Do bờ đắp quanh mỗi thửa ruộng có thể xem như một đường gấp khúc khép kín không tự cắt và đặc biệt ở chỗ các cạnh đều song song với các trục tọa độ và độ dài của M loại dây (M >= 2) được đưa ra theo thứ tự giảm dần nên 2 điểm liên tiếp luôn có hoành độ bằng nhau nếu đoạn thẳng nối 2 điểm là song song với trục tung hoặc tung độ bằng nhau nếu đoạn thẳng nối 2 điểm là song song với trục hoành.

Giả sử 2 điểm liên tiếp là A(x1, y1) và B(x2, y2) thì độ dài AB = sqrt((x2-x1)^2 + (y2 - y1)^2), do x1 = x2 hoặc y1 = y2 nên AB = distance(A,B) = |x2 - x1| + |y2 - y1|.

Giả sử đường gấp khúc khép kín xác định bởi N điểm A1(x1, y1), A2(x2, y2), A3(x3, y3),…, An(xn; yn), thì độ dài đường gấp khúc là D = distance(A1, A2) + distance(A2, A3) + … + distance(An-1, An) + distance(An, A0).

Có M đường gấp khúc => ta tính được tập hợp S = {D1, D2, D3,…, Dn}.

Giả sử tập hợp S = {D1, D2} đã tính thỏa mãn D1 và D2 không là ước của nhau, tính tiếp được D3, duyệt qua các phần tử trong tập hợp, xét D1 ta có 2 trường hợp:

+ Nếu D1 là ước của D3 thì loại D1 khỏi tập hợp lúc này tập hợp còn {D2}, duyệt tiếp D2 tương tự.

+ Nếu D3 là ước của D1 thì dừng duyệt.

Sau khi duyệt xong nếu D3 không là ước của bất kỳ phần tử nào trong tập hợp {D1, D2} thì ta thêm D3 vào tập hợp.

Ví dụ ở đề bài, ban đầu ta có S = {D1} = {36}, tính tiếp được D2 = 18 là ước của D1 dừng duyệt. Tính tiếp D3 = 14 không là ước của 36 => thêm D3 vào có S = {36, 14}. Tính tiếp D4 = 20 không là ước của 36 và 14 => thêm D4 vào có S = {36, 14, 20}. Cuối cùng D5 = 12 là ước của 36 dừng duyệt. Kết quả sắp xếp giảm dần có S = {36, 20, 14}.

\* Việc thêm và loại bỏ phần tử dễ dàng thực hiện với kiểu dữ liệu SET trong STL của C++, hơn nữa SET đã sắp xếp các phần tử mặc định theo thứ tự tăng dần, khi in chỉ cần in ngược lại.