

### Programming 3 Input Examples

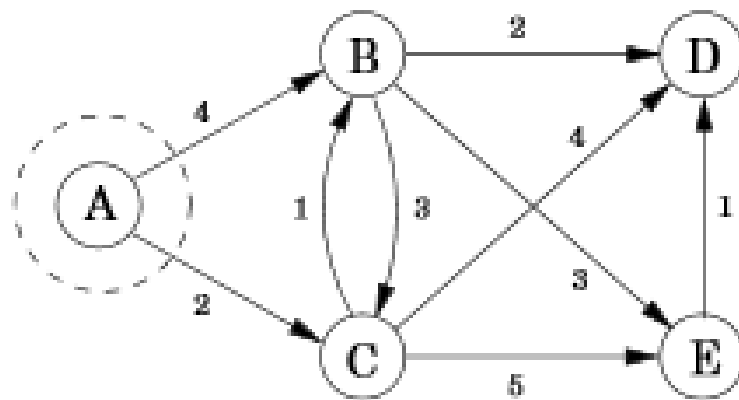


Figure 1 Directed Graph

If the input graph is as above then the corresponding input file will be:

#Input graph File name is input.txt - Single line comment

D

A B 4

A C 2

B C 3

C B 1

B D 2

B E 3

C E 5

C D 4

E D 1

Given the input as above your code should be able to decipher from the first uncommented line that this is a directed graph and the format for the edges is:

Node1<space>Node2<space>Weight<return> Till the EOF is reached.

This means that the graph is directed and that there is an edge directed from Node1 to Node2 with weight=Weight

### Programming 3 Input Examples

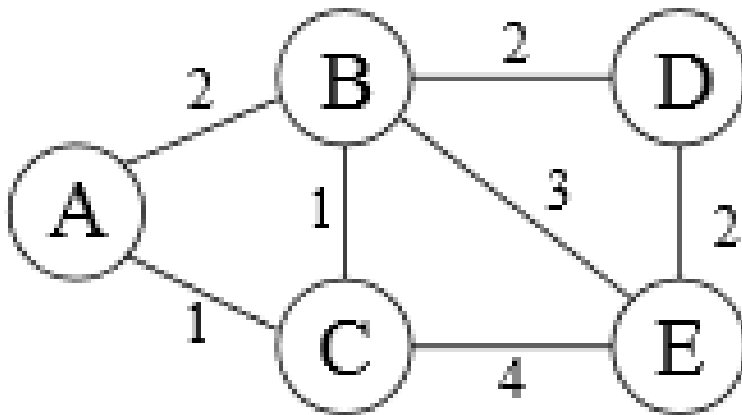


Figure 2 Undirected Graph

For an undirected graph as shown above the figure the corresponding input file will be as below:

#Input graph File name is input.txt - Single line comment

UD

A B 2

A C 1

B C 2

B D 2

B E 3

D E 2

C E 4

This input file can potentially also be:

#Input graph File name is input.txt - Single line comment

UD

B A 2

C A 1

### Programming 3 Input Examples

C B 1

D B 2

E B 3

E D 2

E C 4

This is because as the file represents an undirected graph (as denoted by UD in first uncommented line) AB is same as BA. Thus the format of each line is:

Node1<space>Node2<space>Weight<Return> Till EOF.

This means that there is an undirected edge from Node1 to Node2 OR from Node2 to Node1 with weight=Weight.

The code should just accept the file along with other inputs as mentioned in the write-up. We should be able to supply the start node for Dijkstra and the start node and K for the reliable shortest paths algorithm.