

SOLUTION 08/12 chiều

Bài 1: Chuỗi may mắn

Với subtask 1 đơn giản chỉ cần sử dụng kiểu string của C++ với các phép xóa. Chú ý là khi xóa thì lặp từ cuối xâu về đầu xâu.

Với subtask 2:

Giả sử xâu đã cho có dạng \dots . Khi đó với mỗi vị trí i ta tính $T[i]$ là lượt xóa mà trong lượt này ký tự b_i xóa. Điều này có thể làm bằng cách sử dụng một ngăn xếp:

ST là ngăn xếp.

S xâu ký tự may mắn

FOR $i: 1 \rightarrow \text{length of } S$

IF ($S[i] == '6'$)

ST.push(i)

ELSEIF ($S[i] == '8'$ and ST is not empty):

$U = \text{st} \rightarrow \text{top};$

St.pop

$V = i;$

//U and V is a matched pair

$T = T[V] = \max(T^*, U < i < V) (*)$

ENDIF

ENDFOR

Nếu ký tự không bị xóa thì để giá trị T đủ lớn ($=n$)

Trong đoạn đánh dấu (*) ở trên ta cần tìm giá trị lớn nhất của một đoạn trong mảng T . Điều này có thể được thực hiện bằng cách sử dụng IT khi đó thời gian thực hiện là $O(n \log n)$ (nếu khéo hơn có thể dùng sttack - $O(n)$).

Sau khi đã có mảng T thì dễ thấy với vị trí i nếu nó bị xóa ($T[i] < n$) thì điểm nó được cộng thêm vào kết quả là số lượng giá trị T từ 1 đến i có giá trị lớn hơn hoặc bằng $T[i]$. Ta có thể sử dụng một BIT tổng để làm điều này.

Bài 2: Phòng dịch

Ta xét cạnh giữa 2 đỉnh x và y , đặt cho nó trọng số là $GCD(a_x, a_y)$. Vậy khi đó, nếu ta xét một đồ thị với toàn bộ các cạnh là bội của i , thì nó sẽ tạo thành một rừng cây con. Và số cặp đường đi của đồ thị sẽ chính bằng tổng số cặp đường đi ở các cây con, và bằng số cặp có thể chọn ở các cây con. Đó chính là giá trị $f(i)$ ta cần tìm. Vậy với từng i , ta sẽ dựng một đồ thị với các cạnh có trọng số là bội của i đã được lưu từ trước. Để ý nếu ta DFS thử tất cả các điểm như ta thường làm, độ phức tạp của mỗi lần sẽ lên tới $O(n)$ và hoàn toàn vượt quá giới hạn. Do đó, ta sẽ chỉ DFS từ từng đầu mút của cạnh một trong các cạnh đã cho. Ta để ý số có nhiều ước số nhất trong khoảng 10^5 có 128 ước số, vậy mỗi cạnh sẽ xuất hiện trong tối đa 128 thao tác dựng đồ thị, và đảm bảo thời gian chạy cho chúng ta ở mức an toàn.

Bài 3: Hệ thống đèn

Gọi $a[i]$ là số lần tác động lên những bóng đèn ở hàng dọc thứ i ;

Gọi $b[i]$ là số lần tác động lên những bóng đèn ở hàng ngang thứ i ;

Với mỗi lần tác động, ta cập nhật giá trị mảng a (hay b);

Duyệt tất cả các bóng đèn, với mỗi bóng đèn, gọi $p[i][j]$ là trạng thái ban đầu của bóng đèn hàng i cột j , $a[j] + b[i]$ số lần tác động thêm vào bóng đèn hàng i cột j ,

nếu $p[i][j] + a[j] + b[i]$ chia hết cho 3 thì tức là bóng đèn đó đang ở trạng thái tắt ,
khi đó kết quả cộng lên 1 ;
Độ phức tạp $O (m*n + q)$

-----Hết-----