This assignment asks you to implement Dial's label-setting algorithm for determining a shortest path tree of a directed network G with n nodes and m edges, and nonnegative edge lengths. Use a "circular" address list A to search for the next node to be scanned.

Input to your program consists of a source node s and a destination node t, followed by an edge list representation of G: from_node, to_node, (real) length. You can suppose that the nodes of G are consecutively numbered 1, 2, …, n. If needed by your code, the values n and m can also be input.

Output should consist of a shortest s-t path, giving its total length as well as listing its constituent edges (in their natural order) and their individual lengths.

(1) Run your algorithm on the sample problems (to be supplied) and provide detailed output, as described above.

(2) Carefully analyze the *time* and *space* complexities of your particular implementation, in terms of relevant network parameters.

Things to Keep in Mind:

- make your code as general and modular as possible
- have the main routine be "self-documenting"
- avoid unnecessary special cases
- use a reasonable value for "infinity"
- check the input: e.g., that all edge lengths are in fact nonnegative

Provide well-documented code as well as a clear English language description of your data structures, what your code is doing, and important program variables. Be sure to discuss your initializations and termination conditions.