Name: Pourna Sengupta

ID: 109086577

CSCI 3104, Algorithms
Homework 6 (100 points)

Escobedo & Jahagirdar
Summer 2020, CU-Boulder

Advice 1: For every problem in this class, you must justify your answer: show how you arrived at it and why it is correct. If there are assumptions you need to make along the way, state those clearly.

Advice 2: Verbal reasoning is typically insufficient for full credit. Instead, write a logical argument, in the style of a mathematical proof.

Instructions for submitting your solution:

- The solutions **should be typed**, we cannot accept hand-written solutions. Here's a short intro to **Latex**.
- In this homework we denote the asymptomatic *Big-O* notation by $\mathcal{O}$ and *Small-O* notation is represented as o.
- We recommend using online Latex editor **Overleaf**. Download the **.tex** file from Canvas and upload it on overleaf to edit.
- You should submit your work through **Gradescope** only.
- If you don't have an account on it, sign up for one using your CU email. You should have gotten an email to sign up. If your name based CU email doesn't work, try the identikey@colorado.edu version.
- Gradescope will only accept **.pdf** files (except for code files that should be submitted separately on Canvas if a problem set has them) and **try to fit your work in the box provided**.
- You cannot submit a pdf which has less pages than what we provided you as Gradescope won't allow it.

Name: Pourna Sengupta

ID: 109086577

CSCI 3104, Algorithms **Escobedo & Jahagirdar**

Homework 6 (100 points) **Summer 2020, CU-Boulder**

Piazza threads for hints and further discussion

Piazza Threads
Question 1
Question 2
Question 3
Question 4
Question 5
Question 6
Question 7
Question 8

Recommended reading:

Graph Algorithms Intro: Ch. 22 → 22.1, 22.2, 22.3

Graph Algorithms SSSPs: Ch. 24 → 24.3

Name: Pourna Sengupta

ID: 109086577

CSCI 3104, Algorithms **Escobedo & Jahagirdar**

Homework 6 (100 points) **Summer 2020, CU-Boulder**

1. (5 pts) How many unique MSTs does the following graph have. Show the necessary work to justify your answer.

HW6/p1.PNG

The graph has two unique MSTS. Kruskal's algorithm can be used to solve this by repeatedly adding to F the smallest weight edge that is safe. This way, no loops are created when adding edges. The edges with weight form 1 to 6 are added first since they do not form loops. Safe edges with the weight 7 are added next. The edges B $\rightarrow$ F and B $\rightarrow$ G can be added or B $\rightarrow$ F and C $\rightarrow$ G. Other edges or additions will create loops and therefore do not add safe edges. Therefore, the graph has two unique MSTs.

2. (20 pts) Based on the following graph :

   (a) (10 pts) In what order would Prim's algorithm add edges to the MST if we start at vertex A?

HW6/p2.png

(b) (10 pts) In what order Kruskal's would add the edges to the MST?

Name: Pourna Sengupta

ID: 109086577

CSCI 3104, Algorithms **Escobedo & Jahagirdar**

Homework 6 (100 points) **Summer 2020, CU-Boulder**

3. (10 pts) Suppose that you have calculated the MST of an undirected graph $G = (V, E)$ with positive edge weights. If you increase each edge weight by 5, will the MST change? Prove that it cannot change or give a counterexample if it changes. (Note: Your proof, if there is one, can be a simple logical argument.)

The MST does not change when increasing each edge weight by 6 on an undirected graph G = (V,E). When using Kruskal's, the algorithm traverses the tree to find the edge with the minimum weight and adding it if not already added. This prevents loops. Adding a constant 5 to each edge does not change which edge is the minimum.

Name: Pourna Sengupta
ID: 109086577

CSCI 3104, Algorithms **Escobedo & Jahagirdar**
Homework 6 (100 points) **Summer 2020, CU-Boulder**

4. (15 pts) Suppose you are given the minimum spanning tree T of a given graph G (with n vertices and m edges) and a new edge $e = (u, v)$ of weight w that will be added to G. Give an efficient algorithm to find the MST of the graph $G \cup e$. Your algorithm should run in $O(n)$ time.

```
\\rough psuedocode

effMST(T, u, v){
    \\add edge to T
    edge + T
    find path from u to v
    find max edge weight by comparing each weight to the max
        if greater, set as new max
        continue traversing tree till end
    if max edge weight is greater than new edge
        remove max weight edge and new edge
    else if new edge is greater
        remove new edge
        MST is unchanged
}

This algorithm runs in O(n) because the operations are constant in time complexity
```

Name: Pourna Sengupta
ID: 109086577

5. (20 pts) In a directed graph $G = (V, E)$ with positive edge weights, we define the maximum weight of a path from s to t to be the maximum of the edge weights along the path. For example, if the path from s to t has edges with weights $e_1 = 10, e_2 = 15, e_3 = 5$ then the maximum of the path is $max(e_1, e_2, e_3) = 15$. Give an algorithm to compute the smallest maximum weight paths from a source vertex s to all other vertices. (Hint: Your algorithm should be a modification of Dijkstra's algorithm)

```
smallMaxW(G, s, t){
    for(u != s) {
        int W(s) = infinity;
        int W(u) = 0;
    }
    while(Q != empty) do{
        u = Q.removeMax();
        for(vertexes adjacent to u and in Q){
            if(min(weight(u, v), W(u)) > W(v)){
                W(v) = min(weight(u, v), W(u))
                update v in Q
            }
        }
    }
    return W(t);
}
```

6. (2 pts) What are the two conditions that must be met for the flow on a graph to be valid?

7. (3 pts) What do the edge weights in the residual graph G_f represent? Include the definition of both forward and backward edges.

Name: Pourna Sengupta
ID: 109086577

8. (25 pts) Based on the following network and the given edge capacities answer the following.

HW6/p8.png

(a) (15 pts) Suppose we start the Ford-Fulkerson algorithm with **A** being the source vertex and **F** being the sink and **select the path** $A- > C- > B- > E- > D- > F$ **in the first iteration (Do not chose the first A-F path on your own).** Complete all the iterations of Ford-Fulkerson to find the Max-Flow (including the first round that is incomplete). Clearly show each round with

i. The path that you are selecting in that round.
ii. The bottleneck edge on this path.
iii. The additional flow that you push from the source by augmenting (pushing maximum allowed flow along) this selected augmenting path.
iv. The residual graph with the residual capacities (on both the forward and backward) edges.

Also, report the Max-Flow after the algorithm terminates.

(b) (5 pts) Draw the graph and show the final flow f(e) for the edges of the original graph when the Ford-Fulkerson algorithm terminates.

(c) (5 pts) Find the minimum capacity cut with respect to the capacities on the original graph. Is this minimum capacity equal to the Max-Flow that you earlier identified? Justify your answer in a sentence. Also, list the edges that are part of the min-cut and are saturated (can't carry any more flow).

Min capacity cut = max flow
Edges are spaced by minimum cut and are saturated: CD, DE, EF

Name: Pourna Sengupta
ID: 109086577

9. ***Extra Credit Question 1 (5 pts)*** *For this extra credit question, please refer the leetcode link provided below or click here. Multiple solutions exist to this question ranging from brute force to the most optimal one. Points will be provided based on Time and Space Complexities relative to that of the most optimal solution.*

   Please provide your solution with proper comments which carries points as well.

   https://leetcode.com/problems/cheapest-flights-within-k-stops/

   `Replace this text with your source code inside of the .tex document`

10. ***Extra Credit Question 2 (5 pts)*** *For this extra credit question, please refer the leetcode link provided below or click here. Multiple solutions exist to this question ranging from brute force to the most optimal one. Points will be provided based on Time and Space Complexities relative to that of the most optimal solution.*

    Please provide your solution with proper comments which carries points as well.

    https://leetcode.com/problems/pacific-atlantic-water-flow/

    `Replace this text with your source code inside of the .tex document`