

N개의 피자를 동시에 구울 수 있는 화덕이 있다. 피자는 치즈가 모두 녹으면 화덕에서 꺼내며, 치즈의 양은 피자마다 다르다. 1번부터 M번까지 M개의 피자를 순서대로 화덕에 넣을 때, 치즈의 양에 따라 녹는 시간이 다르기 때문에 꺼내지는 순서는 바뀔 수 있다.

주어진 조건에 따라 피자를 구울 때, 화덕에 가장 마지막까지 남아있는 피자 번호를 알아내는 프로그램을 작성하시오.

- 피자는 1번위치에서 넣거나 뺄 수 있다.
- 화덕 내부의 피자받침은 천천히 회전해서 1번에서 잠시 꺼내 치즈를 확인하고 다시 같은 자리에 넣을 수 있다.
- M개의 피자에 처음 뿌려진 치즈의 양이 주어지고, 화덕을 한 바퀴 돌 때 녹지않은 치즈의 양은 반으로 줄어든다. 이전 치즈의 양을 C라고 하면 다시 꺼냈을 때 $C//2$ 로 줄어든다.
- 치즈가 모두 녹아 0이 되면 화덕에서 꺼내고, 바로 그 자리에 남은 피자를 순서대로 넣는다.

[입력]

첫 줄에 테스트 케이스 개수 T가 주어진다. $1 \leq T \leq 50$

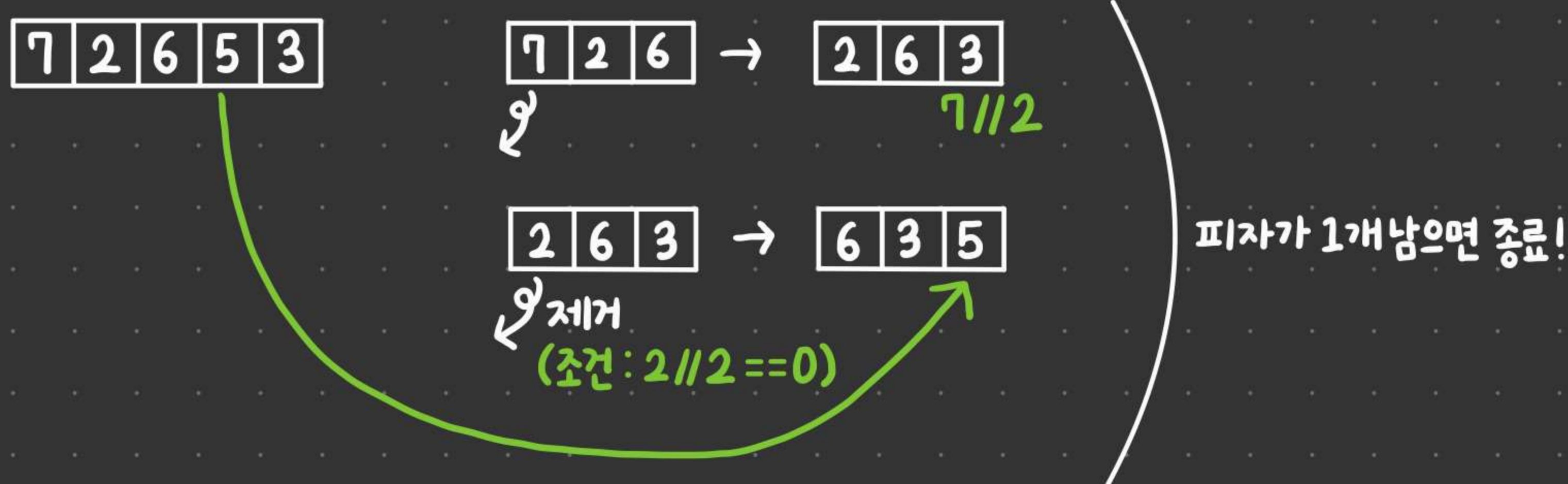
다음 줄부터 테스트 케이스의 첫 줄에 화덕의 크기 N과 피자 개수 M이 주어지고, 다음 줄에 M개의 피자에 뿌려진 치즈의 양을 나타내는 C_i 가 주어진다.

$3 \leq N \leq 20$, $N \leq M \leq 100$, $1 \leq C_i \leq 20$

[출력]

각 줄마다 "#T" (T는 테스트 케이스 번호)를 출력한 뒤, 번호를 출력한다.

예시) $N=3$ $M=5$ $C_i = [7, 2, 6, 5, 3]$



[SWEA] 5102 노드의 거리

V개의 노드 개수와 방향성이 없는 E개의 간선 정보가 주어진다. 주어진 출발 노드에서 최소 몇 개의 간선을 지나면 도착 노드에 갈 수 있는지 알아내는 프로그램을 만드시오. 노드 번호는 1번부터 존재하며, 노드 중에는 간선으로 연결되지 않은 경우도 있을 수 있다.

[입력]

첫 줄에 테스트 케이스 개수 T가 주어진다. $1 \leq T \leq 50$

다음 줄부터 테스트 케이스의 첫 줄에 V와 E가 주어진다. $5 \leq V \leq 50$, $4 \leq E \leq 1000$

테스트케이스의 둘째 줄부터 E개의 줄에 걸쳐, 간선의 양쪽 노드 번호가 주어진다. E개의 줄 이후에는 출발 노드 S와 도착 노드 G가 주어진다.

[출력]

각 줄마다 "#T" (T는 테스트 케이스 번호)를 출력한 뒤, 답을 출력한다. 두 노드 S와 G가 서로 연결되어 있지 않다면, 0을 출력한다.

예시) $V=6$ $E=5$ $arr = [(1,4), (1,3), (2,3), (2,5), (4,6)]$
 $S=1$ $G=6$

i) 인접배열 리스트 생성

	3,4	3,5	1,2	1,6	2	4
	1	2	3	4	5	6

ii) 방문리스트 생성

0	1	0	0	0	0	0
	1	2	3	4	5	6

출발노드

iii) queue

1	3	4
---	---	---

제거

조건) 방문리스트 값이 0

출발지, 도착지 포함하는 최단거리 -2

$N \times N$ 크기의 미로에서 출발지 목적지가 주어진다. 이때 최소 몇 개의 칸을 지나면 출발지에서 도착지에 다다를 수 있는지 알아내는 프로그램을 작성하시오. 경로가 있는 경우 출발에서 도착까지 가는데 지나야 하는 최소한의 칸 수를, 경로가 없는 경우 0을 출력한다.

다음은 5x5 미로의 예이다. 1은 벽, 0은 통로를 나타내며 미로 밖으로 벗어나서는 안된다.

13101
10101
10101
10101
10021

마지막 줄의 2에서 출발해서 0인 통로를 따라 이동하면 맨 윗줄의 3에 5개의 칸을 지나 도착할 수 있다.

[입력]

첫 줄에 테스트 케이스 개수 T 가 주어진다. ($1 \leq T \leq 50$)

다음 줄부터 테스트 케이스의 별로 미로의 크기 N 과 N 개의 줄에 걸쳐 미로의 통로와 벽에 대한 정보가 주어진다. ($5 \leq N \leq 100$)

0은 통로, 1은 벽, 2는 출발, 3은 도착이다.

[출력]

각 줄마다 "#T" (T 는 테스트 케이스 번호)를 출력한 뒤, 답을 출력한다.

예시)

1 3 0 0 1
1 0 1 0 1
1 0 1 0 1
1 0 1 0 1
1 0 0 2 1

i) 출발 인덱스 찾기 [0,1] 혹은 2

ii) delta 이용해서 주변에 값이 0인 경우 찾기
[(0,1), (1,0), (0,-1), (-1,0)]
우 하 좌 상

조건) 배열 범위 내!!

1 3 0 0 1
1 0 1 0 1
1 0 1 0 1
1 0 1 0 1
1 0 0 2 1

iii) 0을 찾으면 기록!! 조건) 이전에 방문하지 않은 곳
→ Queue → 리스트 생성하여 방문여부 관리

(0,2) (1,1)

0	1	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

iv) 0으로 이동! (주변에 0이 여러개일 경우 순차적으로 방문)

0	1	2	0	0
0	2	0	0	0
0	0	0	0	0
0	0	0	0	0

이동하면서 방문했음을 표시
→ 이전 방문 지점의 값 + 1로 표시
거리계산을 하기 위함

혹은 2

v) Queue에 기록된 첫번째 좌표의 주변에 값이 0인 경우 찾기

& Queue에서 제거

1 3 0 0 1
1 0 1 0 1
1 0 1 0 1
1 0 1 0 1
1 0 0 2 1

(0,2) (1,1) (0,3)

← 제거

0	1	2	3	0
0	2	0	0	0
0	0	0	0	0
0	0	0	0	0

반복 (도착지에 도착하면 끝!)

N 장의 카드에 1번 부터 N번 까지 숫자가 부여되어 있으며, 1번 카드가 제일 위에, 2번 카드가 그 다음, N번 카드가 가장 아래인 상태로 쌓여있다. 숫자 카드가 한 장 남을 때까지 다음 동작을 반복하게 된다.

- 가장 위에 있는 카드를 가져와서 버린다.
- 그 다음, 가장 위에 있는 카드를 가장 아래에 있는 카드 밑으로 옮긴다.

N = 4인 경우를 생각해 보자.

카드는 위에서부터 [1, 2, 3, 4] 의 순서로 놓여있다. 1을 버리면 [2, 3, 4]가 남는다. 여기서 2를 아래로 옮기면 [3, 4, 2]가 된다. 3을 버리면 [4, 2]가 되고, 4를 밑으로 옮기면 [2, 4]가 된다. 마지막으로 2를 버리고 나면, 남는 카드는 4가 된다.

N이 주어졌을 때, 가장 마지막에 남는 카드를 구하시오.

[입력]

첫 줄에 테스트 케이스 개수 T가 주어진다. ($1 \leq T \leq 20$)

각 테스트 케이스 마다 한줄에 N 이 주어진다. ($1 \leq N \leq 100000$)

[출력]

각 줄마다 "#T" (T는 테스트 케이스 번호)를 출력한 뒤, 마지막에 남는 카드 번호를 출력한다.

예시) N=4



배열 [0]을 제거!
배열 [1]을 제거후 배열 [-1]로 추가) 반복 → 앞에서부터 제거!
Queue