

그리드 복원 (grid_encoding)

다니엘은 각 셀이 검정색 또는 흰색으로 칠해진 $N \times N$ 그리드를 가지고 있다. 이 그리드의 상태는 각 원소가 0 또는 1인 $N \times N$ 행렬 A 로 표시된다. $0 \leq i, j \leq N - 1$ 에 대해, $A[i][j] = 1$ 은 i 번 행 j 번 열의 셀 (앞으로 이를 편의상 셀 (i, j) 로 표기한다)이 검정색인 경우를, $A[i][j] = 0$ 은 흰색인 경우를 의미한다. 한편, 그리드에는 다음 조건들을 만족하는 (i_1, i_2, j_1, j_2) 가 존재하지 않는 것이 알려져 있다:

- $0 \leq i_1 < i_2 \leq N - 1, 0 \leq j_1 < j_2 \leq N - 1$
- $A[i_1][j_1] = A[i_2][j_2], A[i_1][j_2] = A[i_2][j_1], A[i_1][j_1] \neq A[i_1][j_2]$

다니엘은 그리드를 영욱이에게 전달하고자 한다. 통신은 보안상의 이유로 다음과 같은 절차를 따른다:

- 다니엘은 총 N^2 개의 셀 중에서 몇 개의 셀을 선택한다.
- 통신 시스템은 $(0, 1, 2, \dots, N - 1)$ 의 비밀 순열인 X, Y 을 가지고 있다.
- $N \times N$ 행렬 B 가 영욱이에게 전달된다. 다니엘이 선택한 각 셀 (i, j) 에 대해 $B[X[i]][Y[j]] = A[i][j]$ 가 성립하고, 선택하지 않은 셀 (i, j) 에 대해 $B[X[i]][Y[j]] = -1$ 이 성립한다.

영욱이는 행렬 B 에서 -1 로 채워진 부분을 복구하여 모든 $0 \leq i, j \leq N - 1$ 에 대해 $B[X[i]][Y[j]] = A[i][j]$ 가 성립하도록 만들어야 한다.

함수 목록 및 정의

여러분은 아래 함수들을 구현해야 한다.

```
void send(std::vector<std::vector<int> > A)
```

- A : 크기가 N 인 이차원 벡터. 각 원소인 벡터의 크기 역시 N 이다.
- 이 함수 내에서 `select` 함수를 호출하여 전달할 셀을 선택하여야 한다.

```
void select(int i, int j)
```

- 호출 시 $0 \leq i, j \leq N - 1$ 을 만족해야 한다.
- 이 함수가 호출된 총 횟수를 K 라 한다.

```
vector<vector<int> > reconstruct(vector<vector<int> > B)
```

- grader는 $(0, 1, \dots, N - 1)$ 의 순열 X, Y 를 가지고 있다.
- 원래 send에서 주어진 이차원 벡터 A 에 대해, 아래 조건을 만족하는 B 가 파라미터로 주어진다.
 - B 는 A 와 동일한 모양의 이차원 벡터이다.
 - $0 \leq i, j \leq N - 1$ 에 대해, send 함수에서 $\text{select}(i, j)$ 가 호출된 경우 $B[X[i]][Y[j]] = A[i][j]$ 가 성립한다.
 - $0 \leq i, j \leq N - 1$ 에 대해, send 함수에서 $\text{select}(i, j)$ 가 호출된 적이 없는 경우 $B[X[i]][Y[j]] = -1$ 이 성립한다.
- 함수의 리턴값 C 는 모든 $0 \leq i, j \leq N - 1$ 에 대해 $C[X[i]][Y[j]] = A[i][j]$ 를 만족해야 한다.

send 함수에서의 select 호출 및 reconstruct 함수의 반환값은 주어진 파라미터의 값에만 의존해야 한다. 함수를 같은 파라미터 값으로 여러 번 호출했을 때 다른 행동이 발생하면 오답으로 처리될 수 있다.

채점 과정에서 순열 X, Y 는 미리 결정되어 있고, 비적응적이다(non-adaptive). 그러나 참가자는 이에 접근할 수 없으며, 샘플 그레이더에서는 X 와 Y 가 입력으로 주어진다.

각 입력은 여러 개의 독립적인 테스트 케이스로 이루어진다. 각 테스트 케이스별로 send 및 reconstruct 함수가 한 번씩 호출된다. send 및 reconstruct 함수가 테스트 케이스 순서대로 호출된다는 보장은 하지 않지만, 함수의 호출 및 반환값을 바탕으로 지문에서 제시된 방식대로 동작한다는 것은 보장된다.

Sample grader와 달리, 실제 채점 시 여러분의 프로그램은 각 입력마다 두 번 실행된다. 첫 번째 실행에서는 각 테스트 케이스에 대해 send 함수를 호출하여 실행 결과를 출력하고, 두 번째 실행에서는 첫 번째 실행의 출력 결과를 입력으로 받아 reconstruct 함수를 실행한다. 각 테스트 케이스에 대해 대회 시스템 상에서 수행 시간은 두 번 실행한 프로그램의 수행 시간의 합으로 측정되며, 메모리 사용량 또한 두 번 실행한 프로그램의 메모리 사용량의 합으로 측정된다. 시간 제한과 메모리 제한은 두 번의 실행 결과를 합친 것을 기준으로 한다. 첫 번째 실행에서만 send 함수가 호출되기 때문에 select 함수의 호출 횟수 제약 조건은 두 번 실행되는 것에 영향을 받지 않는다.

제출하는 소스 코드의 어느 부분에서도 입출력 함수를 실행해서는 안 된다.

제약 조건

- $1 \leq N \leq 500$
- send 내에서 select의 호출 횟수 K 는 N^2 를 넘을 수 없다.
- 모든 테스트 케이스에서 N^2 을 합한 값은 10^6 을 넘지 않는다.

부분문제

1. (12점)

- $0 \leq i, j \leq N - 1$ 에 대해, $i \leq j$ 와 $A[i][j] = 1$ 가 동치이다.

2. (35점)

- 그리드가 히스토그램 형태이다. 즉, 모든 $0 \leq j \leq N - 1$ 에 대해 $0 \leq H_j \leq N$ 인 H_j 가 존재하여 $i < H_j$ 이면 $A[i][j] = 1$, 그렇지 않으면 $A[i][j] = 0$ 이 성립한다.

3. (53점)

- 추가적인 제약 조건이 없다.

`reconstruct` 함수의 리턴값이 $C[X[i]][Y[j]] = A[i][j]$ 를 만족하는 경우, 각 부분문제에 대해 다음과 같이 부분점수가 존재한다. 해당 조건을 만족하지 않는 경우에는 0점을 받는다.

A 의 크기 N 과 `select`의 호출 횟수 K 에 대해

- 모든 테스트 케이스에 대해 $K \leq 2N - 1$ 을 만족하면 모든 점수를 받는다.
- 그렇지 않다면 $c \cdot N \geq K$ 를 만족하는 가장 작은 정수 c 에 대해, $c \leq 10$ 인 경우 부분문제 점수의 $\frac{110-9c}{100}$ 만큼 받는다.
- 두 조건에 해당하지 않고, $K \leq \frac{N^2}{2} + 1$ 를 만족하는 경우 부분문제 점수의 $\frac{7}{100}$ 를 받는다.
- 위 세 가지 조건들에 모두 해당하지 않는 경우 해당 부분문제에 대한 점수를 받을 수 없다.

예제 1

$N = 3$, $A = [[1, 1, 1], [1, 1, 0], [0, 1, 0]]$, $X = [2, 1, 0]$, $Y = [0, 1, 2]$

인 경우를 생각해 보자.

그레이더는 먼저 `send` 함수를 호출한다.

```
send([[1, 1, 1], [1, 1, 0], [0, 1, 0]])
```

`send` 함수 내에서의 함수 호출이 다음과 같았다고 하자.

```
select(0, 1)
select(0, 2)
select(1, 0)
select(2, 2)
```

그 후 그레이더는 다음과 같이 함수를 호출한다.

```
reconstruct([[ -1, -1, 0], [1, -1, -1], [ -1, 1, 1]])
```

`reconstruct` 함수의 리턴값 C 는 모든 $0 \leq i, j \leq N - 1$ 에 대해 $C[X[i]][Y[j]] = A[i][j]$ 를 만족해야 하므로, `reconstruct([[-1, -1, 0], [1, -1, -1], [-1, 1, 1]])`는 $[[0, 1, 0], [1, 1, 0], [1, 1, 1]]$ 을 리턴해야 한다.

Sample grader

Sample grader는 테스트 케이스의 개수 T 를 입력 받는다. 그 이후 T 회에 걸쳐 다음과 같은 정보를 입력받는다:

- Line 1: N
- Line $2 + k(0 \leq k \leq N - 1)$: $A[k][0] A[k][1] \cdots A[k][N - 1]$
- Line $N + 2$: $X[0] X[1] \cdots X[N - 1]$
- Line $N + 3$: $Y[0] Y[1] \cdots Y[N - 1]$

Sample grader는 각 테스트 케이스에 대해 다음을 출력한다.

- 만약 호출된 `select(i, j)` 가 $0 \leq i, j \leq N - 1$ 을 만족하지 않는 경우 한 줄에 Wrong Answer [1]를 출력한다.
- `select`함수의 호출 횟수가 N^2 를 초과하는 경우 한 줄에 Wrong Answer [2]를 출력한다.
- `reconstruct`함수의 리턴값이 B 와 동일한 형태가 아닌 경우 한 줄에 Wrong Answer [3]를 출력한다.
- `reconstruct`함수의 리턴값이 조건을 만족하도록 행렬을 복원하지 못한 경우 한 줄에 Wrong Answer [4]를 출력한다.
- 그 외의 경우 K: 10과 같이 `select`함수의 호출 횟수를 출력한다.

Wrong Answer를 출력할 시, Sample grader는 즉시 종료된다.

Sample grader는 실제 채점에서 사용하는 그레이더와 다를 수 있음에 유의하라.