

## HW3 Report

By Clarence Jiang (uni: yj2737)

This assignment focuses on converting each building on Columbia Campus into a human description that should effectively help a visitor identify any of their interested buildings.

### Step1:

In the first step, the inputs include a table that encodes each building with a number and a PGM campus image that stores the location of each building in a 2d map using the encoded number. That is to say, buildings like Pupin could be encoded as the number 9, so in the 2d map where Pupin stays, all pixels in the PGM campus will be written as 9. To read the encoded table, I used the typical “with” keyword in Python and stored the encoded relationship as a dictionary, called “encoded\_dict”. The encoded building number will serve as the key and the building name serves as the value. To read the PGM image, I used the “imageio” python package since it works well with the PGM format, and the 2d map is represented as a 2d numpy array, called “labeled\_map”. These 2 variables will serve as the main inputs for the following exploration.

Also, I will print out some important properties for each building. First, the building number and name could be accessed through the key-value pair of “encoded\_dict”. Second, the center of mass could be calculated by finding the indices of this building in the “labeled\_map” through the “np. where(labeled\_map == building number)” function and then taking the np. mean of row indices and column indices. The average of row indices and column indices serve as the y and x coordinates of the center mass respectively. Third, the total area in pixels could be extracted using the size of the result returned by “np. where” since this function will return a tuple of x and y indices. Fourth, to find the upper left and lower right corner of the Minimum Bounding Rectangle (MBR), I applied np. min and np. max on the row and column indices. For instance, the upper left coordinate is essentially (np. min(column indices), np. min(row indices)). Fifth, the diagonal length is calculated by the Euclidean distance between the upper left and lower right points using “np. sqrt()”. Finally, before calculating the overlap situation of each building, I stored all the building properties in a big dictionary for further usage, called “building\_num\_collective\_info\_dict”. The key is building numbers, and the values are the tuple of (building name, building area in pixels, diagonal length, (upper left, lower right), center of mass coordinate). I printed out the overlap information separately from the previous 5 properties because the overlap results are based on the previous 5 properties. Two buildings overlap if their minimum bounding rectangles overlap. I wrote a “cal\_overlap()” function, where I essentially used a nested for loop to check every pair of buildings. There is an overlap between 2 buildings if the lower right corner of the MBR of the first building is greater than or equal to the upper left of the second building and the upper left corner is less than or equal to the lower right corner of the second building, or if it is the other way around switching building 1 and 2. Remember the upper left and lower right points are stored in my “building\_num\_collective\_info\_dict”. I found “Uris” overlaps with “Mudd”; “Schermerhorn” overlaps with “Mudd”; “EarlHall” overlaps with

“Lewisohn”. Again, throughout this assignment, I realized a lot of information obtained in previous steps might be helpful for exploration in later steps, so I created tons of variables to store all intermediate results I found.

Building number: 9, building name: Pupin  
Center of mass: ( 76.15304878048781, 14.976219512195122 )  
Total area: 1640  
Upper left point coordinates: (39, 3 )  
Lower right point coordinates: (115, 27 )  
Diagonal length: 79.6994353806851

---

Building number: 19, building name: SchapiroCEPSR  
Center of mass: ( 143.0, 20.0 )  
Total area: 1435  
Upper left point coordinates: (123, 3 )  
Lower right point coordinates: (163, 37 )  
Diagonal length: 52.49761899362675

---

Building number: 28, building name: Mudd&EngTerrace&Fairchild&CS  
Center of mass: ( 223.24181101011834, 35.2795403875836 )  
Total area: 5831  
Upper left point coordinates: (166, 3 )  
Lower right point coordinates: (272, 86 )  
Diagonal length: 134.6291201783626

---

Building number: 38, building name: NorthwestCorner  
Center of mass: ( 16.0, 40.5 )  
Total area: 1998  
Upper left point coordinates: (3, 4 )  
Lower right point coordinates: (29, 77 )  
Diagonal length: 77.4919350642375

---

Building number: 47, building name: Uris  
Center of mass: ( 142.52563879714933, 99.46532244046584 )  
Total area: 5753  
Upper left point coordinates: (110, 48 )  
Lower right point coordinates: (175, 147 )  
Diagonal length: 118.43141475132347

---

Building number: 57, building name: Schermerhorn  
Center of mass: ( 233.4469445154692, 120.51009971874201 )  
Total area: 3911  
Upper left point coordinates: (181, 77 )  
Lower right point coordinates: (273, 147 )  
Diagonal length: 115.6027681329474

---

Building number: 66, building name: Chandler&Havemeyer  
Center of mass: ( 37.53086078051481, 119.54248546913922 )  
Total area: 3613  
Upper left point coordinates: (3, 81 )  
Lower right point coordinates: (80, 147 )

Diagonal length: 101.41498903022176

-----  
Building number: 76, building name: OldComputerCenter

Center of mass: ( 96.5, 136.0 )

Total area: 322

Upper left point coordinates: (90, 125 )

Lower right point coordinates: (103, 147 )

Diagonal length: 25.553864678361276

-----  
Building number: 85, building name: Avery

Center of mass: ( 204.04896907216494, 175.9819587628866 )

Total area: 1164

Upper left point coordinates: (191, 151 )

Lower right point coordinates: (215, 201 )

Diagonal length: 55.46169849544819

-----  
Building number: 94, building name: Fayerweather

Center of mass: ( 259.6150592216582, 176.0 )

Total area: 1182

Upper left point coordinates: (247, 151 )

Lower right point coordinates: (272, 201 )

Diagonal length: 55.90169943749474

-----  
Building number: 104, building name: Mathematics

Center of mass: ( 17.0, 182.0 )

Total area: 1191

Upper left point coordinates: (3, 158 )

Lower right point coordinates: (31, 206 )

Diagonal length: 55.569775957799216

-----  
Building number: 113, building name: LowLibrary

Center of mass: ( 135.0, 221.5 )

Total area: 3898

Upper left point coordinates: (101, 187 )

Lower right point coordinates: (169, 256 )

Diagonal length: 96.87620966986684

-----  
Building number: 123, building name: StPaulChapel

Center of mass: ( 226.5538178472861, 222.02115915363385 )

Total area: 1087

Upper left point coordinates: (201, 210 )

Lower right point coordinates: (251, 234 )

Diagonal length: 55.46169849544819

-----  
Building number: 132, building name: EarlHall

Center of mass: ( 49.599472990777336, 221.86297760210803 )

Total area: 759

Upper left point coordinates: (31, 211 )

Lower right point coordinates: (68, 233 )

Diagonal length: 43.04648650006177

-----  
Building number: 142, building name: Lewisohn9

Center of mass: ( 17.0, 259.0 )

Total area: 1307

Upper left point coordinates: (3, 233 )  
Lower right point coordinates: (31, 285 )  
Diagonal length: 59.0592922409336

-----  
Building number: 151, building name: Philosophy  
Center of mass: ( 258.3354838709677, 263.0 )  
Total area: 1085  
Upper left point coordinates: (245, 240 )  
Lower right point coordinates: (272, 286 )  
Diagonal length: 53.33854141237835

-----  
Building number: 161, building name: Buell  
Center of mass: ( 208.0, 253.5 )  
Total area: 340  
Upper left point coordinates: (196, 246 )  
Lower right point coordinates: (220, 261 )  
Diagonal length: 28.30194339616981

-----  
Building number: 170, building name: AlmaMater  
Center of mass: ( 136.0, 276.0 )  
Total area: 225  
Upper left point coordinates: (129, 269 )  
Lower right point coordinates: (143, 283 )  
Diagonal length: 19.79898987322333

-----  
Building number: 179, building name: Dodge  
Center of mass: ( 41.5, 301.09056603773587 )  
Total area: 1590  
Upper left point coordinates: (3, 289 )  
Lower right point coordinates: (80, 311 )  
Diagonal length: 80.0812087820857

-----  
Building number: 189, building name: Kent  
Center of mass: ( 233.0, 300.9 )  
Total area: 1470  
Upper left point coordinates: (194, 290 )  
Lower right point coordinates: (272, 310 )  
Diagonal length: 80.52328855678958

-----  
Building number: 198, building name: CollegeWalk  
Center of mass: ( 137.0, 322.5 )  
Total area: 4950  
Upper left point coordinates: (0, 314 )  
Lower right point coordinates: (274, 331 )  
Diagonal length: 274.5268657162719

-----  
Building number: 208, building name: Journalism&Furnald  
Center of mass: ( 30.58699808795411, 363.9441682600382 )  
Total area: 2615  
Upper left point coordinates: (4, 338 )  
Lower right point coordinates: (81, 414 )  
Diagonal length: 108.18964830333815

-----  
Building number: 217, building name: Hamilton&Hartley&Wallach&JohnJay

Center of mass: ( 240.33885567890692, 416.9472245943638 )

Total area: 5855

Upper left point coordinates: (191, 338 )

Lower right point coordinates: (270, 490 )

Diagonal length: 171.30382365843442

-----

Building number: 236, building name: Lerner

Center of mass: ( 38.5, 446.5 )

Total area: 2940

Upper left point coordinates: (4, 426 )

Lower right point coordinates: (73, 467 )

Diagonal length: 80.26207074328447

-----

Building number: 246, building name: ButlerLibrary

Center of mass: ( 132.0, 460.4146156758803 )

Total area: 5282

Upper left point coordinates: (85, 431 )

Lower right point coordinates: (179, 490 )

Diagonal length: 110.98198051936178

-----

Building number: 255, building name: Carman

Center of mass: ( 38.5, 479.5 )

Total area: 1540

Upper left point coordinates: (4, 469 )

Lower right point coordinates: (73, 490 )

Diagonal length: 72.12489168102785

-----

Here is the overlap information:

Building Mudd&EngTerrace&Fairchild&CS has overlapped building ['Uris', 'Schermerhorn']

Building Uris has overlapped building ['Mudd&EngTerrace&Fairchild&CS']

Building Schermerhorn has overlapped building ['Mudd&EngTerrace&Fairchild&CS']

Building EarlHall has overlapped building ['Lewisohn9']

Building Lewisohn9 has overlapped building ['EarlHall']

-----

## Step 2:

In Step 2, my system outputs a “what” description that contains 3 aspects of information: size, aspect ratio, and geometry. Size includes smallest/small/medium-size/large/largest; aspect ratio includes narrow/medium-width/wide, and geometry includes square/rectangular/I-shaped/C-shaped/L-shaped/asymmetric.

Again, retrieving the shape description only requires the same inputs in Step 1. I used “area in pixels” as the metric to measure size. For each building, I calculated size using 2 threshold values 1000 and 2000. I got these 2 threshold values by observing the area distribution of all buildings. Remember that I have a dictionary that stores collective information for each building and “area” is included in it. The whole list is contained as a comment in my code, but it is clear that there are 2 places of big value jump: from 759 to 1085 and from 1998 to 2615. In other words, the distribution reveals that areas mostly fall in the range of 1000 to 2000, so it is

natural that I call these buildings “medium” size, areas less than 1000 “small”, and areas greater than 2000 “big”. I think my approach is reasonable since normally “medium-size” should be the category that appears the most, and “small/big” is more like the 2 extremes. Thus, finding the 2 threshold ends of the “medium-size” naturally makes everything into any of the 3 categories.

As for the aspect ratio, I applied a similar approach. I first printed out all aspect ratios. The aspect ratio is evaluated by the ratio of the smaller one over the greater one between height and width. If the aspect ratio is close to 1, it means width and height are somewhat similar, so it is a “wide” building. If it’s close to 0, it means one of the “height/width” is significantly smaller than the other. I calculated height and width using the upper left and lower right coordinates stored in my “collective\_info” dictionary. Again, I printed out all aspect ratios and looked through the distribution. I first used 0.5 as a threshold; if the aspect ratio is less than or equal to 0.5, I defined the building as “narrow”. This value means that one of the sides is at least two times longer than the other side, and I define this as sufficiently “narrow”. Then I used “SchapiroCEPSR” and “LowLibrary” as 2 examples of wide buildings, which have 0.85 and 0.98 aspect ratios. However, I thought they are too perfect as examples of wide buildings, so I wanted to leave some room for the definition of “wide”, which drives me to choose 0.7 as another threshold. Then the medium-width building is defined with an aspect ratio between 0.5 and 0.7.

Finally, for geometry, my methodology is to check geometry in this order: Square, Rectangular, I-shaped, L-shaped, C-shaped, and asymmetric. In other words, if any building passes a test, then there is no need to go to the following stages. For example, if the system found a building to be Square, it will not go through the checking process of rectangular. First, for a “square/rectangular/I-shaped” building, its “area in pixels” will be really close to the area of the MBR. By contrast, an “L-shaped/C-shaped/asymmetric” building will have a lot of little “holes” in the MBR. Thus, I calculated this ratio of “area in pixel/ area of MBR” using the area, upper left, and lower right points in my collective dictionary. Then I performed a similar process of printing out the distribution of this ratio and find most buildings that look “square/rectangular/I-shaped” indeed have a high ratio above 0.83, such as “Schapiro” and “Uris”, so I used 0.83 as a threshold.

Then, to further distinguish “I-shaped”, since “I-shaped” is more different than the other two, I designed a complicated algorithm. “I” has two horizontal bars at the top and bottom and a skinny vertical bar in the middle. Imagine the skinny vertical bar of the letter “I” as a rectangle. I will find the upper left and lower right points of this rectangle. Then the difference between the x-coordinates of these 2 points is the width of this skinny body of “I”. Then, I get the x-coordinate of the upper left point and I will check if the x-coordinates are the same for every point with indices “upper left + k \* width (where k from 1 to the height of the skinny body)”. Also, I will do a similar thing for the lower right point. If there is even one point that has a different x-coordinate, I strictly do not call it an “I”, but it works well for Columbia Campus. Essentially, I am testing if the skinny vertical bar structure of “the letter” exists and if there is only one. This whole testing function is called “isIShape()” in my system.

Then, to distinguish a square and a rectangular, I will simply use the aspect ratio obtained previously. If a building is “wide”, it is called a square; otherwise, it is a rectangular building. Thus, the overall logic would be “area pixel/area MBR > 0.83 AND aspect ratio is not wide AND is not I-shape → Rectangular”, “area pixel/area MBR > 0.83 AND aspect ratio is wide AND is not I-shape → Square”, and finally “area pixel/area MBR > 0.83 AND is I-shape → I-shaped”.

If neither of the 3 conditions is satisfied, the system will run into a function called “isL\_or\_C\_or\_Asymmetric\_Shape()”. L-shaped buildings look the same if reflected according to one of the 2 diagonals. One is the diagonal from upper left to lower right, which could be obtained through `np.transpose()`, while the other is from upper right to lower left, tested by “`np.rot90(np.fliplr(sub_array), -1)`”. After getting the reflected building shape, I will compare it with the original building shape and count the overlapped pixels. If the ratio of this count and the total number of pixels, called the “reflection score”, is greater than 0.8, this building is L-shaped. I got this number by checking the distribution of all buildings. In fact, the “Journalism” building is almost a perfect “L” and it has a high score above 0.9, but the other 2 buildings look L-shaped, “Chandler” and “Schermohorn” has an even lower score than “Mudd”. Thus, I know having a low ratio does not mean it is not “L-shaped”. Thus, I further calculated the max length of the continuous horizontal black gap (black color is the area without building pixels, represented as 0 in the 2d numpy array). If the ratio between this length and the width of the MBR is greater than 0.6 but less than or equal to 0.75, it is also an “L-shaped”. It makes sense because the upper right area of the letter “L” will have some long continuous horizontal black parts compared to an asymmetric building like “Mudd”, but at the same time, it should not be too high (> 0.75) because a “C-shaped” building like “Hamilton” will also have a continuous black gap with long length. In this case, I distinguish both L-shaped and C-shaped. Either “reflection score > 0.8” or “reflection score < 0.8 and 0.75 ≥ black line width/width > 0.6” gives an “L-shaped”. Then “reflection score < 0.8 and black line width/width > 0.75” gives a “C-shaped”. The building left will be asymmetric. Again the 0.6, 0.8, and 0.75 are chosen as thresholds because of distribution observation since I want my system to successfully recognize “Chandler” and “Schermohorn” as “L-shaped” but not misclassify the C-shaped “Hamilton” or the asymmetric “Mudd” as “L-shaped”. All these threshold values are well-chosen to make this work.

The confusion part is easy because I also stored the 3 aspects in a dictionary, so I could just run a nested for loop that checks if any pair has the same 3 aspects. In fact, there is a lot of confusion. For example, Pupin can be confused with ['NorthwestCorner', 'Avery', 'Fayerweather', 'StPaulChapel', 'Dodge', 'Kent', 'Carman']. For the minimization process, I followed 2 general principles. First, any of the 3 aspects could be dropped if dropping it will make the “what” description unique. For instance, there is only 1 “asymmetric” building, so dropping size and aspect ratio is acceptable, as this building cannot be confused with any other building. Second, dropping any of the 3 aspects makes the “what” description all the same. For instance, all the 3 “I-shaped” buildings have the same size and aspect ratio, so keeping them does not help us further identify buildings as I-shaped, and we could just call them “I-shaped” buildings.

In addition, I stored the minimized description of this step in a dictionary called “step2\_result”; it has the building name as the key and the string description as values. This dictionary will be helpful in step 5 when we put all things together.

building name: Pupin, size: medium-size, aspect\_ratio: narrow, geometry: Rectangular  
building name: SchapiroCEPSR, size: medium-size, aspect\_ratio: wide, geometry: Square  
building name: Mudd&EngTerrace&Fairchild&CS, size: large, aspect\_ratio: wide, geometry: asymmetric  
building name: NorthwestCorner, size: medium-size, aspect\_ratio: narrow, geometry: Rectangular  
building name: Uris, size: large, aspect\_ratio: medium-width, geometry: Rectangular  
building name: Schermerhorn, size: large, aspect\_ratio: wide, geometry: L-shaped  
building name: Chandler&Havemeyer, size: large, aspect\_ratio: wide, geometry: L-shaped  
building name: OldComputerCenter, size: small, aspect\_ratio: medium-width, geometry: Rectangular  
building name: Avery, size: medium-size, aspect\_ratio: narrow, geometry: Rectangular  
building name: Fayerweather, size: medium-size, aspect\_ratio: narrow, geometry: Rectangular  
building name: Mathematics, size: medium-size, aspect\_ratio: medium-width, geometry: I-shaped  
building name: LowLibrary, size: large, aspect\_ratio: wide, geometry: Square  
building name: StPaulChapel, size: medium-size, aspect\_ratio: narrow, geometry: Rectangular  
building name: EarlHall, size: small, aspect\_ratio: medium-width, geometry: Rectangular  
building name: Lewisohn9, size: medium-size, aspect\_ratio: medium-width, geometry: I-shaped  
building name: Philosophy, size: medium-size, aspect\_ratio: medium-width, geometry: I-shaped  
building name: Buell, size: small, aspect\_ratio: medium-width, geometry: Rectangular  
building name: AlmaMater, size: smallest, aspect\_ratio: wide, geometry: Square  
building name: Dodge, size: medium-size, aspect\_ratio: narrow, geometry: Rectangular  
building name: Kent, size: medium-size, aspect\_ratio: narrow, geometry: Rectangular  
building name: CollegeWalk, size: large, aspect\_ratio: narrow, geometry: Rectangular  
building name: Journalism&Furnald, size: large, aspect\_ratio: wide, geometry: L-shaped  
building name: Hamilton&Hartley&Wallach&JohnJay, size: largest, aspect\_ratio: medium-width, geometry: C-shaped  
building name: Lerner, size: large, aspect\_ratio: medium-width, geometry: Rectangular  
building name: ButlerLibrary, size: large, aspect\_ratio: medium-width, geometry: Rectangular  
building name: Carman, size: medium-size, aspect\_ratio: narrow, geometry: Rectangular

-----  
Pupin can be confused with ['NorthwestCorner', 'Avery', 'Fayerweather', 'StPaulChapel', 'Dodge', 'Kent', 'Carman']  
NorthwestCorner can be confused with ['Pupin', 'Avery', 'Fayerweather', 'StPaulChapel', 'Dodge', 'Kent', 'Carman']  
Uris can be confused with ['Lerner', 'ButlerLibrary']  
Schermerhorn can be confused with ['Chandler&Havemeyer', 'Journalism&Furnald']  
Chandler&Havemeyer can be confused with ['Schermerhorn', 'Journalism&Furnald']  
OldComputerCenter can be confused with ['EarlHall', 'Buell']  
Avery can be confused with ['Pupin', 'NorthwestCorner', 'Fayerweather', 'StPaulChapel', 'Dodge', 'Kent', 'Carman']  
Fayerweather can be confused with ['Pupin', 'NorthwestCorner', 'Avery', 'StPaulChapel', 'Dodge', 'Kent', 'Carman']  
Mathematics can be confused with ['Lewisohn9', 'Philosophy']  
StPaulChapel can be confused with ['Pupin', 'NorthwestCorner', 'Avery', 'Fayerweather', 'Dodge', 'Kent', 'Carman']  
EarlHall can be confused with ['OldComputerCenter', 'Buell']  
Lewisohn9 can be confused with ['Mathematics', 'Philosophy']  
Philosophy can be confused with ['Mathematics', 'Lewisohn9']  
Buell can be confused with ['OldComputerCenter', 'EarlHall']  
Dodge can be confused with ['Pupin', 'NorthwestCorner', 'Avery', 'Fayerweather', 'StPaulChapel', 'Kent', 'Carman']  
Kent can be confused with ['Pupin', 'NorthwestCorner', 'Avery', 'Fayerweather', 'StPaulChapel', 'Dodge', 'Carman']



Journalism&Furnald can be confused with ['Schermerhorn', 'Chandler&Havemeyer']  
Lerner can be confused with ['Uris', 'ButlerLibrary']  
ButlerLibrary can be confused with ['Uris', 'Lerner']  
Carman can be confused with ['Pupin', 'NorthwestCorner', 'Avery', 'Fayerweather', 'StPaulChapel', 'Dodge', 'Kent']  
-----

Minimization description Step 2:

Pupin is a medium-size Rectangular building  
SchapiroCEPSR is a medium-size Square building  
Mudd&EngTerrace&Fairchild&CS is a large asymmetric building  
NorthwestCorner is a medium-size Rectangular building  
Uris is a large medium-width building  
Schermerhorn is a L-shaped building  
Chandler&Havemeyer is a L-shaped building  
OldComputerCenter is a small building  
Avery is a medium-size Rectangular building  
Fayerweather is a medium-size Rectangular building  
Mathematics is a I-shaped building  
LowLibrary is a large Square building  
StPaulChapel is a medium-size Rectangular building  
EarlHall is a small building  
Lewisohn9 is a I-shaped building  
Philosophy is a I-shaped building  
Buell is a small building  
AlmaMater is a smallest Square building  
Dodge is a medium-size Rectangular building  
Kent is a medium-size Rectangular building  
CollegeWalk is a large Rectangular building  
Journalism&Furnald is a L-shaped building  
Hamilton&Hartley&Wallach&JohnJay is a largest C-shaped building  
Lerner is a large medium-width building  
ButlerLibrary is a large medium-width building  
Carman is a medium-size Rectangular building

Step 3:

In Step 3, I performed a similar process as in Step 2. I need to convert the image location on the labeled map into a “where” description that involves 3 aspects: verticality, horizontality, and orientation. Verticality involves 5 options: uppermost, upper, mid-height, lower, and lowermost. Horizontality includes leftmost, left, mid-width, right, and rightmost. Orientation contains vertically-oriented, non-oriented, and horizontally-oriented.

As for horizontality and verticality, I simply divided the whole 495 \* 275 map into 5 \* 5 areas. The width and height are just equally divided into 5 sections, which correspond to the labels from leftmost to rightmost or uppermost to lowermost correspondingly. I just accessed the center of mass coordinates in my collective information dictionary and used the x and y coordinates of the center of mass to assign horizontality and verticality labels. It worked pretty

well since Columbia Campus does not have too many clustered blocks. As for orientation, I first directly retrieved the aspect ratio of each building from my “step2\_result” dictionary. If it’s a wide building, it indicates it is likely a square, so it is not oriented. Then for those narrow buildings, I calculated the width and height, and if the height is greater than the width, it is a vertically-oriented building, and vice versa for horizontally-oriented buildings. Again, I stored this “where” information in a dictionary where the building name and a dictionary of {horizontality, verticality, orientation} serve as key and value respectively.

In Step 3, to find confusion, I applied the same methodology and just replaced the dictionary in Step 2 with my Step 3 dictionary. I discovered that “LowLibrary’s location could be confused with “AlmaMater” and “Lerner” could be confused with “Carman”. As for the minimization step, I just applied the same idea in Step 2. I discovered that the Campus is designed in a way to avoid clustered buildings. Thus, if imagining the Campus being divided into 5 \* 5 little areas, there is likely only 1 building in each little area based on the center of mass, so you essentially can remove most of the orientation information. There are definitely exceptions like in the mid-height on the right, there are both “St.Paul” and “Avery” which can only be distinguished by their orientation. Thus, I carefully designed different cases with a lot of if-else statements (details included in my code implementation). Likewise, the minimized location description is stored in a “step3\_result” dictionary.

building name: Pupin, horizontality: left, verticality: uppermost, orientation: horizontally-oriented  
building name: SchapiroCEPSR, horizontality: mid-width, verticality: uppermost, orientation: non-oriented  
building name: Mudd&EngTerrace&Fairchild&CS, horizontality: rightmost, verticality: uppermost, orientation: non-oriented  
building name: NorthwestCorner, horizontality: leftmost, verticality: uppermost, orientation: vertically-oriented  
building name: Uris, horizontality: mid-width, verticality: upper, orientation: vertically-oriented  
building name: Schermerhorn, horizontality: rightmost, verticality: upper, orientation: non-oriented  
building name: Chandler&Havemeyer, horizontality: leftmost, verticality: upper, orientation: non-oriented  
building name: OldComputerCenter, horizontality: left, verticality: upper, orientation: vertically-oriented  
building name: Avery, horizontality: right, verticality: upper, orientation: vertically-oriented  
building name: Fayerweather, horizontality: rightmost, verticality: upper, orientation: vertically-oriented  
building name: Mathematics, horizontality: leftmost, verticality: upper, orientation: vertically-oriented  
building name: LowLibrary, horizontality: mid-width, verticality: mid-height, orientation: non-oriented  
building name: StPaulChapel, horizontality: rightmost, verticality: mid-height, orientation: horizontally-oriented  
building name: EarlHall, horizontality: leftmost, verticality: mid-height, orientation: horizontally-oriented  
building name: Lewisohn9, horizontality: leftmost, verticality: mid-height, orientation: vertically-oriented  
building name: Philosophy, horizontality: rightmost, verticality: mid-height, orientation: vertically-oriented  
building name: Buell, horizontality: right, verticality: mid-height, orientation: horizontally-oriented  
building name: AlmaMater, horizontality: mid-width, verticality: mid-height, orientation: non-oriented  
building name: Dodge, horizontality: leftmost, verticality: lower, orientation: horizontally-oriented  
building name: Kent, horizontality: rightmost, verticality: lower, orientation: horizontally-oriented  
building name: CollegeWalk, horizontality: mid-width, verticality: lower, orientation: horizontally-oriented  
building name: Journalism&Furnald, horizontality: leftmost, verticality: lower, orientation: non-oriented  
building name: Hamilton&Hartley&Wallach&JohnJay, horizontality: rightmost, verticality: lowermost, orientation: vertically-oriented  
building name: Lerner, horizontality: leftmost, verticality: lowermost, orientation: horizontally-oriented

building name: ButlerLibrary, horizontality: mid-width, verticality: lowermost, orientation: horizontally-oriented  
building name: Carman, horizontality: leftmost, verticality: lowermost, orientation: horizontally-oriented

-----  
LowLibrary can be confused with ['AlmaMater']  
AlmaMater can be confused with ['LowLibrary']  
Lerner can be confused with ['Carman']  
Carman can be confused with ['Lerner']  
-----

Minimization description Step 3:

Pupin is at uppermost and left in Columbia Campus  
SchapiroCEPSR is at uppermost and mid-width in Columbia Campus  
Mudd&EngTerrace&Fairchild&CS is at uppermost and rightmost in Columbia Campus  
NorthwestCorner is at uppermost and leftmost in Columbia Campus  
Uris is at mid-width and upper in Columbia Campus  
Schermerhorn is a building at rightmost and upper and it's non-oriented in Columbia Campus  
Chandler&Havemeyer is a building at leftmost and upper and it's non-oriented in Columbia Campus  
OldComputerCenter is at left and it's vertically-oriented in Columbia Campus  
Avery is at right and it's vertically-oriented in Columbia Campus  
Fayerweather is a building at rightmost and upper and it's vertically-oriented in Columbia Campus  
Mathematics is a building at leftmost and upper and it's vertically-oriented in Columbia Campus  
LowLibrary is at mid-height and mid-width in Columbia Campus  
StPaulChapel is a building at rightmost and mid-height and it's horizontally-oriented in Columbia Campus  
EarlHall is a building at leftmost and mid-height and it's horizontally-oriented in Columbia Campus  
Lewisohn9 is a building at leftmost and mid-height and it's vertically-oriented in Columbia Campus  
Philosophy is a building at rightmost and mid-height and it's vertically-oriented in Columbia Campus  
Buell is at right and it's horizontally-oriented in Columbia Campus  
AlmaMater is at mid-height and mid-width in Columbia Campus  
Dodge is a building at leftmost and lower and it's horizontally-oriented in Columbia Campus  
Kent is a building at rightmost and lower and it's horizontally-oriented in Columbia Campus  
CollegeWalk is at mid-width and lower in Columbia Campus  
Journalism&Furnald is a building at leftmost and lower and it's non-oriented in Columbia Campus  
Hamilton&Hartley&Wallach&JohnJay is at rightmost and lowermost in Columbia Campus  
Lerner is at leftmost and lowermost in Columbia Campus  
ButlerLibrary is at mid-width and lowermost in Columbia Campus  
Carman is at leftmost and lowermost in Columbia Campus

Step 4:

In Step 4, I created a 26 by 26 2d numpy array. Each pair of buildings is represented as row and column indices; the row represents the target and the column represents the source. So position (0, 1) is interpreted as we are interested in the first building and plan to use the second building as a source to locate the first building. The value of this matrix could only be 0 or 1, and 1 means “near”.

To evaluate if one building is near to the other, I first checked their horizontality and verticality. If both of horizontality and verticality of 2 buildings are off by two levels or more (for example, “left” and “leftmost” is off by 1 level, “leftmost” and “mid-width” are off by 2

levels), there are no need to do any calculation, and I will just assign a 0. I do not use an “or” but use an “and” relationship here, because using a less strict exclusion criterion could avoid accidentally removing some actual near cases in a diagonal direction. Then for the rest of the pair, I calculated the Euclidean distance between their center of masses and took the ratio of this distance over the smaller area of the pair, called “distance\_size\_ratio”, since dimension also plays a role. Also, my system does not make this symmetric. The fact that Building A is near Building B does not imply the reverse. For instance, you can say Almamater is near Low Library since Low Library is such a big building, so it makes sense. However, it will sound weird if you say Low Library is near Almamater because Almamater is such a small thing that does not serve well as a source. As a result, I designed my logic using 3 variables: the distance, the distance\_size ratio, and the area of the pair of buildings. There are also 3 cases. First, if the area of the target building is less than  $\frac{1}{5}$  of that of the source building, the distance needs to be less than or equal to 80 for them to be called “near”. This is the case when the target building is really small. The threshold value needs to be smaller as well. This threshold of 80 is selected to help guarantee that small target buildings like “EarlHall” and “OldComputerCenter” have a source. Second, if the distance is less than or equal to 120 and the distance\_size ratio is less than or equal to 0.06, this pair is also considered “near”. These 2 thresholds are based on the “Pupin” and “Schapiro” pair which represents the most standard “near” relationship. This is the case for normal building pairs. Third, if the area of the source building is greater than or equal to  $\frac{1}{5}$  of the target building, it means the source building is not that small, so the system will just use the distance as the sole metric. If it is greater than or equal to 60, it will assign a 1 to the building pair. This condition is to satisfy those cases that the distance\_size\_ratio is not small enough because the area is not big enough but not as small as typical small buildings like “OldComputerCenter”, such as “St.Paul”. The rest of the cases are not considered as near. Finally, we could easily use the binary matrix and iterate row by row to print the result for each building serving as the target and column by column for each building serving as the source.

To solve the confusion part of this step, I utilized the lambda function and max/min function to calculate the source/target element that has the most/least target/source buildings. For minimization, I iterated through each building and also utilized a similar lambda function and max() to find the source that appears the most in other buildings. Besides, I also used 2 dictionaries to store the source and target buildings for each building that acts as a target/source and another dictionary to store the landmark choice for each building.

Pupin as target, it is near to ['SchapiroCEPSR', 'NorthwestCorner']

SchapiroCEPSR as target, it is near to ['Pupin', 'Mudd&EngTerrace&Fairchild&CS', 'Uris']

Mudd&EngTerrace&Fairchild&CS as target, it is near to ['SchapiroCEPSR', 'Uris', 'Schermerhorn']

NorthwestCorner as target, it is near to ['Pupin', 'Chandler&Havemeyer']

Uris as target, it is near to ['SchapiroCEPSR', 'Mudd&EngTerrace&Fairchild&CS', 'Schermerhorn', 'Chandler&Havemeyer']

Schermerhorn as target, it is near to ['Mudd&EngTerrace&Fairchild&CS', 'Uris', 'Avery', 'Fayerweather']

Chandler&Havemeyer as target, it is near to ['NorthwestCorner', 'Uris', 'Mathematics']

OldComputerCenter as target, it is near to ['Uris', 'Chandler&Havemeyer']

Avery as target, it is near to ['Schermerhorn', 'Fayerweather', 'StPaulChapel']  
 Fayerweather as target, it is near to ['Schermerhorn', 'Avery', 'StPaulChapel']  
 Mathematics as target, it is near to ['Chandler&Havemeyer', 'EarlHall']  
 LowLibrary as target, it is near to ['CollegeWalk']  
 StPaulChapel as target, it is near to ['Avery', 'Fayerweather', 'Philosophy', 'Buell']  
 EarlHall as target, it is near to ['Mathematics', 'Lewisohn9']  
 Lewisohn9 as target, it is near to ['EarlHall', 'Dodge']  
 Philosophy as target, it is near to ['StPaulChapel', 'Buell', 'Kent']  
 Buell as target, it is near to ['LowLibrary', 'StPaulChapel', 'Philosophy', 'Kent']  
 AlmaMater as target, it is near to ['LowLibrary', 'CollegeWalk']  
 Dodge as target, it is near to ['Lewisohn9', 'Journalism&Furnald']  
 Kent as target, it is near to ['Philosophy', 'Buell']  
 CollegeWalk as target, it is near to ['LowLibrary', 'Journalism&Furnald']  
 Journalism&Furnald as target, it is near to ['Dodge', 'CollegeWalk', 'Lerner']  
 Hamilton&Hartley&Wallach&JohnJay as target, it is near to ['ButlerLibrary']  
 Lerner as target, it is near to ['Journalism&Furnald', 'ButlerLibrary', 'Carman']  
 ButlerLibrary as target, it is near to ['Hamilton&Hartley&Wallach&JohnJay', 'Lerner']  
 Carman as target, it is near to ['Lerner']

-----  
 Pupin as source, source buildings close to ['SchapiroCEPSR', 'NorthwestCorner']  
 SchapiroCEPSR as source, source buildings close to ['Pupin', 'Mudd&EngTerrace&Fairchild&CS', 'Uris']  
 Mudd&EngTerrace&Fairchild&CS as source, source buildings close to ['SchapiroCEPSR', 'Uris', 'Schermerhorn']  
 NorthwestCorner as source, source buildings close to ['Pupin', 'Chandler&Havemeyer']  
 Uris as source, source buildings close to ['SchapiroCEPSR', 'Mudd&EngTerrace&Fairchild&CS', 'Schermerhorn', 'Chandler&Havemeyer', 'OldComputerCenter']  
 Schermerhorn as source, source buildings close to ['Mudd&EngTerrace&Fairchild&CS', 'Uris', 'Avery', 'Fayerweather']  
 Chandler&Havemeyer as source, source buildings close to ['NorthwestCorner', 'Uris', 'OldComputerCenter', 'Mathematics']  
 OldComputerCenter as source, source buildings close to []  
 Avery as source, source buildings close to ['Schermerhorn', 'Fayerweather', 'StPaulChapel']  
 Fayerweather as source, source buildings close to ['Schermerhorn', 'Avery', 'StPaulChapel']  
 Mathematics as source, source buildings close to ['Chandler&Havemeyer', 'EarlHall']  
 LowLibrary as source, source buildings close to ['Buell', 'AlmaMater', 'CollegeWalk']  
 StPaulChapel as source, source buildings close to ['Avery', 'Fayerweather', 'Philosophy', 'Buell']  
 EarlHall as source, source buildings close to ['Mathematics', 'Lewisohn9']  
 Lewisohn9 as source, source buildings close to ['EarlHall', 'Dodge']  
 Philosophy as source, source buildings close to ['StPaulChapel', 'Buell', 'Kent']  
 Buell as source, source buildings close to ['StPaulChapel', 'Philosophy', 'Kent']  
 AlmaMater as source, source buildings close to []  
 Dodge as source, source buildings close to ['Lewisohn9', 'Journalism&Furnald']  
 Kent as source, source buildings close to ['Philosophy', 'Buell']  
 CollegeWalk as source, source buildings close to ['LowLibrary', 'AlmaMater', 'Journalism&Furnald']  
 Journalism&Furnald as source, source buildings close to ['Dodge', 'CollegeWalk', 'Lerner']  
 Hamilton&Hartley&Wallach&JohnJay as source, source buildings close to ['ButlerLibrary']  
 Lerner as source, source buildings close to ['Journalism&Furnald', 'ButlerLibrary', 'Carman']  
 ButlerLibrary as source, source buildings close to ['Hamilton&Hartley&Wallach&JohnJay', 'Lerner']  
 Carman as source, source buildings close to ['Lerner']  
 -----

Uris as source, has the most targets ['SchapiroCEPSR', 'Mudd&EngTerrace&Fairchild&CS', 'Schermerhorn', 'Chandler&Havemeyer', 'OldComputerCenter']

OldComputerCenter as source, has the least targets []

Uris as target, has the most sources ['SchapiroCEPSR', 'Mudd&EngTerrace&Fairchild&CS', 'Schermerhorn', 'Chandler&Havemeyer']

LowLibrary as target, has the least sources ['CollegeWalk']

-----

#### Minimization Step 4

Pupin as a target, the most-often-used source is SchapiroCEPSR

SchapiroCEPSR as a target, the most-often-used source is Uris

Mudd&EngTerrace&Fairchild&CS as a target, the most-often-used source is Uris

NorthwestCorner as a target, the most-often-used source is Chandler&Havemeyer

Uris as a target, the most-often-used source is Schermerhorn

Schermerhorn as a target, the most-often-used source is Uris

Chandler&Havemeyer as a target, the most-often-used source is Uris

OldComputerCenter as a target, the most-often-used source is Uris

Avery as a target, the most-often-used source is Schermerhorn

Fayerweather as a target, the most-often-used source is Schermerhorn

Mathematics as a target, the most-often-used source is Chandler&Havemeyer

LowLibrary as a target, the most-often-used source is CollegeWalk

StPaulChapel as a target, the most-often-used source is Avery

EarlHall as a target, the most-often-used source is Mathematics

Lewisohn9 as a target, the most-often-used source is EarlHall

Philosophy as a target, the most-often-used source is StPaulChapel

Buell as a target, the most-often-used source is StPaulChapel

AlmaMater as a target, the most-often-used source is LowLibrary

Dodge as a target, the most-often-used source is Journalism&Furnald

Kent as a target, the most-often-used source is Philosophy

CollegeWalk as a target, the most-often-used source is LowLibrary

Journalism&Furnald as a target, the most-often-used source is CollegeWalk

Hamilton&Hartley&Wallach&JohnJay as a target, the most-often-used source is ButlerLibrary

Lerner as a target, the most-often-used source is Journalism&Furnald

ButlerLibrary as a target, the most-often-used source is Lerner

Carman as a target, the most-often-used source is Lerner

#### Step 5:

In Step 5, I try to group all results from previous steps to develop a final human description of a building. First, I will extract the landmark choice for each building from Step 4, then I will simply concatenate the “what\_T, where\_T, what\_S, where\_S” result from my “step2\_result” and “step3\_result” dictionaries.

I discovered that my system will not have any confusion when concatenating the “what” and “where” for a target building. This is intuitive. Remember that in Step 3, I have only 2 pairs of confusion cases if I just applied the “where” information and these 2 pairs of buildings exactly have different “what” information. So my system will have no confusion even without the source “nearTo” information.

However, there is definitely more space for minimization. To some extent, my target "what + where" is already enough, so I could reduce some less important features in the source building. I took 3 main principles. First, Geometry is the most unique "what" property of a building. If the source building is "I", "L", "C" or "Square", I will remove all the other information about the building. Again this is because my system has really strict algorithms to generate these 4 shapes due to the strict thresholds, so the source building, which theoretically is already extraneous, will be fine with just its geometry. The second principle is that there is only one asymmetric building, so if the target is asymmetric, I will just describe it as an asymmetric building. Finally, the last principle is for the last geometry: Rectangular. Unlike the other 5 shapes, which I have discussed in the first and second principles, "Rectangular" is not a geometry that has many characteristics. Thus, to further minimize description, I should not think from using the perspective of geometry anymore, so I just removed the location information of the source building (horizontality and vertically). It is not that useful because the source is always near the target, so visitors could just look around. That is to say, I already have a precise description of the "where" of my target information; recall that there are only 2 pairs of buildings that have confused "where". Thus, removing the "where" of the source building will not affect much because of both the good accuracy of the "where" of the target building and my strict definition of "near".

Finally, for finding the shortest and longest unambiguous description, I put all the keywords in a list and count the total occurrences. The longest description has 7 keywords for "StPaulChapel", while the shortest has 5 for a total of 12 buildings (listed in code, e.g. Pupin and Uris). I think this is a good result because I intentionally set up my ideal threshold to be 5 as the lowest. As Professor points out, people like redundant information. I incorporated the concern that less information might negatively affect a person's locating process. From a real-life perspective, I do not want to throw away too many keywords, and I believe giving more is always better than not giving enough.

Total description without minimization:

a medium-size Rectangular building at uppermost and left near a medium-size Square building at uppermost and mid-width  
a medium-size Square building at uppermost and mid-width near a large medium-width building at mid-width and upper  
a large asymmetric building at uppermost and rightmost near a large medium-width building at mid-width and upper  
a medium-size Rectangular building at uppermost and leftmost near a L-shaped building non-oriented at leftmost and upper  
a large medium-width building at mid-width and upper near a L-shaped building non-oriented at rightmost and upper  
a L-shaped building non-oriented at rightmost and upper near a large medium-width building at mid-width and upper  
a L-shaped building non-oriented at leftmost and upper near a large medium-width building at mid-width and upper  
a small building vertically-oriented at left near a large medium-width building at mid-width and upper  
a medium-size Rectangular building vertically-oriented at right near a L-shaped building non-oriented at rightmost and upper  
a medium-size Rectangular building vertically-oriented at rightmost and upper near a L-shaped building non-oriented at rightmost and upper  
a I-shaped building vertically-oriented at leftmost and upper near a L-shaped building non-oriented at leftmost and upper  
a large Square building at mid-height and mid-width near a large Rectangular building at mid-width and lower  
a medium-size Rectangular building horizontally-oriented at rightmost and mid-height near a medium-size Rectangular building vertically-oriented at right  
a small building horizontally-oriented at leftmost and mid-height near a I-shaped building vertically-oriented at leftmost and upper

a I-shaped building vertically-oriented at leftmost and mid-height near a small building horizontally-oriented at leftmost and mid-height  
a I-shaped building vertically-oriented at rightmost and mid-height near a medium-size Rectangular building horizontally-oriented at rightmost and mid-height  
a small building horizontally-oriented at right near a medium-size Rectangular building horizontally-oriented at rightmost and mid-height  
a smallest Square building at mid-height and mid-width near a large Square building at mid-height and mid-width  
a medium-size Rectangular building horizontally-oriented at leftmost and lower near a L-shaped building non-oriented at leftmost and lower  
a medium-size Rectangular building horizontally-oriented at rightmost and lower near a I-shaped building vertically-oriented at rightmost and mid-height  
a large Rectangular building at mid-width and lower near a large Square building at mid-height and mid-width  
a L-shaped building non-oriented at leftmost and lower near a large Rectangular building at mid-width and lower  
a largest C-shaped building at rightmost and lowermost near a large medium-width building at mid-width and lowermost  
a large medium-width building at leftmost and lowermost near a L-shaped building non-oriented at leftmost and lower  
a large medium-width building at mid-width and lowermost near a large medium-width building at leftmost and lowermost  
a medium-size Rectangular building at leftmost and lowermost near a large medium-width building at leftmost and lowermost

-----  
Confusion:

Ok, there is no confusion  
-----

Minimization Step 5:

a medium-size Rectangular building at uppermost and left near a Square building  
a medium-size Square building at uppermost and mid-width near a large medium-width building  
a asymmetric building near a large medium-width building at mid-width and upper  
a medium-size Rectangular building at uppermost and leftmost near a L-shaped building  
a large medium-width building at mid-width and upper near a L-shaped building  
a L-shaped building non-oriented at rightmost and upper near a large medium-width building  
a L-shaped building non-oriented at leftmost and upper near a large medium-width building  
a small building vertically-oriented at left near a large medium-width building  
a medium-size Rectangular building vertically-oriented at right near a L-shaped building  
a medium-size Rectangular building vertically-oriented at rightmost and upper near a L-shaped building  
a I-shaped building vertically-oriented at leftmost and upper near a L-shaped building  
a large Square building at mid-height and mid-width near a large Rectangular building  
a medium-size Rectangular building horizontally-oriented at rightmost and mid-height near a medium-size building vertically-oriented  
a small building horizontally-oriented at leftmost and mid-height near a I-shaped building  
a I-shaped building vertically-oriented at leftmost and mid-height near a small building horizontally-oriented  
a I-shaped building vertically-oriented at rightmost and mid-height near a medium-size building horizontally-oriented  
a small building horizontally-oriented at right near a medium-size building horizontally-oriented  
a smallest Square building at mid-height and mid-width near a Square building  
a medium-size Rectangular building horizontally-oriented at leftmost and lower near a L-shaped building  
a medium-size Rectangular building horizontally-oriented at rightmost and lower near a I-shaped building  
a large Rectangular building at mid-width and lower near a Square building  
a L-shaped building non-oriented at leftmost and lower near a large Rectangular building  
a largest C-shaped building at rightmost and lowermost near a large medium-width building  
a large medium-width building at leftmost and lowermost near a L-shaped building  
a large medium-width building at mid-width and lowermost near a large medium-width building  
a medium-size Rectangular building at leftmost and lowermost near a large medium-width building

-----  
Longest unambiguous description has count 7

The building is StPaulChapel

Shortest unambiguous description has count 5

The building is Pupin



The building is Mudd&EngTerrace&Fairchild&CS

The building is NorthwestCorner

The building is Uris

The building is OldComputerCenter

The building is Avery

The building is Mathematics

The building is EarlHall

The building is Buell

The building is AlmaMater

The building is CollegeWalk

The building is Lerner