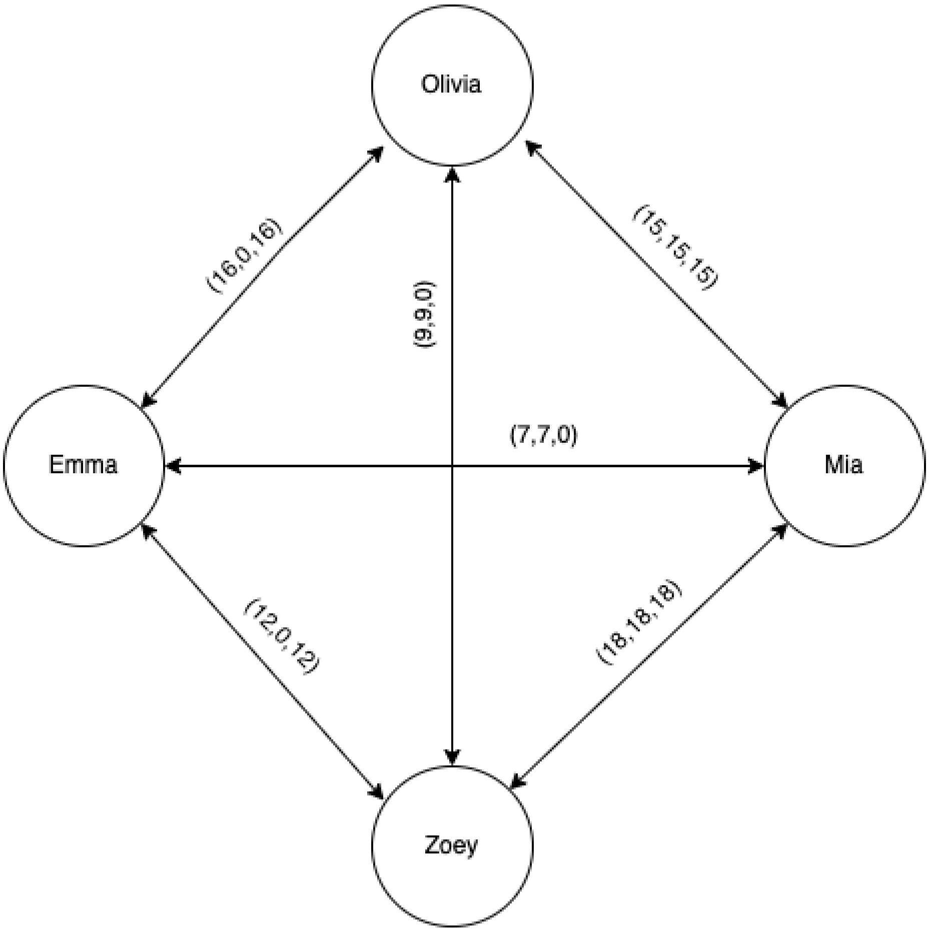
1. Design the architecture of the system component that does the matching and recommends them ladies that they could follow, based on predefined criteria

Having in mind that this component of the system should recommend who to follow and in the end will serve to match ladies for live talking sessions based on their availability, interests and English level, my solution here will be to make a complete, weighted graph to represent ladies and relations between them.

Each **node** in the graph will represent a lady and each **vertice** should represent their connection. The graph is complete, because every lady is connected to each lady, since they are all members of the same platform. The **weight** of a single vertice is a special coefficient that represents how good fit are those two ladies to follow each other or to speak to each other. Following image describes that graph.



In this simple example, four ladies are connected through the platform. The weight of an edge is a tuple of integers denoted with **(f, t, w)** where **f** represents **follow coefficient** and **t** represents **speak coefficient on weekdays** whereas **w** is **speak coefficient on weekends**. They represent how good fit to talk or follow each other are two ladies based on their language level, interests and availability. Follow coefficient is calculated using a simple formula that will be defined as:

$$f = \left[10 - \left|l_1 - l_2\right| + N_t\right]$$

where l_1 represents language level of the first lady (valued 1 to 5) and l_2 represents the language level of the second lady (valued 1 to 5). N_t represents the number of mutual topics of interest (0 to 8, since they get to choose max. 8 topics during the onboarding process).

We can then calculate the **t** as

$$t = f * A_1$$

where A_1 is availability overlap between those to ladies on workdays and can have value of either 1 or 0. When we multiply that by coefficient of matching based on language level and mutual topics we get how likely are the ladies to be a good fit to talk to each other. Similar to that, the **w** is calculated as:

$$w = f * A_2$$

where A_2 is availability overlap between those to ladies on weekends.

After the onboarding is finished, the system should populate the existing graph of members with the newly registered lady. Connect her with existing members and calculate those weights between her and other members.

After that, a simple matching algorithm could be used to determine N ladies that are most suitable for the newly onboarded on to follow or talk to. Algorithm should take all the adjacent nodes (ladies) for the current one and sort descending by their matching coefficient.

An improvement would be to set a threshold or a number of ladies to suggest initially and do the rest of the calculation as a background, asynchronous job. That would still have the same effect on the initial suggestions but would save the time spent on calculations of weights.

Also, the other data that would be useful could be:

- **Age**, so we could connect ladies by similar age if prefered
- **Occupation**, so someone could chose to talk to a programmer on various life and lifestyle topics
- **Extrovert - introvert** scale: an option to assess yourself an introvert or extrovert because someone who is maybe introvert is more comfortable to talking to another introvert person