
Codebase Internal Coffee Chats

Ivon Liu & Ryan Hoque

Introduction

We will try to conduct blind date style coffee chats between two members in Codebase.

Setup

Let E be the set of all members of Codebase that were members since last semester, and N be the set of all members of Codebase that joined this semester. We want to pair members such that for each pair m_1, m_2 , such that if $m_1 \in E$, $m_2 \notin E$. Pairings where $m_1, m_2 \in N$ are allowed. This is to prevent pairings such as Abhi and Tim, because a coffee chat between them would be rather pointless.

Pairing Method

Assuming that $|E| < |N|$, we can pair members using the following algorithm:

```
algorithm pair-members ( $E$ ,  $N$ ):  
    rand_E = randomize( $E$ )  
    rand_N = randomize( $N$ )  
    pairings = []  
    for i = 1 to length(rand_E):  
        pairings.append((rand_E[i], rand_N[i]))  
    for i = length(rand_E)+1 to length(rand_N), skip by 2:  
        pairings.append((rand_N[i], rand_n[i+1]))  
    return pairings
```

Essentially, we are pairing each returning member with a new member. And based on the assumption that there are more new members than existing members, we then pair off the remaining new members. This algorithm will minimize the chances of pairing two people who already know each other, based on the assumption that two members will know each other well enough if and only if they are both returning members.

Post-Pairing

After creating the pairs, we will need each member of the pair to agree on a time and place to meet, without revealing to either party who they are meeting with. To accomplish this, each member of each pair will be sent a link to a when2meet or equivalent, where they will mark the times they are available. Let T_1 be the times where m_1 is available, and T_2 be the times where m_2 is available, for any pair (m_1, m_2) . Then, a random time T will be selected where $T \in T_1 \cap T_2$.

Similarly, each member will be requested to rank their preferences for places where they would like to meet. Then, let $r_{1,i}$ be the ranking for place p_i , where $i \in \{1, \dots, n\}$ for member m_1 of the pair. Similarly, $r_{2,i}$ is the ranking for m_2 . The ranking will be from 1 to n , with 1 being most preferred. Then, the score s_i for place p_i will be $(n - r_{1,i}) + (n - r_{2,i})$. The place with the maximum score will be the place the two members will meet.

Each member will then be sent the time and location of the meeting, as determined by the above method. The two members will then meet and have a nice coffee chat. Ideally, all this will be automated as to minimize the number of people (including internal committee members) that will know who their coffee chat partner is.