Concordia University
COMP 472 Artificial Intelligence – Summer 2020
Assignment 1

Due Date:	By 11:59 PM, Thursday, May 28th 2020
Evaluation:	20% of the final mark
Late Submission:	No late submission will be accepted.
Purpose:	The purpose of this project is to help you learn Python semantics, syntax, libraries such as Numpy, Pandas, Pyshp, Scikit-Learn and design heuristic search algorithms.
Programming Language:	Python
Team:	It is an individual work. No team work is allowed.

General Guidelines When Writing Programs:

Include the following comments at the top of your source codes

```
# ---------------------------------------------------------
# Assignment (include number)
# Written by (include your name and student id)
# For COMP 472 Section (your lab section) – Summer 2020
# ----------------------------------------------------------
```

- Include comments in your program describing the main steps in your program. The Focus in your comments rather on the why than the how.
- Display clear prompts for users when you are expecting the user to enter data from the keyboard if needed.
- All output should be displayed with clear messages and in an easy to read format.
- End your program with a closing message so that the user knows that the program has terminated.

1. Montreal Crime Analytics

In this project, we locate four coordinates ([longitude, latitude]) (-73.59, 45.49), (-73.55, 45.49), (-73.55, 45.53), (-73.59, 45.53) in the center of Montreal downtown area. The blue square in the screenshot (see Figure 1.1) shows the located area. All the crime data in this area are saved in the file "*crime_dt.shp*", which includes all the crime data considered at different points. The dataset is fetched and modified from the City of Montreal's open data portal.

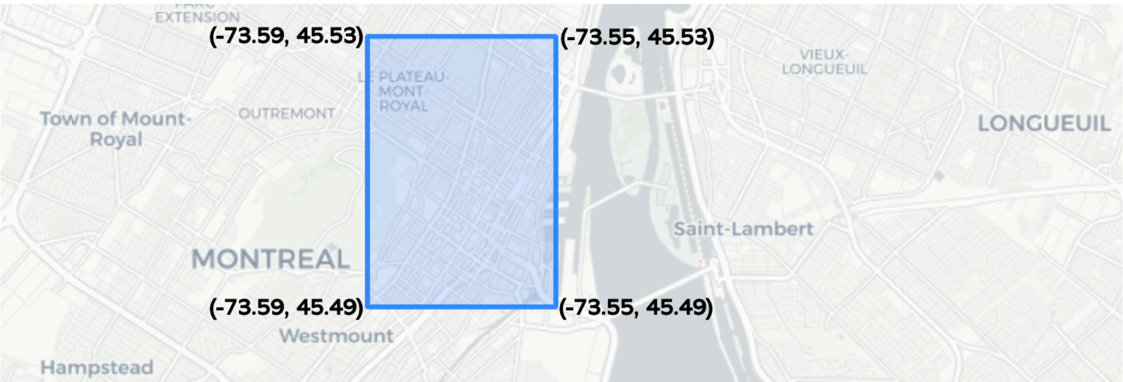
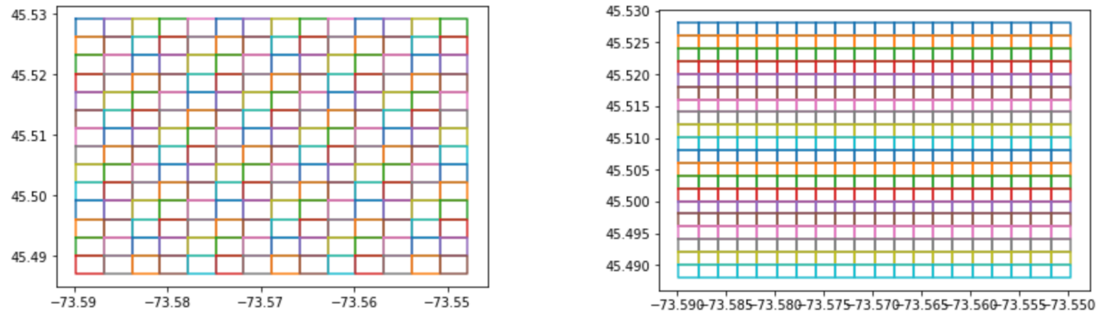


Figure 1.1 Located area in the center of Montreal map

Using this located area, you need to generate grids in this area to compute the crime rate statistics in each grid. The size of each grid could be changed due to the distribution of crime; however, your program should be flexible to change the size of each grid when it is required.

For example, in Figure 1.2 each grid has the size of 0.003 * 0.003 in (a) and 0.002*0.002 in (b). As the size of grids grows, the results of distribution may be focused on each grid evenly. Therefore, to ensure and display the accuracy of the data, smaller size less or equal to 0.002* 0.002 of each grid is recommended.

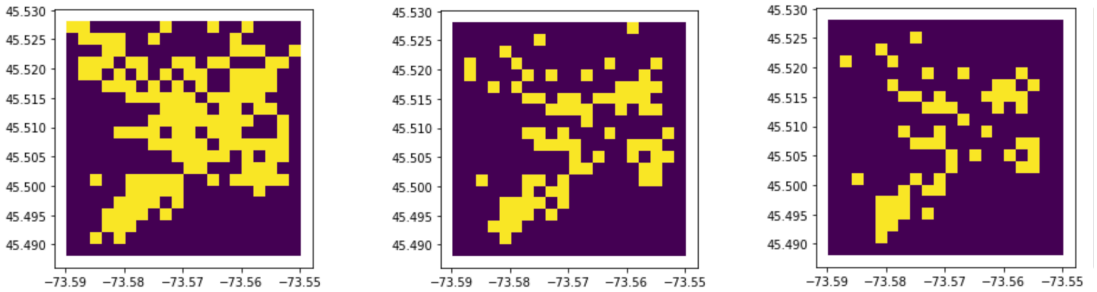


(a) size of each grid 0.003*0.003

(b) size of each grid 0.002*0.002

Figure 1.2 distribution of grids with different sizes

Crime rates are considered as the number of points in each block defined by the grid. The crime rates need to be calculated using statistics such as total count, mean, standard deviation in your results. A threshold of crime rate is introduced to define the high crime rate blocks and low crime rate blocks in the area selected. The threshold would be arbitrarily chosen. For example, if you choose the threshold using the median (the crime rate of a block higher than 50% of the crime rates of total blocks), blocks with the crime rate higher than median are labelled in yellow (highly risky blocks), and blocks with the crime rate lower than median are labelled in purple (less risky blocks). According to different thresholds, the crime rates in the located area will be presented as the following graphs.



(a) threshold 1 (b) threshold 2 (c) threshold 3

Figure 1.3 Using different thresholds to present the crime rates on the map

<u>Definition of Threshold</u>

Assuming all the grids in your program have different number of crime rates, all the numbers are in an descending order. Therefore, if the threshold = 50%, we take the numbers that are greater or equal to median number marked as block area; if the threshold = 75%, we take the numbers that have index (=1, 2…) less or equal to total number of grids*(1-75%) marked as block area; if the threshold = 90%, we take the numbers have index less or equal to total number of grids*(1-90%) marked as block area.

For example, we have the sequence {10, 9, 8, 7, 6, 5, 4, 3, 2, 1}
If the threshold = 50%, the numbers {10, 9, 8, 7, 6} are considered as blocks;
If the threshold = 75%, the numbers {10, 9} are considered as blocks;
If the threshold = 90%, the numbers {10} are considered as blocks.

2. Your Task

In this assignment, you will generate the crime risk map based on the crime rates and implement an A* heuristic search algorithm to find an optimal path between two coordinates on the map.

2.1 Display Results

1. Your program should be able to read the map area located in the area of four coordinates ([longitude, latitude]) (-73.59, 45.49), (-73.55, 45.49), (-73.55, 45.53), (-73.59, 45.53) in the center of Montreal downtown area.
2. Your program can draw the grids and generates graph indicating the crime rates in each grid using the "*crime_dt.shp*" file.
   (**Please note:** to load and read the "*crime_dt.shp*" file, the other four files "*crime_dt.cpg", "crime_dt.dbf", "crime_dt.prj", and "crime_dt.shx"* in the same folder named "Shape" attached should be put in the same local path.)

3. Compute the number of total crimes in each grid, average and standard deviation of all grids and generate graphs displaying the crime rates using different colors based on the local area.
4. Using the graph, which includes yellow blocks considers as obstacles, in the example above illustrates the high crime rate areas (see Figure 1.3), your program should implement an A* heuristic algorithm to find an optimal path from any two coordinates that the user prompt within the map.

   Please note your path should avoid all the high-risk areas in your map and the path could be changed according to the different chosen thresholds. Your program cannot call any existing searching functions to find the optimal path. You have to implemented your original heuristic algorithm.
5. When you design the A* heuristic method and draw the optimal path on the map, here are some rules that should be followed:
   a) Try to avoid the block areas.
   b) All the edges between two block areas and its diagonal are forbidden (see Fig(1) below).
   c) All the edges between one block (yellow) and one low crime rate area (blue) are accessible and the cost of this edge is 1.3 (see Fig(2) below).
   d) All the edges between two low crime rate areas and diagonal are accessible. The cost of this edge = 1 and diagonal cost = 1.5 (see Fig(3) below).

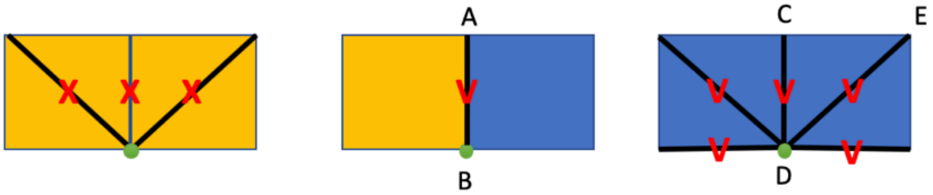


Fig (1) Fig (2) Fig (3)

● - Start Point; X- forbidden; V- Accessible;
Cost: B→A = 1.3; D→C = 1; D→E = 1.5

   e) All the boundary edges of the map are considered as inaccessible.
   f) If the point P is located inside one grid, use the lowest coordinates in this grid as the location. This coordinate is the left bottom coordinates of this grid.

2.2 Programming Details

1. To implement this project, you must use Python.

   Please note the libraries can be used but not limited to Numpy, Pandas, Pyshp, Scikit-Learn. Please list all the required libraries of your program in README.txt.
2. Your program must be able to find the optimal path at most 10 seconds.

   If there is no path will be found from the current map, e.g. the destination are surrounded by blocks, your program should output the message *"Due to blocks, no path is found. Please change the map and try again"*; Otherwise, if the path exists, your program should measure the time when starting to find the optimal path. If it takes more than 10 seconds, your program should output the message *"Time is up. The optimal path is not found."*

3. It is not necessary to have a fancy user-interface. A simple command-line interface with a direct output is sufficient.
4. All the results listed in Section *2.1 Display Details* should be presented in GUI and it should be easy to change the input grid size and the chosen threshold using the command-line interface when it is required.

3. Deliverables

3.1 Demos

Assignment 1 will be demonstrated using Zoom. You will demo the program that was uploaded as the official submission on or before the due date. The schedule of the demos will be posted on Moodle. No special preparation is necessary for the demo (no slides or presentation needed). Your TA will ask you questions about your code, and you will have to answer all the questions. Your code will be checked for similarity.

3.2 Grading Scheme

Grading Criteria	Description	Points
Map (5 pts)	Draw the grids	1 pt
	Generate the map by crime rates	2 pts
	Apply different threshold by prompting user for inputs	1 pt
	Display the number of total crimes in each grid, average and standard deviation of all grids	1 pt
A* Heuristic Method (10 pts)	Heuristic function, total cost, admissibility	4 pts
	Follow the rules in 2.1.5 to design heuristic method	6 pts
Optimal Path (3 pts)	Find the path from any 2 points by prompting user for inputs	2 pts
	Execution time and instant outputs	1 pt
Code Quality	Necessary comments, readability and clarity	1 pt
README.txt	Instructions on how to run your program and list of libraries	1 pt
Total		20 pts

Submitting Assignment 1

Your submission should include the following documents:

1. Read and sign the expectation of originality form (available on Moodle or at https://www.concordia.ca/content/dam/ginacody/docs/Expectations-of-Originality-Feb14-2012.pdf)
2. Create a **README.txt** file, which will contain specific and complete instructions on how to run your program. If the instructions in your file do not work or incomplete, you will not be given the benefit of the doubt.
3. Create one zip file, containing all your code, the README.txt and the expectation of originality.
4. Name your zip file: ***A1_ StudentID.zip***

Note: please check your course Moodle webpage on how to submit the assignment and follow the instructions to submit your assignment. Wrong submission files and codes will not be accepted.

Have fun!!