

## 세계 지도

볼리비아 고고학자인 Mr. Pacha는 티와나쿠 시대(300-1000 CE)의 세계를 설명하는 고대 문서를 발견했다. 티와나쿠 시대에는  $N$ 개의 국가가 있었고 이 국가들은 1번부터  $N$ 번까지로 구분된다.

이 문서에는 인접 국가를 나타내는  $M$ 개의 서로 다른 쌍이 적힌 하나의 목록이 있다:

$$(A[0], B[0]), (A[1], B[1]), \dots, (A[M-1], B[M-1]).$$

각  $i$  ( $0 \leq i < M$ )에 대해, 이 문서는 국가  $A[i]$ 와 국가  $B[i]$ 가 서로 인접했다고 기술하고 있다. 이 목록에 포함되지 않은 국가들의 쌍은 서로 인접하지 않았다.

Mr. Pacha는 티와나쿠 시대의 국가들 사이의 인접성을 정확히 나타내는 세계 지도를 만들고 싶다. 이를 위해, 그는 먼저 양의 정수  $K$ 를 선택한다. 그 다음,  $K \times K$ 개의 정사각형 셀들의 격자로 지도를 그리는데, 이때 행 번호는 0부터  $K-1$ 까지(위에서 아래로)이고 그리고 열 번호는 0부터  $K-1$ 까지(왼쪽에서 오른쪽으로)이다.

그는 지도의 각 셀을  $N$ 개의 색 중 하나로 칠하고 싶다. 이 색들은 1부터  $N$ 까지 번호가 붙어있고, 국가  $j$  ( $1 \leq j \leq N$ )는 색  $j$ 로 표현된다. 색칠은 다음 **조건들**을 모두 만족해야 한다:

- 각  $j$  ( $1 \leq j \leq N$ )에 대해, 색  $j$ 로 칠해진 셀이 최소 하나는 존재한다.
- 각 인접 국가 쌍  $(A[i], B[i])$ 에 대해, 한 셀은  $A[i]$ 로 칠해지고 다른 셀은  $B[i]$ 로 칠해진 인접한 두 셀의 쌍이 최소 하나는 존재한다. 두 셀이 한 변을 공유하는 경우 두 셀이 인접하다고 한다.
- 다른 색으로 칠해진 인접한 두 셀의 쌍 각각에 대해, 이들 두 색으로 표현된 나라들은 티와나쿠 시대에 서로 인접했다.

예를 들어, 만약  $N = 3$ ,  $M = 2$ 이고 인접 국가들의 쌍이  $(1, 2)$ 와  $(2, 3)$ 이라면, 쌍  $(1, 3)$ 은 서로 인접하지 않았고, 크기  $K = 3$ 인 다음 지도는 모든 조건을 만족한다.

2	3	3
2	3	2
1	2	1

특히, 한 국가가 지도에서 연결된 영역 형태일 필요는 **없다**. 위의 지도에서, 국가 3은 하나의 연결된 영역 형태이지만 국가 1과 2는 끊어진 영역 형태이다.

당신은 Mr. Pacha가  $K$  값을 선택하고 지도를 만드는 것을 도와야 한다. 고대 문서는 지도가 존재함을 보장한다. Mr. Pacha는 더 작은 지도를 선호하기에, 마지막 서브태스크에서 당신의 점수는  $K$  값에 의존하고, 더 작은  $K$  값은 더 좋은 점수를 얻게 한다. 하지만, 가능한 최솟값  $K$ 를 찾을 필요는 없다.

## Implementation Details

다음 함수를 구현해야 한다.

```
std::vector<std::vector<int>> create_map(int N, int M,  
    std::vector<int> A, std::vector<int> B)
```

- $N$ : 국가 수
- $M$ : 인접 국가 쌍의 수
- $A$ 와  $B$ : 인접 국가를 나타내는 길이  $M$ 인 배열들
- 이 함수는 각 테스트 케이스에 대해 **최대 50번** 호출된다.

이 함수는 지도를 나타내는 배열  $C$ 를 리턴해야 한다.  $K$ 를  $C$ 의 길이로 두자.

- $C$ 의 각 원소는 1부터  $N$ 까지 중의 정수로 구성되는 길이  $K$ 인 배열이어야 한다.
- $C[i][j]$ 는  $i$ 행  $j$ 열 셀의 색이다( $0 \leq i, j < K$ 인 각  $i$ 와  $j$ 에 대해).
- $K$ 는 240보다 작거나 같아야 한다.

## Constraints

- $1 \leq N \leq 40$
- $0 \leq M \leq \frac{N \cdot (N-1)}{2}$
- $1 \leq A[i] < B[i] \leq N$  ( $0 \leq i < M$ 인 각  $i$ 에 대해)
- 쌍들  $(A[0], B[0]), \dots, (A[M-1], B[M-1])$ 은 서로 다르다.
- 모든 조건을 만족하는 지도가 최소 하나는 존재한다.

## Subtasks and Scoring

Subtask	Score	Additional Constraints
1	5	$M = N - 1, A[i] = i + 1, B[i] = i + 2$ (각 $0 \leq i < M$ 에 대해)
2	10	$M = N - 1$
3	7	$M = \frac{N \cdot (N-1)}{2}$
4	8	국가 1은 다른 모든 국가와 인접하다. 다른 국가 쌍도 인접할 수 있다.
5	14	$N \leq 15$
6	56	추가적인 제한이 없다.

서브태스크 6에서, 당신의 점수는  $K$  값에 의존한다.

- `create_map`이 리턴하는 어떤 지도가 모든 조건을 만족하지 못한다면, 이 서브태스크에서 당신의 점수는 0이 될 것이다.
- 그렇지 않다면, 모든 `create_map` 호출에서  $K/N$ 의 **최댓값**을  $R$ 로 두자. 그러면 당신은 다음 표에 따라 **부분 점수**를 받는다:

Limits	Score
$6 < R$	0
$4 < R \leq 6$	14
$3 < R \leq 4$	28
$2.5 < R \leq 3$	42
$2 < R \leq 2.5$	49
$R \leq 2$	56

## Example

CMS에서, 다음 두 시나리오를 모두 포함하는 테스트 케이스가 있다.

### Example 1

다음 호출을 생각해보자:

```
create_map(3, 2, [1, 2], [2, 3])
```

이것은 문제 설명에 있던 예로, 함수는 다음 지도를 리턴할 수 있다.

```
[
  [2, 3, 3],
  [2, 3, 2],
  [1, 2, 1]
]
```

## Example 2

다음 호출을 생각해보자:

```
create_map(4, 4, [1, 1, 2, 3], [2, 3, 4, 4])
```

이 예에서,  $N = 4$ ,  $M = 4$ 이고 인접 국가들의 쌍이 (1,2), (1,3), (2,4), 그리고 (3,4)이다. 따라서, (1,4)와 (2,3)은 인접하지 않다.

함수는 모든 조건을 만족하는 크기  $K = 7$ 인 다음 지도를 리턴할 수 있다.

```
[
  [2, 1, 3, 3, 4, 3, 4],
  [2, 1, 3, 3, 3, 3, 3],
  [2, 1, 1, 1, 3, 4, 4],
  [2, 2, 2, 1, 3, 4, 3],
  [1, 1, 1, 2, 4, 4, 4],
  [2, 2, 1, 2, 2, 4, 3],
  [2, 2, 1, 2, 2, 4, 4]
]
```

지도는 더 작을 수 있다; 예를 들어, 함수는 크기  $K = 2$ 인 다음 지도를 리턴할 수 있다.

```
[
  [3, 1],
  [4, 2]
]
```

참고로 두 지도 모두  $K/N \leq 2$ 를 만족한다.

## Sample Grader

입력의 첫 줄은 하나의 정수  $T$ 로, 이는 시나리오의 개수이다.  $T$ 개의 시나리오에 대한 내용이 이어서 나오는데, 각 시나리오의 형식은 아래와 같다.

입력 형식:

```
N M
A[0] B[0]
:
A[M-1] B[M-1]
```

출력 형식:

```
P
Q[0] Q[1] ... Q[P-1]

C[0][0] ... C[0][Q[0]-1]
:
C[P-1][0] ... C[P-1][Q[P-1]-1]
```

여기에서,  $P$ 는 `create_map`이 리턴하는 배열  $C$ 의 길이이고,  $Q[i]$  ( $0 \leq i < P$ )는  $C[i]$ 의 길이이다. 참고로 출력 형식의 line 3은 의도적으로 빈 줄로 남겨둔다.