Neighboring Nodes

To demonstrate your analytical skills and creativity, please complete this coding test in either Python or Java. This test should take 1-2 hours to complete, but please take the necessary time to complete it and give some consideration to code optimality.

Submit the results as an archive or link to a repository containing the code.

Please include a readme file documenting major methods/procedures (how to run, required parameters).

Feel free to email any questions. We will review your code in a followup interview!

Task 1: Build the grid

Implement a class named Neighboring Nodes with the following parameters:

- *size* (a positive int)
- *debug* (boolean)

Add a method to Neighboring Nodes that constructs a grid of *size* x *size* nodes. For example, if *size* = 3, Neighboring Nodes will instantiate a 3x3 grid.

Each node in the grid should have the following features:

- *x*:itsx-axis coordinate
- *y*:itsy-axis coordinate
- *i*:index number of the node,in order of creation

If the boolean parameter *debug* is true, the method should print out the (*x* , *y*, *i*) features for each of the nodes after the grid is built.

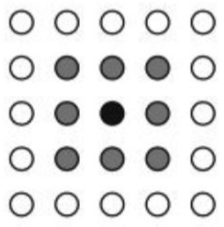
Add a method that accepts the index of a node as its parameter and returns the (*x*, *y*) coordinate of that node.

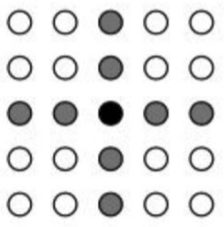
Task 2: Finding neighbors
Add a method to Neighboring Nodes that accepts the following parameters:

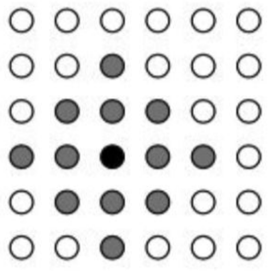
- *x*:x-axis coordinate
- *y*:y-axis coordinate
- *i*:nodeindex
- *m*: neighborhood radius (int, 0 < *m* <= *size/*2)
- *type*:neighborhood type(enum:SQUARE,CROSS,DIAMOND)

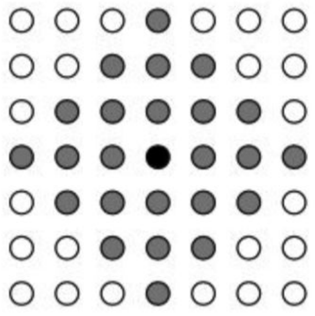
The method should accept either *x* & y OR *i*, not both at the same time. *x* & y or *i* represent the origin node; using the supplied parameters, the method should return coordinates of all the neighboring nodes, based on the provided topological neighborhood type, within distance m.

Some visual examples:


m = 1
type = SQUARE


m = 2
type = CROSS


m = 2
type = DIAMOND


m = 3
type = DIAMOND

Be sure to handle the following cases:

- The distance from the specified node to one or more edges of the grid is < *m*
- *x*,*y* or *i* fall outside the grid-in this case you can optionally find the nearest matching coordinates and return its neighbors (not required)

Task 3: SQL

Given a table having records of user visits in the following format: `user_id, visit_date, url`, find out the longest consecutive days each user has visited