Problem Set 5

This problem set contains **4 problems**.

This problem set is due after the break **at 10:00pm** on **Monday, March 30, 2020**.

Important: Since we significantly extended the due date for this problem set, *you will NOT be able to use any late days for this assignment.* Solutions will be posted early on *March 31st* so that you have time to review the solutions prior to our midterm.

Please make note of the following instructions:

- This assignment, like later assignments, consists of *exercises* and *problems*. **Hand in solutions to the problems only.** However, we strongly advise that you work out the exercises for yourself, since they will help you learn the course material. You are responsible for the material they cover.
- Remember that the problem set must be submitted on Gradescope. If you haven't done so already, please signup for 6.046 Spring 2020 on Gradescope, with the entry code MNEBKP, to submit this assignment.
- We require that the solution to the problems is submitted as a PDF file, **typeset on LaTeX**, using the template available in the course materials. Each submitted solution should start with your name, the course number, the problem number, your recitation section, the date, and the names of any students with whom you collaborated.
- You will often be called upon to "give an algorithm" to solve a certain problem. Your write-up should take the form of a short essay. A topic paragraph should summarize the problem you are solving and what your results are. The body of your essay should provide the following:
  1. A description of the algorithm in English and, if helpful, pseudocode.
  2. A proof (or indication) of the correctness of the algorithm.
  3. An analysis of the asymptotic running time behavior of the algorithm.
  4. Optionally, you may find it useful to include a worked example or diagram to show more precisely how your algorithm works.

EXERCISES (NOT TO BE TURNED IN)

Streaming

- Devise a streaming algorithm that keeps track of the average and standard deviation of a stream of integers.

Fast Fourier Transform

- Do Exercise 30.1-1 in CLRS on page 905.
- Do Exercise 30.1-4 in CLRS on page 905.
- Do Exercise 30.2-4 in CLRS on page 914.

Minimum Spanning Tree

- Do Exercise 23.1-5 in CLRS (pg. 629)
- Do Exercise 23.2-8 in CLRS (pg. 637)

Problem 5-1. An Enemy of My Enemy [50 points]

Fakebook Inc. is a social media website for the less amicable, where instead of making friends, users can designate "enemies." These relationships are encoded in an undirected graph $G = (V, E)$, where V represents the set of users, and two users v_i and v_j are enemies if and only if $(v_i, v_j) \in E$ (indeed, the enemy relationship must be mutually acknowledged).

Seryl, a social analyst, wants to determine whether the old adage "an enemy of an enemy is a friend" applies to Fakebook's social network. In particular, Seryl wants to determine whether we can partition the vertices into two sets $V_1, V_2 \subseteq V$ of "mutual friends", i.e., where no two users in the same set have an edge between them. In other words, Seryl wants to determine if G is bipartite.

Due to a recent surge in hostility, the graph G has suddenly become *dense*, such that $|E| = \omega(|V|)$ (or equivalently, $|V| = o(|E|)$). Unfortunately, this makes it infeasible to store all of G in memory. Your task is to help Seryl by coming up with an efficient algorithm to determine whether G is bipartite, given a stream of edges and only $O(|V|)$ space.

Note: For the sake of simplicity, assume each vertex can be represented in $O(1)$ space by an integer in $\{1, ..., |V|\}$ (i.e. ignore the bit-complexity of storing integers).

(a) [15 points] **Warm Up:** Let E_k represent the first k edges in the stream, and let $G_k = (V, E_k)$. Devise an algorithm that uses $O(|V|)$ space and returns the smallest k such that there exists a cycle in G_k. Be sure to specify which data structures you are using and analyze their space complexity.

(b) [35 points] Devise an algorithm that determines whether or not G is bipartite using only $O(|V|)$ space. For full credit, your algorithm should run in $O(|E| \log |V|)$.

Hint: Consider modifying your algorithm from (a) to detect when certain properties of bipartite graphs are violated.

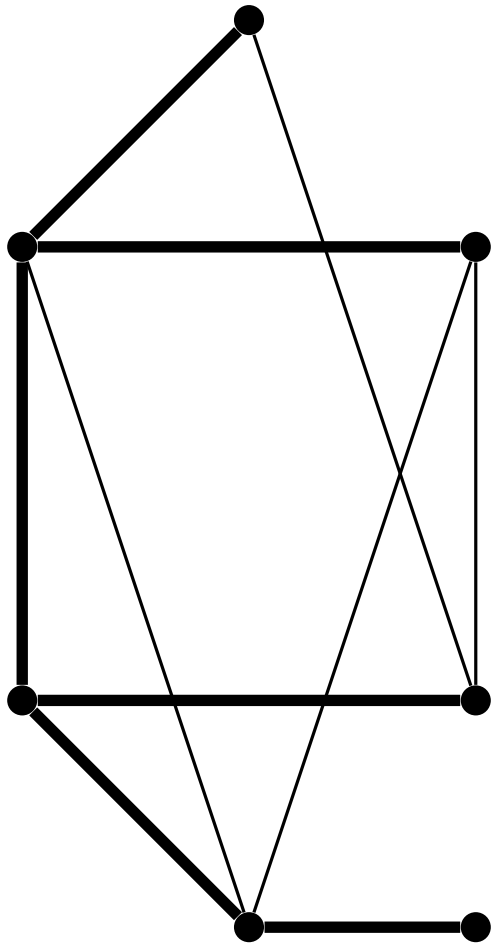
Problem 5-2. Build Your Own MST [15 points] Label the edges of the graph below with weights from the set

$$\{\mathbf{1}, \mathbf{2}, 3, \mathbf{4}, 5, \mathbf{6}, 7, \mathbf{8}, 9, \mathbf{10}\}$$

so that:

- each weight is used *exactly* once;
- the thick edges in the graph below have the weights that are in bold face above;
- the thick edges form an MST in the resulting graph.

No justification is needed (nor will it be considered).



Problem 5-3. The Fellowship of the Ring [25 points]

Balbo Biggins has found three rings covered in mysterious inscriptions. Each ring has a continuous line of n characters drawn from an alphabet Σ.

(a) [10 points] One of the rings seems to only use two of the symbols of Σ. Balbo knows it contains a length-m word s, but wants to find where s appears in the ring. Develop an algorithm that uses FFT to find all occurrences of s in the ring in $O((n+m)\log(n+m))$ time.

(b) [10 points] Balbo inspects the second ring and tries to find a second length-m word s in the inscription, but the ring has several unreadable characters, so he concludes they don't matter. Therefore, an unreadable character on the ring can match *any* one character from s. Develop a $O(|\Sigma|(n+m)\log(n+m))$ algorithm to perform the matching.

(c) [5 points] Balbo looks at the third ring and is relieved that all of the characters are readable! Unfortunately, the length-m word s he wants to match to the ring has some unreadable characters. As before, an unreadable character in s can match *any* one character from the ring. Modify your algorithm from part (b) to solve this under the same runtime constraints.

Problem 5-4. Feedback Form [10 points] Please fill out a feedback form about this problem set at

`https://forms.gle/sB9PwZcWybKS1PHm6`.

It should not take more than a few minutes and will greatly help us improve teaching and material for future semesters!