CIS 121 - Data Structures and Algorithms
Homework Assignment 7

Given: March 13, 2018 **Due:** March 19, 2018

Note: The homework is due **electronically on Gradescope and Canvas** on Monday, March 19 by 11:59 pm EST. For late submissions, please refer to the Late Submission Policy on the course webpage. You may use a maximum of 2 late days on this homework.

A. Gradescope: You must select the appropriate pages on Gradescope. Gradescope makes this easy for you: before you submit, it asks you to associate pages with the homework questions. Failing to do so will get you points off, which cannot be argued against after the fact. Gradescope may prompt you with a warning to select your cover page, please ignore this warning.

B. LaTeX: You must use the `hw121.cls` Latex template provided on the course website, or a harsh penalty will be incurred. Handwritten solutions or solutions not typeset in Latex will not be accepted.

C. Solutions: Please write concise and clear solutions; you will get only a partial credit for correct solutions that are either unnecessarily long or not clear. Please refer to the Written Homework Guidelines for all the requirements.

D. Algorithms: Whenever you present an algorithm, your answer must include 3 separate sections:

1. A precise description of your algorithm in English. No pseudocode, no code.
2. Proof of correctness of your algorithm.
3. Analysis of the running time complexity of your algorithm.

E. Collaboration: You are allowed to discuss **ideas** for solving homework problems in groups of up to 3 people but *you must write your solutions independently.* Also, you must write on your homework the names of the people with whom you discussed. For a clarification on the collaboration policy, please see Piazza @547.

F. Outside Resources: Finally, you are not allowed to use *any* material outside of the class notes and the textbook. Any violation of this policy may seriously affect your grade in the class. If you're unsure if something violates our policy, please ask.

1. [20pts - Arjun's Grand Discovery] During Spring Break, Arjun went exploring the dense forests of Ain-Draham, a town in the North-West of Tunisia. During his exploration, he stumbled upon a suspicious cave. A sign in front of the cave read:

"Proceed at your own peril. Note: this cave consists of a network C of rooms connected with tight, two-way passages. Locals claim that the cave is connected."

Arjun went inside the cave in hopes of making a grand discovery. Arjun did the following: starting at an arbitrary room r, he explored the network C in a depth-first fashion, and recorded the DFS tree T_1 he obtained. He then went back to room r, and explored the network C again but in a breadth-first manner, and recorded the BFS tree T_2 he obtained. He then noticed something interesting: T_1 and T_2 are the same graph T. Arjun vaguely remembers that this means that $C = T$, which meant that he discovered all the passages in the network.

Fearful that he might have missed on a big discovery, Arjun gave you a call and asked you to help him prove that C is indeed equal to T.

2. [20pts - Shirali's Lazy Climb] Having had so much fun camping with the other 121 TAs, Shirali decides to go backpacking in Huffman Park. There are n campsites in the park with trails connecting them, but each trail can only be traveled in a single direction because of their steepness. However, any two campsites, say X and Y might be connected by two trails so that one trail travels from X to Y and the other trail travels from Y to X. Each trail has a positive rating associated with it, used to compare the hiking difficulties of trails.

Huffman Park is constructed in such a way that every campsite is reachable from every other campsite. Shirali wants to visit all the campsites, but she brought a lot of gear with her that she doesnt want to lug around. She decides to leave all her things at campsite H, where she can return to rest. Shirali is also still recovering from a back injury from the bonding activity, so she doesnt want to take the harder routes to get to the campsites.

Help Shirali come up with an efficient algorithm that finds the easiest path of trails (i.e. minimum total difficulty) from every campsite to every other campsite, ensuring that each path passes through campsite H so that she can rest.

3. [15pts - Jess's Nugget-Fry Structure] Deciding to try her hand at *art*, Jess bought n nuggets and $n+8$ fries of distinct lengths (where $n \geq 6$). Using the fries, she joined all the nuggets together to form a giant connected nugget-fry. Every fry has exactly one nugget on each end.

Hungry Steven chanced upon the structure and realized that some fries can still be eaten such that the structure still remains connected. Since he is really hungry, Steven only wants to eat fries that are as long as possible. Give him an $O(n)$ algorithm to find the set of the longest fries that can be eaten, while still leaving the structure connected.

4. [20pts - Hungry Steven Strikes Again] Inspired by the new trend of nugget-fry art, Sam bought n nuggets and m fries of distinct lengths as well to create his own connected nugget-fry structure, such that each fry is connected to two nuggets. Proud of his creation, he left it on display, only for it to also be discovered by Hungry Steven.

This time, Steven wants to eat as many fries as possible such that the remaining nuggets are still connected, and the sum of the lengths of the remaining fries is as small as possible. However, this time he noticed a weird looking fry, f, and decides not to eat it, so f has to remain in the structure. Design an $O(m \lg n)$ algorithm that outputs the structure which remains after Steven eats the fries.

5. [25pts - Roopa Outsmarts Hungry Steven] It seems like nugget-fry art has become the next big thing! Roopa has joined the trend, and decides to try it out herself. She buys herself n nuggets and m fries. Roopa isn't picky though. She doesn't care about the length of her fries.

Roopa makes her nugget-fry structure such that each nugget is connected to at least two other nuggets (via french-fry connections). As if he hadn't already eaten enough, Hungry Steven stumbled across Roopa's structure. This time, Roopa was warned by Hungry Steven's previous victims and imposes some restrictions on the set of nuggets Steven can eat. The conditions that the set of nuggets $N = \{N_1, N_2, ..., N_k\},\ 2 < k \leq n$ Steven can eat must satisfy are:

1. For $1 \leq i < k$, there is a fry between N_i and N_{i+1}, and there is a fry connecting nuggets N_1 and N_k.
2. There are at most $2\lceil \lg n \rceil + 1$ nuggets in the set that are directly connected to three or more other nuggets. The remainder of the nuggets in the set are connected to only two other nuggets.

First prove that such a set N always exists, and then design an algorithm to help Roopa find the set N in linear ($O(m + n)$) time so she can tell Steven what he can eat before the nuggets get cold!