CS 3000: Algorithms & Data — Spring '20 — Jonathan Ullman

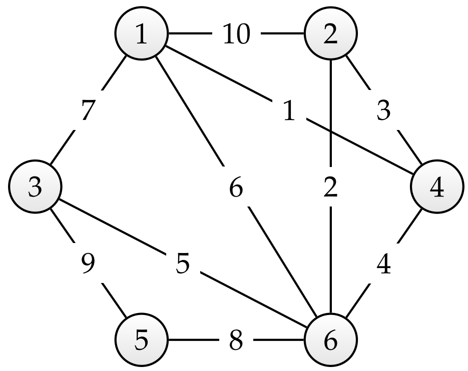
Homework 7
Due Friday Mar 20 at 11:59pm via Gradescope

Name:
Collaborators:

- Make sure to put your name on the first page. If you are using the LaTeX template we provided, then you can make sure it appears by filling in the `yourname` command.
- This assignment is due Friday Mar 20 at 11:59pm via Gradescope. No late assignments will be accepted. Make sure to submit something before the deadline.
- Solutions must be typeset in LaTeX. If you need to draw any diagrams, you may draw them by hand as long as they are embedded in the PDF. I recommend using the source file for this assignment to get started.
- I encourage you to work with your classmates on the homework problems. *If you do collaborate, you must write all solutions by yourself, in your own words.* Do not submit anything you cannot explain. Please list all your collaborators in your solution for each problem by filling in the `yourcollaborators` command.
- Finding solutions to homework problems on the web, or by asking students not enrolled in the class is strictly forbidden.

Problem 1. *MST Practice*

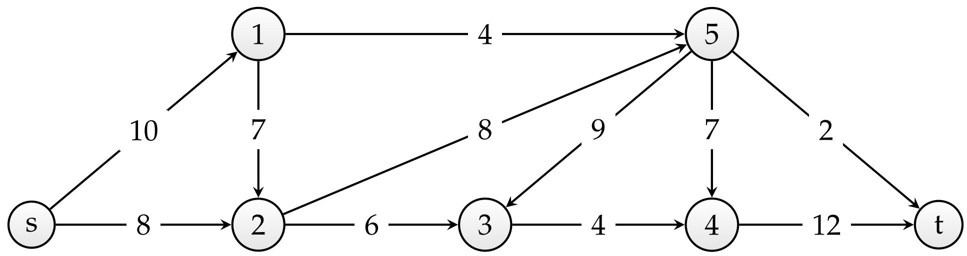
Compute an MST in the following graph. You do not need to justify your answer. **Hint:** Type your solution nearly by copying the LaTeX for the MST and removing some edges.



Solution:

Problem 2. *Flow Practice*

Consider the following flow network $G = (V, E, \{c(e)\}, s, t)$.



(a) Compute a maximum s-t flow in this network and its value. State the sequence of augmenting paths used to obtain the flow. **Hint:** Type your solution nearly by copying the LaTeX for the flow network and labeling edges with their flow values $f(e)$ (preferably even with $f(e)/c(e)$ to help you keep track of the capacity constraints.)

Solution:

(b) Compute a minimum s-t cut in this network and its capacity.

Solution:

Problem 3. *Anti-Kruskal*

In this problem we will see a new algorithm for finding an MST—the *anti-Kruskal* algorithm. Recall that *Kruskal's algorithm* starts wth $T = \emptyset$, considers edges e in *ascending* order of weight, and adds e to T as long as doing so would not create a cycle. The *anti-Kruskal algorithm* starts with $T = E$, considers edges e in *descending* order of weight, and removes e from T as long as doing so would not make T disconnected.

Explain why the anti-Kruskal algorithm outputs an MST T. You may assume that all edge-weights are distinct and you may use the cut and cycle properties of MSTs without proof.

Solution:

Problem 4. *Disabling a Flow Network*

In this problem you will analyze an algorithm for the following problem: you are given a flow network, and you want to remove k edges from the network while reducing the value of the maximum flow by as much as possible. Specifically, you are given the following:

1. a flow network $G = (V, E, s, t, \{c(e)\})$ where every edge has capacity $c(e) = 1$,
2. a maximum s-t flow f^* for G, and
3. a non-negative integer $k \in \mathbb{N}$ such that $k \leq \text{val}(f^*)$.

Given a set of edges $S \subseteq E$, let $G'_S = (V, E \setminus S, s, t, \{c(e)\})$ with the set of edges S removed. The output of the algorithm is the set S containing at most k edges such that the value of the maximum s-t flow in G'_S is as small as possible.

Algorithm 1: Disabling a Flow Network

Function DISABLEFLOW(G, f^*, k)**:**

```
// Find a minimum cut
```

Compute the residual graph G_{f^*}

Let A be the set of nodes reachable from s in G_{f^*} and let $B = V \setminus A$.

```
// Choose any k edges crossing the minimum cut
```

Let S contain any k edges $(u, v) \in E$ such that $u \in A$, $v \in B$

Return S

(a) Analyze the running time of DISABLEFLOW.

Solution:

(b) Prove that the value of the maximum s-t flow in G'_S is at most $\text{val}(f^*) - k$.

Solution:

(c) Prove that the value of the maximum s-t flow in G'_S is at least $\text{val}(f^*) - k$.

Solution: