Homework Assignment #9
Due: April 4, 2025 at 5:00 p.m.

The same rules apply as for Assignment 1. (In particular, you can work in pairs, where each pair submits just one paper.)

Both of the following questions can be answered by constructing a graph and then simply using the graph algorithms from the textbook. This means that no sophisticated algorithms need to be designed, but when you describe the construction of the graph, you should precisely specify:

- whether it is directed or undirected,
- the set of nodes,
- whether each pair of nodes is connected by an edge or not,
- weights on edges (if any),
- any other important details (such as the number of nodes or edges in your graph, which may be useful for bounding the running time of your algorithm).

Then, describe how you would use one of the textbook graph algorithms to solve the problem. Try to make your algorithm as efficient as you can.

[5] **1.** You have n pictures of famous cats, and you would like to figure out which pictures are of the same cat. You decide to use crowdsourcing to classify the cat pictures. Various volunteers are shown two different pictures side by side, and they are asked to say whether they are pictures of the same cat or different cats. Then, you want to determine whether the information gathered during this process is consistent. The pictures are numbered 1 to n. You have a set S of s pairs (i, j) where pictures i and j were judged to be of the same cat, and a set D of d pairs (i, j) where pictures i and j were judged to be of different cats. Your goal is to figure out whether all of the information you have could be true, or if there must be at least one error in your data. Give a good bound on the running time of your algorithm in terms of n, s and d.

[5] **2.** In mediaeval Leutonia, decrees by the king were made public by messengers who would carry copies of the decree from city to city, all across the realm. The king has a map of the cities of the realm, and the roads that connect pairs of cities. For each road connecting city x to city y, $p(x, y)$ is the probability that a brigand will abduct a messenger travelling along that road. Assume the brigands who might attack each road operate completely independently of one another. Messengers belong to the Leutonian Messengers' Guild and charge 1 Leutonian drachma to carry any number of copies of a decree from one city to a neighbouring city along a road. The king wishes to come up with a plan to distribute each decree as cheaply as possible. If none of the messengers get abducted, the decree should reach every city. (If some messengers do get abducted, some cities may not receive the decree.) Among all solutions that have minimum cost, the king wishes to choose the one that has the highest probability of successfully delivering the decree to all cities of the realm.

There are c cities and r roads connecting pairs of cities. Give a good bound on the running time of your algorithm to find the optimal solution.

Hint: use logarithms.