

微積のテストを楽にするアプリ

谷口 輝海

結果

- 打倒阿原Tを掲げて取り組みましたが返り討ちにあいました。
- なので代替りのものを作ってきました。

合コンとかで使えるかも?な マッチングアプリ

谷口 輝海

作ったもの

- n 対 n ($1 \leq n \leq 5$)のマッチングアプリ
- マッチングしたい順に相手を評価したリストに基づいてマッチングさせる。

こんなときにも使えるかも

- 合コンの終盤で連絡先を交換する際に少しゲーム的要素をつけたいとき



※合コン…出会いを求める男女数人が集まって行われる懇親会

使ったツール

- Python3
- PySimpleGUI (<https://pysimplegui.readthedocs.io/en/latest/>)
 - 文字通り簡単にGUIが作れるライブラリ。
 - 辞書とリストを使えれば誰でも簡単にデスクトップアプリが作れる。

使ったアルゴリズム

- Gale-Shapleyのアルゴリズム
 - できるだけ全ての人の希望に沿ったマッチングを求めるためのアルゴリズム
 - 研究室配属にも使われてるらしい

使ったアルゴリズム

Gale-Shapley アルゴリズム

```
1:  $M = \emptyset$  (空集合) とし, 全員をフリーにする.  
2: while 希望リストが空でないフリーの男性がいる  
3:   任意のそのような男性を  $m$  とする.  
4:    $m$  の希望リストの先頭の女性を  $w$  とする.  
5:   if  $w$  がフリー  
6:      $(m, w)$  を  $M$  に加え,  $m$  と  $w$  を婚約中にする.  
7:   else (つまり  $w$  が婚約中)  
8:      $w$  の婚約相手を  $m'$  とする.  
9:     if  $w$  は  $m$  より  $m'$  が好き  
10:       $m$  の希望リストから  $w$  を削除する.  
11:     else  
12:       $M$  から  $(m', w)$  を削除し,  $(m, w)$  を加える.  
13:       $m'$  をフリーに,  $m$  を婚約中にする.  
14:       $m'$  の希望リストから  $w$  を削除する.  
15:     end if  
16:   end if  
17: end while  
18:  $M$  を出力する.
```

こんな感じのやつ

結構シンプル!

実際に使ってみる

The screenshot shows a window titled "Stable Matching" with a dark blue background. It contains two sections of preference lists, separated by a dashed line. The top section is for "グループ1" (Group 1) and the bottom for "グループ2" (Group 2). Each section has five rows, one for each member. Each row consists of a label (e.g., "グループ1/お名前1:"), a text input field for the member's name, a label "希望順:" (Preference Order), and a series of five text input fields for the preference order, separated by greater-than symbols (>). At the bottom, there are two buttons: "マッチング" (Matching) and "クリア" (Clear). Below the buttons is a label "組合せ結果:" (Combination Result) followed by a long, empty text input field.

グループ	お名前	希望順	1	2	3	4	5
グループ1	お名前1:	M1	W1	W2	W3	W4	W0
	お名前2:	M2	W2	W3	W4	W0	W1
	お名前3:	M3	W3	W4	W0	W1	W2
	お名前4:	M4	W4	W0	W1	W2	W3
	お名前5:	M0	W0	W1	W2	W3	W4
グループ2	お名前1:	W1	M1	M2	M3	M4	M0
	お名前2:	W2	M2	M3	M4	M0	M1
	お名前3:	W3	M3	M4	M0	M1	M2
	お名前4:	W4	M4	M0	M1	M2	M3
	お名前5:	W0	M0	M1	M2	M3	M4

マッチング クリア

組合せ結果: _____

実行すると上のようなウィンドウ表示される。

実際に使ってみる

Stable Matching

グループ1/お名前1:	<input type="text" value="A"/>	希望順:	<input type="text" value="W"/>	>	<input type="text" value="Y"/>	>	<input type="text" value="Z"/>	>	<input type="text" value="X"/>	>	<input type="text"/>
グループ1/お名前2:	<input type="text" value="B"/>	希望順:	<input type="text" value="X"/>	>	<input type="text" value="W"/>	>	<input type="text" value="Z"/>	>	<input type="text" value="Y"/>	>	<input type="text"/>
グループ1/お名前3:	<input type="text" value="C"/>	希望順:	<input type="text" value="X"/>	>	<input type="text" value="Z"/>	>	<input type="text" value="W"/>	>	<input type="text" value="Y"/>	>	<input type="text"/>
グループ1/お名前4:	<input type="text" value="D"/>	希望順:	<input type="text" value="Z"/>	>	<input type="text" value="X"/>	>	<input type="text" value="Y"/>	>	<input type="text" value="W"/>	>	<input type="text"/>
グループ1/お名前5:	<input type="text"/>	希望順:	<input type="text"/>	>	<input type="text"/>	>	<input type="text"/>	>	<input type="text"/>	>	<input type="text"/>

グループ2/お名前1:	<input type="text" value="X"/>	希望順:	<input type="text" value="C"/>	>	<input type="text" value="B"/>	>	<input type="text" value="A"/>	>	<input type="text" value="D"/>	>	<input type="text"/>
グループ2/お名前2:	<input type="text" value="Y"/>	希望順:	<input type="text" value="A"/>	>	<input type="text" value="D"/>	>	<input type="text" value="C"/>	>	<input type="text" value="B"/>	>	<input type="text"/>
グループ2/お名前3:	<input type="text" value="Z"/>	希望順:	<input type="text" value="D"/>	>	<input type="text" value="B"/>	>	<input type="text" value="C"/>	>	<input type="text" value="A"/>	>	<input type="text"/>
グループ2/お名前4:	<input type="text" value="W"/>	希望順:	<input type="text" value="B"/>	>	<input type="text" value="D"/>	>	<input type="text" value="C"/>	>	<input type="text" value="A"/>	>	<input type="text"/>
グループ2/お名前5:	<input type="text"/>	希望順:	<input type="text"/>	>	<input type="text"/>	>	<input type="text"/>	>	<input type="text"/>	>	<input type="text"/>

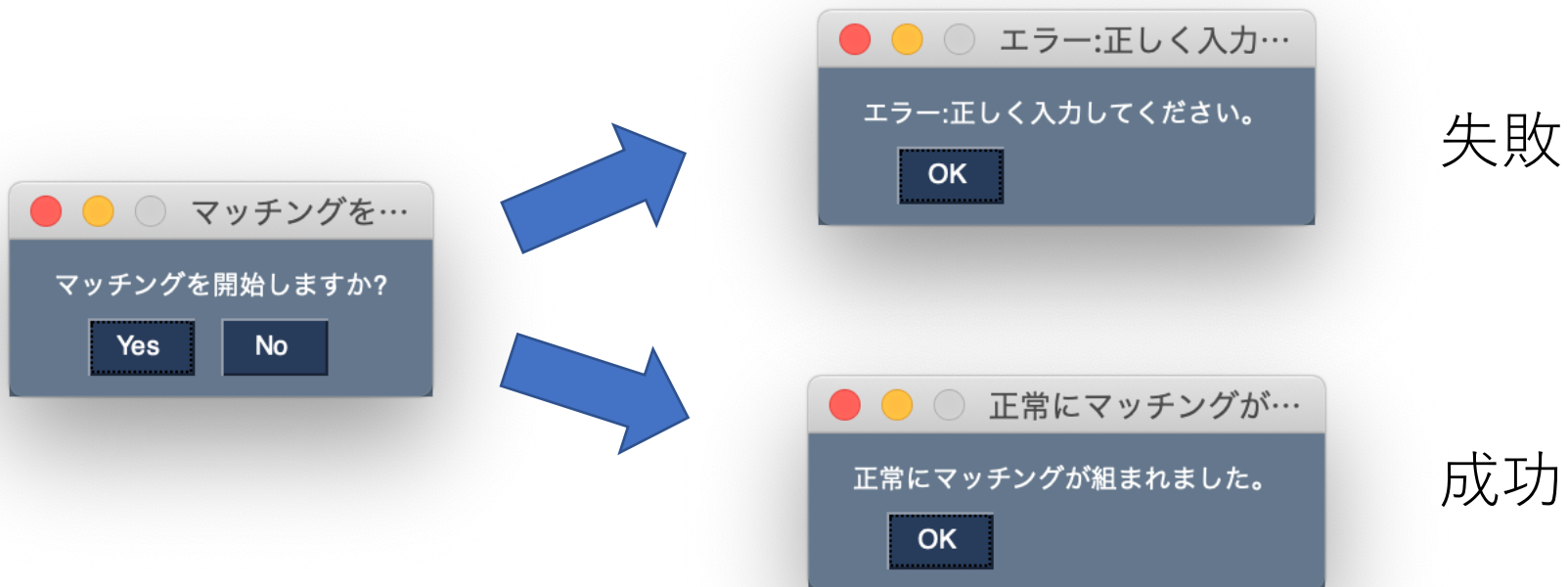
組合せ結果:

出席者の名前と希望順に相手の名前を入力。(5人未満の場合は空欄)

クリアボタンを押せば全リセットされます。

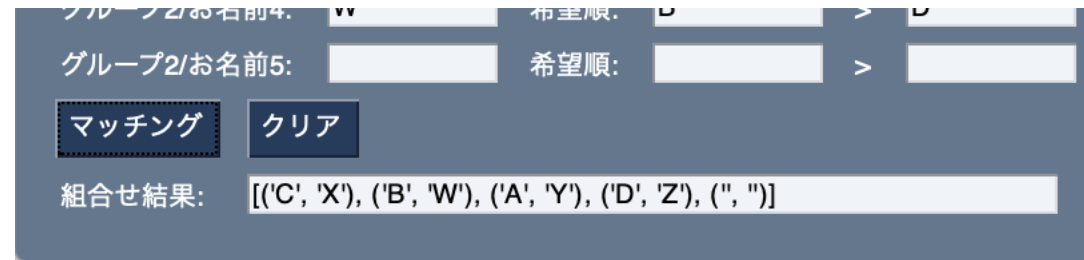
実際に使ってみる

- 全員が入力し終わったら左下のマッチングボタンを押す。



実際に使ってみる

- 成功したら組合せ結果の欄にマッチング結果が表示される。



The screenshot shows a web interface for a matching tool. It features two input fields for group names and two for desired order, with arrows indicating a flow. Below these are two buttons: 'マッチング' (Matching) and 'クリア' (Clear). The 'マッチング' button is highlighted with a dashed border. Below the buttons is a text area labeled '組合せ結果:' (Combination Result) which displays the output: `[('C', 'X'), ('B', 'W'), ('A', 'Y'), ('D', 'Z'), ('', '')]`.

- `[('C', 'X'), ('B', 'W'), ('A', 'Y'), ('D', 'Z'), ('', '')]`が結果となる。

疑問点

入力時に全員のリストが見えたら嘘つく人が出るのでは?
→嘘をついても結果が改善されることはない!

疑問点

全員が第一希望にならないなら裏で勝手に駆け落ちするペアが出てくるのでは？

→「自分の現在の相手よりも希望が上の人」が現在マッチングしている相手は希望リストの順位が自分よりも必ず上!

Ex)

出力マッチング: (m, w) m の希望順: $[a > w > c > d]$

(m', a) のとき a の希望順で $m' > m$

改善点・問題点

- 入力が面倒な点
 - 参加者名だけ書いて希望順は一覧から選択できるようにする？
- プログラムの組み方によってグループごとに有利不利が出る。
 - 相手にアルゴリズムなどに強い人がいないことを祈る

感想・反省

- 思えば合コンたるものに行ったことなかった。
- python は記述がわかりやすくプログラミングが苦手な僕でも調べながら結構進められた。