

철도

사이가 나쁜 A, B 두 나라가 있다. A나라는 B나라를 침략하기 위해 B나라의 철도망을 알아내려 한다. B나라에 여러 차례 스파이를 보냈지만 항상 의미있는 정보를 캐기 전 잡혔기 때문에, A나라가 알고 있는 정보는 다음이 전부이다.

- B나라의 철도망은 모두 N 개의 역으로 이루어져 있고, 각 역은 1부터 N 까지 번호가 붙어 있다.
- 서로 다른 어떤 두 역을 고르더라도 직접 철도로 이어져 있거나, 철도로 이어진 다른 역(들)을 통해서 이어져 있다.
- 어떤 두 역을 고르더라도 이 둘을 연결하는 경로는 정확히 하나이다.
- 자신과 자신을 철도로 직접 이은 경우는 없다.

스파이를 보내는 것은 한계가 있음을 깨닫고, A나라는 B나라 철도회사 고위 간부를 매수하여 철도망을 그린 그림을 얻어내려고 한다. 이 그림을 직접 보내면 배신자가 누구인지 알려질 것이므로, 배신자는 다음과 같이 그림을 고쳐서 A나라에 보낼 것이다.

- 철도망을 그린 그림 위에 K 개의 가짜 철로를 그린다. 즉, 그림에서 철도로 직접 연결되지 않은 서로 다른 두 역 a 와 b 를 골라서, 이 둘을 가짜 철도로 직접 잇는다. 이를 K 번 반복한다.
- 하나의 역에 **특별한 표시**를 한다.
- 마지막으로, 역들의 번호를 모두 지운다.

배신자는 최종적으로 얻은 그림을 A나라에 전송한다. 이 정보만으로는 B나라의 철도망을 그린 그림이라는 것을 알기 어렵기 때문에, 비밀 정보가 유출되었다는 사실을 아무도 모를 것이다.

이 계획이 성공하기 위해서는 다음과 같은 문제가 해결되어야 한다.

- A나라가 받은 그림에는 역들의 번호가 지워져 있고, 또한 어느 철로가 진짜이고 가짜인지 표시되어 있지 않다. 알 수 있는 것은 어느 역에 **특별한 표시**가 되어 있는지, 그리고 가짜 철로를 총 K 군데 놓았다는 사실뿐이다.
- 따라서, 보내는 쪽에서는 받는 쪽이 그림만 보고 어느 철로가 진짜이고 어느 철로가 가짜인지 알 수 있도록 적절한 위치에 가짜 철로를 놓고, 적절한 역에 **특별한 표시**를 해야 한다.
- 또한 받는 쪽에서는 보내는 쪽이 그림을 고친 방법을 이해하고, 받은 그림에서 원래 철도망을 그린 그림을 구해야 한다.

위에서 설명한 것처럼, 철도망을 그린 그림을 고치는 함수와, 이 그림으로부터 진짜 철도망을 구하는 함수들이 필요하다. A나라는 여러분에게 이 일을 맡기려고 한다.

함수 목록 및 정의

여러분은 다음 두 함수를 구현해야 한다.

```
vector< pair<int, int> > encode_map(int N, int K, int &X,
                                   vector< pair<int, int> > E)
```

- 인자로 주어지는 정수 N 은 B나라의 철도망에 있는 역의 개수 N 을 의미한다. 모든 역은 1 이상 N 이하인 정수로 표현된다.
- 인자로 주어지는 정수 K 는 추가할 가짜 철로의 개수 K 를 의미한다.
- X 는 특별한 표시를 할 역의 번호를 기록하기 위한 인자이다. 함수가 종료될 때 X 에는 1 이상 N 이하의 양의 정수가 저장되어 있어야 한다.
- 인자로 주어지는 배열 E 의 크기는 $N - 1$ 이다. E 는 B나라의 철도망을 나타내는데, 순서쌍 (a, b) 가 E 에 저장되어 있다는 것은 역 a 와 역 b 가 직접 철로로 연결되어 있다는 뜻이다.
- 이 함수는 K 개의 순서쌍 (a, b) 가 저장된 배열을 반환하는데, 이는 가짜 철로로 서로 다른 두 역 a 와 b 를 연결한다는 뜻이다. 진짜이든 가짜이든, 이미 철로로 연결된 두 역 a 와 b 를 다시 가짜 철로로 연결할 수는 없다.

```
vector< pair<int, int> > decode_map(int N, int K, int X,
                                   vector< pair<int, int> > E)
```

- 인자로 주어지는 정수 N 은 B나라의 철도망에 있는 역의 개수 N 을 의미한다. 모든 역은 1 이상 N 이하인 정수로 표현된다. 그림을 보내는 과정에서 역의 번호가 모두 지워졌기 때문에, **이 함수에서 사용하는 역의 번호는 실제 역의 번호와 다를 수 있다.** 즉, 이 함수에서 사용하는 역의 번호는 `encode_map` 함수에서 사용한 역의 번호를 어떤 전단사 함수로 계산한 결과이다.
- 인자로 주어지는 정수 K 는 추가된 가짜 철로의 개수 K 를 의미한다.
- 인자로 주어지는 정수 X 는 특별한 표시가 되어 있는 역의 번호를 의미한다.
- 인자로 주어지는 배열 E 의 크기는 $N + K - 1$ 이다. E 는 A나라로 전달된 그림을 나타내는데, 순서쌍 (a, b) 가 E 에 저장되어 있다는 것은 그림에서 역 a 와 역 b 가 직접 진짜/가짜 철로로 연결되어 있다는 뜻이다. 참고로, E 에서 원소의 순서는 아무 의미가 없다.
- 이 함수는 `encode_map` 함수로 만든 그림에서 원래 그림을 복원하는 역할을 한다. 즉, 이 함수는 $N - 1$ 개의 순서쌍 (a, b) 가 저장된 배열을 반환하는데, 여기에 B나라의 철도망이 저장되어 있어야 한다.

제출하는 소스 코드의 어느 부분에서도 입출력 함수를 실행해서는 안 된다.

각각의 테스트 케이스는 하나 이상의 독립적인 시나리오를 포함한다. T 개의 시나리오를 포함하는 테스트 케이스 하나에 대해서, 위의 함수를 호출하는 프로그램은 다음과 같이 정확히 두 번 실행된다.

프로그램이 첫 번째로 실행될 때는 다음과 같다.

- `encode_map` 함수가 T 번 호출된다.
- `encode_map` 함수의 실행 결과가 채점 시스템에 저장된다.
- `decode_map` 함수는 호출되지 않는다.

프로그램이 두 번째로 실행될 때는 다음과 같다.

- `decode_map` 함수가 여러 번 호출된다. 각각의 호출에서 임의의 시나리오가 선택되며, 이 시나리오에서 `encode_map` 함수로 만든 그림이 `decode_map` 함수의 입력으로 사용된다.
- `encode_map` 함수는 호출되지 않는다.

이 문제에서 프로그램의 실행 시간과 메모리 사용량은 두 번의 실행을 합하여 계산된다.

제약 조건

- $1 \leq T \leq 200$
- $3 \leq N \leq 200$
- $1 \leq K < \frac{N}{2}$

부분문제

1. (4점)
 - $N \leq 4$
2. (13점)
 - $K = 1$
3. (11점)
 - 각각의 역에 연결된 철로는 최대 두 개이다.
4. (29점)
 - 임의의 역에서 최대 $\lfloor \frac{N}{2} \rfloor$ 개의 철로를 거쳐 다른 임의의 역으로 이동할 수 있다.
5. (43점)
 - 추가적인 제약 조건이 없다.

채점 기준

`encode_map` 함수에서 올바른 그림을 만들고, `decode_map` 함수에서 철도망을 정확하게 알아낸 경우에만 해당 데이터를 맞힌 것으로 채점된다. 특히, `decode_map` 함수가 리턴한 배열에 가짜 철로가 하나라도 포함되어 있으면 해당 데이터의 점수는 0이다. `encode_map` 함수와 `decode_map` 함수가 리턴하는 배열에서 원소의 순서는 중요하지 않다.

각 부분문제의 점수는 그 부분문제의 모든 데이터에 대한 점수 중 최솟값임에 유의하라.

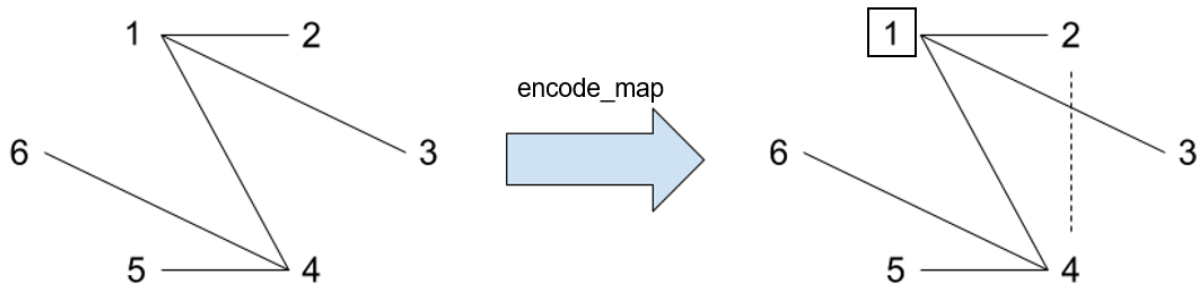
예제

- $N = 6$, $K = 1$, 철로 $E = [(1, 2), (1, 3), (1, 4), (4, 5), (4, 6)]$ 인 경우를 생각해 보자.

그레이더는 다음 함수를 호출한다.

```
encode_map(6, 1, X, [(1, 2), (1, 3), (1, 4), (4, 5), (4, 6)])
```

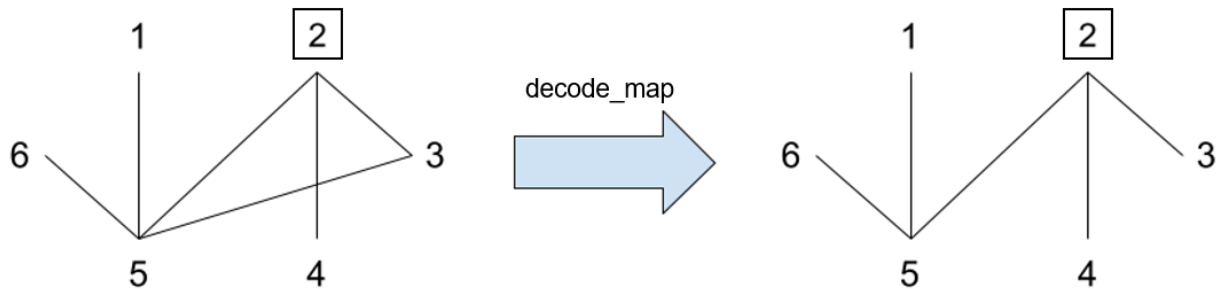
왼쪽 그림은 B나라의 철도망을 나타낸다. 가짜 철로 (4, 2)를 추가하고, 1번 역에 특별한 표시를 하면 오른쪽 그림을 얻을 수 있다. 이 경우 `encode_map` 함수는 [(4, 2)]를 반환하며, X에는 1이 저장된다. 이 외에도 다른 답이 존재할 수 있다.



다음으로, 그레이더는 아래의 함수를 호출한다. 마지막 인자로 주어지는 배열의 원소 순서는 바뀔 수 있다.

```
decode_map(6, 1, 2, [(1, 5), (2, 3), (2, 4), (2, 5), (3, 5), (5, 6)])
```

입력이 나타내는 그림은 다음과 같다. 역의 번호가 `encode_map` 함수에서 사용한 것과 다름에 유의하라.



이 함수는 [(3, 2), (4, 2), (5, 6), (1, 5), (5, 2)]를 반환한다. 원소의 순서는 바뀔 수 있다. 이 예제는 2, 4, 5번 부분문제의 조건을 만족한다.

Sample grader

Sample grader는 아래와 같은 형식으로 입력을 받는다.

- Line 1: T

이후 각각 하나의 시나리오를 나타내는 T 개의 블록이 따라온다. 각 블록의 형식은 다음과 같다.

- Line 1: $N \ K$
- Line $1 + i$ ($1 \leq i \leq N - 1$): $a_i \ b_i$

a_i, b_i 쌍은 a_i 번 역과 b_i 번 역을 잇는 철도가 존재함을 의미한다.

Sample grader는 각 시나리오에 대해 아래와 같은 형식으로 출력한다.

- Line 1: 함수 `encode_map`이 반환한 배열
- Line 2: 함수 `decode_map`이 반환한 배열

Sample grader는 실제 채점에서 사용하는 그레이더와 다를 수 있음에 유의하라.