CS 341, Fall 2019

PROGRAMMING ASSIGNMENT 2

DUE: Wednesday, November 20, 5 PM. DO NOT COPY. ACKNOWLEDGE YOUR SOURCES.

Please read http://www.student.cs.uwaterloo.ca/~cs341 for general instructions and policies.

1. [20 marks] **Shortest simple path.** In this problem, you will implement a program that solves the *Shortest simple path* problem from Question 1 of Assignment 7.

   The input to your program will be a directed graph \( G = (V,E) \) with integer weights on the edges and two vertices \( s,t \in V \). The graph will be given via adjacency lists. The program must output the minimum weight of a simple path from \( s \) to \( t \) in \( G \). Recall that a *simple* path is a path that does not repeat vertices, and that the weight of a path is the sum of the weights of the edges in the path.

   More specifically, the input is formatted as follows:

   - Line 1 consists of three positive integers
     
     \[ n \ s \ t \]

     separated by whitespace. The integer \( n \) indicates that the graph \( G \) has vertex set \( \{1,2,\ldots,n\} \). The integers \( s,t \in \{1,2,\ldots,n\} \) are vertices of \( G \).

   - Lines 2 through \( n + 1 \) each consist of a list of integers
     
     \[ k \ v_1 \ w_1 \ v_2 \ w_2 \ \cdots \ v_k \ w_k \]

     all separated by whitespace. The first integer \( k \) in the list indicates that \( k \) pairs \( (v_j,w_j) \) follow. The pair \( v_j,w_j \) in line \( i + 1 \) indicates that \( G \) has an edge of weight \( w_j \) going from vertex \( i \) to vertex \( v_j \in \{1,2,\ldots,n\} \).

   The output of the program is an integer \( W \) denoting the minimum weight of a simple path from \( s \) to \( t \) in \( G \), if at least one such path exists. If \( G \) contains no path from \( s \) to \( t \), the algorithm should output *No path*.

   Implement a branch-and-bound algorithm that solves the shortest simple path problem described above. You may write your code in C, C++ or Java.

   See the next page for two example instances of the program and expected output.
Example 1. On input

\[
\begin{array}{cccc}
6 & 1 & 3 \\
2 & 2 & 10 & 4 & -1 \\
1 & 3 & 6 \\
1 & 1 & 2 \\
3 & 1 & -2 & 2 & 8 & 5 & 4 \\
2 & 3 & 2 & 4 & 1 \\
0
\end{array}
\]

the correct output is

5

since the input corresponds to a graph whose shortest simple path from vertices 1 to 3 is \(1 \rightarrow 4 \rightarrow 5 \rightarrow 3\) which has total weight \(-1 + 4 + 2 = 5\).

Example 2. On input

\[
\begin{array}{cccc}
4 & 1 & 4 \\
1 & 2 & 1 \\
1 & 3 & 1 \\
1 & 1 & 1 \\
3 & 1 & 1 & 2 & 1 & 3 & 1
\end{array}
\]

the valid output is

No path

since there is no path from vertex 1 to vertex 4 in the graph specified by the input.