Ladder: Gọi f[i] là vị trí xa nhất sao cho đoạn a[i] -> a[f[i]] là dãy không giảm.

Gọi g[i] là vị trí xa nhất sao cho đoạn a[g[i]] -> a[i] là dãy không giảm.

Tính 2 mảng f, g trong O(N) Với mỗi truy vấn (l,r), đoạn a[l] -> a[r] là dãy thang nếu f[l] >= g[r]. Độ phức tạp O(N).

Tính mảng L,R bằng deque while(a[l[i]] <= a[l[i]+1]) l[i] = l[l[i]+1]; cái R tương tự

Kingpath. BFS + Map

Đầu tiên với mỗi đoạn thẳng, nhét vào Map cái vị trí của nó (hàng cột), Push đỉnh (x,y) vào queue. Với mỗi lần pop ra, ta kiểm tra xem các đỉnh kề với (x,y) theo 8 hướng có trong Map hay ko , nếu có thì đẩy vào queue và bỏ mấy đỉnh đó ra khỏi map. Lặp lại quá trình đến khi đến được (x0,y0) nếu không đến được in ra -1

Digit:1->9 mỗi lần độ dài +1, 10->99 + 2, 100->999 +3, cứ thế mà vét …

Domino Dựng lại đồ thị dựa theo các quân domino đã cho, rồi tiến hành tìm đường đi euler, nếu tồn tại 1 đường đi thì in ra đường đi đấy ko thì in ra -1

Tomau: Đầu tiên giả sử ta có hàm DFS(u,bl,wh,last) với ý nghĩa, u đỉnh đang xét, bl số đỉnh đen đã dc tô, wh số đỉnh trắng đã dc tô last đỉnh cuối dc tô là màu gì. Tại đỉnh u, ta xét nếu u là màu đen thì dfs(v,bl,wh+1,trắng), nếu là đen thì ngược lại, và đặt cận nếu số đỉnh chưa tô + số đỉnh bl <= max đen tìm dc thì loại, bl+white = n thì cập nhập bblack

Dua do. Loại toàn bộ những đoạn có l<r (vì ta tiện đường công tác đi từ 1->m luôn nên skip tụi nó)

Việc còn lại sẽ còn lại những đoạn [l,r] với R>l, ta đảo ngược tụi nó lại, sort tăng dần theo x, nếu x bằng nhau tăng dần theo y, ta lần lượt xét lại những đoạn đó, gọi left , right là vị trí trái nhất và phải nhất mà ta có thể mở rộng dc, nếu mà đoạn left,right giao với đoạn thứ i, ta mở rộng tụi nó ra, tới khi nào ko giao nữa thì ta cộng res = (right-left) \* 2; (đi từ đầu này về đầu kia + ngược lại) . xong hết mọi thứ ta cộng thêm res 1 đoạn = m(do đi từ 1->m) res là đáp án bài toán :D . tư tưỡng tham

Scale: Nếu giới hạn 10^13 thì chỉ việc sinh nhị phân ... còn 10^100 em chưa nghĩ

Thu âm: Sparse table min,max :<

Xổ số: Gọi F(a) là số lượng đường ngang của các cột đồ thị có khả năng giao với một trục dọc có tọa độ x=a

Một cột đồ thị có các trục dọc x=A và x=B sẽ tạo ra F(A) + F(B) điểm trúng thưởng

Đoạn nằm ngang (A+1, B-1) bây giờ sẽ có khả năng phát sinh thêm điểm trúng thưởng, nên ta tăng f(x) =f(x)+1, với mọi x trong đoạn [A+1, B-1]

cập nhập F(x) theo từng đoạn

mà đoạn đó có độ dài 256

hay 255 gì quên r

N\*(256\*2+N/256)

sẽ tốn