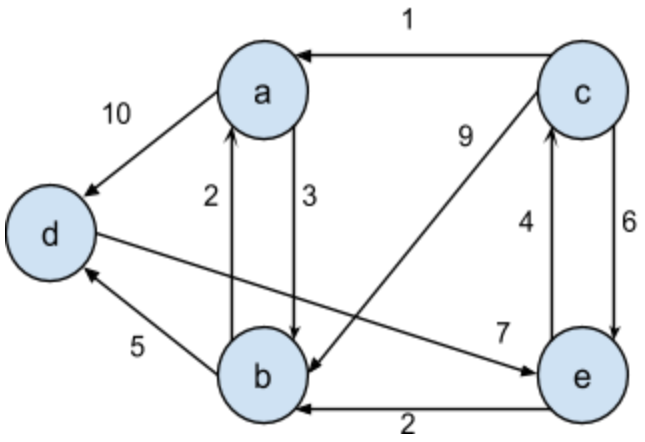
CS3110: Assignment 5
June 20 2017

1. G(V,E) is a directed graph with a shortest path tree T for the source *s*, prove or disprove the following statement:

- If If we replace all the edge weights w(u,v) by $w(u,v)^2$, T remains the shortest path tree (10 marks)

2. Describe an efficient algorithm that given a digraph G(V,E), it can output the shortest paths from each vertex v in V to a specific destination, d. What is the complexity of your algorithm? (20 marks)

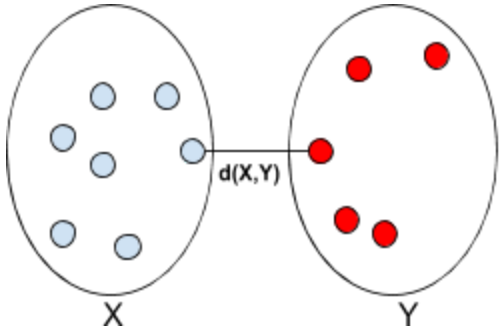
B) Apply your algorithm on the following graph and report the path and cost from *e* and *c* to d (d is the destination) (20 marks)



3. Clustering is one of the most fundamental tasks in data mining. It is defined as:
"...*the task of grouping a set of objects in such a way that objects in the same group (called a* ***cluster****) are more similar (in some sense or another) to each other than to those in other groups (clusters).* " (Wikipedia)

Not everybody agrees on how to define the distance between two clusters, and different definitions can lead to different clustering algorithms. Lets use the following definition:
The difference between two clusters of data is the distance between the closest pair of data, each in one cluster. Mathematically

$$dist(X,Y) = min\ \{dist(x,y) \text{ for all } x \in X \text{ and } y \in Y\}$$



A) You are given a set of objects, with all pairwise distances d(x,y), explain an efficient clustering algorithm in the following cases, **using the idea of MST**:

1. We want to cluster the data into exactly *k* clusters (20 marks)
2. It doesn't matter how many clusters we obtain, but we don't want the distance between any two clusters be less than δ (20 marks)

B) Can you explain why our definition of the distance can be problematic?

C) **Bonus: Implementation (Deadline July 4th) (25 marks).**
Step 1. Data preparation:
Manually create a list of Wikipedia titles (for example around 20), use different subjects (say 5 Hockey players, 10 singers and 5 medicines).
Convert your list to a tsv file; each line a pair of concepts. For example, if your initial list contained three concepts (it's an example, three is too small!):

Sidney_Crosby, David_Beckham *Victoria_Beckham*

You generate something like:

Sidney_Crosby	David_Beckham
Sidney_Crosby	Victoria_Beckham
David_Beckham	Victoria_Beckham

Now give your list to this web service and obtain the pairwise similarities (go to the batch section):
http://cgm6.research.cs.dal.ca/~sajadi/wikisim/

Step 2. Clustering:
Now implement the algorithm discussed in the previous section, cluster your data into some number of clusters (you can guess the number, because you generated the data yourself), and print the data. Try different numbers, different lists, and see how much the clusters make sense (25 marks).
Note: Your algorithm must be based and maximally similar to one of the pseudocodes we discussed in the class (line numbers, variable names and everything be similar in the main function calculating the MST). Obviously unless, you have a good reason to change something, which you are welcome providing a good comment on the code.