In [1]:

```
import arcpy
from urllib.request import urlopen
from bs4 import BeautifulSoup
import requests
import zipfile
```

In [2]:

```
# Get the download link of streams data from MN Geo
response = urlopen("https://gisdata.mn.gov/dataset/env-impaired-streams-2012")
html = response.read()
parser = BeautifulSoup(html.decode("utf-8"), "html.parser")
target = parser.find_all('a', class_='SHP btn btn-primary dropdown-toggle resource-url-anal
for link in target:
    url=link.get('href')
```

In [3]:

```
# Download the streams data and extract to data file
r = requests.get(url, allow_redirects=True)
open('streams.zip', 'wb').write(r.content)
with zipfile.ZipFile(open('streams.zip','rb')) as f:
    f.extractall(r"D:\2021-spring\ArcGIS\Lab2\Path\Path.gdb\data\streams")
```

In [4]:

```
# Get the download link of roads data from MN Geo
response = urlopen("https://gisdata.mn.gov/dataset/trans-roads-mndot-tis")
html = response.read()
parser = BeautifulSoup(html.decode("utf-8"), "html.parser")
target = parser.find_all('a', class_='SHP btn btn-primary dropdown-toggle resource-url-anal
for link in target:
    url=link.get('href')
```

In [5]:

```
# Download the roads data and extract to data file
r = requests.get(url, allow_redirects=True)
open('roads.zip', 'wb').write(r.content)
with zipfile.ZipFile(open('roads.zip','rb')) as f:
    f.extractall(r"D:\2021-spring\ArcGIS\Lab2\Path\Path.gdb\data\roads")
```

In [6]:

```
# Get the download link of DEM data from MN Geo
response = urlopen("https://gisdata.mn.gov/dataset/elev-30m-digital-elevation-model")
html = response.read()
parser = BeautifulSoup(html.decode("utf-8"), "html.parser")
target = parser.find_all('a', class_='fgdb btn btn-primary dropdown-toggle resource-url-ana
for link in target:
    url=link.get('href')
```

In [7]:

```
# Download the DEM data and extract to data file
r = requests.get(url, allow_redirects=True)
open('dem.zip', 'wb').write(r.content)
with zipfile.ZipFile(open('dem.zip','rb')) as f:
    f.extractall(r"D:\2021-spring\ArcGIS\Lab2\Path\Path.gdb\data\dem")
```

In [8]:

```
# Get the download link of cropland data from MN Geo
response = urlopen("https://gisdata.mn.gov/dataset/agri-cropland-data-layer-2018")
html = response.read()
parser = BeautifulSoup(html.decode("utf-8"), "html.parser")
target = parser.find_all('a', class_='fgdb btn btn-primary dropdown-toggle resource-url-ana
for link in target:
    url=link.get('href')
```

In [9]:

```
# Download the cropland data and extract to data file
r = requests.get(url, allow_redirects=True)
open('cropland.zip', 'wb').write(r.content)
with zipfile.ZipFile(open('cropland.zip','rb')) as f:
    f.extractall(r"D:\2021-spring\ArcGIS\Lab2\Path\Path.gdb\data\cropland")
```

In [10]:

```
arcpy.env.workspace = r"D:\2021-spring\ArcGIS\Lab2\Path\Path.gdb"
```

In [11]:

```
# Clip the DEM
arcpy.management.Clip(r"\data\dem\elev_30m_digital_elevation_model.gdb\digital_elevation_mo
                      "563400 4872300 581650 4890600",
                      "DEM",
                      "boundary",
                      "32767", "NONE", "NO_MAINTAIN_EXTENT")
```

Out[11]:

Output

D:\2021-spring\ArcGIS\Lab2\Path\Path.gdb\DEM

Messages

Start Time: 2021年3月3日 上午 01:18:13

Building Pyramids...

Succeeded at 2021年3月3日 上午 01:18:14 (Elapsed Time: 0.75 seconds)

In [12]:

```
# Clip the stream
arcpy.analysis.Clip(r"\data\streams\env_impaired_streams_2012.shp",
                    "boundary",
                    "clip_stream", None)
```

Out[12]:

Output

D:\2021-spring\ArcGIS\Lab2\Path\Path.gdb\clip_stream

Messages

Start Time: 2021年3月3日 上午 01:18:16

Analyzing input features...

Dissolving clip features...

Clipping input features...

Succeeded at 2021年3月3日 上午 01:18:16 (Elapsed Time: 0.35 seconds)

In [13]:

```
# Clip the road
arcpy.analysis.Clip(r"\data\roads\STREETS_LOAD.shp",
                    "boundary",
                    "clip_road", None)
```

Out[13]:

Output

D:\2021-spring\ArcGIS\Lab2\Path\Path.gdb\clip_road

Messages

Start Time: 2021年3月3日 上午 01:18:17

Assembling Features...

Reading Features...

Cracking Features...

Succeeded at 2021年3月3日 上午 01:18:17 (Elapsed Time: 0.52 seconds)

In [14]:

```
# Buffer the road and erase the stream since there are bridges
arcpy.analysis.Buffer("clip_road", "clip_road_Buffer", "50 Meters", "FULL", "ROUND", "NONE"
arcpy.analysis.Erase("clip_stream", "clip_road_Buffer", "stream", None)
```

Out[14]:

Output

D:\2021-spring\ArcGIS\Lab2\Path\Path.gdb\stream

Messages

Start Time: 2021年3月3日 上午 01:18:19

Reading Features...

Cracking Features...

Assembling Features...

Succeeded at 2021年3月3日 上午 01:18:21 (Elapsed Time: 1.56 seconds)

In [15]:

```
# Calculate the values depends on the distance with the roads
accumulation_raster = arcpy.sa.DistanceAccumulation("clip_road_Buffer")
accumulation_raster.save("Distanc_road")
```

In [16]:

```
# Rescale the valuse to 1 to 10
out_raster = arcpy.sa.RescaleByFunction("Distanc_road",
                                        "MSSMALL 0.25 0.25 0 # 1997 #", 10, 1)
out_raster.save("Rescale_road")
```

In [17]:

```
# Calculate the values depends on the slope of the surface
out_raster = arcpy.sa.Slope("DEM", "DEGREE", 1, "PLANAR", "METER")
out_raster.save("Slope_DEM")
```

In [18]:

```
# Rescale the valuse to 1 to 10
out_raster = arcpy.sa.RescaleByFunction("Slope_DEM",
                                        "MSSMALL 1 1 0 # 75 #", 10, 1)
out_raster.save("Rescale_Slop")
```

In [19]:

```
# Clip the cropland
arcpy.management.Clip(r"\data\cropland\agri_cropland_data_layer_2018.gdb\agri_cropland_data
                      "563400 4872300 581650 4890600",
                      "clip_cropland",
                      "boundary",
                      "255", "NONE", "NO_MAINTAIN_EXTENT")
```

Out[19]:

Output

D:\2021-spring\ArcGIS\Lab2\Path\Path.gdb\clip_cropland

Messages

Start Time: 2021年3月3日 上午 01:18:30

Building Pyramids...

Succeeded at 2021年3月3日 上午 01:18:30 (Elapsed Time: 0.92 seconds)

In [20]:

```
# Reclassify the cropland
out_raster = arcpy.sa.Reclassify("clip_cropland", "VALUE",
                                 "1 5;5 5;21 5;23 5;27 5;28 5;36 5;37 5;53 5;59 5;68 5;71 5
                                 "DATA")
out_raster.save("Reclass_cropland")
```

In [21]:

```
# Sum up the road, slope, and cropland
out_raster = arcpy.sa.WeightedSum("Reclass_cropland Value 1;Rescale_road VALUE 1;Rescale_Sl
out_raster.save("Weighted_sum")
```

In [22]:

```
# Get the optimal path
arcpy.sa.OptimalRegionConnections("destination",
                                  "Optimal_path",
                                  "stream",
                                  "Weighted_sum",
                                  None, "PLANAR", "GENERATE_CONNECTIONS")
```

Out[22]:

```
<geoprocessing server result object at 0x1c264cea2a0>
```

In [ ]: