The Python skeleton code is provided to you to get started and automate the correctness testing. You may add your own Python files to your submission, but please ensure that they work with the supplied Python script.

|  |  |
| --- | --- |
| **File** | **Description** |
| nearest\_neigh\_file\_based.py | Code that reads in operation commands from file then executes those on the specified nearest neighbour data structure. *No need to modify this file.* |
| category.py | Class defining the different categories of queries. *No need to modify this file.* |
| point.py | Class representing a point in 2D space. *No need to modify this file.* |
| nearest\_neigh.py | Base class for the two query agents. *No need to modify this file.* |
| naive\_nn.py | Skeleton code that implements a naive implementation of nearest neigh- bour searches. Complete the implementation (implement parts labelled “To be implemented”). |
| kd\_tree\_nn.py | Skeleton code that implements Kd-tree implementation of nearest neigh- bour searches. Complete the implementation (implement parts labelled “To be implemented”). |

To run the code, from the directory where the nearest\_neigh\_file\_based.py is, execute:

* python nearest\_neigh\_file\_based.py [approach] [data filename] [command filename] [output filename]

where,

* approach is one of “naïve” or “kdtree”.
* data filename is the name of the file containing the intial set of points.
* command filename is the name of the file with the commands/operations.
* output filename is where to store the output of program.

python nearest\_neigh\_file\_based.py naive sampleData.txt test1.in output.txt

python nearest\_neigh\_file\_based.py kdtree sampleData.txt test1.in output.txt