

Lời giải tham khảo ACM - ICPC vòng loại miền Trung của team unsigned, Đại học công nghệ, đại học quốc gia Hà Nội.

Bài A:

Tính $row[i][j]$ = số ô có màu j trên hàng i

$col[i][j]$ = số ô có màu j trên cột i

Đáp án bằng = Tổng $row[i][k] * col[j][k]$ với mọi (i,j) sao cho $A[i,j]$ khác 0 và $k = 3 - A[i,j]$.

Bài C:

Team mình nghĩ đây là một bài khá đơn giản, chỉ cần có kiến thức về Persistent segment tree là có thể giải được. <http://vnoi.info/wiki/algo/data-structures/persistent-data-structures>. Mỗi lần gặp truy vấn loại 1 hoặc 2 thì chỉ cần tạo một phiên bản mới rồi update trên phiên bản ấy, còn gặp truy vấn loại 3 thì gán phiên bản hiện tại bằng phiên bản thứ k .

Đối với những bạn không biết Persistent IT thì vẫn có thể giải bài này bằng cách xử lý offline. Ta sẽ xây một cây truy vấn, mỗi nút trên cây là 1 truy vấn.

+ Với mỗi truy vấn i , nếu là loại 1 thì thêm một cạnh nối từ $i - 1$ đến i , và ở đây khi đi từ nút $i - 1$ đến nút i thì phải thay đổi nút v có giá trị mới là y .

+ Với truy vấn loại 2, thì thêm một cạnh từ $i - 1$ đến i và cạnh này không phải làm gì cả.

+ Với truy vấn loại 3, thì thêm một cạnh từ k đến i .

Thực hiện dfs, đi vào một cạnh thì thực hiện thao tác gán nếu cạnh đó loại 1, đi ra khỏi một cạnh thì gán lại giá trị cũ nếu cạnh đó là loại 1. Như vậy ở mỗi nút ta sẽ có thông tin của mảng khi thực hiện tất cả các cạnh từ nút đó đến gốc.

Bài D:

Là một bài siêu dễ, ta chỉ cần kiểm tra 2 đường đi liên tiếp có phải rẽ phải hay không.

Bài E:

Biến đổi bài toán.

Nếu A và B có một cạnh yêu thì ta sẽ cho A và B ở cùng bên trong đồ thị 2 phía, còn ngược lại A và B sẽ phải khác bên trong đồ thị 2 phía.

Đảm bảo thỏa mãn 3 điều kiện của đề bài.

Vậy một cạnh sẽ bị loại bỏ nếu sau khi thêm vào thì nó sẽ tạo ra một chu trình lẻ, hay làm mất tính đồ thị 2 phía của đồ thị.

Bài này ta chỉ cần dùng disjoint set để nối các cạnh vào, đồng thời tô màu cho các đỉnh khi gộp 2 thành phần liên thông. Nếu thêm một cạnh mà nó đã được liên thông thì xem màu của 2 đỉnh ấy có tương ứng với cạnh yêu hay ghét không. Nếu không thì cạnh ấy sẽ bị loại bỏ.

Bài F:

Bài này để AC nhanh thì team mình đã dùng python sinh ra đáp án rồi làm thành mảng hằng, vì mảng chỉ có 25 phần tử.

Bài I:

Bài này chỉ cần xét các vị trí có cùng modul với 3. Và dùng prefix Sum để giải bài toán tìm đoạn con có tổng lớn nhất với mỗi loại vị trí mod 3 dư 0, 1, 2.

Bài G:

Nếu coi mỗi cột là một đỉnh ở bên trái, mỗi hàng là một đỉnh ở bên phải trong một đồ thị 2 phía. Nếu ô (i,j) có hoa thì có một cạnh nối từ i -> j. Thì bài toán trở thành bài tìm tập phủ đỉnh cực tiểu của một đồ thị hai phía, được giải dễ dàng bằng thuật toán tìm cặp ghép cực đại. Để hiểu rõ hơn thuật toán các bạn có thể lên gg search Tập phủ đỉnh cực tiểu.

Bài H:

Bài này ta được cho 3 dãy A[], B[], C[], đếm số cặp chỉ số (i,j) sao cho $A[i] > A[j]$ & $B[i] > B[j]$ & $C[i] > C[j]$.

Xét các bộ quan hệ (<, <, >)..

Gọi $C(?, ?, ?)$ là số cặp thoả mãn bộ quan hệ (?, ?, ?).

Ta tính số cặp thoả mãn bộ quan hệ (<, ?, >), (<, >, ?), (?, <, >) trong đó chiều có dấu ? là chiều mà chúng ta không cần quan tâm.

Gọi S là tổng số cặp thoả mãn 3 bộ quan hệ trên.

Có $S = 2 * (C(<, <, >) + C(<, >, <) + C(<, >, >))$.

Có $C(<, ?, ?) = C(<, <, <) + C(<, <, >) + C(<, >, <) + C(<, >, >) = S / 2$.

Vậy ta có thể dễ dàng tính được $C(<, <, <)$.

Bài J:

Bài này chúng ta sẽ giải đệ quy.

Gọi $F(n,m)$ là kí tự thứ n của xâu có m kí tự được xây theo quy tắc của đề bài (m là một lũy thừa của 2 và $m > n$).

Ta có công thức truy hồi như sau:

+ Nếu $n = 1$ thì $F(n,m) = 0$.

+ Nếu $n \leq m / 2$ thì $F(n,m) = F(n, m / 2)$;

còn không thì $F(n,m) = 1 - F(n - m / 2, m / 2)$;

Gọi M là số lũy thừa của 2 bé nhất mà lớn hơn bằng n.

Đáp án của bài sẽ là $F(n,M)$.

Bài K:

Bài này đơn giản chỉ là giải phương trình

$$+ 2x + 2y = N$$

$$+ 2x + 4y = M.$$

Khi đã biết N và M. Cách giải phương trình này mình nhớ mình đã được học từ hồi lớp 3, với bài toán vừa gà vừa chó 36 con và 100 chân chẵn. Nên mình khá chắc các bạn lớp 3 sẽ giải tốt bài này.

Bài L:

Mỗi xâu Si thì ta sẽ thử xem Si có thể là xâu có thứ tự từ điển tốt nhất nếu có thể hoán vị bằng chữ cái đi một cách hợp lý không.

Đi lần lượt từ đầu đến cuối xâu Si:

+ Gọi c là chữ cái đầu tiên của Si, để Si tốt nhất thì c phải có vị trí bé hơn tất cả các kí tự đầu tiên của các xâu khác.

+ Trong những xâu có kí tự đầu tiên bằng với c, ta lại thêm những điều kiện so sánh giữa các kí tự trong bảng chữ cái. Với mỗi điều kiện kí tự X cần đứng trước kí tự Y ở trong bảng chữ cái sau khi hoán vị lại đi, ta thêm một cạnh từ X -> Y.

+ Cuối cùng kiểm tra xem đồ thị gồm 26 đỉnh từ 'A' đến 'Z' mà ta vừa thêm cạnh có chu trình hay không.

Để thực hiện thuật toán trên hiệu quả, bọn mình đã dùng Trie để đi từ đầu đến cuối một xâu rồi thêm cách điều kiện một cách hiệu quả.