



# 유학생 도우미 Matching

Assignment #6 : 외국인 유학생 도우미 매칭 최적화

# Buddy Matching?

The image shows a screenshot of the Sungshil University (SSU) website's 'Buddy Matching' program page. The page is in Korean and features a header with the university's logo and navigation links. The main content area is titled '참여 프로그램' (Participation Program) and includes a section for '외국인 유학생도우미 모집' (Recruitment of International Student Buddies). Below this, there is a flyer for the '44th Buddy Matching' event, which includes details about the program's purpose, activities, and contact information.

**외국인 유학생도우미 모집**

**모집기간**  
2021.09.06 (월) ~ 2021.09.29 (수) 23:00

**활동기간**  
2021.10.04 (월) ~ 2021.11.19 (금) \*총 7주  
매칭결과 발표 및 오리엔테이션 10월1일(금)  
15:00/Zoom으로 진행, 필참

**모집대상**  
송실대학교 학부생이라면 누구나!  
외국어가능자 우대 (일본어, 중국어, 베트남어, 영어)  
친화력 있는 학생! 꾸준히 참여할 학생!

**지원방법**  
FUN시스템에 신청서 제출 (fun.ssu.ac.kr) > 글로벌  
'2021 가을 학기 한국어 과정 외국인 유학생  
도우미 44기' 신청

**혜택**  
프로그램 수료 도우미 장학금 5만원 수여  
유학생 도우미 44기 수료증 발급  
봉사활동확인서 발급

**문의처** 국제교육원 02-820-0778

❑ 송실대학교 편시시스템에서 봄, 여름, 가을, 겨울 학기마다 일년에 총 4번 외국인 유학생 도우미를 모집한다.

❑ 여기서의 외국인 유학생은 교환학생이 아닌, 일본, 중국, 러시아 등 다양한 국적으로 이루어진 국제교육원 어학당 소속의 유학생들이다.

❑ 코로나 이전에도 도우미 학생(한국인)이 부족하여 오직 신입 유학생만을 대상으로 프로그램을 진행하였지만, 코로나로 인해 학생들의 지원이 더욱 저조해지고 있다.

# What is problem?

## 2. 매칭방식 및 활동내용

Step 1. 송실대 재학생(도우미) : 외국인 유학생 = 1:1~2 매칭 → 1쌍

Step 2. 총 7주 동안 7번 외국인 친구와의 만남 후 간단한 보고서 제출

- 학교생활 및 한국생활 안내
- 캠퍼스투어, 은행 및 도서관 방문 보조, USIM구입 등 생활 보조
- 한국어 학습 도움, 언어교환

※ 활동은 유학생과 의논하여 자유롭게 진행하시면 됩니다.

※ 매 주차 매칭된 학생들과 개별모임을 진행하시면 되고,

이번 학기는 코로나19로 인해 조 편성 및 단체활동(매칭행사 등)을 진행하지 않습니다.

※ (중요!!) 코로나19 사회적 거리두기 단계에 따라 1~1.5단계 시 대면활동 진행, 2단계 이상 시 비대면 활동으로 진행(단, 비대면 활동 시 시간제한 없음) 사회적 거리두기 단계에 따라 대면활동을 진행하지 못할 수도 있으며, 이에 동의하는 학생들만 신청 바랍니다.

<안내공지>

- 기존에는 도우미 학생 1명당 1명의 유학생이 매칭되었지만, 도우미 학생의 지원이 줄어들어 **도우미 학생 1명당 2명의 유학생이 매칭**되고 있다.
- 여기서 도우미들과 잘 맞는 유학생들과의 매칭이 이루어져야 하는데, 100명이 넘는 인원들을 대상으로 수작업을 하여 매칭을 시키다 보니 프로그램 참가자 들의 항의가 늘어나고 있다.
- 1:2매칭이 이루어짐에 따라 유학생들 간의 항의도 늘어나고 있는 실정이다.

=> 학생들의 인적사항, 지원정보를 바탕으로 **최적의 1:1, 1:2 매칭**을 만들

# Data Collection

<인적사항+지원정보>

	A	B	C	D	E	F		A	B	C	D	E	F	G	H
1	유학생	성별	나이	출신언어권	기숙사 거주여부	한국어 급수	1	한국학생	성별	나이	매칭 희망 국가 (1지망)	매칭 희망 국가 (2지망)	매칭 희망 국가 (3지망)	중국어 자격증 or 학과	일본어 자격증 or 학과
2	1	여자	23	대한	o	2	2	1	여자	25	일본	영미	중국	x	o
3	2	여자	26	대한	o	2	3	2	남자	26	일본	영미	국적무관	x	x
4	3	여자	33	대한민국	o	2	4	3	남자	23	일본	중국	국적무관	x	x
5	4	남자	28	독일	x	2	5	4	남자	22	중국	국적무관	국적무관	x	x
6	5	여자	25	러시아	o	2	6	5	여자	22	영미	러시아	일본	x	x
7	6	여자	19	러시아	o	1	7	6	남자	27	일본	영미	러시아	x	x
8	7	여자	19	러시아	o	2	8	7	남자	24	일본	러시아	국적무관	x	x
9	8	여자	20	러시아	o	2	9	8	남자	25	중국	영미	일본	o	x
10	9	여자	21	러시아	x	1	10	9	여자	20	일본	베트남	중국	x	x
11	10	여자	31	러시아	o	1	11	10	여자	23	영미	일본	국적무관	x	x
12	11	여자	21	베트남	o	2	12	11	남자	20	일본	일본	일본	x	x
13	12	여자	26	베트남	o	2	13	12	남자	25	일본	영미	국적무관	x	x
14	13	남자	20	베트남	o	2	14	13	남자	29	일본	베트남	중국	x	x
15	14	여자	21	베트남	o	1	15	14	남자	23	일본	국적무관	국적무관	x	o
16	15	여자	20	베트남	o	2	16	15	남자	24	일본	러시아	영미	x	o
17	16	여자	22	베트남	o	1	17	16	여자	21	중국	영미	국적무관	o	x
18	17	여자	20	베트남	o	2	18	17	남자	22	일본	중국	영미	x	x
19	18	여자	20	베트남	o	2	19	18	남자	26	일본	국적무관	국적무관	x	x
20	19	여자	20	일본	o	2	20	19	남자	23	영미	일본	국적무관	x	x
21	20	여자	22	일본	o	2	21	20	여자	21	중국	영미	일본	o	x
22	21	여자	22	일본	o	1	22	21	남자	24	일본	중국	영미	x	x
23	22	여자	25	일본	o	2	23	22	여자	24	국적무관	국적무관	국적무관	x	x
24	23	남자	29	일본	o	2	24	23	남자	24	국적무관	국적무관	국적무관	x	x
25	24	남자	38	일본	o	2	25	24	남자	26	영미	러시아	일본	x	x
26	25	남자	19	일본	x	2	26	25	남자	22	일본	영미	중국	x	x
27	26	여자	26	일본	o	2	27	26	여자	24	일본	중국	국적무관	x	x
28	27	여자	20	일본	o	1	28	27	남자	25	중국	일본	국적무관	x	x

<유학생data>

<도우미data>

- 현재 한국어학당에서 근로를 하고 있어 해당 프로그램의 담당자 기\*나 선생님의 도움을 얻어 큰 어려움 없이 학생들의 인적사항+지원정보를 얻었다(왼쪽사진)
- 해당 문서는 가을학기 Data이며 안에는 학생들의 나이, 학과, 매칭희망국가(1,2,3지망), 한국어 급수 등의 정보가 있으며 이를 오른쪽과 같이 전처리를 하여 필요한 정보만을 사용하였다.

## 사용한 Data

- > 도우미(한국학생): 순번, 성별, 나이, 매칭희망국가(1,2,3지망), 중국어 자격증or학과 여부, 일본어 자격증or학과 여부
- > 유학생(외국학생): 순번, 성별, 나이, 출신언어권, 기숙사 거주여부, 한국어 급수

# Optimization Target

## 필수 제약

- 1) 중국어, 일본어 자격증이 있거나 관련 학과를 다니는 도우미는 중국, 일본 유학생과 매칭이 되어야한다.
- 2) 1(도우미) : 2(유학생1, 유학생2) 매칭 시 유학생1의 한국어 급수가 1,2급이라면 유학생2는 반드시 1,2급이어야한다.  
↳ (1,2급과, 3~6급의 수업시간이 오전, 오후로 달라 활동하기가 어려움)
- 3) 기타문의로 1:1매칭을 희망하는 유학생이 있어 이를 고려
- 4) 기타문의로 특정 성별과의 매칭을 희망하는 유학생이 있어 이를 고려

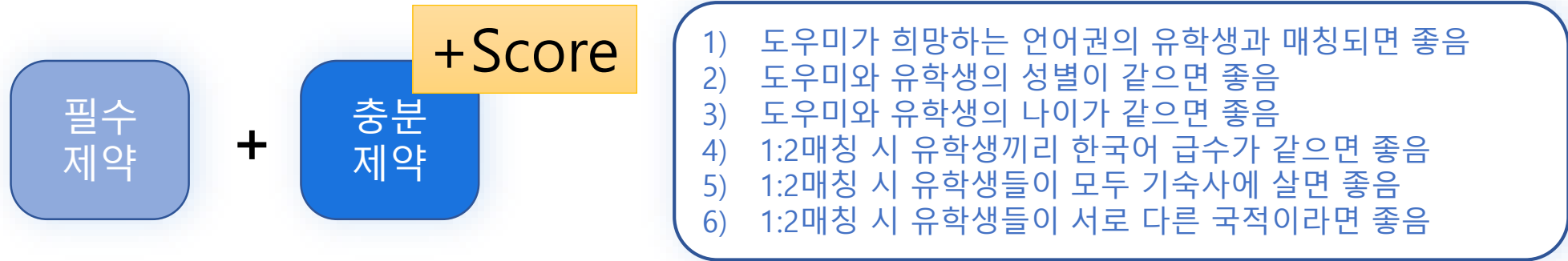
## 충분 제약

- 1) 도우미가 희망하는 언어권의 유학생과 매칭되면 좋음(1,2,3지망)
- 2) 도우미와 유학생의 성별이 같으면 좋음
- 3) 도우미와 유학생의 나이가 같으면 좋음
- 4) 1:2매칭 시 유학생끼리 한국어 급수가 같으면 좋음
- 5) 1:2매칭 시 유학생들이 모두 기숙사에 살면 좋음
- 6) 1:2매칭 시 유학생들이 서로 다른 국적이라면 좋음

❑ 프로그램 담당자의 요청사항을 바탕으로 위와 같이 "필수 제약", 과 "충분 제약"으로 나눔

❑ 필수 제약은 매칭에 무조건 반영 되어야하며, 충분 제약은 학생들의 매칭 만족과 직접적으로 연관되어 많은 조건을 충족시킬 수록, 학생들의 **매칭 만족도가 올라간다.**

# Optimization Target



## Target

각 매칭에서 "충분 제약"을 만족시킨 만큼 Score(가중치)를 부여하고, 최종적으로 Score의 합산이 Max가 되도록 최적화된 1:1, 1:2 매칭이 이루어져야한다.

Object  
(maxScore)

Mpvar  
(m1, m2)

\*기본적으로 Score(가중치)는 +1로 부여

\*도우미의 매칭희망국가(1지망: +3 / 2지망: +2 / 3지망: +1) 제약만 Score를 다르게 부여

# Modeling - Declarations

## !INDEX

K: set of integer !Korean 1~49(도우미 총 49명)  
 I, J: set of integer !Foreigner 1~90(유학생 총 90명)  
 ii: integer ! Data index

## !INPUT DATA

!한국 학생의 성별, 나이, 매칭 희망 국가(1,2,3지망), 중국어 자격증 or 학과, 일본어 자격증 or 학과 여부

K\_SEX: array(K) of string  
 K\_AGE: array(K) of integer  
 K\_LAN\_1: array(K) of string  
 K\_LAN\_2: array(K) of string  
 K\_LAN\_3: array(K) of string  
 K\_CHI: array(K) of string  
 K\_JAP: array(K) of string

	A	B	C	D	E	F	G	H
1	한국학생	성별	나이	매칭 희망 국가 (1지망)	매칭 희망 국가 (2지망)	매칭 희망 국가 (3지망)	중국어 자격증 or 학과	일본어 자격증 or 학과
2	1	여자	25	일본	영미	중국	x	o
3	2	남자	26	일본	영미	국적무관	x	x
4	3	남자	23	일본	중국	러시아	x	x
5	4	남자	22	중국	국적무관	국적무관	x	x
6	5	여자	22	영미	러시아	일본	x	x
7	6	남자	27	일본	영미	러시아	x	x
8	7	남자	24	일본	러시아	국적무관	x	x
9	8	남자	25	중국	일본	영미	o	x
10	9	여자	20	일본	베트남	중국	x	x

!유 학생의 성별, 나이, 출신언어권, 기숙사 거주 여부, 한국어 급수

F\_SEX: array(I) of string  
 F\_AGE: array(I) of integer  
 F\_LAN: array(I) of string  
 F\_DOM: array(I) of string  
 F\_CLASS: array(J) of integer

	A	B	C	D	E	F
1	유학생	성별	나이	출신언어권	기숙사 거주 여부	한국어 급수
2	1	여자	23	대만	o	2
3	2	여자	26	대만	o	2
4	3	여자	33	대한민국	o	2
5	4	남자	28	독일	x	2
6	5	여자	25	러시아	o	2
7	6	여자	19	러시아	o	1
8	7	여자	19	러시아	o	2
9	8	여자	20	러시아	o	2
10	9	여자	21	러시아	x	1



# Modeling - Declarations

*!Set of score(필수조건이 아닌 충분조건으로 각 조건에 대해 score(가중치)를 준 뒤. score를 Maxmise시킴으로써 최적화된 매칭 유도)*

```
SCORE_LAN_1: array(K,I) of integer !한국학생이 원하는 1지망 언어권의 학생과 매칭되면 score +3
SCORE_LAN_2: array(K,I) of integer !한국학생이 원하는 2지망 언어권의 학생과 매칭되면 score +2
SCORE_LAN_3: array(K,I) of integer !한국학생이 원하는 3지망 언어권의 학생과 매칭되면 score +1
SCORE_SEX: array(K,I) of integer !한국학생과 유학생의 성별이 같다면 score +1
SCORE_AGE: array(K,I) of integer !한국학생과 유학생의 나이가 같으면 score +1
```

한국학생-유학생 간의 충분조건 가중치

```
SCORE_CLASS: array(I,J) of integer !1:2매칭 시 유학생들끼리 한국어 급수가 같다면 score +1
SCORE_DOM: array(I,J) of integer !1:2매칭 시 유학생들이 같은 기숙사에 산다면 score +1
SCORE_CONTRY: array(I,J) of integer !1:2매칭 시 유학생들이 다른 언어권이라면 score +1
```

1:2 매칭 시 유학생들간의 충분조건 가중치

```
m1: array(K,I) of mpvar ! Matching 1:1
m2: array(K,I,J) of mpvar ! Matching 1:2
```

1:1매칭(m1) / 1:2매칭(m2) [k:도우미 / i,j:유학생]

의사결정변수

*! 필수 제약식*

```
BINARY_M1: array(K,I) of lincnr
BINARY_M2: array(K,I,J) of lincnr
```

m1, m2는 binary variable: [k,i] or [k,i,j]의 매칭이 이루어지면 1 매칭이 안되면 0

```
COUNT_PEOPLE: lincnr ! 매칭 대상 인원과 매칭된 인원이 같게 한다.(모든 인원이 매칭되게)
```

```
K0_1: array(K) of lincnr ! 한국학생들은 1번씩 매칭되어야 한다
K0_2: array(K) of lincnr ! 중국어 자격증이 있으면 중국인유학생이랑 매칭시켜야 한다
K0_3: array(K) of lincnr ! 일본어 자격증이 있으면 일본유학생이랑 매칭시켜야 한다
```

필수조건 3개 of 도우미

```
F0_1: array(I) of lincnr ! 유학생들은 1번씩 매칭되어야 한다
F0_2: array(I) of lincnr ! 1:2매칭 때 한 유학생1의 급수가 1,2급이라면 3~6급의 유학생과는 매칭이 되면 안된다
F0_3: array(I) of lincnr ! 1:2매칭 때 한 유학생2의 급수가 1,2급이라면 3~6급의 유학생과는 매칭이 되면 안된다
F0_4: array(1..11) of lincnr ! 기타 문의로 1:1매칭만을 희망하는 유학생
F0_5: array(1..11) of lincnr ! 기타 문의로 특정 성별과의 매칭을 희망하는 유학생
```

필수조건 5개 of 유학생

```
maxScore: lincnr ! Set of score(가중치)합이 최대로
end-declarations
```

Object : 각 매칭(m1,m2)이 이루어질 때의 Score 합산



# Modeling – Score calculation

## !3. 모델 수립 및 풀이

### !score calculation

```
forall(k in K, i in I) do
  if K_LAN_1(k)=F_LAN(i) then
    SCORE_LAN_1(k,i) := 3
  else
    SCORE_LAN_1(k,i) := 0
  end-if
end-do
도우미의 희망매칭국가(1지망)과
유학생의 출신언어권이 같으면 Score +3

forall(k in K, i in I) do
  if K_LAN_2(k)=F_LAN(i) then
    SCORE_LAN_2(k,i) := 2
  else
    SCORE_LAN_2(k,i) := 0
  end-if
end-do
도우미의 희망매칭국가(2지망)과
유학생의 출신언어권이 같으면 Score +2

forall(k in K, i in I) do
  if K_LAN_3(k)=F_LAN(i) then
    SCORE_LAN_3(k,i) := 1
  else
    SCORE_LAN_3(k,i) := 0
  end-if
end-do
도우미의 희망매칭국가(3지망)과
유학생의 출신언어권이 같으면 Score +1
```

```
forall(k in K, i in I) do
  if K_SEX(k)=F_SEX(i) then
    SCORE_SEX(k,i) := 1
  else
    SCORE_SEX(k,i) := 0
  end-if
end-do
도우미와 유학생의 성별이 같으면 Score +1

forall(k in K, i in I) do
  if K_AGE(k)=F_AGE(i) then
    SCORE_AGE(k,i) := 1
  else
    SCORE_AGE(k,i) := 0
  end-if
end-do
도우미와 유학생의 나이가 같으면 Score +1
```

```
forall(i in I, j in J | i<>j) do
  if F_CLASS(i)=F_CLASS(j) then
    SCORE_CLASS(i,j) := 1
  else
    SCORE_CLASS(i,j) := 0
  end-if
end-do
유학생1과 유학생2의 한국어 급수가 같으면 Score +1

forall(i in I, j in J | i<>j) do
  if F_DOM(i)=F_DOM(j) then
    SCORE_DOM(i,j) := 1
  else
    SCORE_DOM(i,j) := 0
  end-if
end-do
유학생1과 유학생2가 기숙사에 살면 Score +1

forall(i in I, j in J | i<>j) do
  if F_LAN(i)<>F_LAN(j) then
    SCORE_CONTRY(i,j) := 1
  else
    SCORE_CONTRY(i,j) := 0
  end-if
end-do
유학생1과 유학생2이 다른 언어권이면 Score +1
```

학생들의 data를 바탕으로 [도우미-유학생], [유학생1-유학생2] 간의 Score array를 만들

# Modeling – Constraints1

## □ Binary 제약

```
forall(k in K, i in I) BINARY_M1(k,i):= m1(k,i) is_binary  
forall(k in K, i in I, j in J) BINARY_M2(k,i,j):= m2(k,i,j) is_binary
```

$$\begin{aligned} m1_{ki} &\in \{0,1\} && , \forall k \in K, i \in I \\ m2_{kij} &\in \{0,1\} && , \forall k \in K, i \in I, j \in J \end{aligned}$$

매칭이 되면 1  
매칭이 되지 않으면 0

## □ 매칭 대상 인원과 매칭된 인원이 같게

```
COUNT_PEOPLE:= sum(k in K, i in I) (m1(k,i)*2) + sum(k in K, i in I, j in J) (m2(k,i,j)*3) = getsize(K) + getsize(I)
```

$$\sum_{k \in K, i \in I} (m1_{ki} \times 2) + \sum_{k \in K, i \in I, j \in J} (m2_{kij} \times 3) = |K| + |I|$$

성사된 1:1매칭수 x 2      성사된 1:2매칭수 x 3      49명 + 90명

# Modeling – Constraints2

□ 도우미 제약1: 도우미 한 명은 1:1 or 1:2 매칭에 1번씩 매칭됨

`forall(k in K) K0_1(k) := sum(i in I) m1(k,i) + sum(i in I, j in J | i <> j) m2(k,i,j) = 1`

$$\sum_{i \in I} m1_{ki} + \sum_{i \in I, j \in J : i < j} m2_{kij} = 1, \forall k \in K$$

각 도우미에 대해

m1 or m2 두 경우 중 하나만 1이 된다.

□ 유학생 제약1: 유학생 한 명은 1:1 or 1:2 매칭에 1번씩 매칭됨

`forall(i in I) F0_1(i) := sum(k in K) m1(k,i) + sum(k in K, j in J | i <> j) m2(k,i,j) + sum(k in K, j in J | i <> j) m2(k,j,i) = 1`

$$\sum_{k \in K} m1_{ki} + \sum_{k \in K, j \in J : i < j} m2_{kij} + \sum_{k \in K, j \in J : i < j} m2_{kji} = 1, \forall i \in I$$

각 유학생에 대해

m1 or m2(유학생1일 때) or m2(유학생2일 때) 세 경우 중 하나만 1이 된다.

\* m2<sub>kij</sub> (1:2매칭)에서 유학생은 i 또는 j에 들어갈 수 있어 2가지 상황 모두를 고려  
여기서 i에 위치하는 유학생을 **유학생1**, j에 위치하는 유학생을 **유학생2**라고 칭함

# Modeling – Constraints3

## □ 도우미 제약2: 중국어 자격증이 있거나 관련학과이면 중국 유학생과 매칭

`forall(k in K | K_CHI(k)='o') K0_2(k):= sum(i in I | F_LAN(i)='중국') m1(k,i) + sum(i in I, j in J | i<>j and F_LAN(i)='중국' and F_LAN(j)='중국') m2(k,i,j) = 1`

$$\sum_{i \in I : F\_LAN(i)='중국'} m1_{ki} + \sum_{i \in I, j \in J : F\_LAN(i)='중국', F\_LAN(j)='중국', i <> j} m2_{kij} = 1, \forall k \in K : K\_CHI(k)='o'$$

중국어 자격증 or 학과  
여부가 o인 도우미에 대해

m1(중국 유학생과) or m2(중국 유학생들과) 두 경우 중 하나만 1이 된다.

## □ 도우미 제약3: 일본어 자격증이 있거나 관련학과이면 일본 유학생과 매칭

`forall(k in K | K_JAP(k)='o') K0_3(k):= sum(i in I | F_LAN(i)='일본') m1(k,i) + sum(i in I, j in J | i<>j and F_LAN(i)='일본' and F_LAN(j)='일본') m2(k,i,j) = 1`

$$\sum_{i \in I : F\_LAN(i)='일본'} m1_{ki} + \sum_{i \in I, j \in J : F\_LAN(i)='일본', F\_LAN(j)='일본', i <> j} m2_{kij} = 1, \forall k \in K : K\_CHI(k)='o'$$

일본어 자격증 or 학과  
여부가 o인 도우미에 대해

m1(일본 유학생과) or m2(일본 유학생들과) 두 경우 중 하나만 1이 된다.

# Modeling – Constraints4

- 유학생 제약2(1:2매칭): 한 유학생이 <sup>한국어 급수</sup> 1 or 2 급이라면 <sup>한국어 급수</sup> 3~6급의 유학생과 매칭이 되면 안됨 (수업시간이 다름)

```
forall(i in I | F_CLASS(i)=1 or F_CLASS(i)=2) F0_2(i) := sum(k in K) m1(k,i)
+ sum(k in K, j in J | (F_CLASS(j)=1 or F_CLASS(j)=2) and (i<>j)) m2(k,i,j)
+ sum(k in K, j in J | (F_CLASS(j)=1 or F_CLASS(j)=2) and (i<>j)) m2(k,j,i) = 1
```

$$\sum_{k \in K} m1_{ki} + \sum_{\substack{k \in K, j \in J \\ : (F\_CLASS(i)=1 \text{ or } F\_CLASS(i)=2) \text{ and } i < j}} m2_{kij} + \sum_{\substack{k \in K, j \in J \\ : (F\_CLASS(i)=1 \text{ or } F\_CLASS(i)=2) \text{ and } i < j}} m2_{kji} = 1, \forall i \in I : F\_CLASS(i)=1 \text{ or } F\_CLASS(i)=2$$

1or2급의 유학생에 대해

m1 or m2(1or2급의 유학생2와) or m2(1or2급의 유학생1과) 세 경우 중 하나만 1이 된다.

- 유학생 제약3(1:2매칭): 한 유학생이 3~6급이라면 1 or 2급의 유학생과 매칭이 되면 안됨 (수업시간이 다름)

```
forall(i in I | F_CLASS(i)<>1 and F_CLASS(i)<>2) F0_3(i) := sum(k in K) m1(k,i)
+ sum(k in K, j in J | F_CLASS(j)<>1 and F_CLASS(j)<>2 and i<>j) m2(k,i,j)
+ sum(k in K, j in J | F_CLASS(j)<>1 and F_CLASS(j)<>2 and i<>j) m2(k,j,i) = 1
```

$$\sum_{k \in K} m1_{ki} + \sum_{\substack{k \in K, j \in J \\ : (F\_CLASS(i)<>1 \text{ and } F\_CLASS(i)<>2) \text{ and } i < j}} m2_{kij} + \sum_{\substack{k \in K, j \in J \\ : (F\_CLASS(i)<>1 \text{ and } F\_CLASS(i)<>2) \text{ and } i < j}} m2_{kji} = 1, \forall i \in I : F\_CLASS(i)<>1 \text{ and } F\_CLASS(i)<>2$$

3~6급의 유학생에 대해

m1 or m2(3~6급의 유학생2와) or m2(3~6급의 유학생1과) 세 경우 중 하나만 1이 된다.

# Modeling – Constraints5

□ 유학생 제약4: 기타문의로 1:1 매칭만을 희망하는 유학생 고려(21, 22번 유학생이 희망)

$F0\_4(1) := \sum(k \text{ in } K) m1(k, 21) = 1$   
 $F0\_4(2) := \sum(k \text{ in } K) m1(k, 22) = 1$

21 1대 1을 희망합니다.  
 22 1대1일을 희망합니다. <기타문의>

ex) 21번 학생

$$\sum_{k \in K} m1_{k21} = 1$$

$m1_{(21\text{번 유학생을 갖는})}$ 이 1이어야한다

□ 유학생 제약5: 기타 문의로 특정 성별과의 매칭을 희망하는 유학생(22~31번 유학생이 여성 한국학생과 매칭 희망)

$F0\_5(1) := \sum(k \text{ in } K \mid K\_SEX(k) = '여자') m1(k, 23) + \sum(k \text{ in } K, i \text{ in } I \mid K\_SEX(k) = '여자') m2(k, i, 23) + \sum(k \text{ in } K, j \text{ in } J \mid K\_SEX(k) = '여자') m2(k, 23, j) = 1$   
 $F0\_5(2) := \sum(k \text{ in } K \mid K\_SEX(k) = '여자') m1(k, 24) + \sum(k \text{ in } K, i \text{ in } I \mid K\_SEX(k) = '여자') m2(k, i, 24) + \sum(k \text{ in } K, j \text{ in } J \mid K\_SEX(k) = '여자') m2(k, 24, j) = 1$   
 $F0\_5(3) := \sum(k \text{ in } K \mid K\_SEX(k) = '여자') m1(k, 25) + \sum(k \text{ in } K, i \text{ in } I \mid K\_SEX(k) = '여자') m2(k, i, 25) + \sum(k \text{ in } K, j \text{ in } J \mid K\_SEX(k) = '여자') m2(k, 25, j) = 1$   
 $F0\_5(4) := \sum(k \text{ in } K \mid K\_SEX(k) = '여자') m1(k, 26) + \sum(k \text{ in } K, i \text{ in } I \mid K\_SEX(k) = '여자') m2(k, i, 26) + \sum(k \text{ in } K, j \text{ in } J \mid K\_SEX(k) = '여자') m2(k, 26, j) = 1$   
 $F0\_5(5) := \sum(k \text{ in } K \mid K\_SEX(k) = '여자') m1(k, 27) + \sum(k \text{ in } K, i \text{ in } I \mid K\_SEX(k) = '여자') m2(k, i, 27) + \sum(k \text{ in } K, j \text{ in } J \mid K\_SEX(k) = '여자') m2(k, 27, j) = 1$   
 $F0\_5(6) := \sum(k \text{ in } K \mid K\_SEX(k) = '여자') m1(k, 28) + \sum(k \text{ in } K, i \text{ in } I \mid K\_SEX(k) = '여자') m2(k, i, 28) + \sum(k \text{ in } K, j \text{ in } J \mid K\_SEX(k) = '여자') m2(k, 28, j) = 1$   
 $F0\_5(7) := \sum(k \text{ in } K \mid K\_SEX(k) = '여자') m1(k, 29) + \sum(k \text{ in } K, i \text{ in } I \mid K\_SEX(k) = '여자') m2(k, i, 29) + \sum(k \text{ in } K, j \text{ in } J \mid K\_SEX(k) = '여자') m2(k, 29, j) = 1$   
 $F0\_5(8) := \sum(k \text{ in } K \mid K\_SEX(k) = '여자') m1(k, 30) + \sum(k \text{ in } K, i \text{ in } I \mid K\_SEX(k) = '여자') m2(k, i, 30) + \sum(k \text{ in } K, j \text{ in } J \mid K\_SEX(k) = '여자') m2(k, 30, j) = 1$   
 $F0\_5(9) := \sum(k \text{ in } K \mid K\_SEX(k) = '여자') m1(k, 31) + \sum(k \text{ in } K, i \text{ in } I \mid K\_SEX(k) = '여자') m2(k, i, 31) + \sum(k \text{ in } K, j \text{ in } J \mid K\_SEX(k) = '여자') m2(k, 31, j) = 1$

22 1대1일을 희망합니다.  
 23 동성희망입니다. 저는 애니메이션 보는 것이라 카페 가는 것 좋아요. 같은 취미 있는 친구가 있으면 좋겠어요.  
 24 될 수 있으면 여학생이 더 좋아합니다  
 25 가능한 여학생이 좋습니다.  
 26 도우미 제도 모집에 있어서 가능하다면 여성분과 한국어 회화를 하고 싶은데요, 남자분과 하게되는 경우에는 그  
 27 가능하면 여성을 희망합니다  
 28 여성우선으로 부탁드립니다  
 29 가능하면 여자가 좋습니다  
 30 동성 희망입니다. 저는 카페나 쇼핑하는 것 좋아요. 같은 취미를 가진 친구가 있으면 좋겠어요.  
 31 여자와 공부하고 싶습니다.

<기타문의>

ex) 23번 학생

$$\sum_{k \in K : K\_SEX(k) = '여자'} m1_{k23} + \sum_{k \in K, i \in I : K\_SEX(k) = '여자'} m2_{ki23} + \sum_{k \in K, i \in I : K\_SEX(k) = '여자'} m2_{k23i} = 1$$

$m1_{(여자 도우미와)}$  or  $m2_{(여자 유학생2와)}$  or  $m2_{(여자 유학생1과)}$  세 경우 중 하나만 1이 된다.

# Modeling – Object

□ Object: Score arrays와 m1, m2의 곱의 총합

```
maxScore := sum(k in K, i in I) (m1(k,i)*SCORE_LAN_1(k,i)+m1(k,i)*SCORE_LAN_2(k,i)+m1(k,i)*SCORE_LAN_3(k,i))
+ sum(k in K, i in I) (m1(k,i)*SCORE_SEX(k,i))
+ sum(k in K, i in I) (m1(k,i)*SCORE_AGE(k,i))
+ sum(k in K, i in I, j in J | i <> j) ((m2(k,i,j)*SCORE_LAN_1(k,i)+(m2(k,i,j)*SCORE_LAN_1(k,j)))
+ (m2(k,i,j)*SCORE_LAN_2(k,i)+(m2(k,i,j)*SCORE_LAN_2(k,j)))+(m2(k,i,j)*SCORE_LAN_3(k,i)+(m2(k,i,j)*SCORE_LAN_3(k,j))))
+ sum(k in K, i in I, j in J | i <> j) (m2(k,i,j)*SCORE_SEX(k,i)+(m2(k,i,j)*SCORE_SEX(k,j)))
+ sum(k in K, i in I, j in J | i <> j) (m2(k,i,j)*SCORE_AGE(k,i)+(m2(k,i,j)*SCORE_AGE(k,j)))
+ sum(k in K, i in I, j in J | i <> j) (m2(k,i,j)*SCORE_CLASS(i,j))
+ sum(k in K, i in I, j in J | i <> j) (m2(k,i,j)*SCORE_DOM(i,j))
+ sum(k in K, i in I, j in J | i <> j) (m2(k,i,j)*SCORE_CONTRY(i,j))
```

1:1 매칭의 Score 합

+

1:2 매칭의 Score 합

1:1 매칭 Score 합

$$\sum (m1_{ki}) \times (\text{SCORE\_LAN\_1}_{ki}) + (m1_{ki}) \times (\text{SCORE\_LAN\_2}_{ki}) + (m1_{ki}) \times (\text{SCORE\_LAN\_3}_{ki})$$

$$+ \sum_{k \in K, i \in I} (m1_{ki}) \times (\text{SCORE\_SEX}_{ki}) + \sum_{k \in K, i \in I} (m1_{ki}) \times (\text{SCORE\_AGE}_{ki}) +$$

이어서



# Modeling – Object

□ Object: Score array와 m1, m2의 곱의 총합

```
maxScore := sum(k in K, i in I) (m1(k,i)*SCORE_LAN_1(k,i)+m1(k,i)*SCORE_LAN_2(k,i)+m1(k,i)*SCORE_LAN_3(k,i))
+ sum(k in K, i in I) (m1(k,i)*SCORE_SEX(k,i))
+ sum(k in K, i in I) (m1(k,i)*SCORE_AGE(k,i))
+ sum(k in K, i in I, j in J | i<>j) ((m2(k,i,j)*SCORE_LAN_1(k,i)+(m2(k,i,j)*SCORE_LAN_1(k,j)))
+ (m2(k,i,j)*SCORE_LAN_2(k,i)+(m2(k,i,j)*SCORE_LAN_2(k,j)))+(m2(k,i,j)*SCORE_LAN_3(k,i)+(m2(k,i,j)*SCORE_LAN_3(k,j))))
+ sum(k in K, i in I, j in J | i<>j) (m2(k,i,j)*SCORE_SEX(k,i)+(m2(k,i,j)*SCORE_SEX(k,j)))
+ sum(k in K, i in I, j in J | i<>j) (m2(k,i,j)*SCORE_AGE(k,i)+(m2(k,i,j)*SCORE_AGE(k,j)))
+ sum(k in K, i in I, j in J | i<>j) (m2(k,i,j)*SCORE_CLASS(i,j))
+ sum(k in K, i in I, j in J | i<>j) (m2(k,i,j)*SCORE_DOM(i,j))
+ sum(k in K, i in I, j in J | i<>j) (m2(k,i,j)*SCORE_CONTRY(i,j))
```

1:1 매칭의 Score 합

+

1:2 매칭의 Score 합

1:2 매칭 Score 합

이어서

$$+ \sum_{k \in K, i \in I, j \in J : i < j} (m2_{kij})x(SCORE\_LAN\_1ki) + (m2_{kij})x(SCORE\_LAN\_1kj) \\ + (m2_{kij})x(SCORE\_LAN\_1ki) + (m2_{kij})x(SCORE\_LAN\_1kj) \\ + (m2_{kij})x(SCORE\_LAN\_1ki) + (m2_{kij})x(SCORE\_LAN\_1kj)$$

$$+ \sum_{k \in K, i \in I, j \in J : i < j} (m2_{kij})x(SCORE\_SEXki) + (m2_{kij})x(SCORE\_SEXkj)$$

$$+ \sum_{k \in K, i \in I, j \in J : i < j} (m2_{kij})x(SCORE\_AGEki) + (m2_{kij})x(SCORE\_AGEkj)$$

$$+ \sum_{k \in K, i \in I, j \in J : i < j} (m2_{kij})x(SCORE\_CLASSij) + \sum_{k \in K, i \in I, j \in J : i < j} (m2_{kij})x(SCORE\_DOMij) + \sum_{k \in K, i \in I, j \in J : i < j} (m2_{kij})x(SCORE\_CONTRYij)$$

# Result – Optimized Matching

## □ 결과 출력

```
!4. 해의 상태 확인 및 출력
writeln("-----")
case getprobat of
  XPRS_OPT: writeln('optimal')
  XPRS_INF: writeln('infeasible')
  XPRS_UNB: writeln('unbounded')
  XPRS_UNF: writeln('unfinished')
else
  writeln('unexpected problem status!')
end-case

!5. optimal solution 및 optimal value 출력
writeln("----- SCORE -----")
writeln("MAX SCORE = ", getobjval)
writeln("----- 1:1 MATCH -----")
forall( k in K, i in I | getsol(m1(k,i)) > 0) writeln("[ 도우미", k, " + 유학생", i, " ]")
writeln("----- 1:2 MATCH -----")
forall( k in K, i in I, j in J | getsol(m2(k,i,j)) > 0) writeln("[ 도우미", k, " + 유학생", i, " + 유학생", j, " ]")
```

가을학기 Data(도우미 49명, 유학생 90명)를 바탕으로  
필수조건을 반영하고, 충분조건을 **최대한** 만족하는(max score=189)  
**8개의 1:1매칭, 41개의 1:2매칭** 결과를 얻음

## 최적의 1:1, 1:2 매칭

Running model

optimal

----- SCORE -----

MAX SCORE = 189

----- 1:1 MATCH -----

[ 도우미1 + 유학생21 ]  
[ 도우미8 + 유학생82 ]  
[ 도우미16 + 유학생86 ]  
[ 도우미20 + 유학생1 ]  
[ 도우미26 + 유학생13 ]  
[ 도우미29 + 유학생38 ]  
[ 도우미39 + 유학생22 ]  
[ 도우미48 + 유학생2 ]

----- 1:2 MATCH -----

[ 도우미2 + 유학생81 + 유학생87 ]  
[ 도우미3 + 유학생16 + 유학생18 ]  
[ 도우미4 + 유학생39 + 유학생40 ]  
[ 도우미5 + 유학생31 + 유학생56 ]  
[ 도우미6 + 유학생20 + 유학생65 ]  
[ 도우미7 + 유학생80 + 유학생85 ]  
[ 도우미9 + 유학생3 + 유학생4 ]  
[ 도우미10 + 유학생27 + 유학생29 ]  
[ 도우미11 + 유학생7 + 유학생10 ]  
[ 도우미12 + 유학생8 + 유학생6 ]  
[ 도우미13 + 유학생9 + 유학생5 ]  
[ 도우미14 + 유학생46 + 유학생33 ]  
[ 도우미15 + 유학생73 + 유학생43 ]  
[ 도우미17 + 유학생67 + 유학생50 ]  
[ 도우미18 + 유학생88 + 유학생64 ]  
[ 도우미19 + 유학생52 + 유학생60 ]

[ 도우미21 + 유학생71 + 유학생36 ]  
[ 도우미22 + 유학생30 + 유학생77 ]  
[ 도우미23 + 유학생63 + 유학생78 ]  
[ 도우미24 + 유학생68 + 유학생75 ]  
[ 도우미25 + 유학생76 + 유학생72 ]  
[ 도우미27 + 유학생48 + 유학생37 ]  
[ 도우미28 + 유학생55 + 유학생53 ]  
[ 도우미30 + 유학생83 + 유학생84 ]  
[ 도우미31 + 유학생19 + 유학생90 ]  
[ 도우미32 + 유학생32 + 유학생61 ]  
[ 도우미33 + 유학생17 + 유학생12 ]  
[ 도우미34 + 유학생79 + 유학생42 ]  
[ 도우미35 + 유학생24 + 유학생54 ]  
[ 도우미36 + 유학생34 + 유학생23 ]  
[ 도우미37 + 유학생59 + 유학생45 ]  
[ 도우미38 + 유학생25 + 유학생26 ]  
[ 도우미40 + 유학생11 + 유학생14 ]  
[ 도우미41 + 유학생28 + 유학생49 ]  
[ 도우미42 + 유학생15 + 유학생89 ]  
[ 도우미43 + 유학생44 + 유학생57 ]  
[ 도우미44 + 유학생35 + 유학생70 ]  
[ 도우미45 + 유학생51 + 유학생41 ]  
[ 도우미46 + 유학생58 + 유학생74 ]  
[ 도우미47 + 유학생69 + 유학생66 ]  
[ 도우미49 + 유학생62 + 유학생47 ]