CS 8, Winter 2015
Homework Assignment #? (draft)

Assignment Overview

The goal of this project is to gain more practice with Python and implement some very useful graph algorithms.

Background

A graph data structure consists of a finite (and possibly mutable) set of *vertices* or *nodes* or *points*, together with a set of unordered pairs of these vertices for an undirected graph or a set of ordered pairs for a directed graph. These pairs are known as *edges*, *arcs*, or *lines* for an undirected graph and as *arrows*, *directed edges*, *directed arcs*, or *directed lines* for a directed graph.

A graph is a powerful data structure that can be used to represent many real-world phenomena. For example, a road network connecting multiple cities (vertices) using highways (edges) is a graph. A social network is often represented as a graph where people (vertices) are connected to their friends by edges. Computer networks are graphs where vertices are servers/routers/etc. and edges are direct communication links between these nodes.

Project Specifications

We will implement two very useful graph algorithms: MCST (minimum-cost spanning tree) and SSSP (single-source shortest path). MCST is an algorithm that finds the "skeleton" of a graph that has a minimum cost. A skeleton of a graph comprises all the nodes and a subset of edges so that every node can reach every other node by exactly one path – no redundancy. SSSP returns the shortest path (in terms of some cost measurement) between a source node and all other nodes in a graph.

Deliverables

The deliverable for this assignment is the following file:

graph.py – the source code for your Python program

Be sure to use the specified file name and submit it for grading via the **turnin** system before the project deadline.

Assignment Notes:

1. A file storing an undirected graph has the following fields:
    a. The first line has two numbers: number of vertices (NV) and number of edges (NE), followed by
    b. NV lines of vertex specification. Each line contains two numbers: the x and y coordinates of a vertex, followed by
    c. NE lines of edge specification. Each line contains two numbers: idx1 and idx2 that are the indices (starting at 0) of the two vertices the edge incident upon.

2. We will use the Euclidean distance between two nodes as the cost of the edge incident on these two nods. Not every possible edge exists. Only those specified in the input files are true edges.
3. There are only two exposed functions: MCST and SSSP. MCST takes a single argument that is the filename of a graph. SSSP take two arguments: the first is a filename of a graph and the second is the index of the source vertex. The shortest paths from the source vertex to all other vertices are return.

Sample Outputs:

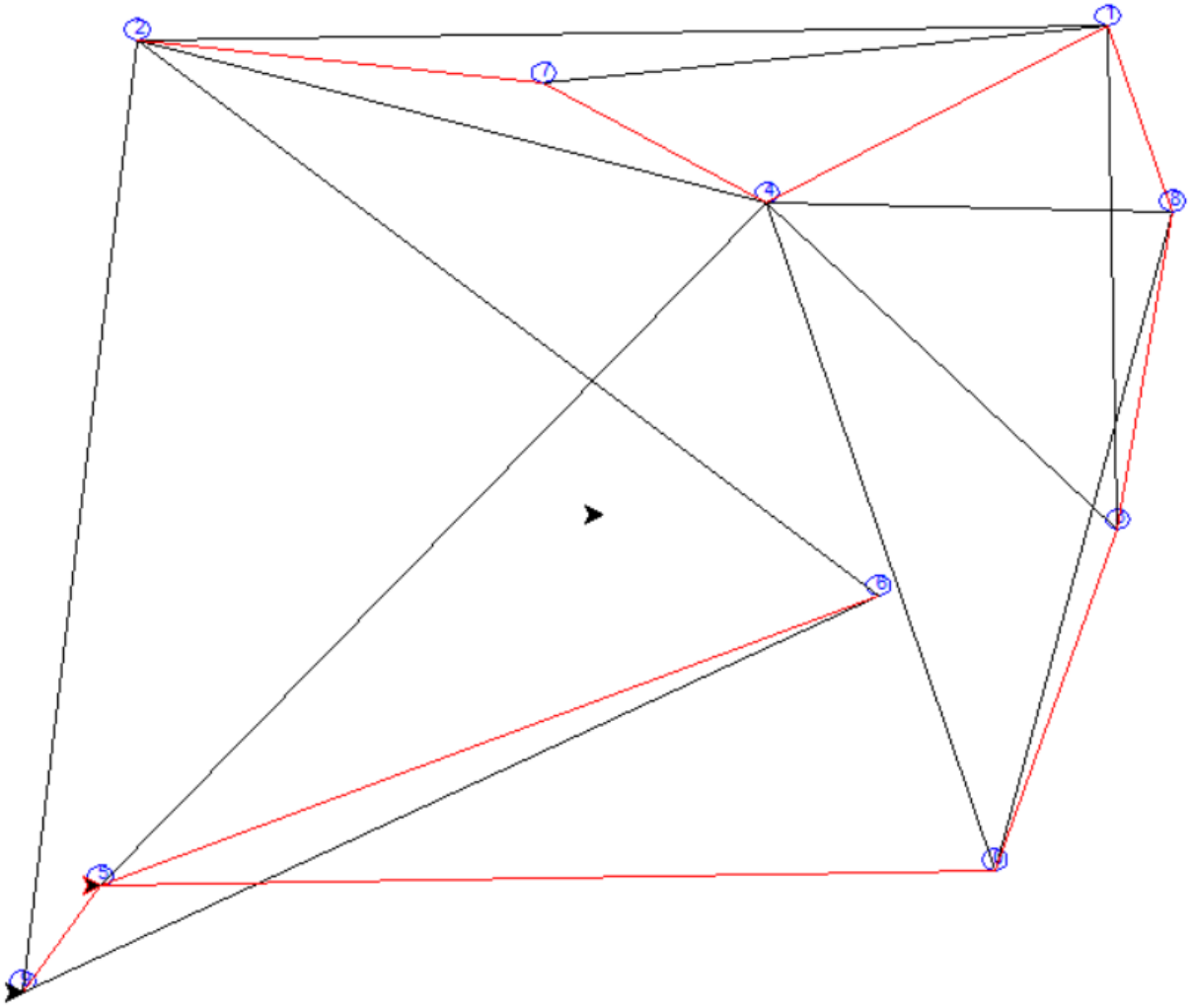

Figure 1: A graph with 10 vertices, with vertices marked in blue circles, edges marked in black lines and edges forming the MCST marked in red

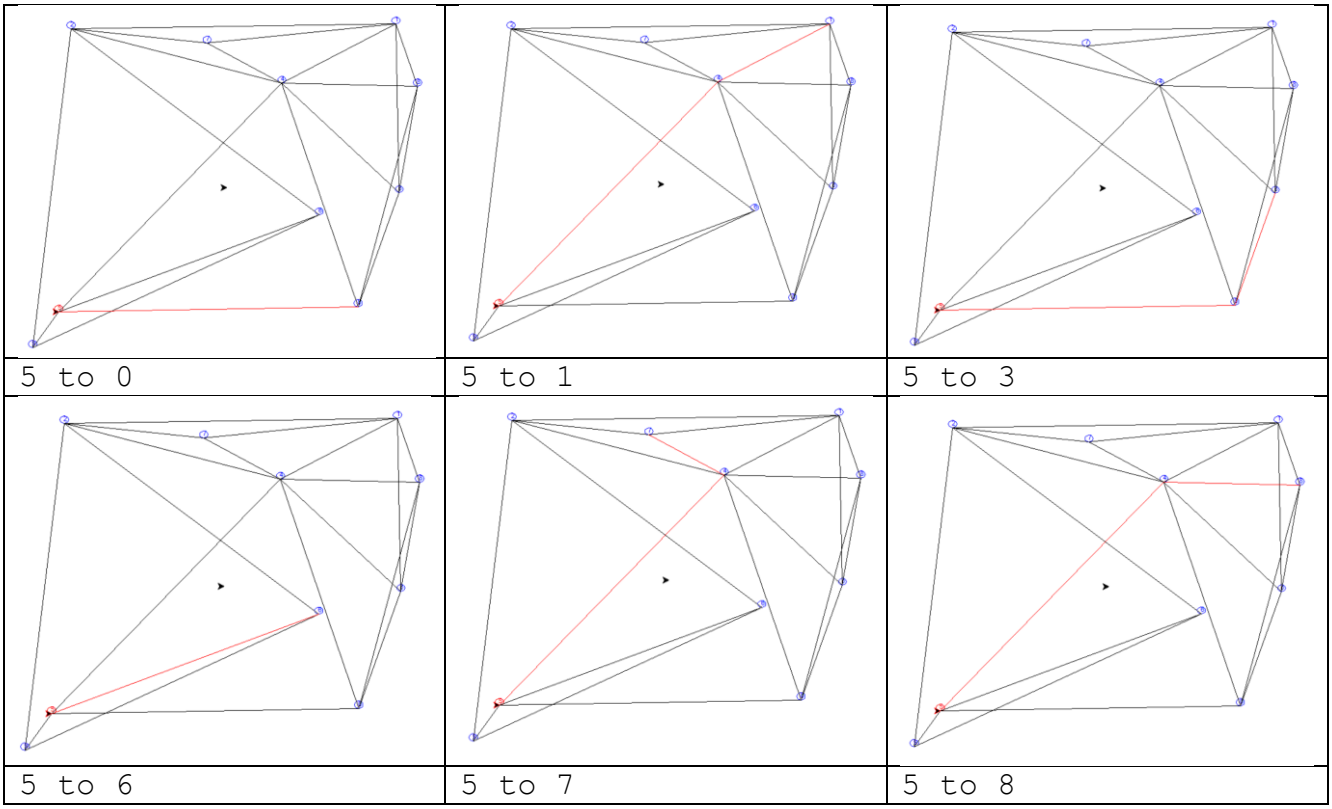

Figure 2 Sample shortest paths from source node 5.