Homework #4

04/20/2017
Due Date: ~~04/18/2017~~

Problem 1 *(60 points)*

You will "manually" simulate the join protocol for constructing a distributed DT. Whenever you need to compute a DT graph, you can use `delaunay` and `triplot` functions in MATLAB, which is available in every UTCS machine. Initially, there are 10 nodes in 2D. Their coordinates are provided in Table 1.

node ID	x	y
1	1	7
2	4	8
3	2	3
4	4	4.5
5	6	1
6	9	4.5
7	11	0
8	16	7
9	15	5
10	17	4

Table 1: Coordinates of Nodes

1. Compute and plot the DT graph of the 10 nodes with coordinates shown in Table 1. Label each node in the graph by its ID.

2. For each node, say u, initialize its candidate set C_u to include both u and its neighbors in the 10-node DT.

3. Consider a new node, node 11 with coordinate $(8,9)$, which joins the existing 10-node DT. Node 11 has found that its closest node is node 2 with coordinate $(4,8)$. Nodes 11 and 2 must be neighbors in the new 11-node DT. Node 11 performs an iterative search for all of its neighbors in the new 11-node DT, one round at a time. In each round, the following steps are performed [by you]:

(a) Node 11 sends a `NB_SET_REQ` to every node in the set of new neighbors it has found. (In round 1, node 11 has found just one new neighbor, i.e., node 2.)

(b) If a node, say u, receives a `NB_SET_REQ` from node 11, node u adds node 11 to its candidate set C_u. Node u then computes $\mathrm{DT}(C_u)$ and sends a `NB_SET_REPLY` to node 11.

[You will compute and plot the $\mathrm{DT}(C_u)$ graph, with nodes labeled by their IDs. Write down the set of neighbors of node 11 in $\mathrm{DT}(C_u)$ included in `NB_SET_REPLY` from node u to node 11.]

Repeat this step for every node to which node 11 has sent a `NB_SET_REQ` in this round.

(c) After receiving a `NB_SET_REPLY` in response to every `NB_SET_REQ` sent in this round, node 11 adds all neighbor candidates (provided in all `NB_SET_REPLY` messages in this round) to its own candidate set C_{11}. Node 11 then computes $\mathrm{DT}(C_{11})$ to see if it has new neighbors. If node 11 has found one or more new neighbors, it initiates another round (go to step (a)); **else, exit**.

[You will compute and plot the $\mathrm{DT}(C_{11})$ graph, with nodes labeled by their IDs. Also write down node 11's set of neighbors in $\mathrm{DT}(C_{11})$, namely N_{11}, separately.]

(d) Plot the global DT after the successful join of node 11.

Note: Your TA has created a function, `plotdt`, which wraps the `delaunay` and `triplot` functions for you to compute and plot a DT graph. Read comments in `plotdt.m` for its usage.