```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

/*
Graph is implemented using the adjacency list
data structure where each node keeps a list of
its neighboring node and weights. The graph data structure
allows functionality such as adding nodes and edges and also remove
nodes and edges
*/
public class Graph
{
  private List<Node> nodes;

  public Graph() {
    nodes = new List<Node>();
  }



  public void addVertex(string data) {
    // create a new node
    Node v =  new Node(data);
    // add it to the nodes storage
    nodes.Add(v);
  }

  // getter for list of nodes of the graph
  public List<Node> getVertices() {
    return nodes;
  }

  /*
  Update both source and destionation's neighbor since this is
  an undirected graph
  */
  public void addEdge(Node source, Node destination, int cost) {
    source.addNeighbor(destination, cost);
    destination.addNeighbor(source, cost);
  }

  /*
  An auxiliary function that return a node with given data as its value
  */
  public Node getNodeByValue(string data) {
    foreach(Node v in nodes) {
      if(v.getData() == data) {
        return v;
      }
    }
    return null;
  }

}

```