```
{
 "cells": [
  {
   "cell_type": "markdown",
   "metadata": {},
   "source": [
    "# Cost Surface Analysis Operations"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 4,
   "metadata": {},
   "outputs": [],
   "source": [
    "import arcpy\n",
    "import requests\n",
    "import io\n",
    "import zipfile"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {},
   "source": [
    "## Display Dory's Farm and North Picnic Area in ArcGIS Pro from CSV Table to XY to Point Tool"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 2,
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/html": [
       "<div class='gpresult'><h2>Messages</h2><div id='messages' data-messages='[\"Start Time: Friday, October 28, 2022 4:39:44 PM\",\"Succeeded at Friday, October 28, 2022 4:39:45 PM (Elapsed Time: 1.00 seconds)\"]' data-show='true'><div id = 'default' /></div></div>"
      ],
      "text/plain": [
       "<Result 'C:\\\\Users\\\\Alexander Danielson\\\\Desktop\\\\Fall 2022Spring2023\\\\ArcGIS I\\\\Lab2\\\\Lab2\\\\Lab2.gdb\\\\DisplayXYPathDoryFarmtoNPA'>"
      ]
     },
     "execution_count": 2,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "arcpy.management.XYTableToPoint(r\"C:\\Users\\Alexander Danielson\\Desktop\\Fall 2022Spring2023\\ArcGIS I\\Lab2\\Lab2\\XYTable.csv\", r\"C:\\Users\\Alexander Danielson\\Desktop\\Fall 2022Spring2023\\ArcGIS I\\Lab2\\Lab2\\Lab2.gdb\\DisplayXYPathDoryFarmtoNPA\", \"Y\", \"X\", 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')"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {},
   "source": [
    "## Form Optimal Route For Dory's Given Extent of Area Using Data From MNGEO and
```

```
        Performing Raster Analysis"
       ]
      },
      {
       "cell_type": "markdown",
       "metadata": {},
       "source": [
        "## Extract County Data from MNGEO (including MN NLCD, DEM, Streams, Impervious
        Surfuces/Roads) to Create Optimal Route"
       ]
      },
      {
       "cell_type": "code",
       "execution_count": 41,
       "metadata": {},
       "outputs": [],
       "source": [
        "CountyBoundsLink =
        \"https://resources.gisdata.mn.gov/pub/gdrs/data/pub/us_mn_state_dnr/bdry_counties_in
        _minnesota/shp_bdry_counties_in_minnesota.zip\""
       ]
      },
      {
       "cell_type": "code",
       "execution_count": 42,
       "metadata": {},
       "outputs": [
        {
         "data": {
          "text/plain": [

           "'https://resources.gisdata.mn.gov/pub/gdrs/data/pub/us_mn_state_dnr/bdry_counties
           _in_minnesota/shp_bdry_counties_in_minnesota.zip'"
          ]
         },
         "execution_count": 42,
         "metadata": {},
         "output_type": "execute_result"
        }
       ],
       "source": [
        "CountyBoundsLink"
       ]
      },
      {
       "cell_type": "code",
       "execution_count": 43,
       "metadata": {},
       "outputs": [],
       "source": [
        "OutputSource = requests.post(CountyBoundsLink)"
       ]
      },
      {
       "cell_type": "code",
       "execution_count": 44,
       "metadata": {},
       "outputs": [],
       "source": [
        "Sources = OutputSource.content"
       ]
      },
      {
       "cell_type": "code",
       "execution_count": 45,
       "metadata": {},
       "outputs": [],
       "source": [
        "Zipp = zipfile.ZipFile(io.BytesIO(Sources))"
```

```
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 21,
   "metadata": {},
   "outputs": [],
   "source": [
    "Zipp.extractall(r'C:\\Users\\Alexander Danielson\\Desktop\\Fall
    2022Spring2023\\ArcGIS I\\Lab2\\Lab2')"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 69,
   "metadata": {},
   "outputs": [],
   "source": [
    "CountyNLCDLink =
    \"https://resources.gisdata.mn.gov/pub/gdrs/data/pub/us_mn_state_dnr/biota_landcover_
    nlcd_mn_2019/tif_biota_landcover_nlcd_mn_2019.zip\""
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 70,
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/plain": [
       "'https://resources.gisdata.mn.gov/pub/gdrs/data/pub/us_mn_state_dnr/biota_landcov
       er_nlcd_mn_2019/tif_biota_landcover_nlcd_mn_2019.zip'"
      ]
     },
     "execution_count": 70,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "CountyNLCDLink"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 65,
   "metadata": {},
   "outputs": [],
   "source": [
    "OutputNLCDSource = requests.post(CountyNLCDLink)"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 66,
   "metadata": {},
   "outputs": [],
   "source": [
    "SourcesNLCD = OutputNLCDSource.content"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 67,
   "metadata": {},
   "outputs": [],
   "source": [
```

```
      "ZippNLCD = zipfile.ZipFile(io.BytesIO(SourcesNLCD))"
     ]
    },
    {
     "cell_type": "code",
     "execution_count": 68,
     "metadata": {},
     "outputs": [],
     "source": [
      "ZippNLCD.extractall(r'C:\\Users\\Alexander Danielson\\Desktop\\Fall
      2022Spring2023\\ArcGIS I\\Lab2\\Lab2')"
     ]
    },
    {
     "cell_type": "code",
     "execution_count": 71,
     "metadata": {},
     "outputs": [],
     "source": [
      "CountyDEMLink =
      'https://resources.gisdata.mn.gov/pub/gdrs/data/pub/us_mn_state_dnr/elev_30m_digital_
      elevation_model/fgdb_elev_30m_digital_elevation_model.zip'"
     ]
    },
    {
     "cell_type": "code",
     "execution_count": 73,
     "metadata": {},
     "outputs": [
      {
       "data": {
        "text/plain": [

         "'https://resources.gisdata.mn.gov/pub/gdrs/data/pub/us_mn_state_dnr/elev_30m_digi
         tal_elevation_model/fgdb_elev_30m_digital_elevation_model.zip'"
        ]
       },
       "execution_count": 73,
       "metadata": {},
       "output_type": "execute_result"
      }
     ],
     "source": [
      "CountyDEMLink "
     ]
    },
    {
     "cell_type": "code",
     "execution_count": 74,
     "metadata": {},
     "outputs": [],
     "source": [
      "OutputDEMSource = requests.post(CountyDEMLink)"
     ]
    },
    {
     "cell_type": "code",
     "execution_count": 75,
     "metadata": {},
     "outputs": [],
     "source": [
      "SourcesDEM = OutputDEMSource.content"
     ]
    },
    {
     "cell_type": "code",
     "execution_count": 76,
     "metadata": {},
     "outputs": [],
```

```
   "source": [
    "ZippDEM = zipfile.ZipFile(io.BytesIO(SourcesDEM))"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 77,
   "metadata": {},
   "outputs": [],
   "source": [
    "ZippDEM.extractall(r'C:\\Users\\Alexander Danielson\\Desktop\\Fall 2022Spring2023\\ArcGIS I\\Lab2\\Lab2')"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 83,
   "metadata": {},
   "outputs": [],
   "source": [
    "CountyStreamLink = 'https://resources.gisdata.mn.gov/pub/gdrs/data/pub/us_mn_state_dnr/water_strahler_stream_order/shp_water_strahler_stream_order.zip'"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 84,
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/plain": [
       "'https://resources.gisdata.mn.gov/pub/gdrs/data/pub/us_mn_state_dnr/water_strahler_stream_order/shp_water_strahler_stream_order.zip'"
      ]
     },
     "execution_count": 84,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "CountyStreamLink"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 90,
   "metadata": {},
   "outputs": [],
   "source": [
    "OutputStreamSource = requests.post(CountyStreamLink)"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 91,
   "metadata": {},
   "outputs": [],
   "source": [
    "SourcesStream = OutputStreamSource.content"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 92,
   "metadata": {},
```

```
   "outputs": [],
   "source": [
    "ZippStream = zipfile.ZipFile(io.BytesIO(SourcesStream))"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 93,
   "metadata": {},
   "outputs": [],
   "source": [
    "ZippStream.extractall(r'C:\\Users\\Alexander Danielson\\Desktop\\Fall
    2022Spring2023\\ArcGIS I\\Lab2\\Lab2')"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {},
   "source": [
    "## Dissolve Optimal Counties and Geoprocess Other Ancillary Data Then Transform
    Features to Reclassify for Weights "
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {},
   "source": [
    "### Dissolve Operation for Optimal Study Area"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 94,
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/html": [
       "<div class='gpresult'><h2>Messages</h2><div id='messages'
       data-messages='[\"Start Time: Friday, October 28, 2022 6:54:28
       PM\",\"Dissolving...\",\"Succeeded at Friday, October 28, 2022 6:54:29 PM
       (Elapsed Time: 0.54 seconds)\"]' data-show='true'><div id = 'default'
       /></div></div>"
      ],
      "text/plain": [
       "<Result 'C:\\\\Users\\\\Alexander Danielson\\\\Desktop\\\\Fall
       2022Spring2023\\\\ArcGIS
       I\\\\Lab2\\\\Lab2\\\\Lab2.gdb\\\\OptimalAreaCounties_Dissolve'>"
      ]
     },
     "execution_count": 94,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "arcpy.management.Dissolve(\"OptimalAreaCounties\", r\"C:\\Users\\Alexander
    Danielson\\Desktop\\Fall 2022Spring2023\\ArcGIS
    I\\Lab2\\Lab2\\Lab2.gdb\\OptimalAreaCounties_Dissolve\", None, None, \"MULTI_PART\",
    \"DISSOLVE_LINES\", '')"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {},
   "source": [
    "### Extract by Mask (NLCD) using Optimal Study Area"
   ]
  },
  {
```

```
    {
     "cell_type": "code",
     "execution_count": 96,
     "metadata": {},
     "outputs": [],
     "source": [
      "out_raster = arcpy.sa.ExtractByMask(r\"C:\\Users\\Alexander
      Danielson\\Desktop\\Fall 2022Spring2023\\ArcGIS I\\Lab2\\Lab2\\NLCD 2019 - Land
      Cover.lyr\", \"OptimalAreaCounties_Dissolve\", \"INSIDE\", '524966.6376 4853462.8394
      637916.1448 4922619.9426
      PROJCS[\"NAD_1983_UTM_Zone_15N\",GEOGCS[\"GCS_North_American_1983\",DATUM[\"D_North_A
      merican_1983\",SPHEROID[\"GRS_1980\",6378137.0,298.257222101]],PRIMEM[\"Greenwich\",0
      .0],UNIT[\"Degree\",0.0174532925199433]],PROJECTION[\"Transverse_Mercator\"],PARAMETE
      R[\"False_Easting\",500000.0],PARAMETER[\"False_Northing\",0.0],PARAMETER[\"Central_M
      eridian\",-93.0],PARAMETER[\"Scale_Factor\",0.9996],PARAMETER[\"Latitude_Of_Origin\",
      0.0],UNIT[\"Meter\",1.0]]'); out_raster.save(r\"C:\\Users\\Alexander
      Danielson\\Desktop\\Fall 2022Spring2023\\ArcGIS
      I\\Lab2\\Lab2\\Lab2.gdb\\NLCDExtraction\")"
     ]
    },
    {
     "cell_type": "markdown",
     "metadata": {},
     "source": [
      "### Extract by Mask (NLCD Roads) using Optimal Study Area"
     ]
    },
    {
     "cell_type": "code",
     "execution_count": 97,
     "metadata": {},
     "outputs": [],
     "source": [
      "out_raster = arcpy.sa.ExtractByMask(r\"C:\\Users\\Alexander
      Danielson\\Desktop\\Fall 2022Spring2023\\ArcGIS
      I\\Lab2\\Lab2\\NLCD_2019_Land_Cover_Impervious.tif\",
      \"OptimalAreaCounties_Dissolve\", \"INSIDE\", '524966.6376 4853462.8394 637916.1448
      4922619.9426
      PROJCS[\"NAD_1983_UTM_Zone_15N\",GEOGCS[\"GCS_North_American_1983\",DATUM[\"D_North_A
      merican_1983\",SPHEROID[\"GRS_1980\",6378137.0,298.257222101]],PRIMEM[\"Greenwich\",0
      .0],UNIT[\"Degree\",0.0174532925199433]],PROJECTION[\"Transverse_Mercator\"],PARAMETE
      R[\"False_Easting\",500000.0],PARAMETER[\"False_Northing\",0.0],PARAMETER[\"Central_M
      eridian\",-93.0],PARAMETER[\"Scale_Factor\",0.9996],PARAMETER[\"Latitude_Of_Origin\",
      0.0],UNIT[\"Meter\",1.0]]'); out_raster.save(r\"C:\\Users\\Alexander
      Danielson\\Desktop\\Fall 2022Spring2023\\ArcGIS
      I\\Lab2\\Lab2\\Lab2.gdb\\NLCDExtractionRoads\")"
     ]
    },
    {
     "cell_type": "markdown",
     "metadata": {},
     "source": [
      "### Extract by Mask (DEM) using Optimal Study Area"
     ]
    },
    {
     "cell_type": "code",
     "execution_count": 98,
     "metadata": {},
     "outputs": [],
     "source": [
      "out_raster = arcpy.sa.ExtractByMask(r\"C:\\Users\\Alexander
      Danielson\\Desktop\\Fall 2022Spring2023\\ArcGIS
      I\\Lab2\\Lab2\\elev_30m_digital_elevation_model.gdb\\digital_elevation_model_30m\",
      \"OptimalAreaCounties_Dissolve\", \"INSIDE\", '524966.6376 4853462.8394 637916.1448
      4922619.9426
      PROJCS[\"NAD_1983_UTM_Zone_15N\",GEOGCS[\"GCS_North_American_1983\",DATUM[\"D_North_A
      merican_1983\",SPHEROID[\"GRS_1980\",6378137.0,298.257222101]],PRIMEM[\"Greenwich\",0
      .0],UNIT[\"Degree\",0.0174532925199433]],PROJECTION[\"Transverse_Mercator\"],PARAMETE
```

```
R[\"False_Easting\",500000.0],PARAMETER[\"False_Northing\",0.0],PARAMETER[\"Central_Meridian\",-93.0],PARAMETER[\"Scale_Factor\",0.9996],PARAMETER[\"Latitude_Of_Origin\",0.0],UNIT[\"Meter\",1.0]]'); out_raster.save(r\"C:\\Users\\Alexander Danielson\\Desktop\\Fall 2022Spring2023\\ArcGIS I\\Lab2\\Lab2\\DEMExtraction\")"
```


```
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {},
   "source": [
    "### Clip (Streams) using Optimal Study Area"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 99,
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/html": [
```


```
       "<div class='gpresult'><h2>Messages</h2><div id='messages' data-messages='[\"Start Time: Friday, October 28, 2022 7:14:27 PM\",\"Analyzing input features...\",\"Dissolving clip features...\",\"Clipping input features...\",\"Succeeded at Friday, October 28, 2022 7:14:27 PM (Elapsed Time: 0.72 seconds)\"]' data-show='true'><div id = 'default' /></div></div>"
```


```
      ],
      "text/plain": [
```


```
       "<Result 'C:\\\\Users\\\\Alexander Danielson\\\\Desktop\\\\Fall 2022Spring2023\\\\ArcGIS I\\\\Lab2\\\\Lab2\\\\Lab2.gdb\\\\ClipedStreams'>"
```


```
      ]
     },
     "execution_count": 99,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
```


```
    "arcpy.analysis.Clip(r\"C:\\Users\\Alexander Danielson\\Desktop\\Fall 2022Spring2023\\ArcGIS I\\Lab2\\Lab2\\streams_with_strahler_stream_order.shp\", \"OptimalAreaCounties_Dissolve\", r\"C:\\Users\\Alexander Danielson\\Desktop\\Fall 2022Spring2023\\ArcGIS I\\Lab2\\Lab2\\Lab2.gdb\\ClipedStreams\", None)"
```


```
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {},
   "source": [
    "### Create Slope out of DEM "
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 100,
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/html": [
```


```
       "<div class='gpresult'><h2>Messages</h2><div id='messages' data-messages='[\"Start Time: Friday, October 28, 2022 7:20:19 PM\",\"Succeeded at Friday, October 28, 2022 7:20:21 PM (Elapsed Time: 2.10 seconds)\"]' data-show='true'><div id = 'default' /></div></div>"
```


```
      ],
      "text/plain": [
```


```
       "<Result 'C:\\\\Users\\\\Alexander Danielson\\\\Desktop\\\\Fall 2022Spring2023\\\\ArcGIS I\\\\Lab2\\\\Lab2\\\\Lab2.gdb\\\\Slope_DEMExt1'>"
```


```
      ]
     },
```

```
       "execution_count": 100,
       "metadata": {},
       "output_type": "execute_result"
      }
     ],
     "source": [
      "arcpy.ddd.Slope(\"DEMExtraction\", r\"C:\\Users\\Alexander Danielson\\Desktop\\Fall 2022Spring2023\\ArcGIS I\\Lab2\\Lab2\\Lab2.gdb\\Slope_DEMExt1\", \"PERCENT_RISE\", 1, \"PLANAR\", \"METER\")"
     ]
    },
    {
     "cell_type": "markdown",
     "metadata": {},
     "source": [
      "### Convert Feature to Raster (From Clipped Streams)"
     ]
    },
    {
     "cell_type": "code",
     "execution_count": 101,
     "metadata": {},
     "outputs": [
      {
       "data": {
        "text/html": [
         "<div class='gpresult'><h2>Messages</h2><div id='messages' data-messages='[\"Start Time: Friday, October 28, 2022 7:25:12 PM\",\"Succeeded at Friday, October 28, 2022 7:25:14 PM (Elapsed Time: 1.59 seconds)\"]' data-show='true'><div id = 'default' /></div></div>"
        ],
        "text/plain": [
         "<Result 'C:\\\\Users\\\\Alexander Danielson\\\\Desktop\\\\Fall 2022Spring2023\\\\ArcGIS I\\\\Lab2\\\\Lab2\\\\Lab2.gdb\\\\RasterizedStreams'>"
        ]
       },
       "execution_count": 101,
       "metadata": {},
       "output_type": "execute_result"
      }
     ],
     "source": [
      "arcpy.conversion.FeatureToRaster(\"ClipedStreams\", \"FW_ID\", r\"C:\\Users\\Alexander Danielson\\Desktop\\Fall 2022Spring2023\\ArcGIS I\\Lab2\\Lab2\\Lab2.gdb\\RasterizedStreams\", 276.583641600005)"
     ]
    },
    {
     "cell_type": "markdown",
     "metadata": {},
     "source": [
      "## Reclassifcation"
     ]
    },
    {
     "cell_type": "markdown",
     "metadata": {},
     "source": [
      "### Relcassify NLCD for Roads/Bridges"
     ]
    },
    {
     "cell_type": "code",
     "execution_count": null,
     "metadata": {},
     "outputs": [],
     "source": [
      "out_raster = arcpy.sa.Reclassify(\"NLCDExtraction\", \"Value\", \"11 1;11 22 2;22 24 3;24 31 4;31 41 5;41 43 6;43 52 7;52 71 8;71 82 9;82 95 10\", \"DATA\");
```

```
        out_raster.save(r\"C:\\Users\\Alexander Danielson\\Desktop\\Fall
        2022Spring2023\\ArcGIS I\\Lab2\\Lab2\\Lab2.gdb\\ReClassNLCD\")"
       ]
      },
      {
       "cell_type": "markdown",
       "metadata": {},
       "source": [
        "### Relcassify NLCD for Land Use"
       ]
      },
      {
       "cell_type": "code",
       "execution_count": 102,
       "metadata": {},
       "outputs": [],
       "source": [
        "out_raster = arcpy.sa.Reclassify(\"NLCDExtraction\", \"Value\", \"11 1;11 22 2;22
        24 3;24 31 4;31 41 5;41 43 6;43 52 7;52 71 8;71 82 9;82 95 10\", \"DATA\");
        out_raster.save(r\"C:\\Users\\Alexander Danielson\\Desktop\\Fall
        2022Spring2023\\ArcGIS I\\Lab2\\Lab2\\Lab2.gdb\\ReClassNLCD\")"
       ]
      },
      {
       "cell_type": "markdown",
       "metadata": {},
       "source": [
        "### Relcassify Slope"
       ]
      },
      {
       "cell_type": "code",
       "execution_count": 103,
       "metadata": {},
       "outputs": [],
       "source": [
        "out_raster = arcpy.sa.Reclassify(\"Slope_DEMExt1\", \"Value\", \"11 1;11 22 2;22 24
        3;24 31 4;31 41 5;41 43 6;43 52 7;52 71 8;71 82 9;82 95 10\", \"NODATA\");
        out_raster.save(r\"C:\\Users\\Alexander Danielson\\Desktop\\Fall
        2022Spring2023\\ArcGIS I\\Lab2\\Lab2\\Lab2.gdb\\ReClassNLCDSlope\")"
       ]
      },
      {
       "cell_type": "markdown",
       "metadata": {},
       "source": [
        "### Relcassify Streams"
       ]
      },
      {
       "cell_type": "code",
       "execution_count": 104,
       "metadata": {},
       "outputs": [],
       "source": [
        "out_raster = arcpy.sa.Reclassify(\"RasterizedStreams\", \"Value\", \"0 1;103383
        2;103761 3;104589 4;104621 5;104624 6;104626 7;104687 8;104692 9;104695 10;104696
        11;104702 12;104703 13;104705 14;104706 15;104707 16;104708 17;104709 18;104710
        19;104711 20;104712 21;104714 22;104715 23;104717 24;104718 25;104719 26;104721
        27;104722 28;104724 29;104725 30;104735 31;104736 32;104737 33;104738 34;104740
        35;104741 36;104742 37;104743 38;104744 39;104745 40;104746 41;104747 42;104748
        43;104749 44;104751 45;104752 46;104753 47;104754 48;104755 49;104806 50;104807
        51;104812 52;104813 53;104814 54;104821 55;104856 56;104857 57;104858 58;104859
        59;104860 60;104861 61;104862 62;104863 63;104864 64;104865 65;104867 66;104868
        67;104869 68;104870 69;104871 70;104872 71;104873 72;104874 73;104875 74;104876
        75;104877 76;104878 77;104879 78;104880 79;104881 80;104882 81;104883 82;104884
        83;104885 84;104886 85;104887 86;104888 87;104889 88;104890 89;104891 90;104895
        91;104896 92;104897 93;104898 94;104899 95;104900 96;104901 97;104902 98;104904
        99;104906 100;104910 101;104911 102;104912 103;104914 104;104916 105;104919
```

106;104920 107;104921 108;104922 109;104923 110;104924 111;104925 112;104926
113;104927 114;104928 115;104929 116;104930 117;104931 118;104932 119;104933
120;104934 121;104935 122;104936 123;104937 124;104938 125;104939 126;104940
127;104941 128;104942 129;104943 130;104944 131;104945 132;104946 133;104947
134;104948 135;104949 136;104951 137;104952 138;104953 139;104954 140;104955
141;104956 142;104957 143;104958 144;104959 145;104960 146;104961 147;104962
148;104963 149;104964 150;104965 151;104966 152;104967 153;104968 154;104969
155;104970 156;104971 157;104972 158;104973 159;104974 160;104975 161;104976
162;104977 163;104978 164;104979 165;104980 166;104982 167;104983 168;104984
169;104985 170;104986 171;104987 172;104988 173;104989 174;104990 175;104991
176;104992 177;104993 178;104994 179;104995 180;104996 181;104997 182;104999
183;105000 184;105001 185;105002 186;105003 187;105004 188;105005 189;105006
190;105007 191;105008 192;105009 193;105010 194;105038 195;105039 196;105040
197;105042 198;105043 199;105044 200;105045 201;105047 202;105061 203;105063
204;105064 205;105065 206;105066 207;105067 208;105069 209;105070 210;105071
211;105072 212;105073 213;105074 214;105075 215;114869 216;119024 217;119027
218;119028 219;119029 220;119033 221;119037 222;119038 223;119040 224;119041
225;119042 226;119043 227;119044 228;119050 229;119051 230;119052 231;119054
232;119056 233;119057 234;119058 235;119059 236;119060 237;119061 238;119062
239;119063 240;119064 241;119065 242;119066 243;119067 244;119069 245;119070
246;119071 247;119072 248;119073 249;119074 250;119075 251;119076 252;119077
253;119078 254;119079 255;119081 256;119082 257;119083 258;119097 259;119099
260;119104 261;119105 262;119106 263;119107 264;119108 265;119109 266;119110
267;119111 268;119112 269;119113 270;119114 271;119115 272;119117 273;119118
274;119119 275;119120 276;119121 277;119122 278;119123 279;119124 280;119125
281;119126 282;119127 283;119128 284;119130 285;119131 286;119134 287;119135
288;119136 289;119137 290;119138 291;119141 292;119145 293;119146 294;119147
295;119148 296;119149 297;119150 298;119153 299;119154 300;119155 301;119156
302;119157 303;119158 304;119159 305;119160 306;119161 307;119162 308;119164
309;119168 310;119169 311;119170 312;119171 313;119172 314;119173 315;119174
316;119175 317;119177 318;119179 319;119180 320;119181 321;119183 322;119185
323;119186 324;119187 325;119188 326;119189 327;119191 328;119192 329;119193
330;119194 331;119195 332;119196 333;119197 334;119198 335;119199 336;119200
337;119201 338;119202 339;119203 340;119204 341;119205 342;119206 343;119207
344;119209 345;119210 346;119211 347;119212 348;119213 349;119216 350;119218
351;119219 352;119222 353;119227 354;119235 355;119237 356;119238 357;119239
358;119242 359;119243 360;119244 361;119245 362;119246 363;119247 364;119248
365;119249 366;119252 367;119253 368;119254 369;119255 370;119256 371;119258
372;119259 373;119263 374;119264 375;119267 376;119268 377;119269 378;119271
379;119272 380;119273 381;119274 382;119275 383;119276 384;119278 385;119281
386;119282 387;119284 388;119287 389;119291 390;119293 391;119294 392;119296
393;119297 394;119298 395;119299 396;119300 397;119301 398;119302 399;119303
400;119304 401;119305 402;119306 403;119307 404;119308 405;119309 406;119310
407;119313 408;119314 409;119315 410;119316 411;119317 412;119318 413;119319
414;119321 415;119322 416;119323 417;119324 418;119325 419;119326 420;119327
421;119328 422;119329 423;119330 424;119331 425;119332 426;119333 427;119334
428;119345 429;119348 430;119349 431;119350 432;119351 433;119352 434;119353
435;119354 436;119355 437;119356 438;119358 439;119359 440;119360 441;119361
442;119362 443;119363 444;119364 445;119365 446;119366 447;119367 448;119368
449;119369 450;119370 451;119371 452;119372 453;119373 454;119374 455;119375
456;119376 457;119377 458;119378 459;119381 460;119382 461;119383 462;119384
463;119387 464;119388 465;119389 466;119390 467;119391 468;119392 469;119393
470;119394 471;119395 472;119396 473;119397 474;119398 475;119399 476;119400
477;119402 478;119407 479;119408 480;119409 481;119410 482;119411 483;119412
484;119413 485;119414 486;119415 487;119416 488;119417 489;119418 490;119419
491;119420 492;119421 493;119422 494;119423 495;119424 496;119425 497;119426
498;119428 499;119429 500;119430 501;119431 502;119432 503;119433 504;119434
505;119435 506;119436 507;119437 508;119438 509;119439 510;119440 511;119441
512;119442 513;119443 514;119444 515;119445 516;119446 517;119447 518;119448
519;119449 520;119450 521;119451 522;119452 523;119453 524;119454 525;119455
526;119456 527;119457 528;119458 529;119459 530;119460 531;119461 532;119462
533;119463 534;119464 535;119465 536;119466 537;119467 538;119468 539;119469
540;119471 541;119473 542;119474 543;119475 544;119476 545;119477 546;119479
547;119480 548;119483 549;119484 550;119486 551;119487 552;119488 553;119489
554;119490 555;119491 556;119492 557;119493 558;119494 559;119495 560;119496
561;119502 562;119503 563;119504 564;119505 565;119508 566;119516 567;119517
568;119520 569;119521 570;119523 571;119524 572;119525 573;119526 574;119527
575;119528 576;119529 577;119531 578;119532 579;119533 580;119534 581;119535
582;119537 583;119538 584;119539 585;119540 586;119541 587;119542 588;119543

```
687;119827 688;119828 689;119830 690;119831 691;119838 692;119841 693;119843
694;119845 695;119846 696;119847 697;119849 698;119850 699;119851 700;119852
701;119853 702;119855 703;119860 704;119861 705;119862 706;119863 707;119864
708;119866 709;119867 710;119868 711;119870 712;119871 713;119872 714;119873
715;119874 716;119875 717;119876 718;119878 719;119879 720;119880 721;119881
722;119883 723;119884 724;119885 725;119886 726;119887 727;119888 728;119889
729;119890 730;119891 731;119893 732;119894 733;119895 734;119896 735;119897
736;119898 737;119900 738;119901 739;119902 740;119903 741;119904 742;119905
743;119906 744;119910 745;119911 746;119912 747;119916 748;119919 749;119921
750;119924 751;119925 752;119927 753;119928 754;119930 755;119932 756;119937
757;119939 758;119940 759;119942 760;119943 761;119944 762;119945 763;119947
764;119948 765;119950 766;119953 767;119954 768;119959 769;119962 770;119970
771;119972 772;119975 773;119976 774;119977 775;119978 776;119979 777;119980
778;119981 779;119982 780;119983 781;119984 782;119986 783;119993 784;119997
785;120002 786;120003 787;120012 788;120015 789;120018 790;120020 791;120023
792;120024 793;120025 794;120026 795;120028 796;120029 797;120030 798;120031
799;120032 800;120033 801;120034 802;120035 803;120036 804;120037 805;120039
806;120041 807;120042 808;120044 809;120052 810;120059 811;120060 812;120061
813;120062 814;120063 815;120064 816;120067 817;120068 818;120069 819;120071
820;120077 821;120092 822;120097 823;120107 824;120108 825;120109 826;120110
827;120112 828;120113 829;120121 830;120122 831;120304 832;120305 833;120306
834;120309 835;120311 836;120541 837;120542 838;120546 839;120548 840;120549
841;120550 842;120554 843;120555 844;120556 845;120557 846;120558 847;120559
848;120560 849;120561 850;120563 851;120565 852;120566 853;120567 854;120568
855;120569 856;120571 857;120572 858;120574 859;120580 860;120581 861;120582
862;120583 863;120584 864;120585 865;120586 866;120587 867;120589 868;120593
869;120594 870;120595 871;120599 872;120600 873;120601 874;120602 875;120603
876;120604 877;120606 878;120607 879;120631 880;120633 881;120636 882;120637
883;120638 884;120639 885;120640 886;120641 887;120644 888;120645 889;120646
890;120649 891;120652 892;120656 893;120659 894;120672 895;120674 896;120676
897;120677 898;120678 899;120679 900;120680 901;120687 902;120693 903;120697
904;120698 905;120699 906;120700 907;120701 908;120703 909;120704 910;120705
911;120706 912;120708 913;120709 914;120710 915;120713 916;120714 917;120715
918;120716 919;120723 920;120730 921;120731 922;120732 923;120734 924;120738
925;120739 926;120740 927;120741 928;120742 929;120743 930;120744 931;120745
932;120746 933;120747 934;120748 935;120749 936;120750 937;120752 938;120753
939;120755 940;120756 941;121075 942;121079 943;121080 944;121084 945;121086
946;121091 947;121097 948;121099 949;121100 950;121102 951;121103 952;121108
953;121109 954;121110 955;121112 956;121126 957;121127 958;121128 959;121272
960;121273 961;121274 962;121281 963;121282 964;121291 965;121292 966;121293
967;121294 968;121295 969;121296 970;121297 971;121298 972;121300 973;121301
974;121302 975;121304 976;121305 977;121308 978;121309 979;121310 980;121312
981;121313 982;121314 983;121315 984;121316 985;121317 986;121320 987;121321
988;121322 989;121323 990;121324 991;121328 992;121329 993;121332 994;121333
995;121334 996;121335 997;121336 998;121337 999;121338 1000;121339 1001;121889
1002;121890 1003;121891 1004;121892 1005;121893 1006;122271 1007;122399 1008;122401
1009;122404 1010;123664 1011;123665 1012;123673 1013;123682 1014;123684 1015;123686
1016;123693 1017;123696 1018;123704 1019;123705 1020;123706 1021;123707 1022;123708
1023;123709 1024;123710 1025;123712 1026;123713 1027;123714 1028;123715 1029;123716
1030;123717 1031;123718 1032;123719 1033;123720 1034;123722 1035;123723 1036;123724
1037;123725 1038;123726 1039;123727 1040;123728 1041;123750 1042;123751 1043;123752
1044;123775 1045;123776 1046;123777 1047;123778 1048;123779 1049;123781 1050;123782
1051;123783 1052;123784 1053;123785 1054;123786 1055;123787 1056;123788 1057;123789
1058;123790 1059;123791 1060;123792 1061;123793 1062;123794 1063;123795 1064;123796
1065;123797 1066;123801 1067;123802 1068;123803 1069;123804 1070;123808 1071;123809
1072;123810 1073;123811 1074;123812 1075;123813 1076\", \"NODATA\");
out_raster.save(r\"C:\\Users\\Alexander Danielson\\Desktop\\Fall
2022Spring2023\\ArcGIS I\\Lab2\\Lab2\\Lab2.gdb\\ReClassStreams\")"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {},
   "source": [
    "### Use Raster Calculator For Streams"
   ]
  },
  {
   "cell_type": "code",
```

```
   "execution_count": 106,
   "metadata": {},
   "outputs": [
    {
     "ename": "SyntaxError",
     "evalue": "invalid syntax (<string>, line 1)",
     "output_type": "error",
     "traceback": [
      "\u001b[0;31m-------------------------------------------------------------------------
      -----\u001b[0m\n\u001b[0;31mSyntaxError\u001b[0m
      Traceback (most recent call last)",
      "File \u001b[0;34mC:\\Program
      Files\\ArcGIS\\Pro\\bin\\Python\\envs\\arcgispro-py3\\lib\\ast.py\u001b[0m, in
      \u001b[0;32mparse\u001b[0m:\nLine \u001b[0;34m50\u001b[0m:
      \u001b[34mreturn\u001b[39;49;00m \u001b[36mcompile\u001b[39;49;00m(source,
      filename, mode, flags,\n",
      "\u001b[0;31mSyntaxError\u001b[0m: invalid syntax (<string>, line
      1)\n\u001b[0;31m-------------------------------------------------------------------------
      --------\u001b[0m"
     ]
    }
   ],
   "source": [
    "output_raster = arcpy.ia.RasterCalculator('StreamOutput =
    Con(IsNull(\"ReClassStreams\"), 0 , \"ReClassStreams\" )');
    output_raster.save(r\"C:\\Users\\Alexander Danielson\\Desktop\\Fall
    2022Springoutput_raster = arcpy.ia.RasterCalculator('StreamOutput =
    Con(IsNull(\"ReClassStreams\"), 0 , \"ReClassStreams\" )');
    output_raster.save(r\"C:\\Users\\Alexander Danielson\\Desktop\\Fall
    2022Spring2023\\ArcGIS
    I\\Lab2\\Lab2\\Lab2.gdb\\RasterCalculatorStreams\")2023\\ArcGIS
    I\\Lab2\\Lab2\\Lab2.gdb\\RasterCalculatorStreams\")"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {},
   "source": [
    "# Input Reclassifed Rasters to Be Weighted and Create Cost Surface Analysis"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {},
   "source": [
    "## Create Weighted Raster"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 107,
   "metadata": {},
   "outputs": [],
   "source": [
    "out_raster = arcpy.sa.WeightedOverlay(r\"('C:\\Users\\Alexander
    Danielson\\Desktop\\Fall 2022Spring2023\\ArcGIS
    I\\Lab2\\Lab2\\Lab2.gdb\\RasterCalculatorStreams' 100 'Value' (0 1; 1 1; 2 2; 3 3; 4
    4; 5 6; 6 7; 7 8; 8 9; 9 10; 10 1; 11 1; 12 1; 13 1; 14 1; 15 1; 16 1; 17 1; 18 1;
    19 1; 20 1; 21 1; 22 1; 23 1; 24 1; 25 1; 26 1; 27 1; 28 1; 29 1; 30 1; 31 1; 32 1;
    33 1; 34 1; 35 1; 36 1; 37 1; 38 1; 39 1; 40 1; 41 1; 42 1; 43 1; 44 1; 45 1; 46 1;
    47 1; 48 1; 49 1; 50 1; 51 1; 52 1; 53 1; 54 1; 55 1; 56 1; 57 1; 58 1; 59 1; 60 1;
    61 1; 62 1; 63 1; 64 1; 65 1; 66 1; 67 1; 68 1; 69 1; 70 1; 71 1; 72 1; 73 1; 74 1;
    75 1; 76 1; 77 1; 78 1; 79 1; 80 1; 81 1; 82 1; 83 1; 84 1; 85 1; 86 1; 87 1; 88 1;
    89 1; 90 1; 91 1; 92 1; 93 1; 94 1; 95 1; 96 1; 97 1; 98 1; 99 1; 100 1; 101 1; 102
    1; 103 1; 104 1; 105 1; 106 1; 107 1; 108 1; 109 1; 110 1; 111 1; 112 1; 113 1; 114
    1; 115 1; 116 1; 117 1; 118 1; 119 1; 120 1; 121 1; 122 1; 123 1; 124 1; 125 1; 126
    1; 127 1; 128 1; 129 1; 130 1; 131 1; 132 1; 133 1; 134 1; 135 1; 136 1; 137 1; 138
    1; 139 1; 140 1; 141 1; 142 1; 143 1; 144 1; 145 1; 146 1; 147 1; 148 1; 149 1; 150
```

1; 151 1; 152 1; 153 1; 154 1; 155 1; 156 1; 157 1; 158 1; 159 1; 160 1; 161 1; 162
1; 163 1; 164 1; 165 1; 166 1; 167 1; 168 1; 169 1; 170 1; 171 1; 172 1; 173 1; 174
1; 175 1; 176 1; 177 1; 178 1; 179 1; 180 1; 181 1; 182 1; 183 1; 184 1; 185 1; 186
1; 187 1; 188 1; 189 1; 190 1; 191 1; 192 1; 193 1; 194 1; 195 1; 196 1; 197 1; 198
1; 199 1; 200 1; 201 1; 202 1; 203 1; 204 1; 205 1; 206 1; 207 1; 208 1; 209 1; 210
1; 211 1; 212 1; 213 1; 214 1; 215 1; 216 1; 217 1; 218 1; 219 1; 220 1; 221 1; 222
1; 223 1; 224 1; 225 1; 226 1; 227 1; 228 1; 229 1; 230 1; 231 1; 232 1; 233 1; 234
1; 235 1; 236 1; 237 1; 238 1; 239 1; 240 1; 241 1; 242 1; 243 1; 244 1; 245 1; 246
1; 247 1; 248 1; 249 1; 250 1; 251 1; 252 1; 253 1; 254 1; 255 1; 256 1; 257 1; 258
1; 259 1; 260 1; 261 1; 262 1; 263 1; 264 1; 265 1; 266 1; 267 1; 268 1; 269 1; 270
1; 271 1; 272 1; 273 1; 274 1; 275 1; 276 1; 277 1; 278 1; 279 1; 280 1; 281 1; 282
1; 283 1; 284 1; 285 1; 286 1; 287 1; 288 1; 289 1; 290 1; 291 1; 292 1; 293 1; 294
1; 295 1; 296 1; 297 1; 298 1; 299 1; 300 1; 301 1; 302 1; 303 1; 304 1; 305 1; 306
1; 307 1; 308 1; 309 1; 310 1; 311 1; 312 1; 313 1; 314 1; 315 1; 316 1; 317 1; 318
1; 319 1; 320 1; 321 1; 322 1; 323 1; 324 1; 325 1; 326 1; 327 1; 328 1; 329 1; 330
1; 331 1; 332 1; 333 1; 334 1; 335 1; 336 1; 337 1; 338 1; 339 1; 340 1; 341 1; 342
1; 343 1; 344 1; 345 1; 346 1; 347 1; 348 1; 349 1; 350 1; 351 1; 352 1; 353 1; 354
1; 355 1; 356 1; 357 1; 358 1; 359 1; 360 1; 361 1; 362 1; 363 1; 364 1; 365 1; 366
1; 367 1; 368 1; 369 1; 370 1; 371 1; 372 1; 373 1; 374 1; 375 1; 376 1; 377 1; 378
1; 379 1; 380 1; 381 1; 382 1; 383 1; 384 1; 385 1; 386 1; 387 1; 388 1; 389 1; 390
1; 391 1; 392 1; 393 1; 394 1; 395 1; 396 1; 397 1; 398 1; 399 1; 400 1; 401 1; 402
1; 403 1; 404 1; 405 1; 406 1; 407 1; 408 1; 409 1; 410 1; 411 1; 412 1; 413 1; 414
1; 415 1; 416 1; 417 1; 418 1; 419 1; 420 1; 421 1; 422 1; 423 1; 424 1; 425 1; 426
1; 427 1; 428 1; 429 1; 430 1; 431 1; 432 1; 433 1; 434 1; 435 1; 436 1; 437 1; 438
1; 439 1; 440 1; 441 1; 442 1; 443 1; 444 1; 445 1; 446 1; 447 1; 448 1; 449 1; 450
1; 451 1; 452 1; 453 1; 454 1; 455 1; 456 1; 457 1; 458 1; 459 1; 460 1; 461 1; 462
1; 463 1; 464 1; 465 1; 466 1; 467 1; 468 1; 469 1; 470 1; 471 1; 472 1; 473 1; 474
1; 475 1; 476 1; 477 1; 478 1; 479 1; 480 1; 481 1; 482 1; 483 1; 484 1; 485 1; 486
1; 487 1; 488 1; 489 1; 490 1; 491 1; 492 1; 493 1; 494 1; 495 1; 496 1; 497 1; 498
1; 499 1; 500 1; 501 1; 502 1; 503 1; 504 1; 505 1; 506 1; 507 1; 508 1; 509 1; 510
1; 511 1; 512 1; 513 1; 514 1; 515 1; 516 1; 517 1; 518 1; 519 1; 520 1; 521 1; 522
1; 523 1; 524 1; 525 1; 526 1; 527 1; 528 1; 529 1; 530 1; 531 1; 532 1; 533 1; 534
1; 535 1; 536 1; 537 1; 538 1; 539 1; 540 1; 541 1; 542 1; 543 1; 544 1; 545 1; 546
1; 547 1; 548 1; 549 1; 550 1; 551 1; 552 1; 553 1; 554 1; 555 1; 556 1; 557 1; 558
1; 559 1; 560 1; 561 1; 562 1; 563 1; 564 1; 565 1; 566 1; 567 1; 568 1; 569 1; 570
1; 571 1; 572 1; 573 1; 574 1; 575 1; 576 1; 577 1; 578 1; 579 1; 580 1; 581 1; 582
1; 583 1; 584 1; 585 1; 586 1; 587 1; 588 1; 589 1; 590 1; 591 1; 592 1; 593 1; 594
1; 595 1; 596 1; 597 1; 598 1; 599 1; 600 1; 601 1; 602 1; 603 1; 604 1; 605 1; 606
1; 607 1; 608 1; 609 1; 610 1; 611 1; 612 1; 613 1; 614 1; 615 1; 616 1; 617 1; 618
1; 619 1; 620 1; 621 1; 622 1; 623 1; 624 1; 625 1; 626 1; 627 1; 628 1; 629 1; 630
1; 631 1; 632 1; 633 1; 634 1; 635 1; 636 1; 637 1; 638 1; 639 1; 640 1; 641 1; 642
1; 643 1; 644 1; 645 1; 646 1; 647 1; 648 1; 649 1; 650 1; 651 1; 652 1; 653 1; 654
1; 655 1; 656 1; 657 1; 658 1; 659 1; 660 1; 661 1; 662 1; 663 1; 664 1; 665 1; 666
1; 667 1; 668 1; 669 1; 670 1; 671 1; 672 1; 673 1; 674 1; 675 1; 676 1; 677 1; 678
1; 679 1; 680 1; 681 1; 682 1; 683 1; 684 1; 685 1; 686 1; 687 1; 688 1; 689 1; 690
1; 691 1; 692 1; 693 1; 694 1; 695 1; 696 1; 697 1; 698 1; 699 1; 700 1; 701 1; 702
1; 703 1; 704 1; 705 1; 706 1; 707 1; 708 1; 709 1; 710 1; 711 1; 712 1; 713 1; 714
1; 715 1; 716 1; 717 1; 718 1; 719 1; 720 1; 721 1; 722 1; 723 1; 724 1; 725 1; 726
1; 727 1; 728 1; 729 1; 730 1; 731 1; 732 1; 733 1; 734 1; 735 1; 736 1; 737 1; 738
1; 739 1; 740 1; 741 1; 742 1; 743 1; 744 1; 745 1; 746 1; 747 1; 748 1; 749 1; 750
1; 751 1; 752 1; 753 1; 754 1; 755 1; 756 1; 757 1; 758 1; 759 1; 760 1; 761 1; 762
1; 763 1; 764 1; 765 1; 766 1; 767 1; 768 1; 769 1; 770 1; 771 1; 772 1; 773 1; 774
1; 775 1; 776 1; 777 1; 778 1; 779 1; 780 1; 781 1; 782 1; 783 1; 784 1; 785 1; 786
1; 787 1; 788 1; 789 1; 790 1; 791 1; 792 1; 793 1; 794 1; 795 1; 796 1; 797 1; 798
1; 799 1; 800 1; 801 1; 802 1; 803 1; 804 1; 805 1; 806 1; 807 1; 808 1; 809 1; 810
1; 811 1; 812 1; 813 1; 814 1; 815 1; 816 1; 817 1; 818 1; 819 1; 820 1; 821 1; 822
1; 823 1; 824 1; 825 1; 826 1; 827 1; 828 1; 829 1; 830 1; 831 1; 832 1; 833 1; 834
1; 835 1; 836 1; 837 1; 838 1; 839 1; 840 1; 841 1; 842 1; 843 1; 844 1; 845 1; 846
1; 847 1; 848 1; 849 1; 850 1; 851 1; 852 1; 853 1; 854 1; 855 1; 856 1; 857 1; 858
1; 859 1; 860 1; 861 1; 862 1; 863 1; 864 1; 865 1; 866 1; 867 1; 868 1; 869 1; 870
1; 871 1; 872 1; 873 1; 874 1; 875 1; 876 1; 877 1; 878 1; 879 1; 880 1; 881 1; 882
1; 883 1; 884 1; 885 1; 886 1; 887 1; 888 1; 889 1; 890 1; 891 1; 892 1; 893 1; 894
1; 895 1; 896 1; 897 1; 898 1; 899 1; 900 1; 901 1; 902 1; 903 1; 904 1; 905 1; 906
1; 907 1; 908 1; 909 1; 910 1; 911 1; 912 1; 913 1; 914 1; 915 1; 916 1; 917 1; 918
1; 919 1; 920 1; 921 1; 922 1; 923 1; 924 1; 925 1; 926 1; 927 1; 928 1; 929 1; 930
1; 931 1; 932 1; 933 1; 934 1; 935 1; 936 1; 937 1; 938 1; 939 1; 940 1; 941 1; 942
1; 943 1; 944 1; 945 1; 946 1; 947 1; 948 1; 949 1; 950 1; 951 1; 952 1; 953 1; 954
1; 955 1; 956 1; 957 1; 958 1; 959 1; 960 1; 961 1; 962 1; 963 1; 964 1; 965 1; 966
1; 967 1; 968 1; 969 1; 970 1; 971 1; 972 1; 973 1; 974 1; 975 1; 976 1; 977 1; 978

```
        9 10; 10 1; NODATA NODATA));1 10 1\"); out_raster.save(r\"C:\\Users\\Alexander
        Danielson\\Desktop\\Fall 2022Spring2023\\ArcGIS
        I\\Lab2\\Lab2\\Lab2.gdb\\Weighte_Rast1\")"
       ]
      },
      {
       "cell_type": "markdown",
       "metadata": {},
       "source": [
        "## Create Cost Surface (Distance) Raster"
       ]
      },
      {
       "cell_type": "code",
       "execution_count": 108,
       "metadata": {},
       "outputs": [],
       "source": [
        "out_distance_raster = arcpy.sa.CostDistance(\"DisplayXYPathDoryFarmtoNPA\",
        \"WeightedRaster\", None, None, None, None, None, None, '');
        out_distance_raster.save(r\"C:\\Users\\Alexander Danielson\\Desktop\\Fall
        2022Spring2023\\ArcGIS I\\Lab2\\Lab2\\Lab2.gdb\\CostDis_Disp1\")"
       ]
      },
      {
       "cell_type": "markdown",
       "metadata": {},
       "source": [
        "## Create Back Link "
       ]
      },
      {
       "cell_type": "code",
       "execution_count": 109,
       "metadata": {},
       "outputs": [],
       "source": [
        "out_backlink_raster = arcpy.sa.CostBackLink(\"DisplayXYPathDoryFarmtoNPA\",
        \"WeightedRaster\", None, None, None, None, None, None, '');
        out_backlink_raster.save(r\"C:\\Users\\Alexander Danielson\\Desktop\\Fall
        2022Spring2023\\ArcGIS I\\Lab2\\Lab2\\Lab2.gdb\\CostSurfaceBackLink\")"
       ]
      },
      {
       "cell_type": "markdown",
       "metadata": {},
       "source": [
        "## Create Cost Surface Allocation Raster"
       ]
      },
      {
       "cell_type": "code",
       "execution_count": 110,
       "metadata": {},
       "outputs": [],
       "source": [
        "out_allocation_raster = arcpy.sa.CostAllocation(\"DisplayXYPathDoryFarmtoNPA\",
        \"WeightedRaster\", None, None, \"OBJECTID\", None, None, None, None, None, None,
        ''); out_allocation_raster.save(r\"C:\\Users\\Alexander Danielson\\Desktop\\Fall
        2022Spring2023\\ArcGIS I\\Lab2\\Lab2\\Lab2.gdb\\CosrSurfaceAllo\")"
       ]
      },
      {
       "cell_type": "markdown",
       "metadata": {},
       "source": [
        "## Create Cost Path for Final Route"
       ]
      },
      {
```

```
   {
    "cell_type": "code",
    "execution_count": 111,
    "metadata": {},
    "outputs": [],
    "source": [
     "out_raster = arcpy.sa.CostPath(\"DisplayXYPathDoryFarmtoNPA\",
     \"CostSurfaceDistance\", \"CostSurfaceBackLink\", \"EACH_CELL\", \"OBJECTID\",
     \"INPUT_RANGE\"); out_raster.save(r\"C:\\Users\\Alexander Danielson\\Desktop\\Fall
     2022Spring2023\\ArcGIS I\\Lab2\\Lab2\\Lab2.gdb\\CostPathRoute\")"
    ]
   },
   {
    "cell_type": "code",
    "execution_count": null,
    "metadata": {},
    "outputs": [],
    "source": []
   }
  ],
  "metadata": {
   "kernelspec": {
    "display_name": "ArcGISPro",
    "language": "Python",
    "name": "python3"
   },
   "language_info": {
    "file_extension": ".py",
    "name": "python",
    "version": "3"
   }
  },
  "nbformat": 4,
  "nbformat_minor": 2
 }
```