Dept. of Computer Science and Engineering
CS69011: Computing Lab 1

Assignment 1 — Maximum Marks: 30

Consider a situation where there are M mobile base stations with 2D locations $X = \{ (x_i, y_i), i = 1,\ldots,M \}$, and N mobile phones with 2D locations $X' = \{ (x'_j, y'_j), j=1,\ldots,N \}$. Also, assume that each mobile phone connects to its closest base station according to Euclidean distance:

$$D(x_i, y_i, x'_i, y'_i) = \sqrt{(x_i - x'_i)^2 + (y_i - y'_i)^2}$$

Part 1: (10 marks)

The first task is to write a program which takes the sets X and X' as input and calculates the nearest mobile base station in X to each mobile phone in X'. You can use a naive algorithm for the same. What is the time complexity of your program?

> *Hint about program structure*:
> You can store each set of points as an array of structures of type "`point`". You can write a function, `search_nearest()`, which takes a set of points X and a query point (x, y) as input and returns the nearest point in X.

Input format:
The input will be given in a file of the format (the angle bracket indicates values of the parameter will be provided):

M
x_1, y_1
...
x_M, y_M
N
x'_1, y'_1
...
x'_N, y'_N

Part 2: ***Incremental nearest neighbour*** (10 marks)

In the second part you are given subsets of X and X', $Z = \{(x_i,y_i), i=1,\ldots,M'\}$ and $Z' = \{(x'_j,y'_j), j=1,\ldots,N'\}$ respectively, whose coordinates have changed. You also know the existing nearest mobile base stations for each mobile phone in X' (according to the old locations). You have to write a program which can calculate the nearest mobile base station in set X according to their new locations, for each mobile phone in set X' also considering their new locations. What is the time complexity of this algorithm?

Input format:
You will be given both old locations and changed locations of points which have changed as input. For the changed points, index of the original point is also provided. The input will be given in a file of the format (the angle bracket indicates values of the parameter will be provided):

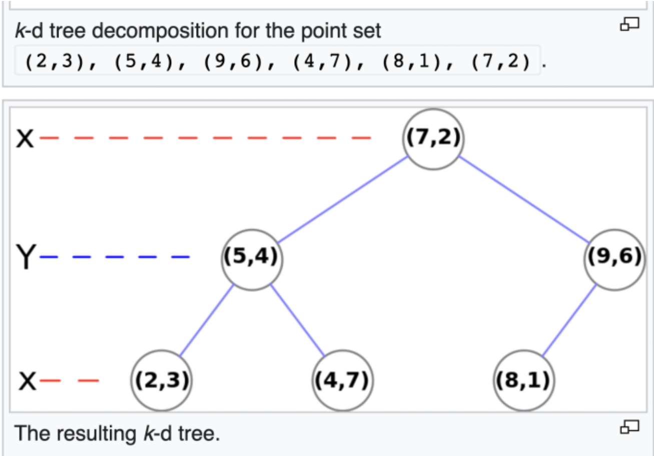
M
x_1, y_1
...

x_M, y_M
N
x'_1, y'_1
...
x'_N, y'_N
M'
i_1, x_{i1}, y_{i1}
...
$i_{M'}, x_{iM'}, y_{iM'}$
N'
j_1, x'_{j1}, y'_{j1}
...
$j_{N'}, x'_{jN'}, y'_{jN'}$

Part 3: ***Faster nearest neighbour search*** (10 marks)

In this part, you are needed to write a program which solves the problem described in part 1 using an algorithm which is O(N log M + M log M). Use the k-d tree algorithm (https://en.wikipedia.org/wiki/K-d_tree) to generate an efficient search data structure for the points in X (mobile base stations) and use it to implement the `search_nearest()` function described in Part 1. The broad algorithm is as follows:

1. Generate k-d tree for the dataset X:
   (a) Sort the points in X according to both dimensions.
   (b) Recursively split the set Z at the current node using the median coordinate in the current dimension to create children, where the points in left children have coordinate value less than median, and those in right children have coordinate value greater than the median. Generally, we cycle through the dimensions. The figure below shows an example k-d tree.
2. Search for the points nearest to each point (x',y') in X'.



The resulting *k*-d tree.

Query algorithm: Traverse the tree using the query point, starting with the root node to a leaf node. The point with the minimum distance among this set is the nearest point.

Submission:
Submit the programs corresponding to the three parts in three different files.