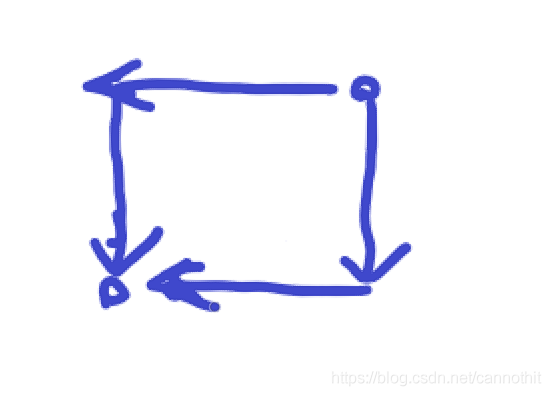


题意：给出A，B两点求两点之间的最短距离，但不能经过F

做法：

第一种情况：

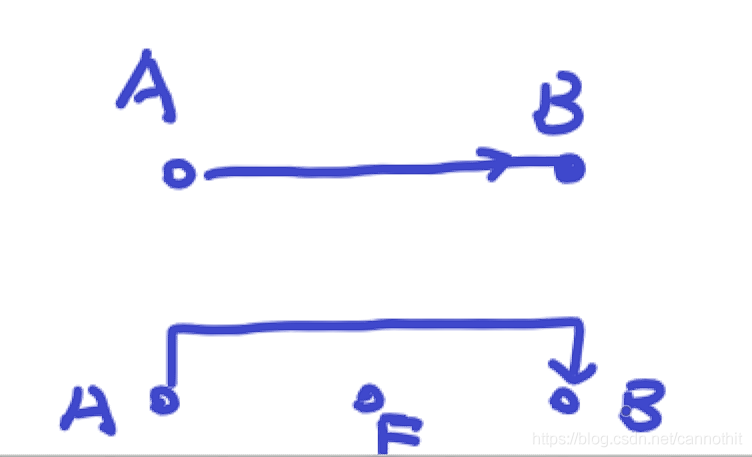
当A,B不在同一条直线上时，不用绕过F，结果就是横纵坐标的之和



第二种情况：（以平行于X轴为例）

1. 当A，B在同一条直线上时，但F不在A，B之间，直线最短
2. 当A，B在同一条直线上时，但F在A，B之间，绕过F点距离加2

平行于y轴亦然



// codeforces.cpp : 此文件包含 "main" 函数。程序执行将在此处开始并结束。

//

#include <iostream>

#include<math.h>

#include<algorithm>

#include<vector>

using namespace std;

struct Pos

{

int x;

int y;

};

int shortest(Pos pos1, Pos pos2) {

return abs(pos1.x - pos2.x) + abs(pos1.y - pos2.y);

}

int main()

{

int n, i = 0;

cin >> n;

while (i < n)

{

cout << endl;

Pos pos1, pos2, pos3;

int x1, x2, x3, y1, y2, y3;

cin >> x1; cin >> y1; cin >> x2; cin >> y2; cin >> x3; cin >> y3;

pos1.x = x1; pos1.y = y1;

pos2.x = x2; pos2.y = y2;

pos3.x = x3; pos3.y = y3;

if (x1 == x2 && x2 == x3) {

if ((y1 < y2 && y3 < y2 && y3 > y1) || y1 > y2 && y3 > y2 && y3 < y1) {

cout << shortest(pos1, pos2) + 2;

}

else {

cout << shortest(pos1, pos2);

}

}

if (y1 == y2 && y2 == y3) {

if ((x1 < x2 && x3 < x2 && x3 > x1) || x1 > x2 && x3 > x2 && x3 < x1) {

cout << shortest(pos1, pos2) + 2 ;

}

else {

cout << shortest(pos1, pos2);

}

}

if ((x3 != x2 || x3 != x1) && (y3 != y2 || y3 != y1)) {

cout << shortest(pos1, pos2);

}

i++;

}

}

#include<iostream>

#include<math.h>

#include <algorithm>

using namespace std;

int main() {

int t;

cin >> t;

int x1, x2, x3, y1, y2, y3;

while (t--) {

cin >> x1 >> y1 >> x2 >> y2 >> x3 >> y3;

int p = abs(x1 - x2) + abs(y1 - y2);

p =

(x1 != x2 && y1 != y2)

|| (y1 == y2) && x3 < min(x1, x2) || (y1 == y2) && x3 > max(x1, x2)

|| (x1 == x2) && y3 < min(y1, y2) || (x1 == x2) && y3 > max(y1, y2)

? p : p + 2;

cout << p << endl;

}

}