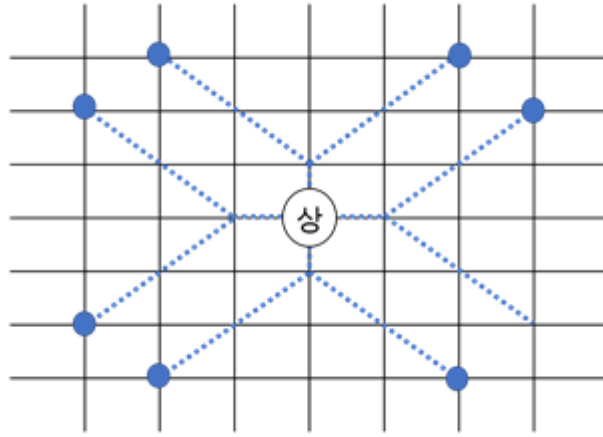


## 문제1] 상 움직이기

장기의 말 중에는 상이 있다. 상의 이동 방법은 한 칸 직진 후 대각으로 2칸 이동하여 움직인다.

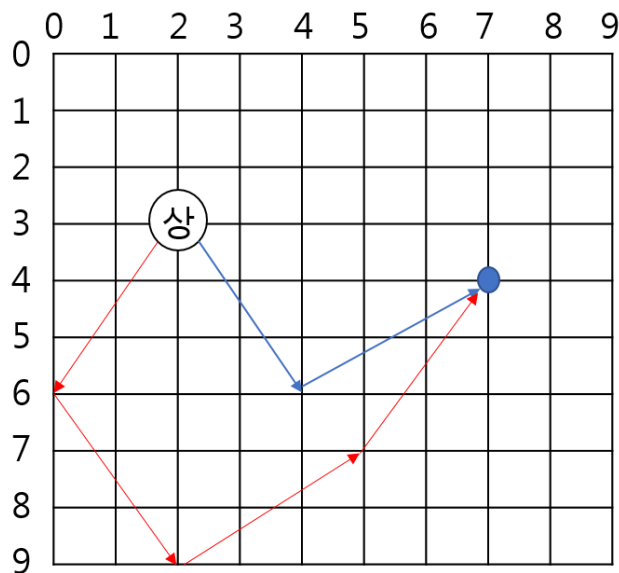
따라서 어느 한곳에서 이동할 수 있는 위치는 다음 그림과 같이 8곳이다.



[그림1]

상의 위치가 주어지고 이동하고자 하는 위치가 주어 질 때 최소 몇 번을 움직여 이동할 수 있는지 이동 횟수의 최소값을 출력하는 프로그램을 작성하시오.

예를 들어 아래 그림과 같이 상의 위치가 (3, 2)이고 움직이고자 하는 위치가 (4, 7)일 경우 최소 2번을 움직여 이동할 수 있으므로 답은 2가 된다. 그림에서 다른 이동 방법의 예도 보여주고 있다.



[그림 2]

단, 상은 게임 판의 범위를 벗어 날수 없다.

## 입력

첫 줄에 테스트 케이스 개수  $T$ 가 주어진다.  $1 \leq T \leq 10$

다음 줄부터 테스트 케이스의 별로 첫 째 줄에는 게임 판의 크기  $N$ 과 ( $10 \leq N \leq 20$ ) 두 번째 줄에는 상의 위치  $x, y$ 와 이동하려는 위치  $tx, ty$ 가 주어진다. ( $0 \leq x, y, tx, ty < N$ )

## 출력

#과 1번부터인 테스트케이스 번호, 빈칸에 이어 상이 처음 위치에서 이동하고자 하는 위치까지 이동 가능한 경우 중 이동 횟수가 최소인 값을 출력한다.

단, 10개의 테스트케이스를 수행하여 답을 출력하는 시간은 10초를 넘지 않아야 한다.

## 입력 예시

```
3                # 테스트 케이스 수

10              # 1번 테스트 케이스의 N

0 1 8 4         # 1번 테스트 케이스의 상의 위치 x, y, 이동하려는 위치 tx, ty

10

0 0 2 3

10

4 4 3 3
```

## 출력 예시

```
#1 3

#2 1

#3 2
```

## 문제2] 6등분하기

정수 값이 채워져 있는 N, M 크기의 2차 배열이 주어 진다. 가로로 한번 구역을 나누고 세로로 두 번 구역을 나누면 6개의 작은 구역으로 나누어 진다. 예를 들어 아래 그림은 3행 8열 크기의 2차 배열을 6구역으로 나누는 하나의 예를 보여 주고 있다.

1	3	8	4	-3	-3	5	1
8	-2	5	5	4	-9	-8	-1
4	9	2	-1	0	4	3	2

[그림1]

6 구역의 요소들을 더하면

$$1\text{구역} : 1 + 3 = 4$$

$$2\text{구역} : 8 + 4 + (-3) = 9$$

$$3\text{구역} : -3 + 5 + 1 = 3$$

$$4\text{구역} : 8 + (-2) + 4 + 9 = 19$$

$$5\text{구역} : 5 + 5 + 4 + 2 + (-1) + 0 = 15$$

$$6\text{구역} : -9 + (-8) + (-1) + 4 + 3 + 2 = -9$$

6개의 구역 합을 구할 수 있다.

이 6개 중 임의로 3개를 선택하여 두 구역 간의 차이의 절대 값을 합산해 볼 수 있다. 예를 들어 1구역, 2구역 3구역을 선택하여 두 구역 간의 차이의 절대 값을 합산하면 다음과 같다.

$$|4 - 9| + |9 - 3| + |3 - 4| = 12$$

같은 방법으로 2구역, 3구역, 5구역을 선택하여 계산하면 다음과 같다.

$$|9 - 3| + |3 - 15| + |15 - 9| = 24$$

따라서 첫 번째 계산 결과 보다 두 번째 계산 결과가 크다.

예로 주어진 배열에서 위와 같은 방법으로 계산하였을 때 가장 절대값이 큰 경우는 다음과 같다.

1	3	8	4	-3	-3	5	1
8	-2	5	5	4	-9	-8	-1
4	9	2	-1	0	4	3	2

[그림2]

1구역부터 6구역까지의 구역 합은 각각 33, -15, 0, 14, 7, 2 이며 이중 다음과 같이 세 구역의 합을 선택하여 각각 차이의 절대값의 합을 계산하면 96로 가장 큰 값을 구할 수 있다.

$$|33 - (-15)| + |-15 - 0| + |0 - 33| = 96$$

주어진 배열을 6등분하여 6구역을 만들고 각 구역의 합을 구한 후 6개의 합 중 임의의 3개를 선택하여 각각의 절대 값을 차이를 계산할 때 가장 큰 값을 구하는 프로그램을 작성하시오.

## 입력

첫 줄에 테스트 케이스 개수 T가 주어진다.  $1 \leq T \leq 10$

다음 줄부터 테스트 케이스의 별로 첫째 줄에는 2차 배열의 행의 크기 N과 열의 크기 M ( $3 \leq N, M \leq 10$ )이 주어지고 두 번째 줄부터 N줄에 걸쳐 M개의 정수 값이 주어진다. 정수 값의 범위는 -100 ~ 100 사이의 값이다.

## 출력

#과 1번부터인 테스트케이스 번호, 빈칸에 문제에서 설명한 최대값을 출력한다.

단. 10개의 테스트케이스를 수행하여 답을 출력하는 시간은 2초를 넘지 않아야 한다.

입력 예시

```
3                               # 테스트 케이스 수

3 8                             # 1번 테스트 케이스의 N, M

1 3 8 4 -3 -3 5 1

8 -2 5 5 4 -9 -8 -1

4 9 2 -1 0 4 3 2

3 8

4 0 1 -9 9 6 -5 -5

2 10 -8 6 9 -10 -3 0

9 5 -10 10 -9 -6 2 -8

3 8

-10 -5 8 1 -10 -3 -10 4

4 3 9 -6 -4 2 -3 -9

-4 7 10 3 1 5 9 1
```

출력 예시

```
#1 96

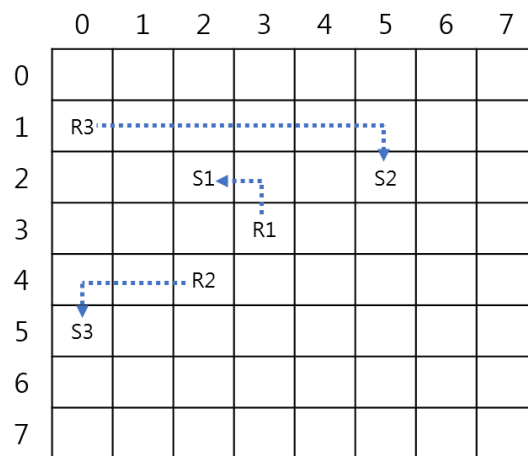
#2 102

#3 114
```

## 문제3] 로봇 과자 먹기

과자를 먹는 로봇이 있다. 지도 상에 과자와 로봇이 각각 N개씩 있다. 하나의 로봇은 한 개의 과자로 이동하여 과자를 먹고 남은 로봇 중 다른 로봇은 남은 과자 중 하나로 이동하여 과자를 먹는다. 즉 로봇 하나는 하나의 과자를 먹는다. 로봇의 이동하는 방법은 상하좌우 직진으로 움직인다. 예를 들어 아래 그림1은 로봇 3기(R1, R2, R3)와 과자 3개(S1, S2, S3)의 위치를 보여 주고 있다. 이때 R1이 S1으로 이동하여 과자를 먹고 R2가 S3로 이동하여 과자를 먹고 R3가 S2로 이동하여 과자를 먹는 총거리를 계산하면 다음과 같다.

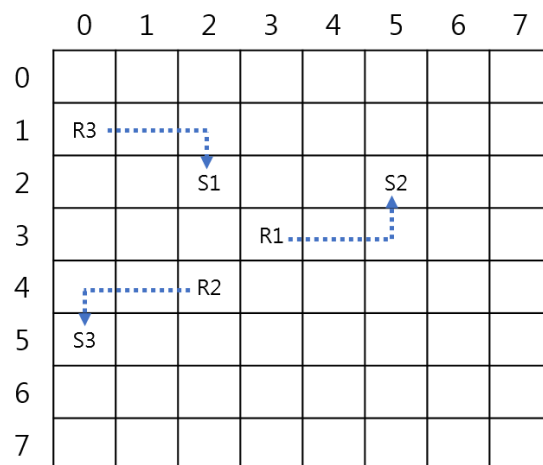
$R1 \rightarrow S1 + R2 \rightarrow S3 + R3 \rightarrow S2 = 2 + 3 + 6 = 11$  이 된다.



[그림1]

또 다른 방법으로 이동하면 총 이동 거리가 달라 질 수 있는데 아래 그림2와 같이 이동할 경우 총 9만큼의 계산 결과가 나오면 이 경우가 가장 짧은 이동거리이다.

$R1 \rightarrow S2 + R2 \rightarrow S3 + R3 \rightarrow S1 = 3 + 3 + 3 = 9$



[그림 2]

N 개의 로봇의 위치가 주어지고 N개의 과자의 위치가 주어 질 때, 서로 다른 로봇이 서로 다른 과자를 하나씩 먹기 위해 이동하는 총거리가 가장 짧은 거리를 계산하는 프로그램을 작성하시오.단 로봇이 먹으려는 과자로 이동하는 경로에 다른 과자나 로봇이 있을 경우 무시하고 지나갈 수 있다고 가정한다.

## 입력

첫 줄에 테스트 케이스 개수 T가 주어진다.  $1 \leq T \leq 10$

다음 줄부터 테스트 케이스의 별로 첫 째 줄에는 로봇과 과자의 개수 N과 ( $3 \leq N \leq 10$ ) 두 번째 줄에는 과자의 좌표 값을 나타내는 정수 쌍이 N개 주어진다. 좌표의 범위는 0이상 100 이하이다. 세번째 줄에는 로봇의 좌표 값을 나타내는 정수 쌍이 N개 주어진다. 좌표의 범위는 0이상 100 이하이다.

## 출력

#과 1번부터인 테스트케이스 번호, 빈칸에 이어 N개의 로봇이 N개의 과자를 먹기 위해 이동한 총 거리 중 가장 총 거리가 최소인 값을 출력한다.

단. 10개의 테스트케이스를 수행하여 답을 출력하는 시간은 10초를 넘지 않아야 한다.

## 입력 예시

```
3                # 테스트 케이스 수

3                # 1번 테스트 케이스의 N
2 2 2 5 5 0      # 1번 테스트 케이스의 과자의 위치 N개
3 3 4 2 1 0      # 1번 테스트 케이스의 로봇의 위치 N개

4

1 1 3 5 3 9 4 1

9 1 11 0 0 4 7 8

4

1 1 3 3 5 5 7 7

2 2 4 4 6 6 8 8
```

출력 예시

#1 9

#2 25

#3 8