Lab 2: Cost Surfaces

Step 1: Download required data

1. Request required data
2. Save and unzip to file

Step 2: Clip data

1. Establish boundaries
2. Clip data

Step 3: Create cost accumulation surface

1. Reclassify component rasters
2. Add rasters together

Step 4: Create source and destination rasters

1. Create point feature classes
2. Add defined points to feature classes

Step 5: Calculate distance accumulation function and backlink raster

1. Both use source raster and cost surface

Step 6: Calculate optimal path function

Details:

1. Create ETL for data to go into cost surface model
2. Create cost surface model and justify how you created cost surface
3. Map optimal path from two points over the constructed cost surface
4. (44.127985, -92.148796) to North Picnic area (44.05438888888889 -92.04483333333333)

Specific preferences:

1. Not walk through any farm fields
2. Doesn’t like crossing water bodies if no bridge
3. Needs path that is the most gradual slopewise

Step 1: DOWNLOAD DATA

In [21]:
```
# DEM: https://gisdata.mn.gov/dataset/elev-30m-digital-elevation-model
# Crop Area: https://gisdata.mn.gov/dataset/agri-cropland-data-layer-2018
#crop area has water on it!
import requests, json, zipfile
import json
import zipfile

def CKAN_retrieval(search_query, result_num, resource_num):

#call API to search packages with search query
    big_url = 'https://gisdata.mn.gov/api/3/action/package_search?q=' + search
_query

#send a request to the API address, do not verify security
    response = requests.get(big_url, verify = False)

#turn result into JSON dictionary
    json_response = json.loads(response.content)

#dig down through dictionary layers to find the right resource
    result_options = json_response['result']['results']
    chosen_result = result_options[result_num]
    resources_under_result= chosen_result['resources'][resource_num]
    chosen_resource = resources_under_result['url']
    print(chosen_resource)
    URL_request = requests.get(chosen_resource)

# define a save file name and write data to it, close file
    save_path = search_query[0:8] + ".zip"
    with open(save_path, 'wb') as f:
        f.write(URL_request.content)
        f.close()

#unzip the file into the same directory
    with zipfile.ZipFile(save_path,"r") as zip_ref:
        zip_ref.extractall()
    print('Complete')

#function calls
CKAN_retrieval('elev-30m-digital-elevation-model',0,1)
CKAN_retrieval('agri-cropland-data-layer-2018',0,1)
```

```
C:\Program Files\ArcGIS\Pro\bin\Python\envs\arcgispro-py3\lib\site-packages\u
rllib3\connectionpool.py:1004: InsecureRequestWarning: Unverified HTTPS reque
st is being made to host 'gisdata.mn.gov'. Adding certificate verification is
strongly advised. See: https://urllib3.readthedocs.io/en/latest/advanced-usag
e.html#ssl-warnings
  InsecureRequestWarning,

https://resources.gisdata.mn.gov/pub/gdrs/data/pub/us_mn_state_dnr/elev_30m_d
igital_elevation_model/fgdb_elev_30m_digital_elevation_model.zip
Complete. Check notebook folder

C:\Program Files\ArcGIS\Pro\bin\Python\envs\arcgispro-py3\lib\site-packages\u
rllib3\connectionpool.py:1004: InsecureRequestWarning: Unverified HTTPS reque
st is being made to host 'gisdata.mn.gov'. Adding certificate verification is
strongly advised. See: https://urllib3.readthedocs.io/en/latest/advanced-usag
e.html#ssl-warnings
  InsecureRequestWarning,

https://resources.gisdata.mn.gov/pub/gdrs/data/pub/us_mn_state_mda/agri_cropl
and_data_layer_2018/fgdb_agri_cropland_data_layer_2018.zip
Complete. Check notebook folder
```

Step 2: CLIP DATA

In [9]:
```
import arcpy

# define a clip extent
clip = "560000 4850000 600000 4900000"

# clip each layer by clip extent and save output to Clip_xxx
arcpy.Clip_management("agri_cropland_data_layer_2018.gdb/agri_cropland_data_la
yer_2018",
                      clip, "Lab2.gdb/Clip_Ag_Wtr")
arcpy.Clip_management("elev_30m_digital_elevation_model.gdb/digital_elevation_
model_30m",
                      clip, "Lab2.gdb/Clip_DEM")
```

Out[9]:

Output

Lab2.gdb\Clip_DEM

Messages

Start Time: Thursday, February 25, 2021 10:00:36 PM

Building Pyramids...

Succeeded at Thursday, February 25, 2021 10:00:37 PM (Elapsed Time: 0.84 seconds)

Step 3: CREATE ACCUMULATION SURFACE

In [2]:
```
##### turns clipped files into merged cost accumulation surface
import arcpy
from arcpy.sa import *
#the DEM needs to be turned to slope and rescaled

# re-establish workspace
arcpy.env.workspace = "C://Users/Cole/Documents/GitHub/GIS5572/Lab2/Lab2.gdb"

#create a slope raster from the clipped elevation model
#the other raster will be standardized to this
#wait a sec, this might yeild negative values, which error later
arcpy.Slope_3d("Clip_DEM", "Slope")
#arcpy.Rescale

#reclassify

#standardize the agriculture/water layer
#values: 1-71, 205-246 are crops, unsuitable
#values: 111 are water, unsuitalbe
#values 72-110, 112-204 are suitable
#remap as 1 and 100
class1 = arcpy.sa.Reclassify("Clip_Ag_Wtr", "VALUE", RemapRange([[0, 72, 100],
[205, 247, 100],[110,115,100],[72,110,1],[115,204,1]]))
class1.save("Ag_Wtr_Reclass")
#add the weighted, standardized rasters together to create a cost/accumulation
surface
merged_raster = Raster("Ag_Wtr_Reclass") + Raster("Slope")
merged_raster.save("Merged_Surface")
```

Step 4: CREATE SOURCE/DEST FEATURE CLASSES

In [11]:
```
import arcpy

arcpy.env.workspace = "C://Users/Cole/Documents/GitHub/GIS5572/Lab2/Lab2.gdb"

#gather the spatial referenence from the cost surface
spatial_ref = arcpy.Describe("Ag_Wtr_Reclass").spatialReference

#create a new feature class for the source point
arcpy.CreateFeatureclass_management("Lab2.gdb", "SourcePoint", "POINT",
                                    spatial_reference = spatial_ref)

#create a new feature class for the dest point
arcpy.CreateFeatureclass_management("Lab2.gdb", "DestPoint", "POINT",
                                    spatial_reference = spatial_ref)

#add a point to the new source feature class
feature_class_sour = "SourcePoint"
cursor = arcpy.da.InsertCursor(feature_class_sour,"SHAPE@XY")
xy = arcpy.Point(568097.73, 4886440.22)
cursor.insertRow([xy])
del cursor

#add a point to the new dest feature class
feature_class_dest = "DestPoint"
cursor = arcpy.da.InsertCursor(feature_class_dest,"SHAPE@XY")
xy = arcpy.Point(576512.44, 4878357.35)
cursor.insertRow([xy])
del cursor
```

Step 5: DISTANCE ACCUMULATION FUNCTION + BACKLINK

In [3]:
```
import arcpy
from arcpy.sa import *

arcpy.env.workspace = "C://Users/Cole/Documents/GitHub/GIS5572/Lab2/Lab2.gdb"

#execute distance accumulation function with source raster and cost surface
cost_surface = DistanceAccumulation("SourcePoint", in_cost_raster = "Merged_Su
rface")

#create a backlink raster using source raster and cost surface
BackLink = CostBackLink("SourcePoint", "Merged_Surface")
```

Step 6: OPTIMAL PATH FUNCTION

In [5]:
```
import arcpy

#execute optimal distance function using the distance accumulation, destinatio
n raster, and backlink
OptimalPathAsLine("DestPoint", "Distanc_Sour1", "CostBac_Sour1", "paths2")
```

Out[5]:
```
<geoprocessing server result object at 0x2e6590ff030>
```

In [ ]:
```

```