**Robot**

Gọi F[x][y][l] là số lệnh ít nhất để đi đến ô (x,y) mà chỉ dùng các lệnh thuộc l lệnh đầu tiên.

Với giới hạn nhỏ, ta có thể dùng quy hoạch động để giải nhưng bài này(các em có thể cái bằng cách này trước). Tuy nhiên, nhưng bài này có giới hạn lớn hơn nên số trạng thái sẽ là rất lớn, ta không thể lưu hết tất cả các trạng thái có thể mà chỉ lưu nhưng trạng thái có thể đến được từ trạng thái xuất phát( có thể sử dụng map). Để duyệt hết những trạng thái có thể đi đến từ trạng thái xuất phát, ta sử dụng BFS bằng queue 2 đầu (thư viện deque). Từ một trạng thái A, ta duyệt hết tất cả các trạng thái B có thể đến được từ A, nếu B chưa có ở queue, thêm B vào đầu queue nếu F[B] = F[A] và thêm B vào cuối queue nếu F[B] = F[A] + 1;

**POLY**

Sub 1, 2: Nén tọa độ rồi kết hợp segment tree để tính kết quả.

Sub 3: Cắt đa giác thành các hình chữ nhật rồi quy về sub 2. Để cắt thì quét từ trái sang phải, giữa 2 tọa độ x thì quét từ dưới lên, đếm số cạnh đã đi qua, nếu số cạnh lẻ thì đó là cạnh trên của hcn, nếu chẵn thì là cạnh dưới.

Sub 4:

Bước 1: Khôi phục đa giác.

Xét từng dòng điểm, trên cùng 1 dòng, điểm có tọa độ x nhỏ nhất sẽ nói với đỉnh có tọa độ nhỏ thứ hai, đỉnh có tọa độ nhỏ thứ ba nối với đỉnh có tọa độ nhỏ thứ 4,…

Làm tương tự với các cột.

Bước 2: Tính diện tích quay về sub 3

**suduko**

Bài này là một bài thi thử IOI, để làm các subtask cuối thì không có thuật toán chuẩn, bài này ta chỉ có thể sử dụng cặp ghép để làm tham. Vì số đỉnh rất nhiều nên phải ghép tham trước để tránh bị TLE. (Tham khảo tại đây <https://apps.topcoder.com/forums/?module=Thread&threadID=684427>). Code thì có thể xem tại đây <http://ioi2017.org/contest/practice/>