W and 120°W, northern hemisphere between equator and 84°N, onshore and offshore. Canada - British
Columbia (BC); Northwest Territories (NWT); Nunavut; Yukon. United States (USA) - Alaska (AK).
- bounds: (-126.0, 0.0, -120.0, 84.0)
Coordinate Operation:
- name: UTM zone 10N
- method: Transverse Mercator
Datum: World Geodetic System 1984 ensemble
- Ellipsoid: WGS 84
- Prime Meridian: Greenwich
```

In [13]:
```
edges.head()
```

Out[13]:

			name	length	geometry
u	**v**	**key**			
0	**1**	**0**	CABIN CREEK S.R.A. CONN #1	172.406720	LINESTRING (474471.652 4813047.859, 474471.427...
	96	**0**	CABIN CREEK S.R.A. CONN #1	199.893815	LINESTRING (474434.459 4812856.736, 474459.957...
		1	CABIN CREEK S.R.A. CONN #2	195.040370	LINESTRING (474434.459 4812856.736, 474451.459...
1	**4**	**0**	PACIFIC	4982.726110	LINESTRING (474446.660 4813218.049, 474441.478...
	0	**0**	CABIN CREEK S.R.A. CONN #1	172.406720	LINESTRING (474471.652 4813047.859, 474471.427...

In [14]:
```
edges['length'].min()
```

Out[14]:
```
1.370217966253487
```

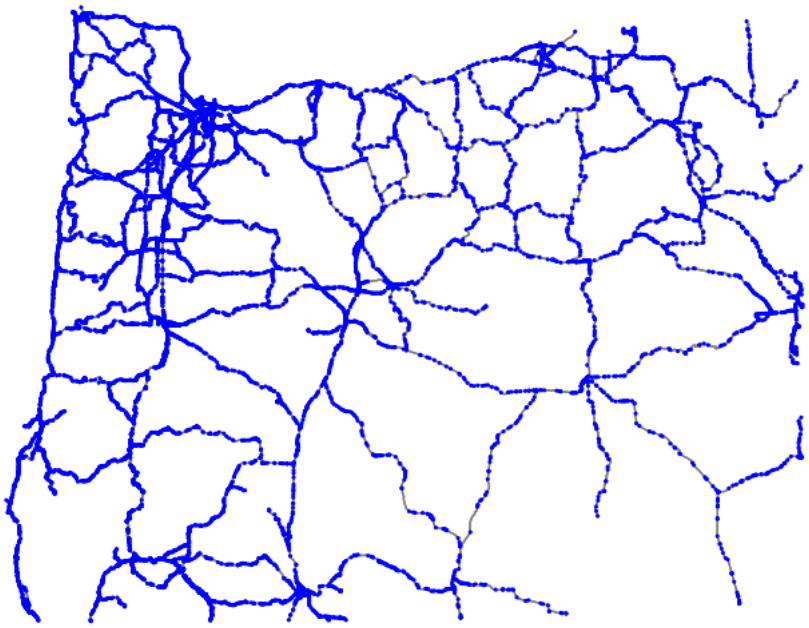
In [15]:
```
edges['length'].max()
```

Out[15]:
```
21730.128883116628
```

In [16]:
```
edges['length'].mean()
```

Out[16]:
```
674.3728080973483
```

In [17]:
```
fig, ax = ox.plot_graph(graph, bgcolor='white', node_color='blue', edge_color='grey', node_size=5)
```



In [18]:
```
import os
os.environ['USE_PYGEOS'] = '0'
import geopandas as gpd

cities = gpd.read_file('data/oregon_cities.shp')
cities.head()
```

Out[18]:

	City	lat	lon	geometry
0	Adair Village city	44.67	-123.22	POINT (-123.22000 44.67000)
1	Adams	45.77	-118.56	POINT (-118.56000 45.77000)
2	Adrian	43.74	-117.07	POINT (-117.07000 43.74000)
3	Albany	44.63	-123.10	POINT (-123.10000 44.63000)
4	Aloha	45.49	-122.87	POINT (-122.87000 45.49000)

In [19]:
```
cities.crs
```

Out[19]:
```
<Geographic 2D CRS: EPSG:4326>
Name: WGS 84
Axis Info [ellipsoidal]:
- Lat[north]: Geodetic latitude (degree)
- Lon[east]: Geodetic longitude (degree)
Area of Use:
- name: World.
- bounds: (-180.0, -90.0, 180.0, 90.0)
Datum: World Geodetic System 1984 ensemble
- Ellipsoid: WGS 84
- Prime Meridian: Greenwich
```

In [20]:
```
cities_reproject = cities.to_crs('EPSG:32610')
cities_reproject.crs
```

Out[20]:
```
<Derived Projected CRS: EPSG:32610>
Name: WGS 84 / UTM zone 10N
Axis Info [cartesian]:
- E[east]: Easting (metre)
- N[north]: Northing (metre)
Area of Use:
- name: Between 126°W and 120°W, northern hemisphere between equator and 84°N, onshore and offshore. Canada - British
Columbia (BC); Northwest Territories (NWT); Nunavut; Yukon. United States (USA) - Alaska (AK).
- bounds: (-126.0, 0.0, -120.0, 84.0)
Coordinate Operation:
- name: UTM zone 10N
- method: Transverse Mercator
Datum: World Geodetic System 1984 ensemble
- Ellipsoid: WGS 84
- Prime Meridian: Greenwich
```

In [21]:
```
cities_reproject
```

Out[21]:

	City	lat	lon	geometry
0	Adair Village city	44.67	-123.22	POINT (482561.392 4946316.184)
1	Adams	45.77	-118.56	POINT (845212.127 5078087.252)
2	Adrian	43.74	-117.07	POINT (977541.425 4860113.062)
3	Albany	44.63	-123.10	POINT (492067.910 4941854.290)
4	Aloha	45.49	-122.87	POINT (510158.282 5037393.753)
...	...	...	...	...
372	Wood Village	45.54	-122.42	POINT (545281.434 5043103.988)
373	Woodburn	45.15	-122.86	POINT (511005.313 4999623.251)
374	Yachats	44.31	-124.10	POINT (412268.825 4906893.114)
375	Yamhill	45.34	-123.19	POINT (485113.923 5020738.776)
376	Yoncalla	43.60	-123.29	POINT (476593.869 4827488.132)

377 rows × 4 columns

In [22]:
```
city1 = cities_reproject[cities_reproject['City'] == 'Adams']
```

In [23]:
```
city2 = cities_reproject[cities_reproject['City'] == 'Adrian']
```

In [24]:
```
city3 = cities_reproject[cities_reproject['City'] == 'Albany']
```

In [25]:
```
city4 = cities_reproject[cities_reproject['City'] == 'Aloha']
```

In [39]:
```
adams = cities_reproject[cities_reproject['City'] == 'Adams'].reset_index()
adrian = cities_reproject[cities_reproject['City'] == 'Adrian'].reset_index()
albany = cities_reproject[cities_reproject['City'] == 'Albany'].reset_index()
aloha = cities_reproject[cities_reproject['City'] == 'Aloha'].reset_index()
```

In [40]:
```
dis1 = adams.distance(adrian).values[0] / 1000
dis1
```

Out[40]:
```
254.997629078076
```

In [41]:
```
dis2 = adams.distance(albany).values[0] / 1000
dis2
```

Out[41]:
```
378.51057794288084
```

In [42]:
```
dis3 = adams.distance(aloha).values[0] / 1000
dis3
```

Out[42]:
```
337.51598488788926
```

In [43]:
```
dis4 = adrian.distance(albany).values[0] / 1000
dis4
```

Out[43]:
```
492.30697951205997
```

In [44]:
```
dis5 = adrian.distance(aloha).values[0] / 1000
dis5
```

Out[44]:
```
499.87543072256335
```

In [45]:
```
dis6 = albany.distance(aloha).values[0] / 1000
dis6
```

Out[45]:
```
97.23708377262562
```

In [46]:
```
adams_coord_x = city1["geometry"].x
```

In [47]:
```
adams_coord_x
```

Out[47]:
```
1    845212.126836
dtype: float64
```

In [48]:
```
adams_coord_y = city1["geometry"].y
```

In [49]:
```
adams_coord_y
```

Out[49]:
```
1    5.078087e+06
dtype: float64
```

In [50]:
```
adams_target_node = ox.distance.nearest_nodes(graph, X=adams_coord_x, Y=adams_coord_y, return_dist=False)[0]
```

In [51]:
```
adams_target_node
```

Out[51]:
```
4276
```

In [52]:
```
adrian_x = city2["geometry"].x
```

In [53]:
```
adrian_y = city2["geometry"].y
```

In [54]:
```
adrian_target_node = ox.distance.nearest_nodes(graph, X=adrian_x, Y=adrian_y, return_dist=False)[0]
```

In [55]:
```
adrian_target_node
```

Out[55]:
```
15186
```

In [56]:
```
albany_x = city3["geometry"].x
```

In [57]:
```
albany_y = city3["geometry"].y
```

In [58]:
```
albany_target_node = ox.distance.nearest_nodes(graph, X=albany_x, Y=albany_y, return_dist=False)[0]
```

In [59]:
```
albany_target_node
```

Out[59]:
```
16435
```

In [60]:
```
aloha_x = city4["geometry"].x
```

In [61]:
```
aloha_y = city4["geometry"].y
```

In [62]:
```
aloha_target_node = ox.distance.nearest_nodes(graph, X=aloha_x, Y=aloha_y, return_dist=False)[0]
```

In [63]:
```
aloha_target_node
```

Out[63]:
```
11650
```

In [64]:
```
# Calculate the shortest path
route3 = nx.shortest_path(graph, source=albany_target_node , target=aloha_target_node, weight='length')
length = nx.shortest_path_length(graph, source=albany_target_node, target=aloha_target_node, weight='length')
print("Shortest path distance = {t:.1f} km.".format(t=length/1000))
```

```
Shortest path distance = 112.2 km.
```

In [65]:
```
print("Shortest path distance = {t:.1f} km.".format(t=length/1000))
```

```
Shortest path distance = 112.2 km.
```

In [66]:
```
# Calculate the shortest path
route2 = nx.shortest_path(graph, source=albany_target_node, target=adrian_target_node, weight='length')
length = nx.shortest_path_length(graph, source=albany_target_node, target=adrian_target_node, weight='length')
print("Shortest path distance = {t:.1f} km.".format(t=length/1000))
```

```
Shortest path distance = 640.9 km.
```

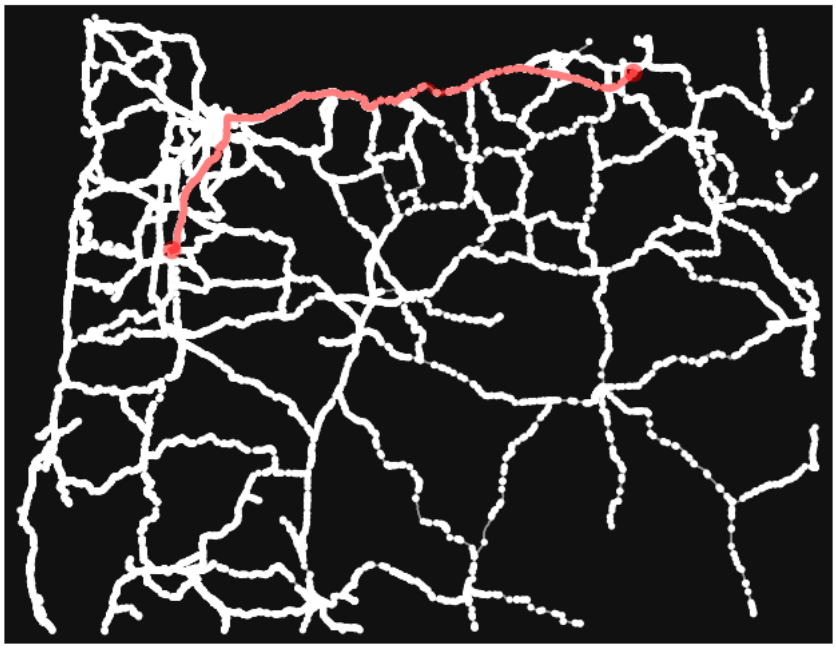
In [67]:
```
print("Shortest path distance = {t:.1f} km.".format(t=length/1000))
```

```
Shortest path distance = 640.9 km.
```

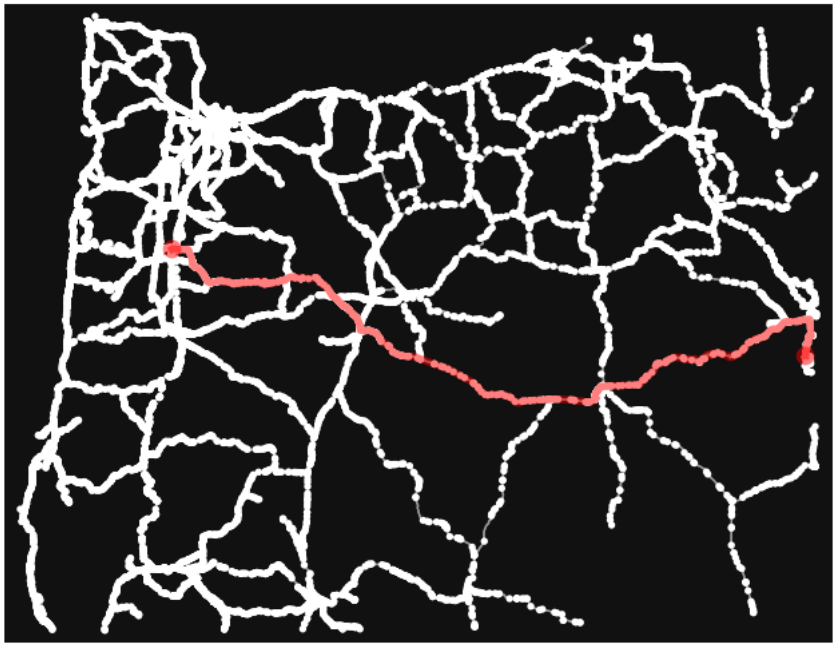
In [68]:
```
# Calculate the shortest path
route1 = nx.shortest_path(graph, source=albany_target_node, target=adams_target_node, weight='length')
length = nx.shortest_path_length(graph, source=albany_target_node, target=adams_target_node, weight='length')
print("Shortest path distance = {t:.1f} km.".format(t=length/1000))
```

```
Shortest path distance = 465.5 km.
```

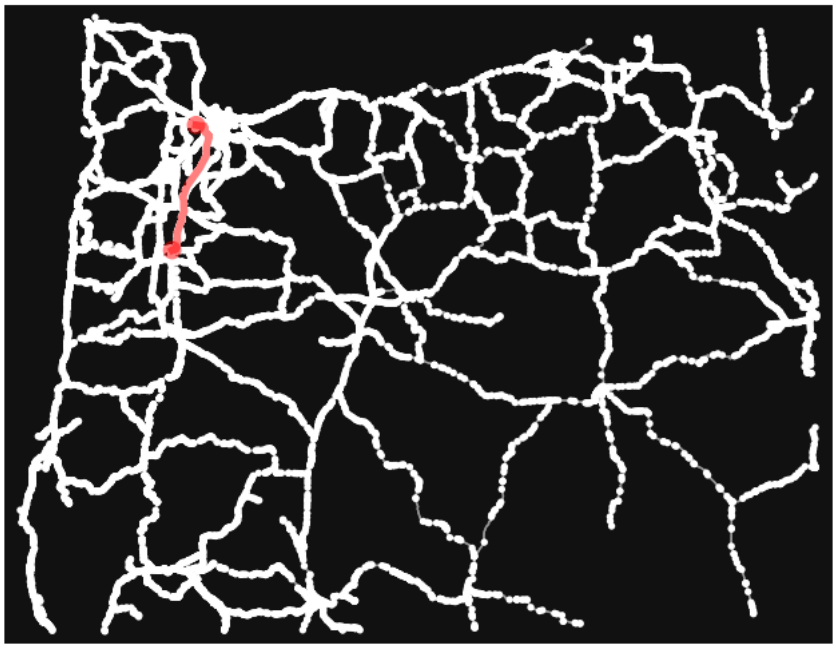
In [69]:
```
fig, ax = ox.plot_graph_route(graph, route1)
```



In [70]:
```
fig, ax = ox.plot_graph_route(graph, route2)
```



In [71]:
```
fig, ax = ox.plot_graph_route(graph, route3)
```



In [72]:
```
cities_reproject
```

Out[72]:

	City	lat	lon	geometry
0	Adair Village city	44.67	-123.22	POINT (482561.392 4946316.184)
1	Adams	45.77	-118.56	POINT (845212.127 5078087.252)
2	Adrian	43.74	-117.07	POINT (977541.425 4860113.062)
3	Albany	44.63	-123.10	POINT (492067.910 4941854.290)
4	Aloha	45.49	-122.87	POINT (510158.282 5037393.753)
...	...	...	...	...
372	Wood Village	45.54	-122.42	POINT (545281.434 5043103.988)
373	Woodburn	45.15	-122.86	POINT (511005.313 4999623.251)
374	Yachats	44.31	-124.10	POINT (412268.825 4906893.114)
375	Yamhill	45.34	-123.19	POINT (485113.923 5020738.776)
376	Yoncalla	43.60	-123.29	POINT (476593.869 4827488.132)

377 rows × 4 columns

In [74]:
```
edges.head()
```

Out[74]:

			name	length	geometry
u	**v**	**key**			
0	**1**	**0**	CABIN CREEK S.R.A. CONN #1	172.406720	LINESTRING (474471.652 4813047.859, 474471.427...
	96	**0**	CABIN CREEK S.R.A. CONN #1	199.893815	LINESTRING (474434.459 4812856.736, 474459.957...
		1	CABIN CREEK S.R.A. CONN #2	195.040370	LINESTRING (474434.459 4812856.736, 474451.459...
1	**4**	**0**	PACIFIC	4982.726110	LINESTRING (474446.660 4813218.049, 474441.478...
	0	**0**	CABIN CREEK S.R.A. CONN #1	172.406720	LINESTRING (474471.652 4813047.859, 474471.427...

In [76]:
```
travel_speed = 60
meters_per_minute = travel_speed * 1000 / 60  # km per hour to m per minute

for u, v, data in graph.edges.data():
    data['time'] = data['length'] / meters_per_minute
```

In [77]:
```
list(graph.edges.data())[0]
```

Out[77]:
```
(0,
 1,
 {'name': 'CABIN CREEK S.R.A. CONN #1',
  'length': 172.40672049682195,
  'geometry': <LINESTRING (474471.652 4813047.859, 474471.427 4813067.65, 474470.456 48130...>,
  'time': 0.17240672049682196})
```

In [ ]:
```

```