

UE19CS251 DAA

Assignment II

Max Marks: 20(will be scaled down)

Implement Dijkstra's algorithm to solve **Single Destination Shortest Path** problem. Single destination shortest path is finding shortest path from all the vertices to the given vertex

Input graph to be read from the file adjacencylist.txt

Input file

- > Vertices are numbered 1 to n
- > First line represents number of vertices
- This is followed by a set of lines
- ➤ Each line starts with a integer (any integer from 1 to n in any order) which represents vertex number (v_id) followed by space followed by a set of d pairs where d represents outdegree of the vertex v_id. First number in the pair represents neighbour vertex id and second number weight of the edge which connects vertex to the neighbour. Weight is always a nonnegative integer

Sample Input file

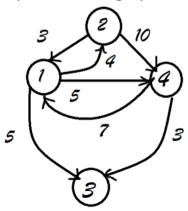
4

1243545

41733

213410

Represents the graph





The input file provided in the assignment folder should be read to create a graph which should be represented in memory using **adjacency list representation**

Implementation

- Program should consider the last vertex (v_id = n) as the destination (Example destination is 4 for the graph mentioned in example above)
- Using Dijkstra's algorithm Program should determine shortest distance from all the vertices to the destination
- Priority queue used by dijkstra's algorithm should be implemented using min heap. Each node in the heap contains vertex number, d value (distance of path from source to the vertex), and p value (Vertex id of predecessor vertex on the path from the source to the vertex). d value should be used as key to perform various operations (insert/delete/update)
- Program should print shortest path from all the vertices to the destination.

Output Format

- ➤ Each paths to be printed on new line (starting with path from 1 to destination then from 2,3-----n-1). If no path exists from a particular vertex to the destination vertex 'NO PATH' should be printed on the respective line number
- Each line should print a single path and length of the path (space separated), where the vertices in a path are space separated (No other format for output is accepted)

Sample output:

1145

22148

3 NO PATH



C programming language should be used for implementation

Implementation should consists of three files

- > Header file (function prototypes, user defined data type definitions)
- > Implementation File (function definitions)
- Client file (Driver function)

The file names should be

SRN_H.h

SRN F.c

SRN C.c

List of functionalities your code should support

- Reading data from file and creating Graph (Adjacency list and not adjacency matrix)
- 2. Implementation for Priority queue using min heap (Insert, delete, search update)
- 3. Implementation of Dijkstra's algorithm to determine shortest path from all vertices to nth vertex in the list (Single Destination Shortest path)
- 4. Determining shortest path distance and shortest path from all the vertices to destination vertex (Dijkstra's module to be invoked only once)
- 5. Printing shortest path and shortest path distance from all the vertices to the destination starting with vertex numbered 1,2,3---n-1 as per the output format specified

Marks distribution

Functionality Max Marks 1+5 5 marks

2, 3, 4 5 marks each

