

Problem

Result

※ 문제 및 풀이에 대해 **사외 온라인/오프라인에 게시/공유 하는 것은 금지**되어 있습니다.
SW Expert 아카데미의 문제는 **삼성전자 직원**에게만 오픈 되어 있습니다.

[제한 사항]

| | |
|-----|--|
| 시간 | 10개 테스트케이스를 합쳐서 C++ 의 경우 1초 / Java 의 경우 2초 |
| 메모리 | 힙, 정적 메모리 합쳐서 256MB 이내, 스택 메모리 1MB 이내 |

※ 본 문제는 임직원의 S/W 문제해결 역량 향상을 위한 **Professional 실전형 연습문제**이며
함께 제공되는 **Testcase** 또한 **학습을 위한 연습용**입니다.

① C 또는 C++로 답안을 작성하시는 응시자께서는 검정시스템에 제출 시, Language 에서 C++ 를 선택하신 후 제출하시기 바랍니다.

② Main 과 User Code 부분으로 구성되어 있습니다.

A. Main : 수정할 수 없는 코드이며, 채점 시 비 정상적인 답안 검출 등 평가를 위한 로직이 추가 될 수 있습니다.

B. User Code : 실제 응시자가 작성해야 하는 코드이며, 제출 시에는 표준 입출력 함수가 포함되어 있으면 안 됩니다.

③ Local PC 에서 프로그래밍 시 유의 사항

A. 2개의 파일을 생성하셔야 합니다. (main.cpp / solution.cpp 또는 Solution.java / UserSolution.java)

B. Main 부분의 코드를 main.cpp 또는 Solution.java 에 복사해서 사용하시기 바랍니다.

C. sample_input.txt 를 사용하기 위해서는 Main 부분의 코드 내에

표준 입력을 파일로 전환하는 코드 (주석처리 되어 있음) 의 주석을 풀어서 사용하시면 됩니다.

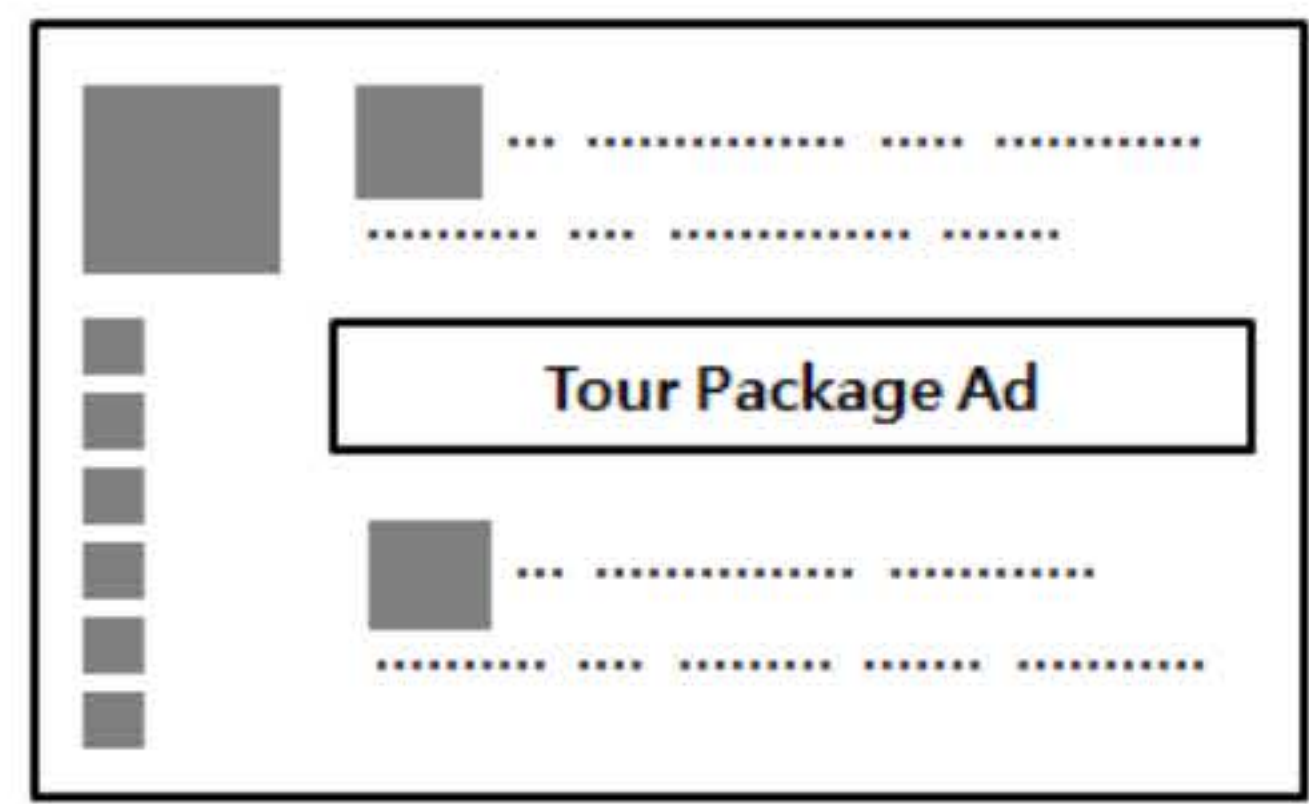
D. User Code 부분의 코드를 작성하신 후 서버에 제출하실 때,

디버깅을 위한 표준 입출력 함수를 모두 삭제 또는 주석 처리해 주셔야 합니다.

④ 문제 내에 제약조건을 모두 명시하지 않으므로 주어진 코드를 분석하셔야 합니다.

⑤ 코드는 개발 언어에 따라 상이할 수 있으므로, 작성할 언어를 기준으로 분석하셔야 합니다.

[문제 설명]



[Fig. 1]

N 명의 사용자들이 있는 소셜 미디어 사이트가 있다.

사용자들끼리 친구가 될 수 있다.

또한, 이 사이트는 여행 상품 목록에 여행 상품을 등록하고 사용자들에게 여행 상품을 판매한다.

사용자가 예약한 여행 상품은 여행 상품 목록에서 삭제되고 더 이상 판매되지 않는다.

각 여행 상품에는 상품 ID, 지역, 가격 정보가 있다.

이 사이트에서 사용자가 가장 선호할 만한 여행 상품을 선정하고 그 사용자에게 여행 상품을 추천하는 기능을 제공하고자 한다.

사용자에게 가장 선호할 만한 여행 상품은 다음과 같이 선정된다.

- ㉠ 사용자와 사용자의 친구들이 예약한 여행 상품 중 가장 많이 예약한 지역(들)을 찾는다.
- ㉡ ㉠ 과정에서 찾은 지역을 가지는 여행 상품들 중 가격이 가장 싼 여행 상품을 선정한다.
- ㉢ ㉡ 과정에서 찾은 가격이 가장 싼 여행 상품이 다수인 경우, 그 상품들 중에서 상품 ID가 가장 작은 여행 상품을 선정한다.
- ㉣ 만약 ㉡ 과정에서 해당하는 여행 상품이 없는 경우, 사용자와 사용자의 친구들이 두번째로 많이 예약한 지역에 대해서 ㉡와 ㉢ 과정을 수행한다.
만약 그런 경우에도 해당하는 여행 상품이 없는 경우, 사용자와 사용자의 친구들이 세번째로 많이 예약한 지역에서 대해서 진행된다.
이렇게 여행 상품이 선정될 때까지 사용자와 사용자의 친구들이 많이 예약한 지역의 순서로 가장 선호할 만한 여행 상품을 찾는다.

이와 같이 사용자에게 가장 선호할 만한 여행 상품을 선정하고 추천하는 기능을 수행하는 API를 구현하라.

다음은 구현해야 할 API이다.

※ 아래 함수 signature는 C/C++에 대한 것으로 Java에 대해서는 제공되는 Solution.java와 UserSolution.java를 참고하라.

| |
|--|
| void init(int N, int M) |
| 테스트 케이스에 대한 초기화를 하는 함수. 각 테스트 케이스의 맨 처음에 1회 호출된다. 소셜 미디어 사이트에 N 명의 사용자가 있다. 사용자는 1 이상 N 이하의 숫자로 구분된다. 이를 사용자 ID라고 한다. M은 여행 상품의 지역의 수를 의미한다. 여행 상품의 지역은 1 이상 M 이하의 정수 값으로 나타낸다. 각 테스트 케이스 초기에는 어느 누구도 친구가 된 사용자는 없고 등록된 여행 상품도 없다. <i>Parameters</i> N : 사이트의 사용자 수 ($5 \leq N \leq 1,000$) M : 여행 상품의 지역의 수 ($2 \leq M \leq 10$) |
| void befriend(int uid1, int uid2) |
| 사용자 uid1과 사용자 uid2는 서로 친구가 된다. 즉, 사용자 uid1은 사용자 uid2의 친구가 되고 사용자 uid2는 사용자 uid1의 친구가 된다. uid1과 uid2는 사용자 ID를 의미한다. uid1과 uid2가 서로 같은 경우는 없다. 이미 친구가 된 사용자 uid1과 uid2에 대해 다시 호출되는 경우는 없다. <i>Parameters</i> uid1, uid2 : 서로 친구가 되고자 하는 사용자 ID ($1 \leq uid1, uid2 \leq N$) |
| void add(int pid, int area, int price) |
| 상품 ID가 pid이고 지역이 area이고 가격이 price인 여행 상품이 여행 상품 목록에 등록된다. 여행 상품 목록에 등록된 상품은 사용자에게 판매된다. 상품 ID는 여행 상품에 대한 고유한 값이다. 예약된 상품과 여행 상품 목록에 있는 상품 중 같은 상품 ID를 가지는 여행 상품은 없다. <i>Parameters</i> pid : 상품 ID ($1 \leq pid \leq 1,000,000,000$) area : 지역 정보 ($1 \leq area \leq M$) price : 여행 상품의 가격 ($1 \leq price \leq 1,000$) |
| void reserve(int uid, int pid) |
| 사용자 ID가 uid인 사용자가 상품 ID가 pid인 여행상품을 예약한다. 예약된 상품은 여행 상품 목록에서 삭제되고 더 이상 판매될 수 없다. 함수 호출 시, 상품 ID가 pid인 여행 상품은 여행 상품 목록에 반드시 있다. <i>Parameters</i> uid : 여행 상품을 예약하려는 사용자 ID ($1 \leq uid \leq N$) |

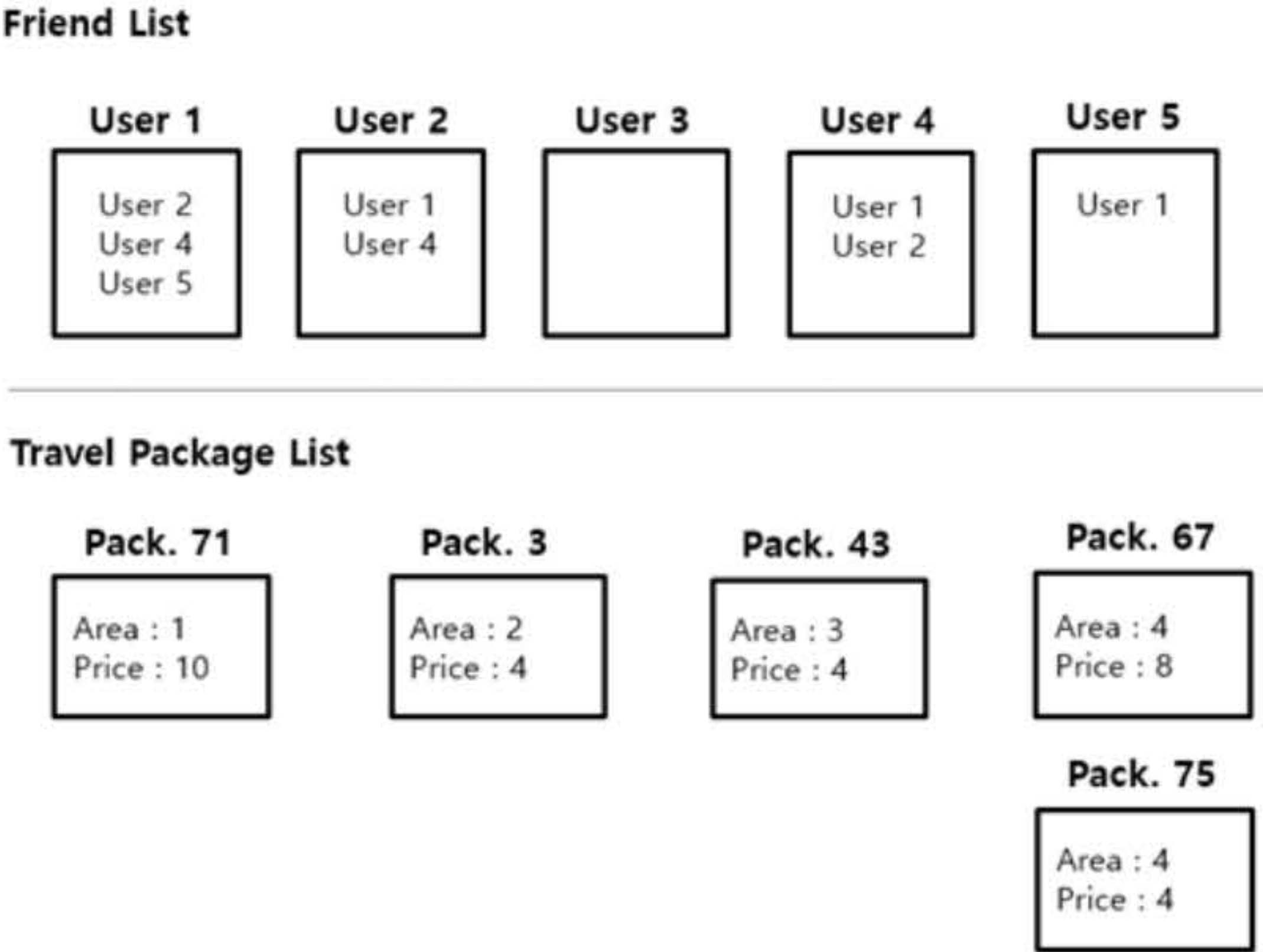
| |
|--|
| pid : 상품 ID ($1 \leq \text{pid} \leq 1,000,000,000$) |
| int recommend(int uid) |
| 사용자 uid에게 가장 선호할 만한 여행 상품을 찾아 그 상품 ID를 반환한다. |
| 해당 함수가 호출될 때, 여행 상품 목록에 여행 상품이 1 개 이상 있다. |
| Parameters uid : 사용자 ID ($1 \leq \text{uid} \leq N$) |
| Return 사용자 uid에게 가장 선호할 만한 여행 상품의 상품 ID |

[Table 1]과 같이 API가 호출되는 경우는 생각해 보자.

| # | API Call | Description | Return | Figure |
|----|----------------|----------------------------------|--------|----------|
| 1 | init(5, 4) | 테스트 케이스 초기화, N=5, M=4 | | |
| 2 | befriend(5, 1) | 사용자 5와 사용자 1이 서로 친구가 된다. | | |
| 3 | befriend(2, 4) | | | |
| 4 | befriend(1, 4) | | | |
| 5 | befriend(2, 1) | | | |
| 6 | add(71, 1, 10) | 상품 ID=71, 지역=1, 가격=10인 상품이 추가된다. | | |
| 7 | add(67, 4, 8) | | | |
| 8 | add(43, 3, 4) | | | |
| 9 | add(3, 2, 4) | | | |
| 10 | add(75, 4, 4) | | | [Fig. 2] |
| 11 | recommend(4) | 사용자 4에 대한 여행 상품을 추천한다. | 3 | |
| 12 | reserve(4, 75) | 사용자 4가 상품 75을 예약한다. | | [Fig. 3] |
| 13 | recommend(1) | | 67 | |
| 14 | add(81, 2, 10) | | | |
| 15 | reserve(2, 67) | | | |
| 16 | add(69, 1, 5) | | | |
| 17 | add(55, 3, 6) | | | [Fig. 4] |
| 18 | recommend(2) | | 3 | |
| 19 | reserve(5, 69) | | | [Fig. 5] |
| 20 | recommend(5) | | 71 | |
| 21 | befriend(3, 5) | | | |
| 22 | befriend(3, 4) | | | |
| 23 | add(28, 4, 10) | | | [Fig. 6] |
| 24 | recommend(3) | | 28 | |

[Table 1]

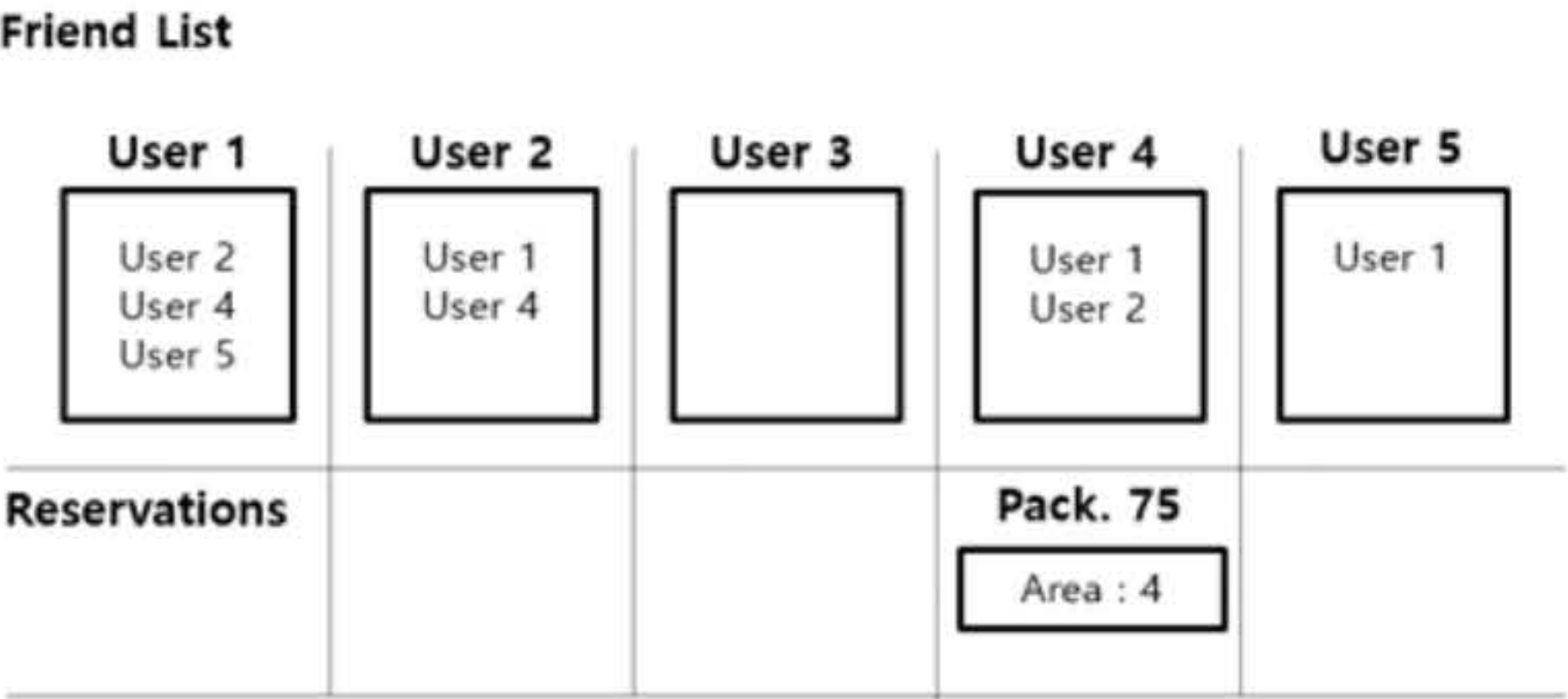
10번째 add(75, 4, 4) 함수가 호출된 후에 사용자에게 대한 친구 리스트와 판매하고 있는 여행 상품은 [Fig. 2]와 같다.



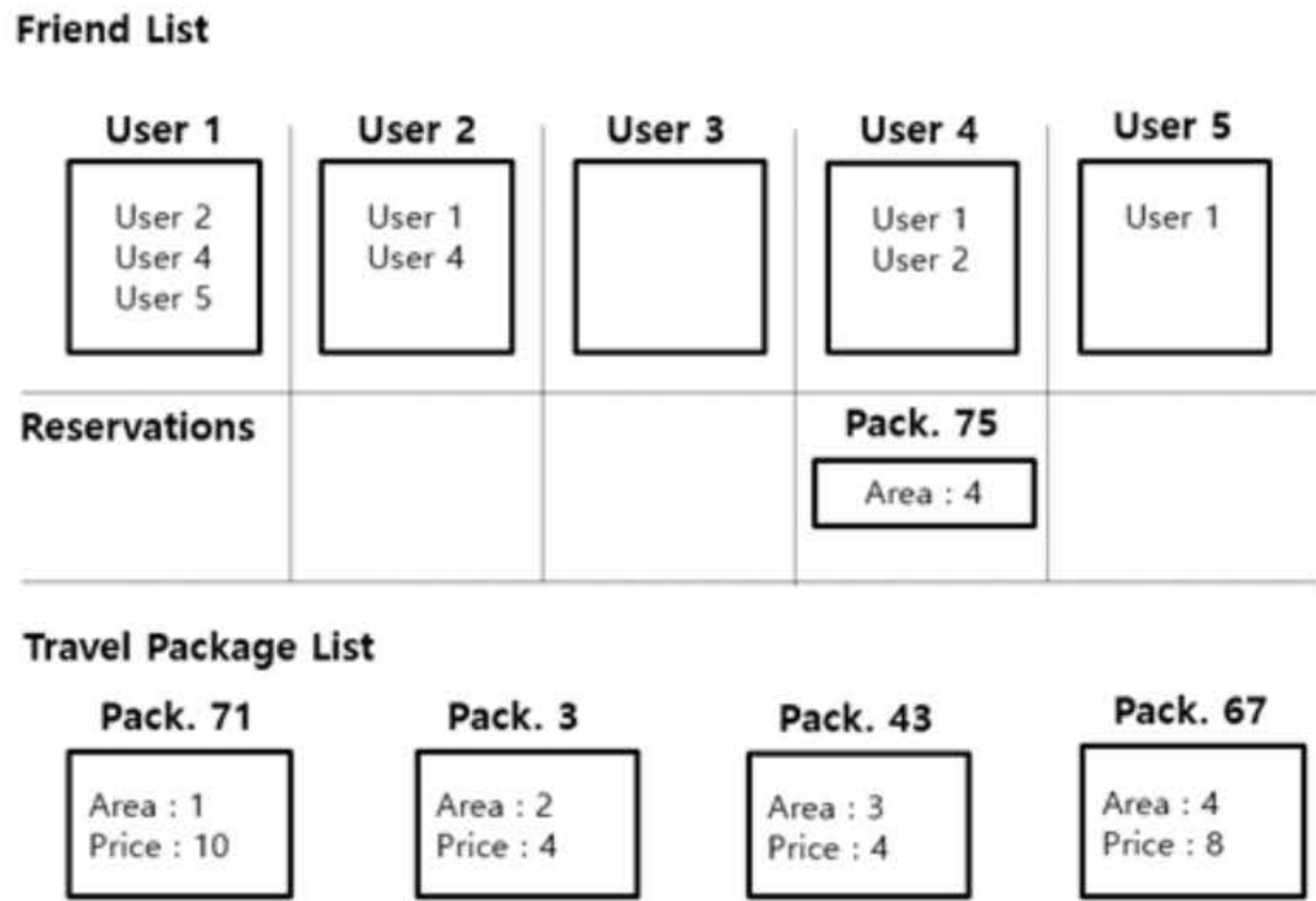
[Fig. 2]

11번째에 recommend(4)가 호출될 때 사용자 4와 그의 친구들이 여행 상품을 예약한 경우가 없다. 그러므로, 가장 많이 예약한 지역은 모든 지역이 된다. 따라서, 여행 상품 목록에 있는 여행 상품 중 가장 싼 여행 상품은 상품 3, 상품 43, 상품 75이다. 상품 ID가 가장 작은 여행 상품은 상품 3이다. 상품 3을 추천한다.

12번째에 reserve(4, 75)가 호출되면 상품 75는 예약되고 여행 상품 목록에서 상품 75는 삭제된다.



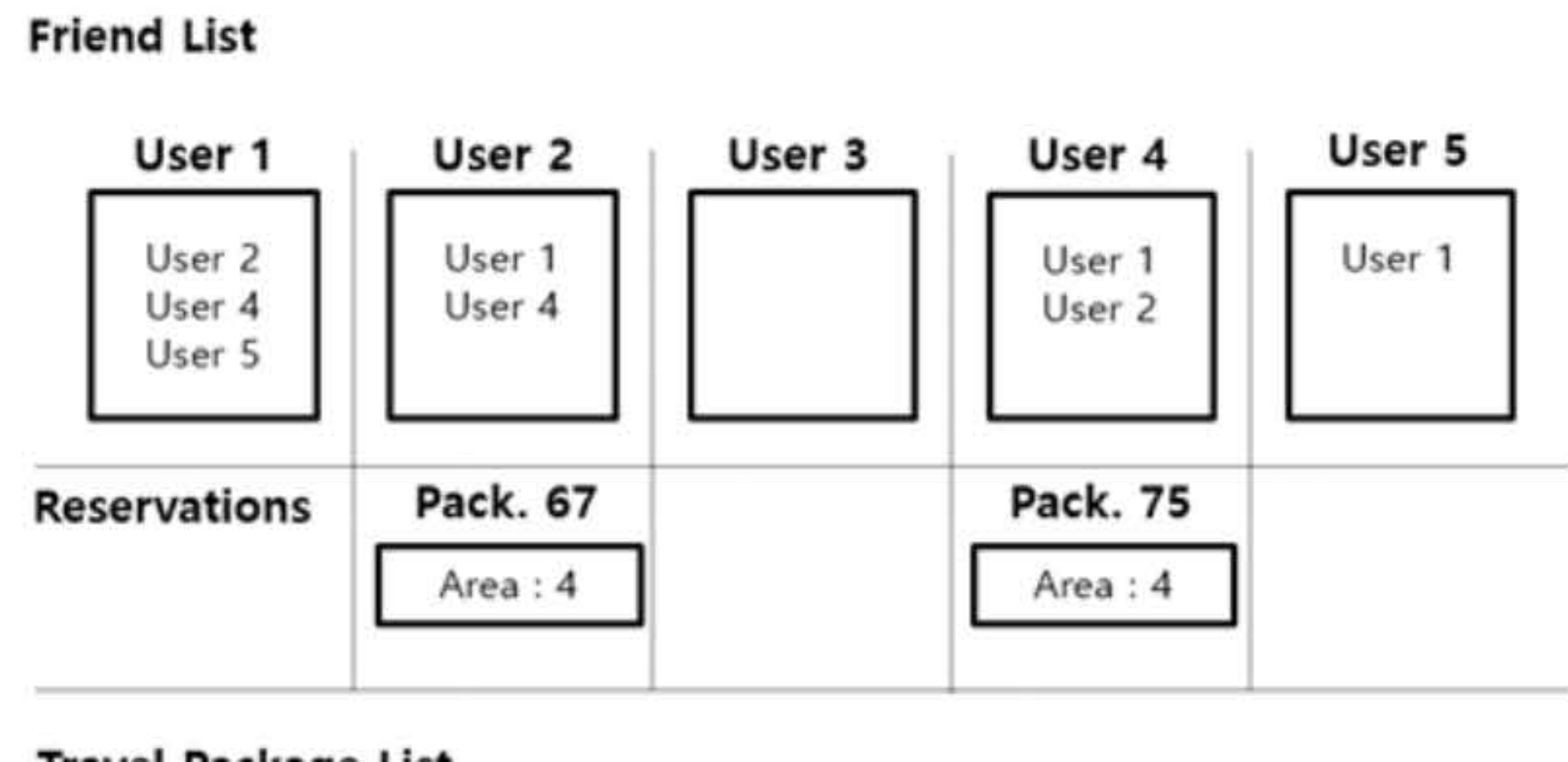
12번째에 reserve(4, 75)가 호출되면 상품 75는 예약되고 여행 상품 목록에서 상품 75는 삭제된다.



[Fig. 3]

13번째에 recommend(1)이 호출될 때 사용자 1과 그의 친구들이 가장 많이 예약한 지역은 4이다.
이 경우에 지역이 4이고 가장 싼 여행 상품은 상품 67이므로 상품 67을 추천한다.

17 번째 add(55, 3, 6) 함수가 호출된 후 [Fig. 4]와 같이 된다.
사용자 2가 상품 67을 예약했기 때문에 상품 67은 여행 상품 목록에서 삭제되었다.
상품 69, 상품 81, 상품 55가 여행 상품 목록에 추가되었다.



Friend List

| User 1 | User 2 | User 3 | User 4 | User 5 |
|----------------------------|----------------------|--------|----------------------|--------|
| User 2 User 4 User 5 | User 1 User 4 | | User 1 User 2 | User 1 |
| Reservations | Pack. 67 Area : 4 | | Pack. 75 Area : 4 | |

Travel Package List

| | | |
|------------------------------------|------------------------------------|-----------------------------------|
| Pack. 71 Area : 1 Price : 10 | Pack. 3 Area : 2 Price : 4 | Pack. 43 Area : 3 Price : 4 |
| Pack. 69 Area : 1 Price : 5 | Pack. 81 Area : 2 Price : 10 | Pack. 55 Area : 3 Price : 6 |

[Fig. 4]

18번째에 recommend(2) 함수가 호출될 때 사용자 2와 그의 친구들이 가장 많이 예약한 지역은 4이다. 그러나, 지역이 4인 여행 상품은 여행 상품 목록에 없다. 그 다음으로 많이 예약한 지역은 1, 2, 3이다. 이 지역을 가지고 가장 싼 여행 상품은 상품 3과 상품 43이다. 상품 ID가 작은 상품인 상품 3이 추천된다.

19번째 reserve(5, 69) 함수가 호출이 된 후 [Fig. 5]와 같이 된다. 사용자 5는 상품 69를 예약하고 여행 상품 목록에서 상품 69는 삭제 된다.

Friend List

| User 1 | User 2 | User 3 | User 4 | User 5 |
|----------------------------|------------------|--------|------------------|--------|
| User 2 User 4 User 5 | User 1 User 4 | | User 1 User 2 | User 1 |

Friend List

| User 1 | User 2 | User 3 | User 4 | User 5 |
|----------------------------|------------------|--------|------------------|--------|
| User 2 User 4 User 5 | User 1 User 4 | | User 1 User 2 | User 1 |

| Reservations | Pack. 67 | | Pack. 75 | Pack. 69 |
|--------------|----------|--|----------|----------|
| | Area : 4 | | Area : 4 | Area : 1 |

Travel Package List

| | | |
|------------------------|------------------------|-----------------------|
| Pack. 71 | Pack. 3 | Pack. 43 |
| Area : 1 Price : 10 | Area : 2 Price : 4 | Area : 3 Price : 4 |
| | Pack. 81 | Pack. 55 |
| | Area : 2 Price : 10 | Area : 3 Price : 6 |

[Fig. 5]

20번째에 recommend(5)이 호출될 때, 사용자 5와 그의 친구가 가장 많이 예약한 지역은 1이다.

여행 상품 목록에 지역이 1인 여행 상품은 상품 71이므로 상품 71이 추천된다.

23번 add(28, 4, 10) 함수가 호출된 후 [Fig. 6]과 같이 된다.

Friend List

| User 1 | User 2 | User 3 | User 4 | User 5 |
|----------------------------|------------------|------------------|----------------------------|------------------|
| User 2 User 4 User 5 | User 1 User 4 | User 4 User 5 | User 1 User 2 User 3 | User 1 User 3 |

| Reservations | Pack. 67 | | Pack. 75 | Pack. 69 |
|--------------|----------|--|----------|----------|
| | Area : 4 | | Area : 4 | Area : 1 |

Travel Package List

| | | | |
|----------|----------|----------|----------|
| Pack. 71 | Pack. 3 | Pack. 43 | Pack. 28 |
| Area : 1 | Area : 2 | Area : 3 | Area : 4 |

| Friend List | | | | |
|----------------------------|------------------|------------------|----------------------------|------------------|
| User 1 | User 2 | User 3 | User 4 | User 5 |
| User 2 User 4 User 5 | User 1 User 4 | User 4 User 5 | User 1 User 2 User 3 | User 1 User 3 |
| Reservations | Pack. 67 | | Pack. 75 | Pack. 69 |
| | Area : 4 | | Area : 4 | Area : 1 |

| Travel Package List | | | |
|------------------------|------------------------|-----------------------|------------------------|
| Pack. 71 | Pack. 3 | Pack. 43 | Pack. 28 |
| Area : 1 Price : 10 | Area : 2 Price : 4 | Area : 3 Price : 4 | Area : 4 Price : 10 |
| | Pack. 81 | Pack. 55 | |
| | Area : 2 Price : 10 | Area : 3 Price : 6 | |

[Fig. 6]

24번째에 recommend(3) 함수가 호출될 때 사용자 3과 그의 친구들이 가장 많이 예약한 지역은 1과 4이다.

지역이 1 또는 4를 가지고 가장 싼 여행 상품은 상품 71과 상품 28이다.

상품 ID가 가장 작은 상품은 상품 28이므로 상품 28이 추천된다.

[제약사항]

1. 사용자의 수 N 은 5 이상 1,000 이하이다. (5 ≤ N ≤ 1,000)
2. 사용자 ID는 1 부터 N 까지 있다.
3. 지역의 수 M 은 2 이상 10 이하이다. (2 ≤ M ≤ 10)
4. 여행 상품의 지역은 1 이상 M 이하의 정수 값을 가진다.
5. 여행 상품의 상품 ID는 1 이상 1,000,000,000 이하의 정수 값을 가진다.
6. 서로 다른 여행 상품이 같은 상품 ID 값을 가질 수 없다.
7. 여행 상품의 가격은 1 이상 1,000 이하의 정수 값을 가진다.
8. 각 테스트 케이스에서 한 사용자의 친구의 수는 최대 20이다.
9. 각 테스트 케이스에 대해서 모든 함수의 총 호출 횟수는 100,000 회 이하이다.

- 9. 각 테스트 케이스에 대해서 모든 함수의 총 호출 횟수는 100,000 회 이하이다.
- 10. 각 테스트 케이스에 대해서 add() 함수는 40,000 회 이하 호출된다.
- 11. 각 테스트 케이스에 대해서 recommend() 함수는 60,000 회 이하 호출된다.

[입출력]

입출력은 제공되는 Main 부분의 코드에서 처리하므로 User Code 부분의 코드에서는 별도로 입출력을 처리하지 않는다.

sample input에 대한 정답 출력 결과는 아래와 같은 형태로 보여진다.

```
#1 100
#2 100
#3 100
#4 100
#5 100
```

연관 MOOC

S/W 문제해결 : [H1921] Recommend a Holiday Package



연관 TALK

[H1921] 여행상품추천

29 Comments

최신순 | 인기순

고정닉네임 ▼

0 / 1000 char

등록