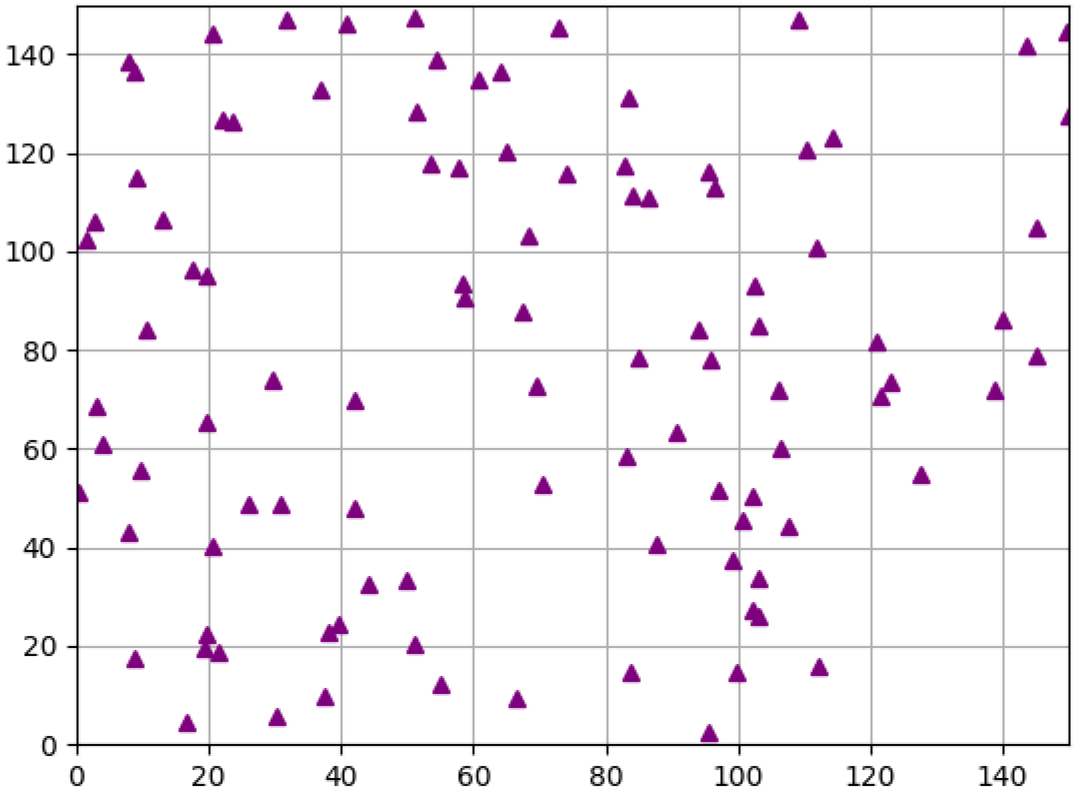
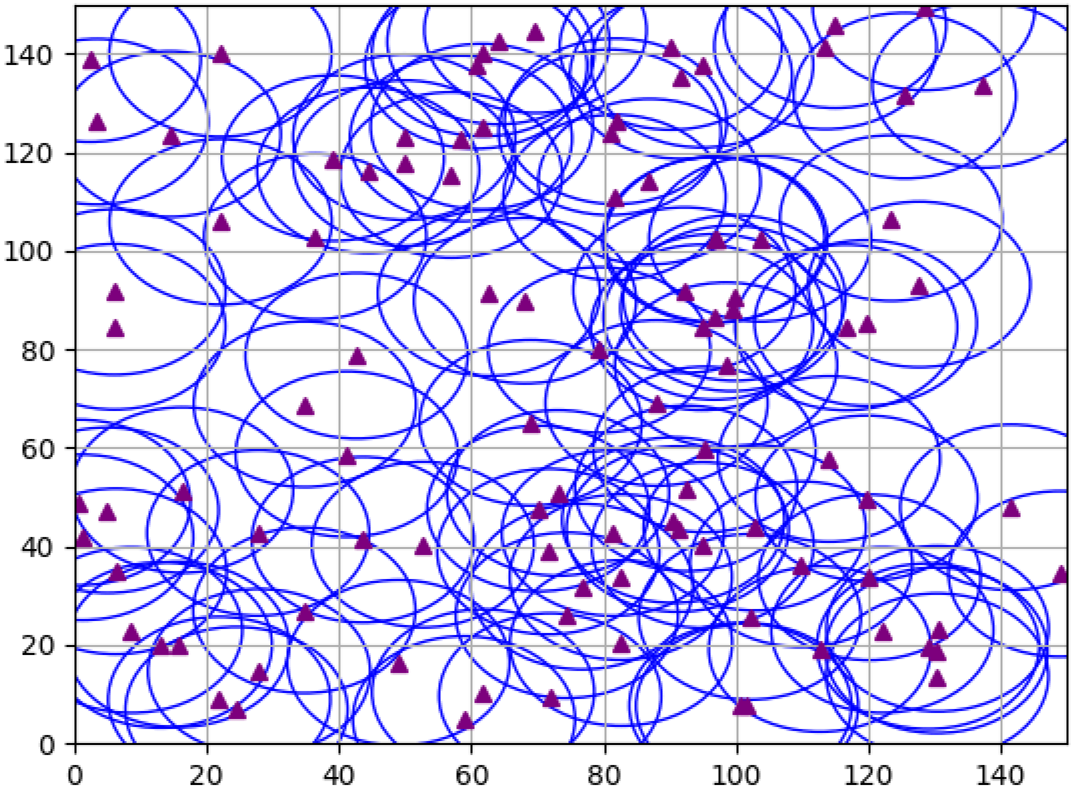
Joey Isola
CS 455

Homework 1

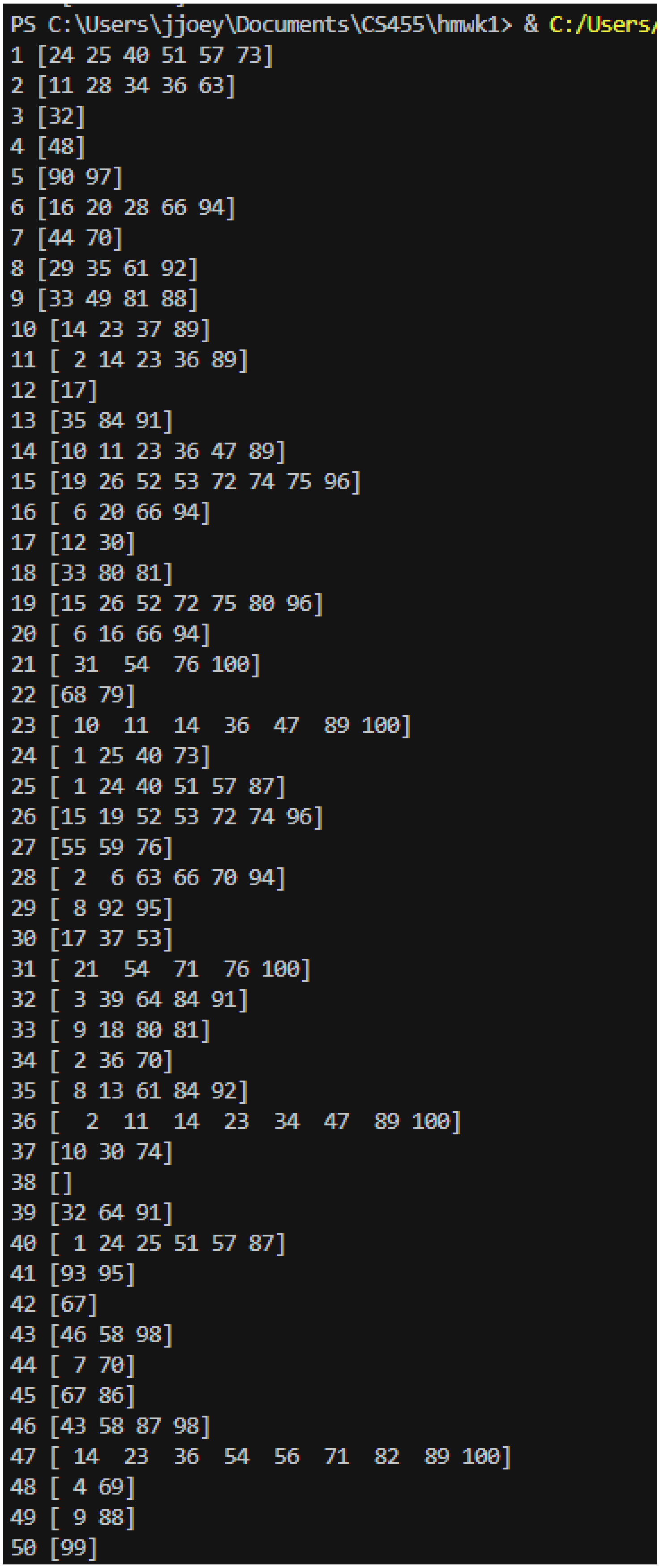
1. To make a plot of 100 sensor nodes in a 150x150 grid, I used the “matplotlib” library in python. I set up my x and y coordinates as random, uniformly spaced numpy arrays. Then I set up my xlim and ylim to be the length of 150x150, and I plotted the coordinates onto the “ax”. Purple Triangles are set using ‘^’ and color=”purple”



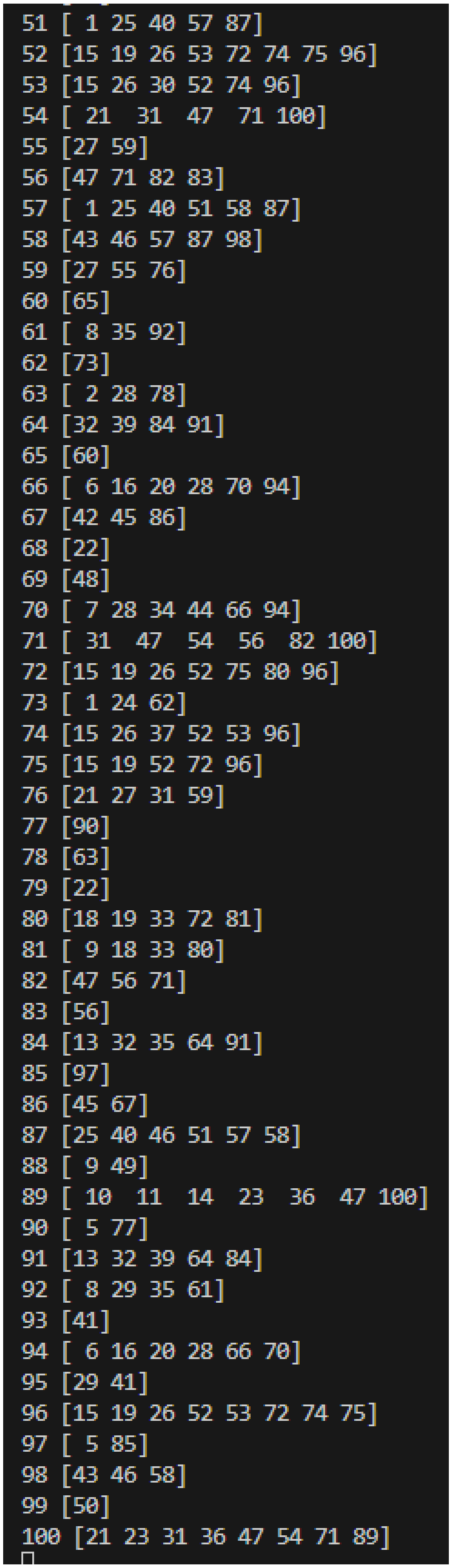
2. To add blue circles acting as each node’s interaction range, a patches.circle is added to the matplotlib object with parameters: coordinates, radius, color, fill.



3. To Generate a list of each node's respective neighbors within the set interaction range: first, a NxN array of zeroes is intialized as an adjacency matrix, then the program is set to iterate through each node and check all the nodes ahead of it whether the given node is within its distance. To accomplish this, the x, and y coordinates of each node are put into the following formula $((x_2 - x_1) + (y_2 - y_1))^{\frac{1}{2}}$. If the neighboring node's magnitude is less than or equal to the interaction range, (i, j) and (j, i) are added to the adjacency matrix. To print out neighboring nodes, iterate through the rows of the adjacency matrix with this line of python code : `np.nonzero(matrix[node])[0] + 1`



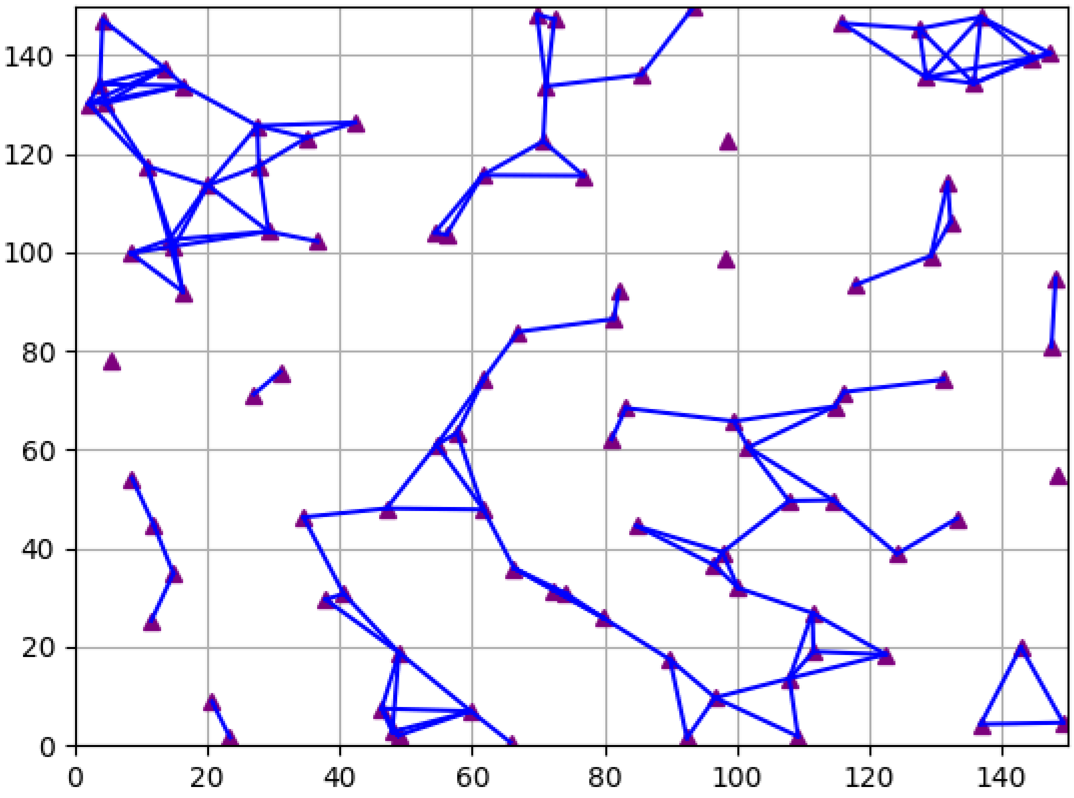
```
PS C:\Users\jjoey\Documents\CS455\hmwk1> & C:/Users/
1 [24 25 40 51 57 73]
2 [11 28 34 36 63]
3 [32]
4 [48]
5 [90 97]
6 [16 20 28 66 94]
7 [44 70]
8 [29 35 61 92]
9 [33 49 81 88]
10 [14 23 37 89]
11 [ 2 14 23 36 89]
12 [17]
13 [35 84 91]
14 [10 11 23 36 47 89]
15 [19 26 52 53 72 74 75 96]
16 [ 6 20 66 94]
17 [12 30]
18 [33 80 81]
19 [15 26 52 72 75 80 96]
20 [ 6 16 66 94]
21 [ 31  54  76 100]
22 [68 79]
23 [ 10  11  14  36  47  89 100]
24 [ 1 25 40 73]
25 [ 1 24 40 51 57 87]
26 [15 19 52 53 72 74 96]
27 [55 59 76]
28 [ 2  6 63 66 70 94]
29 [ 8 92 95]
30 [17 37 53]
31 [ 21  54  71  76 100]
32 [ 3 39 64 84 91]
33 [ 9 18 80 81]
34 [ 2 36 70]
35 [ 8 13 61 84 92]
36 [  2  11  14  23  34  47  89 100]
37 [10 30 74]
38 []
39 [32 64 91]
40 [ 1 24 25 51 57 87]
41 [93 95]
42 [67]
43 [46 58 98]
44 [ 7 70]
45 [67 86]
46 [43 58 87 98]
47 [ 14  23  36  54  56  71  82  89 100]
48 [ 4 69]
49 [ 9 88]
50 [99]
```



```
51 [ 1 25 40 57 87]
52 [15 19 26 53 72 74 75 96]
53 [15 26 30 52 74 96]
54 [ 21  31  47  71 100]
55 [27 59]
56 [47 71 82 83]
57 [ 1 25 40 51 58 87]
58 [43 46 57 87 98]
59 [27 55 76]
60 [65]
61 [ 8 35 92]
62 [73]
63 [ 2 28 78]
64 [32 39 84 91]
65 [60]
66 [ 6 16 20 28 70 94]
67 [42 45 86]
68 [22]
69 [48]
70 [ 7 28 34 44 66 94]
71 [ 31  47  54  56  82 100]
72 [15 19 26 52 75 80 96]
73 [ 1 24 62]
74 [15 26 37 52 53 96]
75 [15 19 52 72 96]
76 [21 27 31 59]
77 [90]
78 [63]
79 [22]
80 [18 19 33 72 81]
81 [ 9 18 33 80]
82 [47 56 71]
83 [56]
84 [13 32 35 64 91]
85 [97]
86 [45 67]
87 [25 40 46 51 57 58]
88 [ 9 49]
89 [ 10  11  14  23  36  47 100]
90 [ 5 77]
91 [13 32 39 64 84]
92 [ 8 29 35 61]
93 [41]
94 [ 6 16 20 28 66 70]
95 [29 41]
96 [15 19 26 52 53 72 74 75]
97 [ 5 85]
98 [43 46 58]
99 [50]
100 [21 23 31 36 47 54 71 89]
```

4. To plot blue lines between adjacent nodes, iterate through the rows of the adjacency matrix and add the code:

```
ax.plot([x_coordinates[i], x_coordinates[j]], [y_coordinates[i],
y_coordinates[j]], 'b-')
```



0
20
40
60
80
100
120
140
0
20
40
60
80
100
120
140