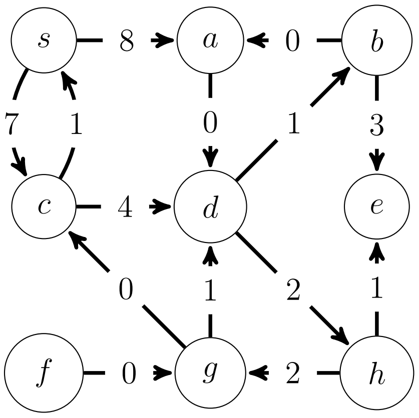
Problem Set 7

All parts are due on Novemeber 1, 2019 at 6PM. 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.mit.edu`.

Problem 7-1. [10 points] Dijkstra Practice

(a) [8 points] Run Dijkstra on the following graph from vertex s to every vertex in $V = \{a, b, c, d, e, f, g, h, s\}$. Write down (1) the minimum-weight path weight $\delta(s, v)$ for each vertex $v \in V$, and (2) the order that vertices are removed from Dijkstra's queue.



(b) [2 points] Change the weight of edge (g, c) to -6. Identify a vertex v for which running Dijkstra from s as in part (a) would change the shortest path estimate to v at a time when v is not in Dijkstra's queue.

Problem 7-2. [10 points] Faster SSSP with Negative Weights

Given a weighted **undirected dense**[1] graph $G = (V, E)$ (where edge weights are integers that may be either positive or negative), describe an $O(|V|^2)$-time algorithm to solve single-source shortest paths from a given source vertex $s \in V$.

[1]Recall, a graph is dense if the number of edges is at least asymptotically quadratic in the number of vertices, i.e., $|E| = \Omega(|V|^2)$.

Problem 7-3. [10 points] **Weighted Graph Radius**

In a weighted directed graph $G = (V, E)$, the **weighted eccentricity** $\epsilon(u)$ of a vertex $u \in V$ is the shortest weighted distance to its farthest vertex v, i.e., $\epsilon(u) = \max\{\delta(u, v) \mid v \in V\}$. The **weighted radius** $R(G)$ of a weighted directed graph $G = (V, E)$ is the smallest eccentricity of any vertex, i.e., $R(G) = \min\{\epsilon(v) \mid u \in V\}$. Given a weighted directed graph G, where edge weights may be positive or negative but G contains no negative-weight cycle, describe an $O(|V|^3)$-time algorithm to determine the graph's weighted radius (compare with Problem 2 from PS5).

Problem 7-4. [15 points] **Under Games**

Atniss Keverdeen is a rebel spy who has been assigned to go on a reconnaissance mission to the mansion of the tyrannical dictator, President Rain. To limit exposure, she has decided to travel via an underground sewer network. She has a map of the sewer, composed of n bidirectional pipes which connect to each other at junctions. Each junction connects to at most four pipes, and every junction is reachable from every other junction via the sewer network. Every pipe is marked with its positive integer length, while some junctions are marked as containing identical motion sensors, any of which will be able to sense Atniss if her distance to that sensor (as measured along pipes in the sewer network) is too close. Unfortunately, Atniss does not know the sensitivity of the sensors. Describe an $O(n \log n)$-time algorithm to find a path along pipes from a given entrance junction to the junction below President Rain's mansion that stays as far from motion sensors as possible.

Problem 7-5. [15 points] **Critter Collection**

Ashley Getem (from PS3) is trying to walk from Trundle Town to Blue Bluff, which are both clearings in the Tanko region. She has a map of all clearings and two-way trails in Tanko. Each of the n clearings connects to at most five trails, while each trail t directly connects two clearings and is marked with its length ℓ_t and capacity c_t of critters living on it, both positive integers. Ashley is a compulsive collector and will collect every critter she comes across by throwing an empty Pocket Sphere at it (which will fill the Pocket Sphere so it can no longer be used). If she encounters a critter without an empty Pocket Sphere, she will be sad. Whenever Ashley reaches a clearing, all critters on all trails will respawn to max capacity. Some clearings contain stores where Ashley can buy empty Pocket Spheres, and deposit full ones. Ashley has more money than she knows what to do with, but her backpack can only hold k Pocket Spheres at a time. Given Ashley's map, describe an $O(nk \log(nk))$-time algorithm to return the shortest route for Ashley to walk from Trundle Town (with a backpack full of empty Pocket Spheres) to Blue Bluff without ever being sad, or return that sadness is unavoidable.

Problem 7-6. [40 points] **Shipping Servers**

The video streaming service UsTube has decided to relocate across country, and needs to ship their servers by truck from San Francisco, CA to Cambridge, MA. They will pay third-party trucking companies to transport servers from city to city. An intern at UsTube has compiled a list R of all n available trucking routes; each available route $r_i \in R$ is a tuple (s_i, t_i, w_i, c_i) where s_i and t_i are respectively the names[2] of the starting and ending cities of the trucking route, w_i is the positive integer weight capacity of the truck, and c_i is the positive integer cost of shipping along that route (cost is the same for shipping any weight from 0 to w_i). Note that the existence of a shipping route from s_i to t_i does not imply a shipping route from t_i to s_i. Some of UsTube's servers are too heavy to fit on any truck, so they will need to transfer them to smaller servers for the move. Assume that it is possible to ship some finite weight from San Francisco to Cambridge via some routes in R. Help UsTube evaluate their shipping options.

(a) [5 points] (Useful Digression) Given a weighted path π, its **bottleneck** is the minimum weight of any edge along the path. Given a directed graph containing vertices s and t, let $b(s,t)$ denote the maximum bottleneck of any path from s to t, and let $I(t)$ denote the set of incoming neighbors of t. Argue that $b(s,t) \geq \min(b(s,v), w(v,t))$ for every $v \in I(t)$, and that $b(s,t) = \min(b(s,v^*), w(v^*,t))$ for at least one $v^* \in I(t)$.

(b) [10 points] Assuming that the number of cities appearing in any of the n trucking routes is less than $3\sqrt{n}$, describe an $O(n)$-time algorithm to return both: (1) the weight w^* of the largest single server that can be shipped from San Francisco to Cambridge via a sequence of trucking routes, and (2) the minimum cost to ship such a server with weight w^*.

(c) [25 points] Write a Python function `ship_server_stats(R, s, t)` that implements your algorithm from (b). You can download a code template containing some test cases from the website. Submit your code online at `alg.mit.edu`.

```python
def ship_server_stats(R, s, t):
    '''
    Input:  R | a list of route tuples
            s | string name of origin city
            t | string name of destination city
    Output: w | maximum weight shippable from s to t
            c | minimum cost to ship weight w from s to t
    '''
    w, c = 0, 0
    ##################
    # YOUR CODE HERE #
    ##################
    return w, c
```

[2]Assume names are ASCII strings that can each be read in constant time.