Computer Science 130B
Fall 2021
Programming Assignment #2

Due: **11:59pm**, October 31, Sunday

Your task here is to implement the minimum cost spanning tree algorithms. You can choose to implement either Kruskal's or Prim's algorithm.

You read input data from `cin` and output your result on `cout`. The format will be

n	/* number of vertices */
x_0 y_0	/* coordinates of the $0th$ vertex */
. . .	. . .
x_{n-1} y_{n-1}	/* coordinates of the $(n-1)th$ vertex */
m	/* number of edges */
v_0^0 v_1^0	/* indices of the first edge, or the first edge goes from vertex v_0^0 to v_1^0 */
. . .	. . .
v_0^{m-1} v_1^{m-1}	/* indices of the $(m-1)th$ edge, or the $(m-1)th$ edge goes from vertex v_0^{m-1} to v_1^{m-1} */

where $0 \le v_0^i, v_1^i < n, 0 \le i < m$. A special case is when $m = 0$, in that case, the graph is a complete graph and there exists an edge in between every pair of vertices. The "cost" or "weight" of an edge is conveniently defined to be the Euclidean distance between the two vertices. Also note that the edges are not directional (a two-way street).

Your program should output (to `cout`) the following information:

V_0^0 V_1^0	/* indices of the first MCST edge that goes from vertex V_0^0 to V_1^0 */
. . .	. . .
V_0^{n-1} V_1^{n-1}	/* indices of the $(n-1)th$ MCST edge that goes from vertex V_0^{n-1} to V_1^{n-1} */

For ease of grading, The MCST edges must be written out in the order of the vertex of the smaller index (e.g., edge (1,3) before edge (2,3)). If the vertices of the smaller index are the same for two MCST edges, the tie is broken by the other vertex (e.g., edge (1,2) before edge (1,3)).

Again, while some standard test data are provided, you should also generate your own data (random points in 2D with random connection patterns). To see if your program produces the correct results, you can use a plotting program, such as gnuplot, to plot input points and the MCST.