CIS 1210 — Data Structures and Algorithms
Homework Assignment 7

Assigned: March 25, 2025 **Due:** April 07, 2025

Note: The homework is due **electronically on Gradescope** on April 07, 2025 by 11:59 pm ET. For late submissions, please refer to the Late Submission Policy on the course webpage. You may submit this assignment up to 2 days late.

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. Forgetting to do so will incur a 5% penalty, which cannot be argued against after the fact.

B. LaTeX: You must use the LaTeX template provided on the course website, or a 5% 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. Ed will also contain a complete sample solution in a pinned post.

D. Algorithms: Whenever you present an algorithm, your answer must include 3 separate sections. Please see Ed for an example complete solution.

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 more on the collaboration policy, please see the course webpage.

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. [20 pts] Negative Start?

Coach Bruno is leading Team1210 through March Madness, and needs to travel to n different arenas, using the least amount of energy possible. The n arenas are connected by m one-way roads, each requiring a positive amount of energy to travel.

The team starts their journey at Penn Arena, where their sponsor, Rajiv, gives them a special energy drink that gives them energy instead of consuming it for the first road they take. Thus, they use a "negative" amount of energy for the first road they take out of Penn Arena. However, they use positive amounts of energy for all subsequent roads.

Coach Bruno needs to make sure the team conserves the most amount of energy for their championship game, so he decides to use Dijkstra's algorithm to find the minimal energy path to all n arenas from Penn Arena using the m roads connecting them. Coach Bruno needs to determine whether or not Dijkstra's algorithm will still work on this graph, even with the negative initial weight edge. Help our Coach decide if he should use Dijkstra's Algorithm by proving or disproving whether using Dijkstra's on these roads is still valid (that is, whether or not Dijkstra's will correctly return the shortest paths from the source to all other vertices).

Note: Only the first road traveled leaving Penn Arena will be of negative weight. You may assume that there are no negative weight cycles.

2. [20 pts] Selina's Game Strategy

In preparation for Team 1210's upcoming game, Selina, the head strategist, is responsible for making sure every player on the team is locked in and ready to execute the latest playbook update before tip-off.

There are n players on the team. Based on pre-game analysis, Selina has mapped out how information flows between players: if player x talks to player y, then x can explain the playbook to y before the game starts. There are m such communication links between players. Importantly, this communication isn't always symmetric, so if player x can effectively explain the play to player y, that doesn't mean y can effectively explain it to x.

Selina only has time to directly brief one player before the game. From there, players will communicate the playbook to their teammates based on the communication links. She defines a player as "critical" if briefing that one player results in every player on the team eventually learning the play.

Given the n players and the m communication links, design an $O(n + m)$ algorithm to help Selina identify all critical players on the team. **No proof of correctness is required, but please do justify the runtime of your algorithm.**

3. [20 pts] Daniel's Tournament Bracket

Esteemed Coach Daniel has been given the highly coveted responsibility of organizing the official March Madness bracket for the 2025 season! He is given the set T of n teams which are competing and the set G of possible games, where each game is between two teams in T. In addition, each game is associated with the distance that the two teams have to travel to play that game which Daniel wants to minimize across all games. Note that a possible game does not necessarily exist between every two teams in the competition, but that for every two teams it is possible to reach one from another through some set of games.

However, there exists a subset of teams $U \subseteq T$ who are underdogs, as this is their first time competing in March Madness! Daniel is a fervent supporter of any underdog team, so despite strict ethical guidelines, he wants to maximize their odds of winning. To do this, he wants to ensure that each underdog team plays no more than one game.

Propose an $O((n+m)\log n)$ time algorithm that computes a set of games, where every team is connected to every other team via some set of games, the total travel distance is minimized, and all the underdog teams in U play at most one game. You may also return that such a set does not exist.

4. [20 pts] Full Court Trap

To prepare for this year's tournament frenzy, the host cities have set up $n \geq 20$ team hubs equipped with courts, trainers, and media facilities. These team hubs are connected by $n + 67$ undirected shuttle routes, such that it is currently possible to travel from any hub x to any other hub y using the current shuttle routes.

Unfortunately, Alan, infamously known as the "BracketBlaster", has been infiltrating this network. Word in the locker rooms is that Alan has been hijacking shuttle routes to tamper with team logistics and gain advantages.

To ensure team safety, "Buzzer-beater" Bruno has analyzed the risk levels across the shuttle routes. He assigned a positive cost to each route that is directionally proportional to its disruption potential. That is, if a sabotage is more likely to occur on the shuttle between hubs p and q than between hubs r and s, then Bruno assigns route (p, q) a higher cost of travel than route (r, s).

Bruno proposes to deactivate certain routes to reduce the disruption risk as much as possible while preserving the property that it is possible to travel between any two hubs. Help Bruno come up with an O(n) time algorithm that finds and returns the set of routes such that their removal will reduce the total risk the most while maintaining connectivity. You may assume access to a list containing information about each route's endpoints and its corresponding disruption potential.

5. [20 pts] Coach Chen's Hub of Madness

During March Madness, Coach Chen is cooking up a plan for the ultimate tournament showdown. Across the country, there are n arenas connected by m travel routes - each with its own travel time (so the trip from Arena A to Arena B might not take the same time as the reverse). Every arena can reach every other arena via some series of routes, ensuring that all teams can make it to the action.

Now, Coach Chen has a bold idea: he wants to designate one arena as the "Madness Hub" so that every game between any two arenas must funnel through this central venue. The idea is that by forcing all teams to pass through this hub, the electric atmosphere and buzz of March Madness will reach every corner of the tournament.

However, not just any arena will do - only k select arenas are in the running to become this hub. To maximize the excitement, the coach wants to choose the hub such that the total travel time for every team - when routed from their starting arena to the hub and then to their destination - is minimized. That is, he wants to choose some location x as the Madness Hub such that the sum of the travel times from a to x to b for all distinct pairs of locations a and b is minimized.

Your challenge is to design an algorithm that, in $O(k\, m \log n)$ time, determines which one of these k arenas should serve as the Madness Hub. You may assume that there is exactly one optimal arena.