Connect to MN Geo FTP

In [48]:
```
from ftplib import FTP
ftp = FTP('ftp.lmic.state.mn.us')  # connect to host, default port
ftp.login()                        # login anonymous
'230 Login successful.'
```

Out[48]:
```
'230 Login successful.'
```

View & select files in FTP directory

In [49]:
```
# list the files in the ftp folder for Winona county DEM
ftp.cwd('/pub/data/elevation//lidar/county/winona/')   # change to "winona" directory
ftp.retrlines('LIST')              # list directory contents
```

```
-rw-rw-r--    1 12102    5070        43935 Mar 10  2012 Minnesota Dep
artment of Natural Resources General Geographic Data License Agreemen
t.rtf
-rw-rw-r--    1 12102    5070        15360 Mar 10  2012 Thumbs.db
-rw-rw-r--    1 12102    5070       773981 Mar 10  2012 Winona_county
_validation_report.pdf
-rw-rw-r--    1 12102    5070      1037759 Mar 10  2012 county_mosaic
_LiDAR_Data_README.rtf
-rw-rw-r--    1 12102    5070     11005858514 Jul 11  2014 elevation_
data.gdb.zip
drwxrwxr-x    2 12110    5070        12288 Jun 25  2014 geodatabase
drwxrwxr-x    2 12102    5070        12288 Jun 11  2014 laz
-rwxr--r--    1 12102    5070        30993 Apr 28  2014 lidar_semn200
8.html
-rwxr--r--    1 12102    5070        17423 Apr 28  2014 lidar_semn200
8.xml
-rw-rw-r--    1 12102    5070        54265 Mar 12  2012 raw_LiDAR_Dat
a_README.rtf
-rw-rw-r--    1 12102    5070      3849941 Mar 12  2012 tile_index_ma
p.pdf
-rw-r--r--    1 12102    5070         2453 May 22  2015 winona_tile_l
ist.txt
```

Out[49]:
```
'226 Directory send OK.'
```

In [50]:

```
# list the files in the ftp folder for Wabasha county DEM
ftp.cwd('/pub/data/elevation//lidar/county/wabasha/')
ftp.retrlines('LIST')   # list directory contents
```

```
-rw-rw-r--    1 12102    5070         43935 Mar 10  2012 Minnesota Dep
artment of Natural Resources General Geographic Data License Agreemen
t.rtf
-rw-rw-r--    1 12102    5070        509694 Mar 10  2012 Wabasha_count
y_validation_report.pdf
-rw-rw-r--    1 12102    5070       1037759 Mar 10  2012 county_mosaic
_LiDAR_Data_README.rtf
-rw-rw-r--    1 12102    5070      7615268596 Jul 11  2014 elevation_d
ata.gdb.zip
drwxrwxr-x    2 12110    5070         12288 Jun 25  2014 geodatabase
drwxrwxr-x    2 12102    5070         12288 Jun 11  2014 laz
-rwxr--r--    1 12102    5070         30993 Apr 28  2014 lidar_semn200
8.html
-rwxr--r--    1 12102    5070         17423 Apr 28  2014 lidar_semn200
8.xml
-rw-rw-r--    1 12102    5070         54265 Mar 12  2012 raw_LiDAR_Dat
a_README.rtf
-rw-rw-r--    1 12102    5070       3471740 Mar 12  2012 tile_index_ma
p.pdf
-rw-r--r--    1 12102    5070          2134 May 22  2015 wabasha_tile_
list.txt
```

Out[50]: '226 Directory send OK.'

In [51]:

```
# list the files in the ftp folder for MN cropland
ftp.cwd('/pub/gdrs/data/pub/us_mn_state_mda/agri_cropland_data_layer_20
18/')                # change into "debian" directory
ftp.retrlines('LIST')            # list directory contents
```

```
-rw-r--r--    1 12110    5070          1080 Aug 29  2019 dataResource.
xml
-rw-r--r--    1 12110    5070      69374754 Sep 05  2019 fgdb_agri_cro
pland_data_layer_2018.zip
drwxr-xr-x    2 12110    5070          4096 Aug 29  2019 metadata
drwxr-xr-x    2 12110    5070          4096 Sep 05  2019 shp
```

Out[51]: '226 Directory send OK.'

Download & extract layer zip files for Winona & Wabasha DEMs, & Cropland shp

In [22]:

```
with open('elevation_data.gdb.zip', 'wb') as fp:  # Download the files
in binary
    ftp.retrbinary('RETR elevation_data.gdb.zip', fp.write)

# Unzip file Winona county DEM
from zipfile import ZipFile
with ZipFile('elevation_data.gdb.zip', 'r') as zipObj:
    zipObj.extractall('data')  # Save to data folder in Pro project
```

In [34]:

```
with open('elevation_data.gdb.zip', 'wb') as fp:  # Download the files in binary
    ftp.retrbinary('RETR elevation_data.gdb.zip', fp.write)

# Unzip Wabasha county zip DEM
from zipfile import ZipFile
with ZipFile('elevation_data.gdb.zip', 'r') as zipObj:
    zipObj.extractall('data\wabasha') # Save to Pro project folder
```

In [ ]:

```
with open('fgdb_agri_cropland_data_layer_2018.zip', 'wb') as fp:  # Download the files in binary
    ftp.retrbinary('RETR fgdb_agri_cropland_data_layer_2018.zip', fp.write)

# Unzip croplands zip
from zipfile import ZipFile
with ZipFile('fgdb_agri_cropland_data_layer_2018.zip', 'r') as zipObj:
    zipObj.extractall('data')
```

Create & Reclass Slope for Area of Interest (AOI)

In [ ]:

```
# Create a new raster for both county DEMs
arcpy.management.MosaicToNewRaster("winona_Slope_dem_3m1;Wabasha_Slope_dem_3m1", r"C:\Users\mmMary\Documents\GIS_Classes\GIS_5572\Labs\Lab2_Part2\5572_lab2_part2", "AOI_DEM", None, "8_BIT_UNSIGNED", None, 1, "FIRST", "FIRST")

# Generate Slope layer from AOI DEM
out_raster = arcpy.sa.Slope("AOI_DEM", "DEGREE", 0.304800609601219, "PLANAR", "METER"); out_raster.save(r"C:\Users\mmMary\Documents\GIS_Classes\GIS_5572\Labs\Lab2_Part2\5572_lab2_part2\5572_lab2_part2.gdb\AOI_Slope")

# Reclassify to disqualify steeper slopes
out_raster = arcpy.sa.Reclassify("AOI_Slope", "VALUE", "0 1.720000 1;1.720000 3.430000 1;3.430000 5.710000 4;5.710000 8.530000 4;8.530000 11.300000 6;11.300000 14.040000 6;14.040000 16.700000 10;16.700000 21.800000 10;21.800000 30.960000 10;30.960000 45 10;45 90 11", "DATA"); out_raster.save(r"C:\Users\mmMary\Documents\GIS_Classes\GIS_5572\Labs\Lab2_Part2\5572_lab2_part2\5572_lab2_part2.gdb\AOI_Slope_Reclass")
```

Create Stream Barrier Layer

- Buffer roads layer by a full pixel size - 150 meters - to ensure no mixed pixels will be considered in the cost surface.
- Erase buffered roads with streams to eliminate any potential crossings unless there is a bridge over the stream.

In [ ]:

```
# Buffer roads
arcpy.analysis.Buffer("Roads_AOI", r"C:\Users\mmMary\Documents\GIS_Classes\GIS_5572\Labs\Lab2_Part2\5572_lab2_part2\5572_lab2_part2.gdb\Roads_AOI_Buffer_150meter", "150 Meters", "FULL", "ROUND", "NONE", None, "PLANAR")

# Erase stream/roads intersections
arcpy.analysis.Erase("Streams_AOI", "Roads_AOI_Buffer_150meter", r"C:\Users\mmMary\Documents\GIS_Classes\GIS_5572\Labs\Lab2_Part2\5572_lab2_part2\5572_lab2_part2.gdb\stream_Rd150meter_Erase", None)

# Reclassify water features as a high cost & no water as NO DATA
stream_barrier_reclass = arcpy.sa.Reclassify("stream_Barrier_Raster", "Water", "water 10", "DATA"); out_raster.save(r"C:\Users\mmMary\Documents\GIS_Classes\GIS_5572\Labs\Lab2_Part2\5572_lab2_part2\5572_lab2_part2.gdb\stream_barrier_reclass")
```

Create Cost Surface

In [ ]:

```
# Reclassify land use - pasture and coniferous forests are acceptable paths
out_raster = arcpy.sa.Reclassify("agri_cropland_data_layer_2018", "CLASS_NAME", "Background 0;Corn 5;Sorghum 5;Soybeans 5;Sunflower 5;'Pop or Orn Corn' 5;Barley 5;'Durum Wheat' 5;'Spring Wheat' 5;'Winter Wheat' 5;'Dbl Crop WinWht/Soybeans' 5;Rye 5;Oats 5;Millet 5;Canola 5;Flaxseed 5;Mustard 5;Alfalfa 5;'Other Hay/Non Alfalfa' 5;Buckwheat 5;Sugarbeets 5;'Dry Beans' 5;Potatoes 5;'Other Crops' 5;'Misc Vegs & Fruits' 5;Peas 5;Herbs 5;Clover/Wildflowers 0;'Sod/Grass Seed' 2;Switchgrass 5;'Fallow/Idle Cropland' 0;Apples 0;'Christmas Trees' 0;'Other Tree Crops' 0;'Open Water' 5;'Developed/Open Space' 0;'Developed/Low Intensity' 0;'Developed/Med Intensity' 0;'Developed/High Intensity' 0;Barren 0;'Deciduous Forest' 1;'Evergreen Forest' 1;'Mixed Forest' 1;Shrubland 0;Grass/Pasture 0;'Woody Wetlands' 7;'Herbaceous Wetlands' 7;Triticale 7;Strawberries 5;Squash 5;Pumpkins 2;'Dbl Crop Corn/Soybeans' 5;Radishes 5", "DATA"); out_raster.save(r"C:\Users\mmMary\Documents\GIS_Classes\GIS_5572\Labs\Lab2_Part2\5572_lab2_part2\5572_lab2_part2.gdb\Ag_Reclass")
```

In [ ]:

```
# Use Weighted Overlay Tool to scale all inputs equally.
out_raster = arcpy.sa.WeightedOverlay("('AOI_Slope_Reclass' 33 'Value' (1 1; 4 4; 6 6; 10 1; 11 1; NODATA NODATA); 'Reclass_Barrier_Stream' 33 'Value' (1 1; 10 Restricted; NODATA NODATA); 'Ag_Reclass' 34 'Value' (0 1; 1 1; 2 2; 5 5; 7 7; NODATA NODATA));1 9 1"); out_raster.save(r"C:\Users\mmMary\Documents\GIS_Classes\GIS_5572\Labs\Lab2_Part2\5572_lab2_part2\5572_lab2_part2.gdb\Weight_Over")
```

Create Origin & Destination Points

In [ ]:

```
arcpy.management.XYTableToPoint("Sheet1$", r"C:\Users\mmMary\Documents\GIS_Classes\GIS_5572\Labs\Lab2_Part2\5572_lab2_part2\5572_lab2_part2.gdb\Origin", "long", "lat", None, "GEOGCS['GCS_WGS_1984',DATUM['D_WGS_1984',SPHEROID['WGS_1984',6378137.0,298.257223563]],PRIMEM['Greenwich',0.0],UNIT['Degree',0.0174532925199433]];-400 -400 1000000000;-100000 10000;-100000 10000;8.98315284119521E-09;0.001;0.001;IsHighPrecision")
arcpy.management.XYTableToPoint("Sheet2$", r"C:\Users\mmMary\Documents\GIS_Classes\GIS_5572\Labs\Lab2_Part2\5572_lab2_part2\5572_lab2_part2.gdb\Destination", "long", "lat", None, "GEOGCS['GCS_WGS_1984',DATUM['D_WGS_1984',SPHEROID['WGS_1984',6378137.0,298.257223563]],PRIMEM['Greenwich',0.0],UNIT['Degree',0.0174532925199433]];-400 -400 1000000000;-100000 10000;-100000 10000;8.98315284119521E-09;0.001;0.001;IsHighPrecision")
```

Determine Least Cost Path

- Create Cost Distance
- Create Cost Back Link

In [ ]:

```
# Create a Cost Distance
out_distance_raster = arcpy.sa.CostDistance("Origin", "Weight_Over", None, r"C:\Users\mmMary\Documents\GIS_Classes\GIS_5572\Labs\Lab2_Part2\5572_lab2_part2\5572_lab2_part2.gdb\Cost_Backlink_Orig", None, None, None, None, ''); out_distance_raster.save(r"C:\Users\mmMary\Documents\GIS_Classes\GIS_5572\Labs\Lab2_Part2\5572_lab2_part2\5572_lab2_part2.gdb\Cost_Dist_Orig")
```

In [ ]:

```
# Create a Cost Back Link
out_backlink_raster = arcpy.sa.CostBackLink("Destination", "Weight_Over", None, None, None, None, None, None, ''); out_backlink_raster.save(r"C:\Users\mmMary\Documents\GIS_Classes\GIS_5572\Labs\Lab2_Part2\5572_lab2_part2\5572_lab2_part2.gdb\Cost_Backlink_Dest")
```

In [ ]:

```
# Create the Lost Cost Path
out_raster = arcpy.sa.CostPath("Origin", "Cost_Dist_Orig", "Cost_Backlink_Dest", "EACH_CELL", "OBJECTID", "INPUT_RANGE"); out_raster.save(r"C:\Users\mmMary\Documents\GIS_Classes\GIS_5572\Labs\Lab2_Part2\5572_lab2_part2\5572_lab2_part2.gdb\Cost_Path_3")
```