Practice Problems 7

Problem 1. Floyd-Warshall Recursion.

The top-down Floyd-Warshall Algorithm is given as follows:

Algorithm 1 Floyd-Warshall(w)

```
1:  for i ← 1 to n do
2:      for j ← 1 to n do
3:          d[i, j] ← w[i, j]
4:          h[i, j] ← ∅
5:  for k ← 1 to n do
6:      for i ← 1 to n do
7:          for j ← 1 to n do
8:              newCost ← d[i, k] + d[k, j]
9:              if newCost < d[i, j] then
10:                 d[i, j] ← newCost
11:                 h[i, j] ← k
12: return d[ ], h[ ]
```

Algorithm 2 get-path(i, j)

```
1: if h[i, j] = ∅ then
2:     return (i, j)
3: else
4:     mid ← h[i, j]
5:     return get-path(i, mid) ⊕ get-path(mid, j)
```

It was noted in the notes that if the "$<$" on line 9 of the Floyd-Warshall algorithm was changed to a $\leq$, the get-path function could cycle endlessly.

1. Explain what the $h[i, j]$ array represents in the output of the Floyd-Warshall algorithm.
2. Provide an example of a graph for which this change would result in infinite recursion in the get-path function. Demonstrate the Floyd-Warshall algorithm on this graph, and show how it results in an infinite recursion in the get-path function on some vertices i, j.

Problem 2. Network Flows: Alternative Cases

1. Note that the definition of a flow f requires that $f(e) \geq 0$. Consider what the problem with allowing negative flow would be. Draw a graph for which allowing negative flow would make the question of maximum s, t flow unanswerable.

2. We consider an **Undirected Flow Network** to be an undirected graph $G = (V, E)$ with source and sink s, t in which each undirected edge e has a capacity $c(e)$. We let G^D represent the **directed extension** of G, where for each edge $e = (u, v)$, G^D instead has directed edges (u, v) and (v, u). A flow f on an undirected G is defined as a flow on the directed G^D.

   Show that if G^D has a flow f of value k, then there exists a flow f' of value k such that for each edge $e = (u, v)$, at least one of $f'(u, v)$ or $f'(v, u)$ is zero.

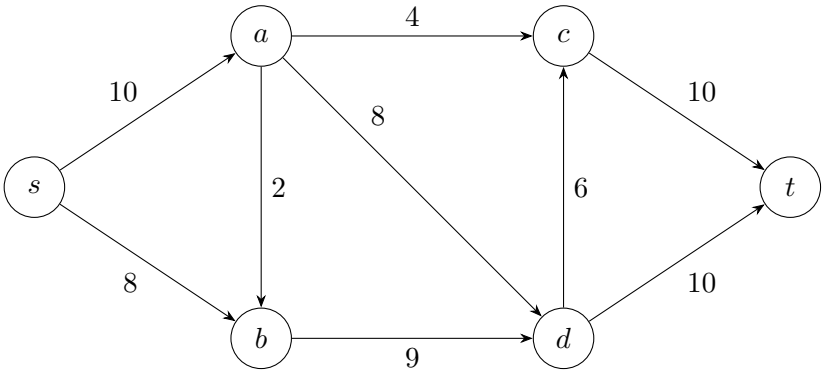
   Hint: Consider what it means for flow to be going both ways between two vertices

3. Give an example of a directed graph $G = (V, E)$, such that if G was made into an undirected graph G', the maximum s, t flow of G'^D would be greater than that of G.

4. Suppose you are given a device which is able to find the maximum flow for a graph $G = (V, E)$ with source and sink s, t. However, this device requires that $c(e) = 1$ for all edges. Come up with an efficient procedure so that you could use this device to find maximum s, t flow for any graph.

Problem 3. Network Flows: A Worked Example:

Demonstrate the Ford-Fulkerson Algorithm as given in class on the following graph, and determine the flow function f which obtains the optimal s, t flow.



Problem 4. Increased Capacity

Suppose you are given a graph $G = (V, E)$ with capacity function c, source and sink s, t, and maximum s, t flow k. You increase the capacity of a single edge e by r, creating new capacity function c'. Show that the maximum s, t flow under c' can be at most $k + r$.

Hint: Consider the how this affects the residual graph for the initial maximum flow.