Problem Set 7

All parts are due on November 9, 2017 at 11:59PM. Please write your solutions in the LaTeX and Python templates provided. Aim for concise solutions; convoluted and obtuse descriptions might receive low marks, even when they are correct. Solutions should be submitted on the course website, and any code should be submitted for automated checking on alg.csail.mit.edu .

Problem 7-1. [30 points] **Weighted Shortest Paths**

(a) [20 points] **Dijkstra Practice:** Below are representions for three weighted graphs. Each representation is a list of triples. The first two items in the triple are integer labels of two vertices connected by an edge, while the third item is a positive numerical weight of the edge. The first two graphs are directed, with a triple (a, b, w) representing a directed edge from vertex v_a to v_b, while the last graph is undirected. For each of graphs G_1 and G_2, perform Dijkstra's algorithm starting from vertex v_1 by doing the following:

1. Draw the weighted graph
2. List one possible order that edges could be first touched by Dijkstra
3. List the shortest path distance $\delta(v_i)$ from v_1 to each vertex $v_i \in V$

For example, a list of edges in graph G_0 that could be first touched by Dijkstra is $(\{e_1, e_2\}, e_3)$ where the first two edges could be touched in either order, with shortest path distances $(\delta(v_1), \delta(v_2), \delta(v_3)) = (0, 3, 1)$.

```
G_0 = [(1, 2, 3),  G_1 = [(1, 5, 5),  G_2 = [(1, 2, 2),   e_1
       (1, 3, 1),         (1, 6, 2),         (1, 4, 1),   e_2
       (2, 3, 2)]         (1, 7, 1),         (1, 5, 5),   e_3
                          (2, 3, 2),         (2, 3, 4),   e_4
                          (4, 5, 1),         (2, 4, 2),   e_5
                          (4, 8, 5),         (2, 5, 1),   e_6
                          (5, 6, 1),         (2, 6, 3),   e_7
                          (7, 2, 4),         (3, 5, 2),   e_8
                          (7, 3, 7),         (3, 6, 2),   e_9
                          (7, 4, 2),         (4, 5, 3),   e_10
                          (7, 6, 2),         (5, 6, 1)]   e_11
                          (7, 8, 5),                      e_12
                          (8, 1, 1)]                      e_13
```

(b) [10 points] **Minimal Destination Paths:** Let $G = (V, E, w)$ be a weighted, directed graph with weight function $w : E \to \mathbb{R}$, containing negative weights but no negative weight cycles. A **destination path** to vertex $v \in V$ is a path with smallest weight starting from any vertex $u \in V$ that terminates at vertex v. Note that vertex u may be v, resulting in a 0 weight destination path. A destination path is **minimal** if its weight is less than or equal to the weight of any other destination path terminating at v. Describe an algorithm to compute the weight of a minimal destination path for every vertex in V, all in $O(|V||E|)$ time.

Problem 7-2. [30 points] **Consulting**

For each of the following scenerios, provide the fastest algorithm you can think of to solve your client's problem.

(a) [10 points] **Downhill Skiing:** Warbler is a ski resort bidding to be the location for next year's downhill ski competition. The resort contains numerous downhill trails weaving down the mountain, with each trail reachable from the lodge at the peak. Warbler has collected trail reviews from customers, and has compiled an average rating for each trail between -5 and 5. The steering committee for the competition has scheduled a visit to evaluate the resort. They will have time for one evaluation downhill ski tour starting at the peak. For the tour, Warbler wants to choose a path of downhill trails beginning at the peak that maximizes the sum of trail ratings along the path. Describe an algorithm to quickly find such a path.

(b) [10 points] **Driving Options:** 6006LE Maps is a popular application that helps users navigate from one location to another. Given a starting location and an ending location, 6006LE currently provides users with the shortest driving route between two locations. However, 6006LE would like to give their users an additional option to choose from; specifically, they would also like to provide the second shortest route. Describe an algorithm to find both the shortest and second shortest routes between a given pair of cities, so that 6006LE may quickly return them to their customer. You may assume that every road supports traffic in both directions.

(c) [10 points] **Currency Exchange:** Travelwhy is a currency exchange company that buys and sells currencies at different prices, with a possibly different fixed rate every day. For instance, today they might purchase $\$1$ US Dollar from you and give you €0.8 Euros in return, though they might also purchase €1 Euro from you in exchange for only $\$1.10$ US Dollars. In this case, we would say the exchange rate from US Dollars to Euros is 0.8, while the exchange rate from Euros to US Dollars is 1.1. Travelwhy is concerned that the daily prices automatically generated by their computer might allow someone to make money off of them, simply by buying and selling the right sequence of currencies on their exchange. Describe an algorithm to check if there's a way to make money on their exchange on a given day.

Problem 7-3. [40 points] **Catwidth**

ComCat is a internet service provider that wants to optimize their network to bring you cat videos as fast as they can. Their network is comprised of thousands of servers storing millions of cat videos. Each user is connected to a single server, and servers are connected to each other, though not every pair of servers share a connection. Each direct server-server connection has a maximum bandwidth of cat videos that each may stream to the other; bandwidth may be different for different pairs of servers. Cat videos might need to pass through multiple servers to reach a user. The **catwidth** of a sequence of server-server connections is the smallest bandwith of any server-server connection in the sequence. There are s servers in the network, each labeled with a unique server ID number between 0 and $s - 1$.

(a) [10 points] ComCat wants to direct traffic in the network along routes that maximize catwidth. Given a pair of servers, describe an algorithm to compute a route with the largest catwidth in order to direct cat videos between them.

(b) [5 points] Assume that the number of server-server connections is large compared to the number of servers, i.e. asymptotically quadratic in the number of servers. Discuss the running time of your algorithm; specifically, can you simplify the implimentation of data structures used by your algorithm based on the assumed server-server connection density?

(c) [25 points] Implement function `catwidth` that takes as input a list of server-server connection triples (each consisting of two server IDs and the bandwidth between them) and a pair of server IDs, and returns the maximum catwidth of any route between them, assuming the number of server-server connections is asymptotically quadratic. Please submit code for this problem in a single file to alg.csail.mit.edu. ALG will only import your implementation of `catwidth`.