



Data Structure

Lab Session #12: Graphs

U Kang
Seoul National University



Goals

- Implement the Dijkstra's algorithm.
 - Fill your codes in “Dijkstra.java” and “Graph.java”.
 - Use an adjacency list structure to store the edges.
 - Use a priority queue to implement ‘minVertex’ operation.
- Print the sample output corresponding to the sample input.



Notice

- After implementing “Dijkstra”, check if your program works well
 - Check sample input and output files in the ‘testdata’ folder
 - Test your program by using them
- Please raise your hand and ask to T.A. if you have any question regarding the problems
- You need to stay for at least an hour



Build a Project

- Download the project for the lab from eTL
- Extract the project, and open it in IntelliJ
 - See the slide of 1st lab session to check how to open the project in IntelliJ



Functions to Implement

- *checkWeightValue(Edge testEdge)*
 - Checks whether a given weight has a negative value whenever a new edge is being inserted
 - If its weight has a negative value, then returns false

- *delEdge(int i, int j)*
 - Removes an edge if there is one from vertex i to vertex j

- *calculateShortestPath(Graph G, int start)*
 - Given a source vertex *src*, calculates the shortest path from *src* to all other vertices



Functions to Implement

- *printPathToEnd(Graph G, int end)*
 - Prints the shortest path from *src* to *end* and sum of weights after calculating the shortest path by *calculateShortestPath()* function

- *printAllPath(Graph G, int src)*
 - Prints All the shortest path from the source vertex *src*

- *minVertex(Graph G)*
 - Determines the next closest vertex
 - You can implement the function using *distanceQueue*



I/O Specification

■ n

Input form	Output form
n (#vertices)	
Description	
<ul style="list-style-type: none">- Creates a directed and weighted graph of size (#vertices).- All the vertices have a unique id starting from 0.- ‘n’ command appears at the first line of input.- ‘n’ command doesn’t appear multiple times.	
Example Input	Example Output
n 10	



I/O Specification

■ edge

Input form	Output form
edge (src) (dst) (w)	
Description	
<ul style="list-style-type: none">- Adds an edge from vertex (src) to vertex (dst) with weight (w).- (src) and (dst) are between 0 and (#vertices) – 1.	
Example Input	Example Output
edge 0 7 1.5	



I/O Specification

■ shortestpath

Input form	Output form
<code>shortestpath (src)</code>	<code>PATH (src) (dst): (dist)</code> <code>(src) ... (dst)</code>
Description	
<ul style="list-style-type: none">- Prints all the paths from (src).- (dist) is the sum of weights along the shortest path.	
Example Input	Example Output
<code>shortestpath 0</code>	<code>PATH 0 1: 0.1 0 1</code> <code>PATH 0 2: 0.7 0 1 2</code>



Sample Input and Output

- n 5
- edge 0 1 10
- edge 0 2 3
- edge 0 2 -3
- edge 0 3 20
- edge 1 3 5
- edge 2 1 2
- edge 2 4 15
- edge 3 4 11
- shortestpath 0
- The weight has a negative value.
- PATH 0 0: 0.0 0
- PATH 0 1: 5.0 0 2 1
- PATH 0 2: 3.0 0 2
- PATH 0 3: 10.0 0 2 1 3
- PATH 0 4: 18.0 0 2 4



Questions?