MASSEY UNIVERSITY
TE KUNENGA KI PŪREHUROA
UNIVERSITY OF NEW ZEALAND

159.201 Algorithms & Data Structures
S1 2015

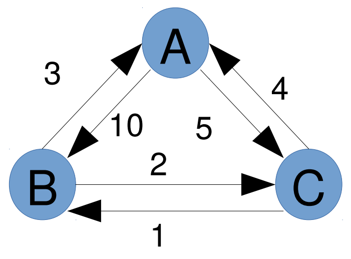
Assignment 6

Write a C++ program to implement Dijkstra's algorithm that can find the minimum costs from node A to all other nodes. A generic graph with an arbitrary number of nodes is written into a text file, and the program should be able to read it and insert nodes and costs into a class called Graph (see week 6 slides).

The txt files contain nodes and direct costs from a node to some other node. An example with 3 nodes is:

```
...
Node A
Node B
Node C
...
A B 10
A C 5
B A 3
B C 2
C A 4
C B 1
```

The text file above would represent the following graph:



After running Dijkstra's algorithm, the output should be the minimum cost from node A to every other node. An example of the format for the output is shown below:

```
From A to B: 6
From A to C: 5
```

There are 5 sample txt files called graph*.txt available on Stream. The number of vertices in those graphs vary from 3 to 20.

It is important that the number of nodes and the number of paths can be allocated dynamically. Therefore, your Graph class needs to use an ***adjacency list***. There are a number of options to achieve that using vectors and/or linked-lists. We have seen an example using vectors for the vertices, with elements that contain a pointer to a linked-list, which then contains the costs to reach a certain vertex. For more details on how to implement this assignment, refer to the material of **tutorial 8**.

Use our virtual machine to mark your submissions (host name **vm000296**). The input/output requirements are essential, please follow them carefully to avoid losing marks. Spaces matter and text is case sensitive.
After you are satisfied with the performance of your code as tested in the virtual machine, submit a **one source file** code on Stream by **Friday 15th of May 2015.** Your **name** and **ID number** must appear **on top of the file as comments.** If you are working in a group, then ***all*** names and IDs must appear on top of the file as comments, but you still need to **submit individually** in both the virtual machine and Stream.