



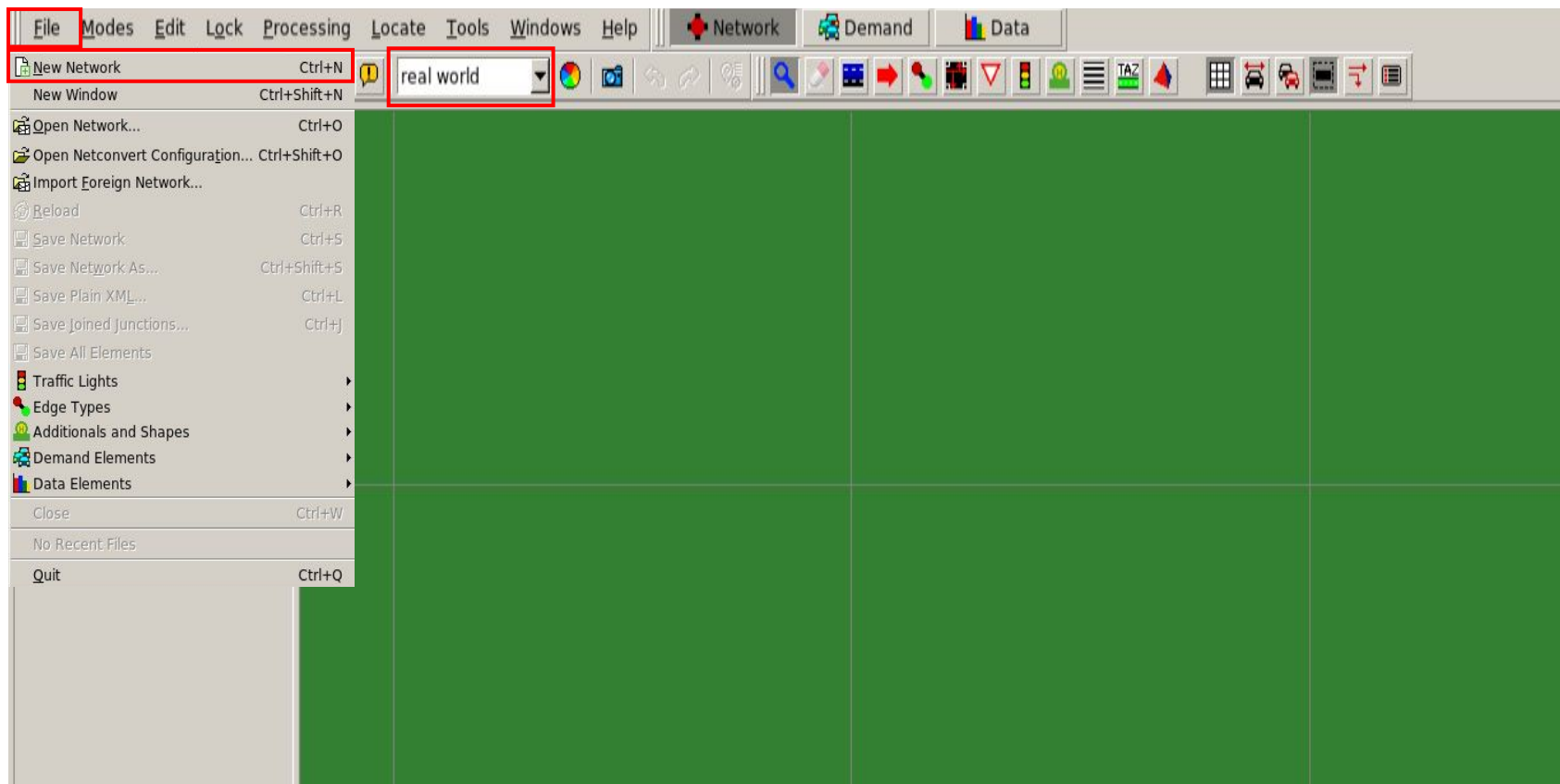
SUMO繪製及生成車流



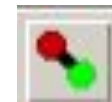
✓ 利用 NetEdit 繪製路網

(1) 建立新檔

✓ 開啟NetEdit, File□New Network建立新檔案, 修改背景為real world。



移動物件



繪製路口與路段



查看物件屬性或移動底圖



刪除物件

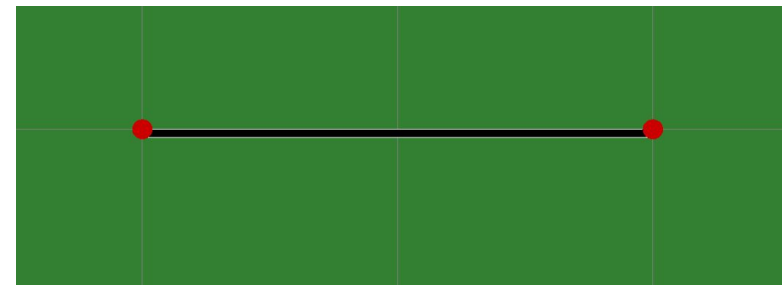



✓ 利用 NetEdit 繪製路網

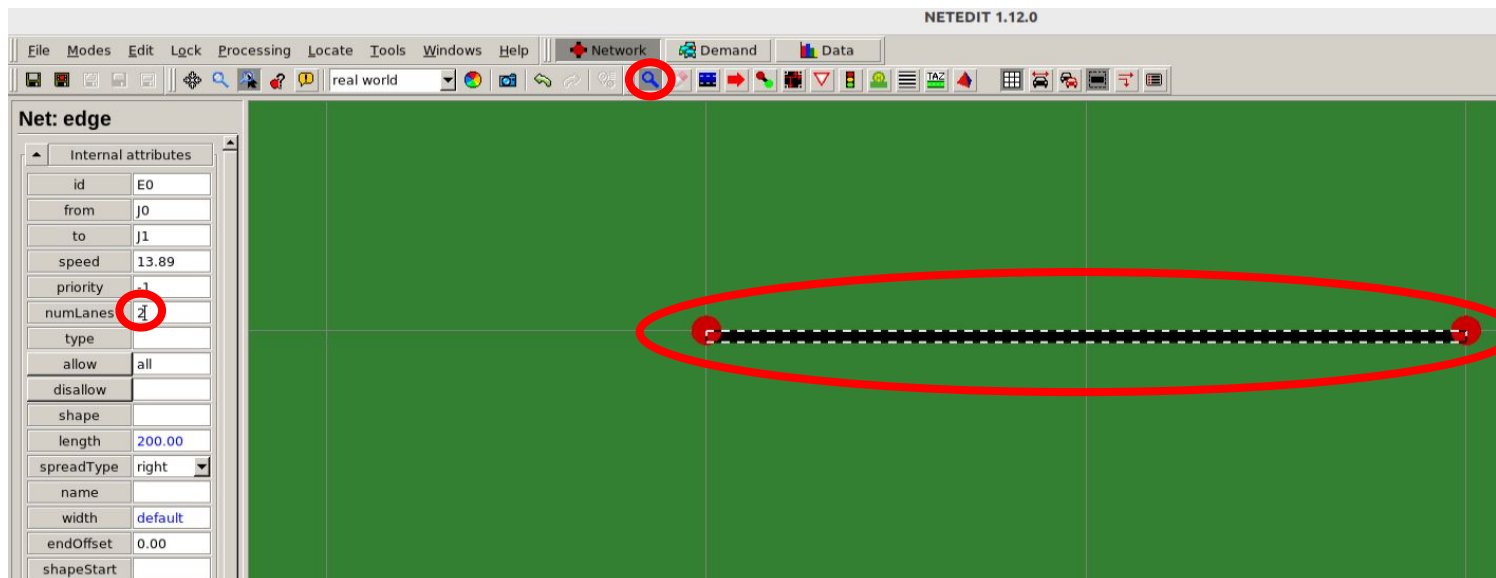
(2)手繪路網

- ✓ 建立簡易的雙路口路網，路段皆為雙向各2車道

Step1: 選  在底圖點2個位置(起點與終點)，畫出一條路段



Step2: 選  點路段將跳出路網的屬性設定值，修改numLanes數值為2，變為2車道。





✓ 利用 NetEdit 繪製路網


(2)手繪路網

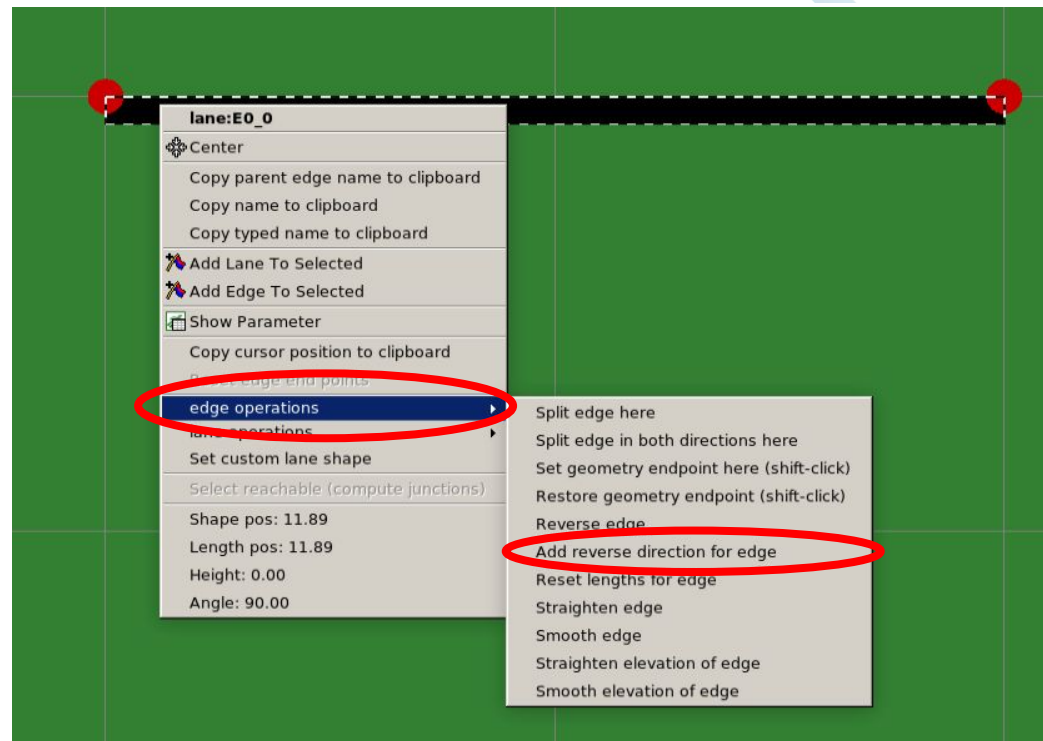
- ✓ 建立簡易的雙路口路網，路段皆為雙向各2車道

Step3: 選  在路段按滑鼠右鍵，選edge operations

- ☐ Add reverse direction for edge, 將自動建立相同車道數的對向車道。

Step4: 選  可將  最兩端的紅色節點左右移動，以延長路段長度。(單位為公尺)

Step5: 重複Step1~Step4 繪製更多路段。過程中如果有畫錯，點選  刪除物件。





✓ 利用 NetEdit 繪製路網

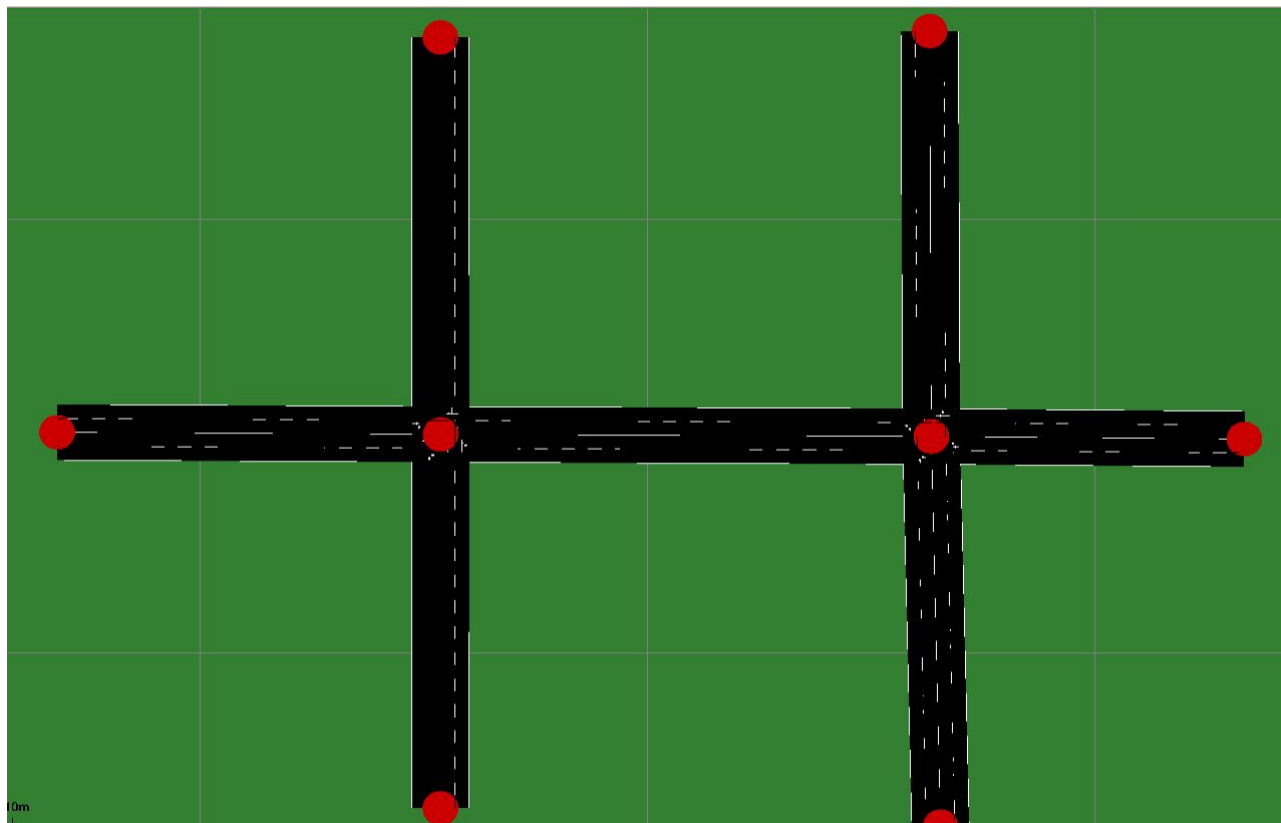
(2)手繪路網

- ✓ 建立簡易的雙路口路網，路段皆為雙向各2車道

【練習】重複Step1~Step4 繪製出右圖。

▲ 每條路段皆為2車道雙向道(共4條車道)

▲ 圖形包含8個節點



✓ 利用 NetEdit 繪製路網

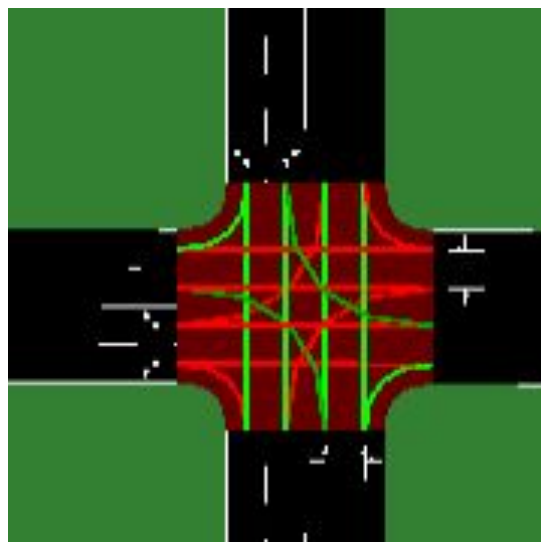
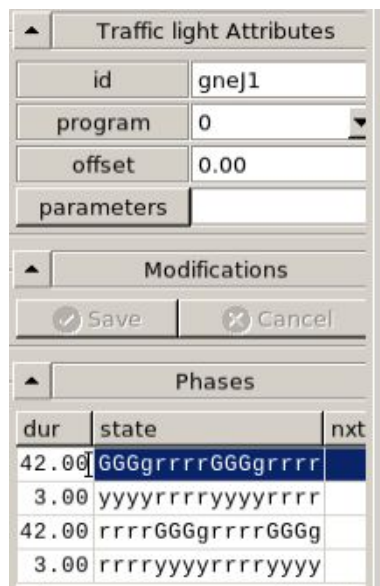
(2)手繪路網

✓ 建立路口號誌

Step1: 選  接著點左側的路口，將跳出設定號誌屬性的畫面



Step2: 在設定畫面選Create TLS，自動預設號誌時制計畫。Phases 方塊內任意上下點選，可看到路口內各流向顏色的變化。



【說明】


第一個時相為42秒，並自動設定一串「GGGgrrrrGGGgrrrr」，共16個英文字，代表16個流向對應的號誌燈。G或g為綠燈，r為紅燈。大寫G指衝突車流時具有優先權；小寫g則為不具優先權，需禮讓其他車流。

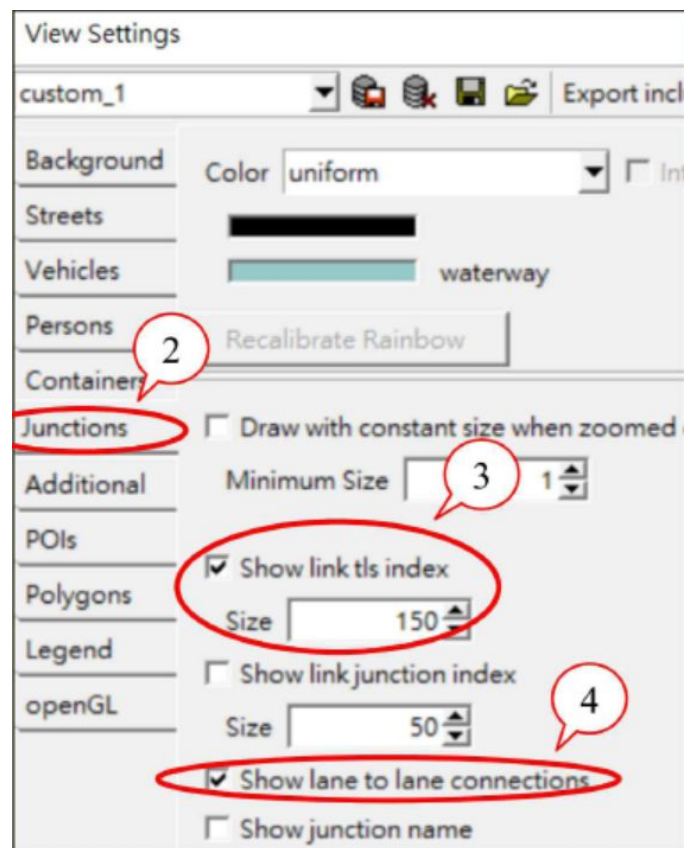
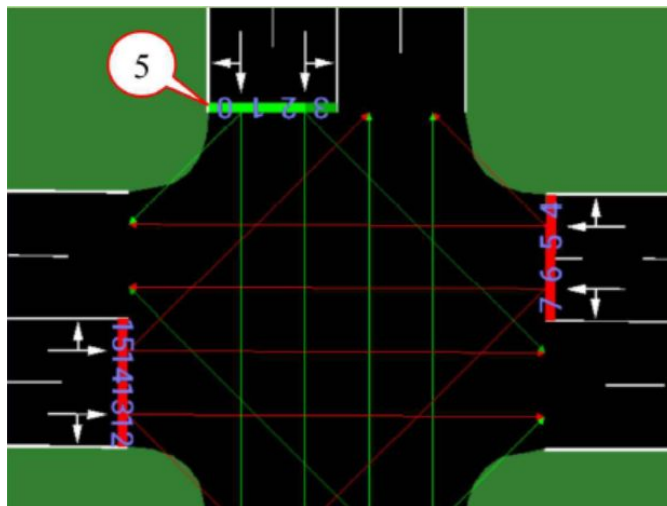
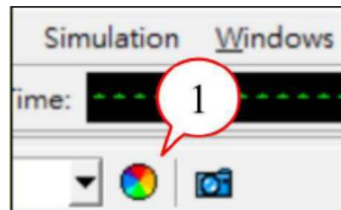


✓ 利用 NetEdit 繪製路網

(2)手繪路網

✓ 建立路口號誌

Step3: 如果想要查詢各流向的順序, 需開啟 sumo-gui, File □ Open Network, 在工具中選 , 跳出 View Settings, 在左側的 Junction □ 勾選 Show link tls index □ 將 Size 數值調大 □ 勾選 Show lane to lane connections。由圖可知, 路口往南的流向, index:0、1、2、3 分別為右轉、直行、直行、左轉。





✓ 利用 NetEdit 繪製路網

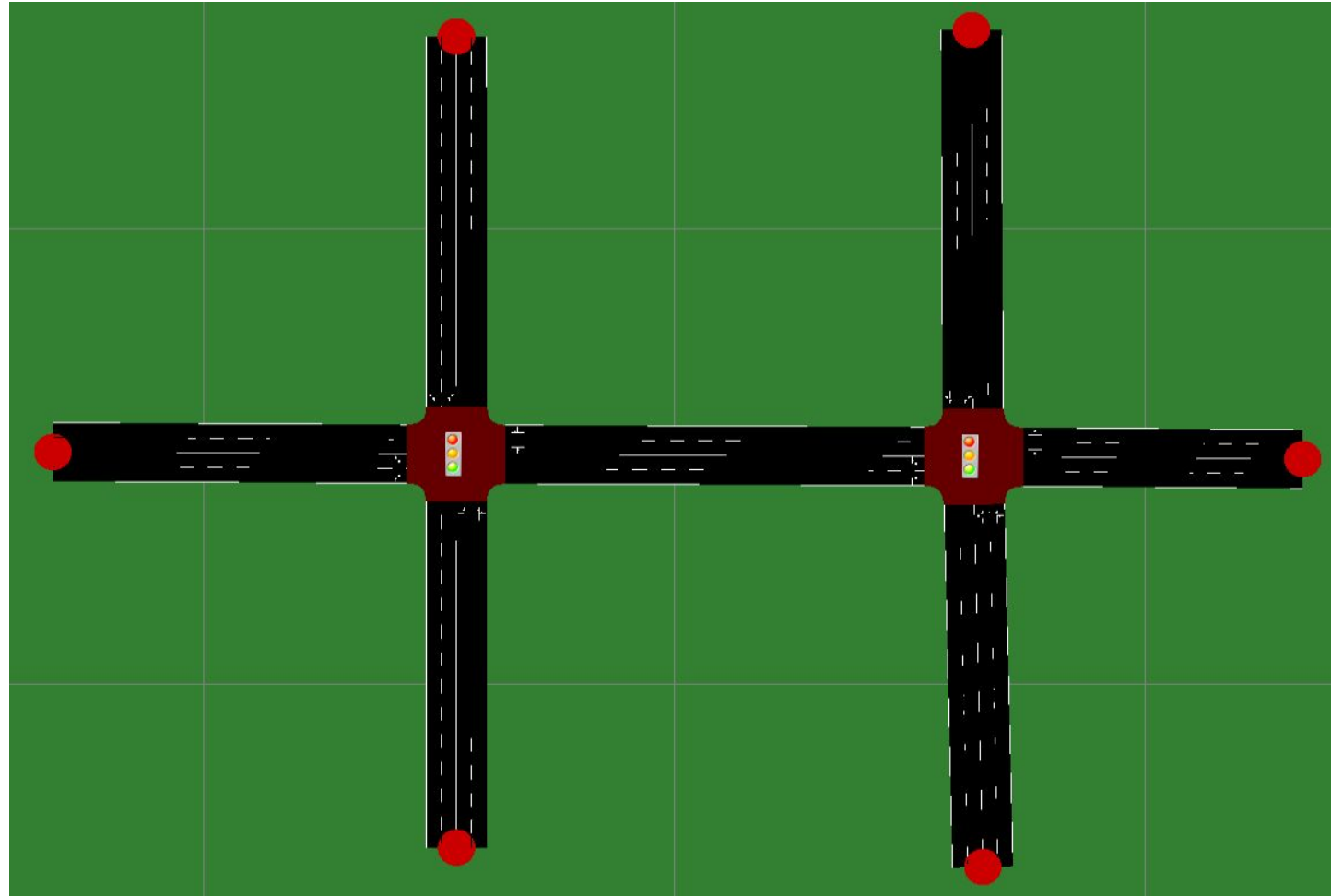
(2)手繪路網

✓ 建立路口號誌

【練習】重複Step1~Step2 繪製出右圖。

▲ 完成兩個路口的號誌設定

▲ 檔案儲存為：學號_HW2_v1.net.xml





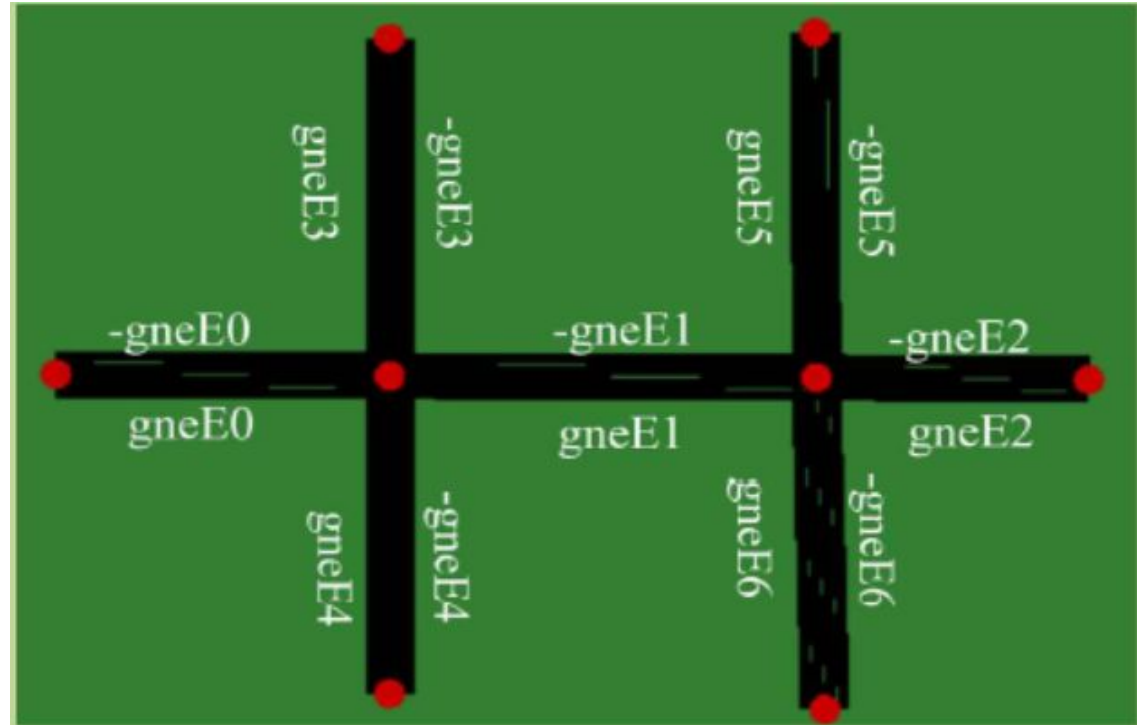
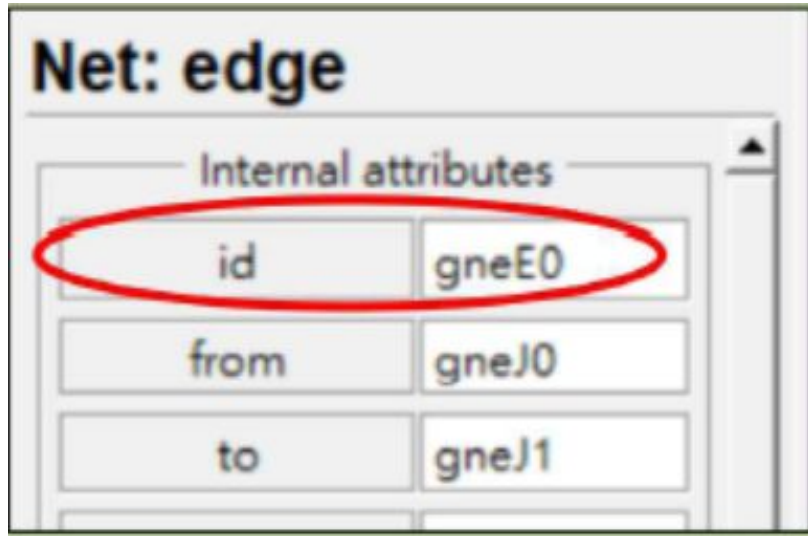
✓ 輸入交通流量 自動產生路徑

✓ 查詢路段名稱

在NetEdit中，選擇



點路段，將於左邊顯示屬性資料，其中id即路段名稱，並可修改。



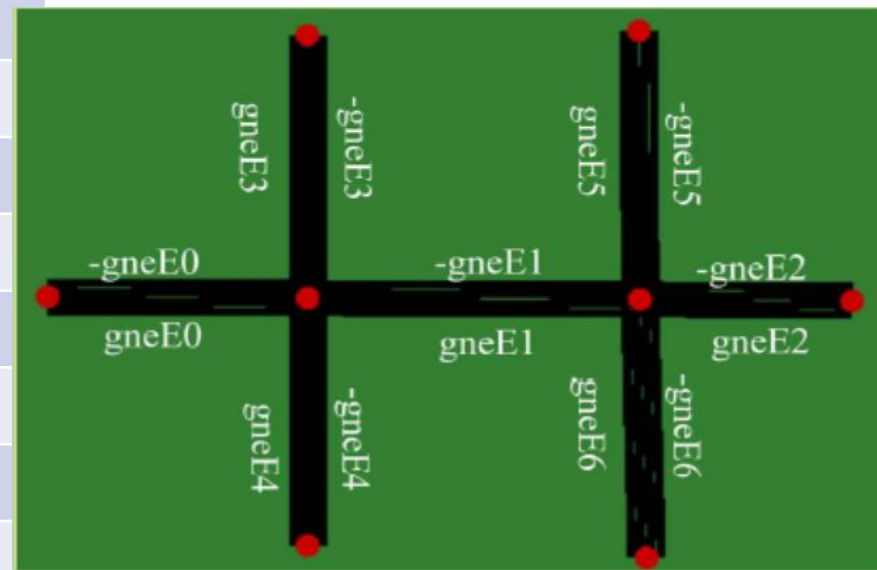


✓ 輸入交通流量

自動產生路徑

- ✓ 建立OD分布表:假設某日上午7點至8點, 交通量分布如下表, 車種為CarA與CarB。

序號	O	D	車種	Volume(hr)
1	genE0	genE2	CarA	800
2	genE0	-genE5	CarA	120
3	genE0	genE4	CarA	100
4	-genE2	-genE0	CarA	500
5	-genE2	genE4	CarA	100
6	genE3	genE4	CarA	200
7	-genE4	-genE3	CarA	200
8	genE5	genE6	CarA	300
9	-genE6	-genE5	CarA	250
10	genE3	genE6	CarB	150





✓ 輸入交通流量

自動產生路徑

- ✓ 利用duarouter.exe, 自動尋找最短路徑。

Step1: 建立檔名為dua_flow2route.duacfg

, 格式如下, net-file、route-file是指定的

輸入檔, output-file是指定的輸出檔。

```
<configuration>
```

```
<input>
```

```
<net-file value="M1354030_HW2_v1.net.xml"/>
```

```
<route-files value="dua_flow2route.xml"/>
```

```
</input>
```

```
<output>
```

```
<write-license value="true"/>
```

```
<output-file value="dua_rou.xml"/>
```

```
</output>
```

```
<report>
```

```
<xml-validation value="never"/>
```

```
<no-step-log value="true"/>
```

```
</report>
```

```
</configuration>
```



✓ 輸入交通流量

自動產生路徑

- ✓ 利用duarouter.exe, 自動尋找最短路徑。

Step2: 依據交通量分布及車種, 建route-files所需的檔案, 格式如下, 檔名為dua_flow2route.xml。

```
<routes>
  <vType id="CarA" accel="3.0" decel="6.0" length="5.0" minGap="2.5" maxSpeed="10.0" sigma="0.5"/>
  <vType id="CarB" accel="2.0" decel="6.0" length="7.5" minGap="2.5" maxSpeed="8.0" sigma="0.5"/>

  <flow id="1" type="CarA" from="gneE0" to="gneE2" begin="0" end="3600" number="800"/>
  <flow id="2" type="CarA" from="gneE0" to="-gneE5" begin="0" end="3600" number="120"/>
  <flow id="3" type="CarA" from="gneE0" to="gneE4" begin="0" end="3600" number="100"/>
  <flow id="4" type="CarA" from="-gneE2" to="-gneE0" begin="0" end="3600" number="500"/>
  <flow id="5" type="CarA" from="-gneE2" to="gneE4" begin="0" end="3600" number="100"/>
  <flow id="6" type="CarA" from="gneE3" to="gneE4" begin="0" end="3600" number="200"/>
  <flow id="7" type="CarA" from="-gneE4" to="-gneE3" begin="0" end="3600" number="200"/>
  <flow id="8" type="CarA" from="gneE5" to="gneE6" begin="0" end="3600" number="300"/>
  <flow id="9" type="CarA" from="-gneE6" to="-gneE5" begin="0" end="3600" number="250"/>
  <flow id="10" type="CarB" from="gneE3" to="gneE6" begin="0" end="3600" number="150"/>
</routes>
```



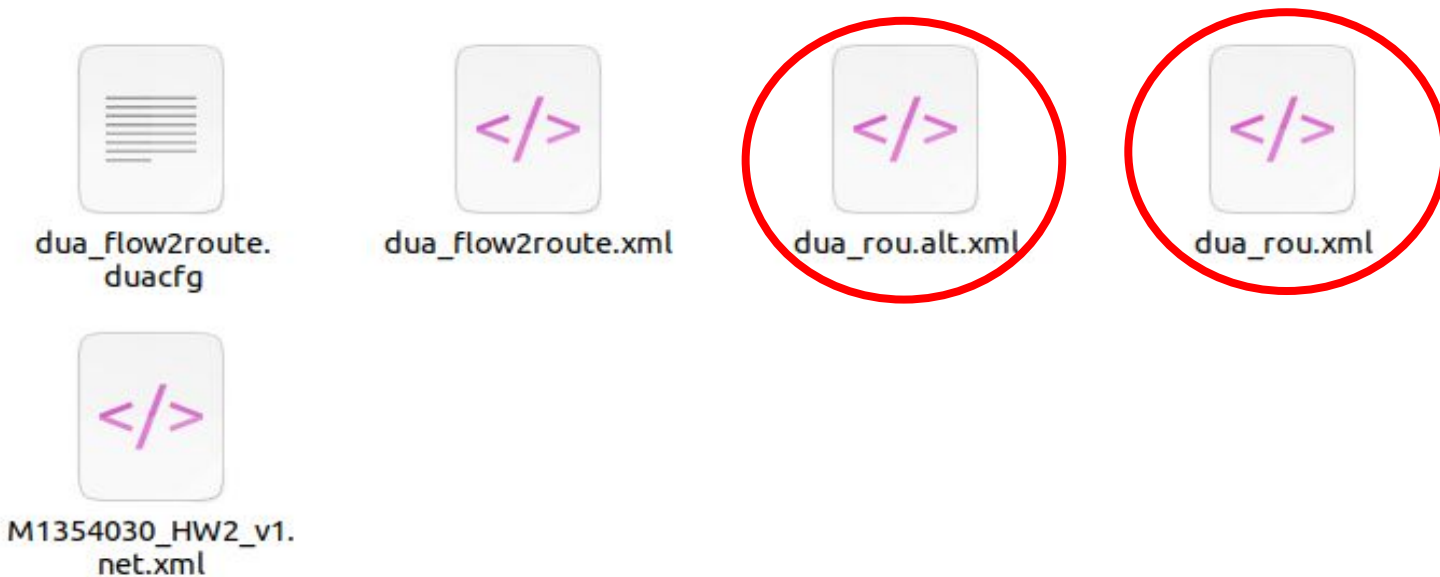
✓ 輸入交通流量

自動產生路徑

- ✓ 利用duarouter.exe, 自動尋找最短路徑。

Step3: 輸入指令 `~/desktop/sumo/bin/duarouter -c dua_flow2route.duacfg`

完成後在目錄下會產出dua.rou.xml及dua.rou.alt.xml共兩個檔案。





✓ 執行模擬

建立模擬設定檔

- ✓ 再執行sumo-gui之前，請先設定參數，格式如下，輸入net-file及route-files對應的檔案，儲存成simulate.sumocfg

```
<?xml version="1.0" encoding="UTF-8"?>  
  
<configuration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
  xsi:noNamespaceSchemaLocation="http://sumo.dlr.de/xsd/sumoConfiguration.xsd">  
  
  <input>  
    <net-file value="M1354030_HW2_v1.net.xml"/>  
    <route-files value="dua_rou.xml"/>  
  </input>  
  
</configuration>
```

- ✓ 開啟sumo-gui執行simulate.sumocfg。



✓ 執行模擬

產出交通績效

- ✓ 新增simulate2.sumocfg。

```
<?xml version="1.0" encoding="UTF-8"?>

<configuration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="http://sumo.dlr.de/xsd/sumoConfiguration.xsd">

  <input>
    <net-file value="M1354030_HW2_v1.net.xml"/>
    <route-files value="dua_rou.xml"/>
    <additional-files value="output_setting.xml"/>
  </input>

</configuration>
```

- ✓ 新增output_setting.xml。

```
<add>
  <edgeData id="01" freq="1800" file="edgedata.xml" excludeEmpty="true"/>
  <laneData id="02" freq="1800" file="lanedata.xml" excludeEmpty="true"/>
</add>
```

▲ freq：彙整頻率

▲ 模擬完後，交通績效會輸出至edgedata.xml與lanedata.xml中

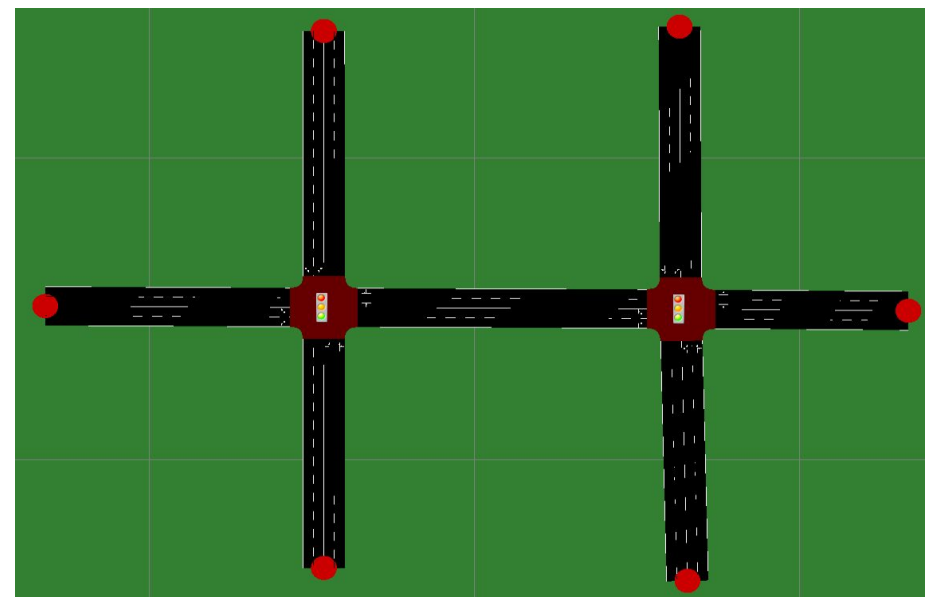
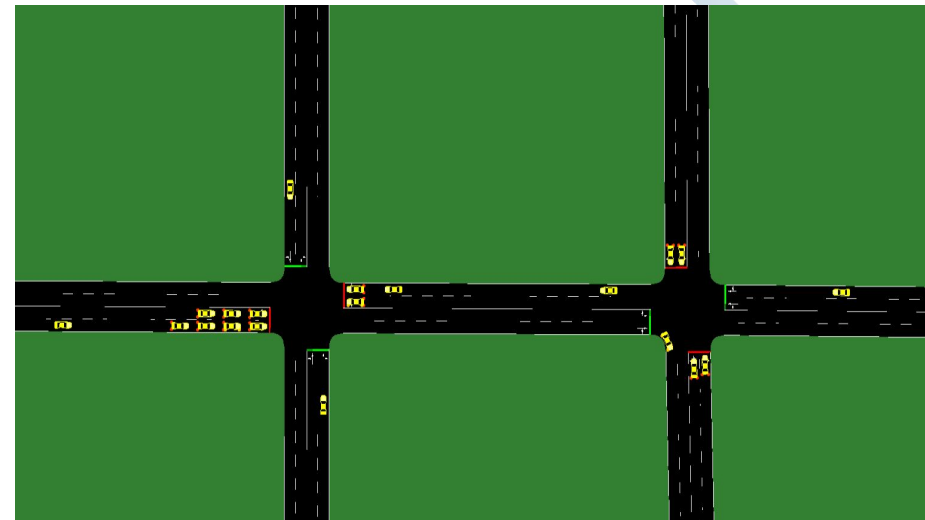
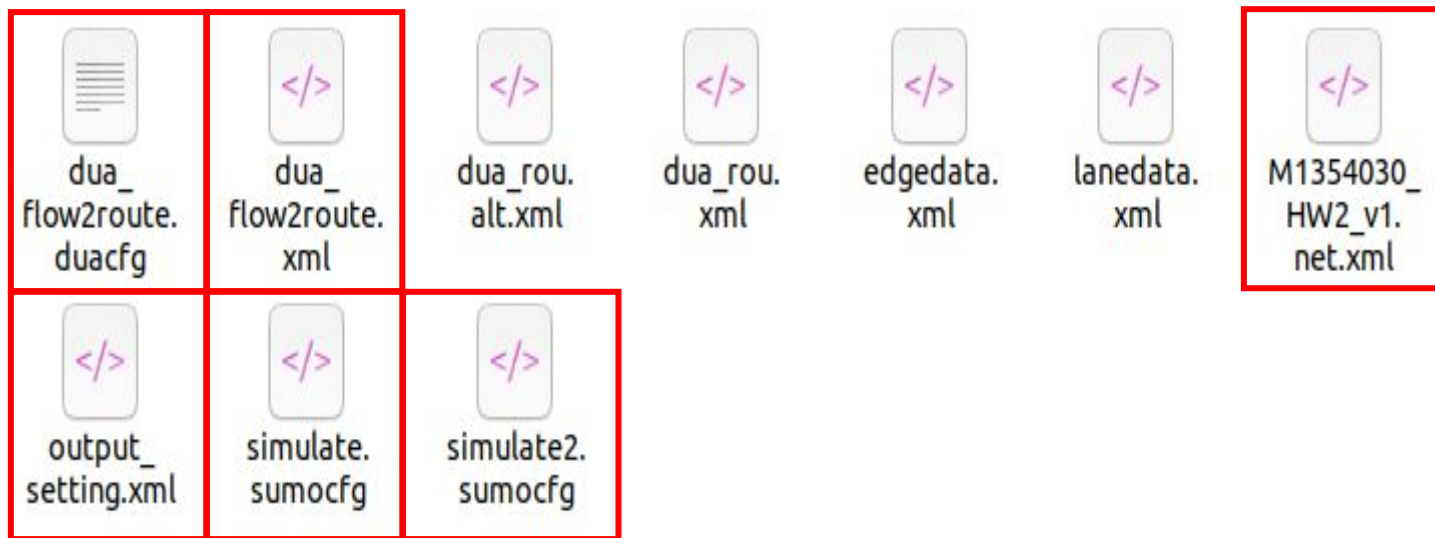


✓ 實作演練

完成 學號_HW2_v1.net.xml 路網之交通模擬，並產出績效。

▲ 將此10個檔案壓縮在 學號_HW2_v1.zip 資料夾中，和作業v2一併上傳繳交。

▲ simulate.sumocfg與simulate2.sumocfg差異為有無產出交通績效。

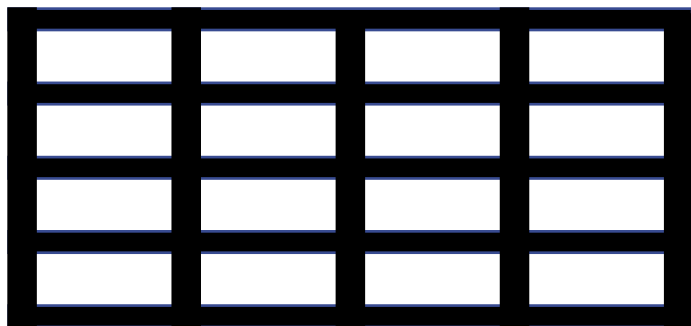




作業二

1. 將實作範例的 10個檔案壓縮為 **學號_HW2_v1.zip** 檔
2. 繪製(圖一)中的路網圖, 每一條道路皆為三車道雙線道, 且每一十字路口皆須設置紅綠燈。
3. 將(圖一)的路網圖產出交通模擬狀況及交通績效, 並於 說明文檔中, 明確寫出每一條道路的編號及其長寬, 以及每一路段的車流量 (投影片 p10的OD表) #至少兩種車
4. 將(圖二)的路網圖產出交通模擬狀況及交通績效 (共9個檔案)壓縮為 **學號_HW2_v2.zip** 檔
5. 需繳交檔案: 1. 紙本檔案、2. 電子檔案上傳 FTP、3. 檔名為 : **學號_HW2.zip** (**學號_HW2_v1.zip**、**學號_HW2_v2.zip**)

上傳 : 120.107.172.19 使用者名稱 : 1132VANET 密碼 : 1132student



圖一



圖二