

Họ tên: Võ Minh Quân
MSSV: 21520093
Lớp: KHTN2021
Tài khoản Codeforces: [omlgg](#)

BÁO CÁO ĐỒ ÁN CUỐI KỲ IT003

Chủ đề: contest codeforces div.2

1. [Codeforces Round #789 \(Div. 2\)](#)(4245 điểm)

1.1 [Problem A:](#)

1.1.1 Tóm tắt đề bài:

Cho một mảng a ($0 \leq a_i \leq 100$) gồm n phần tử ($2 \leq n \leq 100$), với mỗi bước thực hiện, người dùng chọn hai số a_i và a_j (i khác j , $1 \leq i, j \leq n$):

- Nếu $a_i = a_j$, đổi một trong hai số thành 0
- Gán cả hai số bằng $\min(a_i, a_j)$

Tìm số bước ít nhất để tất cả phần tử của mảng đều bằng 0.

1.1.2 Giải thuật:

Ta chia thành các trường hợp:

- TH1. Trong mảng có ít nhất một phần tử bằng 0: lần lượt chọn cặp giữa 0 và phần tử khác 0 trong mảng \Rightarrow số bước thực hiện bằng số phần tử khác 0.
- TH2. Trong mảng có ít nhất hai phần tử bằng nhau và không có phần tử bằng 0: chọn cặp hai phần tử bằng nhau ta được phần tử 0 trong mảng và quay lại trường hợp 1 \Rightarrow số bước thực hiện bằng n
- TH3. Các phần tử trong mảng khác nhau đôi một và không có phần tử bằng 0: chọn cặp hai phần tử khác nhau, khiến chúng bằng nhau, ta mất một bước và quay lại trường hợp 2 \Rightarrow số bước thực hiện bằng $n+1$

1.1.3 Source code:

<https://github.com/omlgg/Codeforces-Round-789-Div.2-/blob/main/A.cpp>

1.2 [Problem B1:](#)

1.2.1 Tóm tắt đề bài:

Cho chuỗi nhị phân s có độ dài n (n chẵn, $2 \leq n \leq 2e5$). Chuỗi s được gọi là tốt nếu chia s thành ít nhất các chuỗi con liên tiếp sao cho mỗi chuỗi con có độ dài chẵn và tất cả các bit của một chuỗi con đó bằng nhau.

Người ta biến s thành chuỗi tốt bằng cách thay đổi giá trị một số bit trong s . Mỗi bước có thể đổi giá trị của s_i thành '0' hoặc '1'. Tìm số bước ít nhất để biến chuỗi s thành chuỗi tốt.

1.2.2 Giải thuật:

Gọi $f(i, j)$ ($1 \leq i \leq n$, $0 \leq j \leq 1$) là số bước ít nhất cần thực hiện để chuỗi con $s(1, i-i\%2)$ trở thành chuỗi tốt.

Gọi $\text{diff}(x,y)$ là hàm trả về sự khác nhau giữa 2 kí tự x và y , nếu x khác y thì $\text{diff}(x,y)$ trả về 1 và ngược lại.

Ta có: $f(i,j) = f(i-1,j) + \text{diff}(s[i],j)$ với i chẵn

$$f(i,j) = \min(f(i-1,j), f(i-1,1-j)) + \text{diff}(s[i],j) \text{ với } i \text{ lẻ}$$

1.2.3 Source code:

<https://github.com/omlgg/Codeforces-Round-789-Div.2-/blob/main/B1.cpp>

1.3 [Problem B2:](#)

1.3.1 Tóm tắt đề bài:

Cho chuỗi nhị phân s có độ dài n (n chẵn, $2 \leq n \leq 2e5$). Chuỗi s được gọi là tốt nếu chia s thành ít nhất các chuỗi con liên tiếp sao cho mỗi chuỗi con có độ dài chẵn và tất cả các bit của một chuỗi con đó bằng nhau.

Người ta biến s thành chuỗi tốt bằng cách thay đổi giá trị một số bit trong s . Mỗi bước có thể đổi giá trị của s_i thành '0' hoặc '1'. Tìm số bước ít nhất để biến chuỗi s thành chuỗi tốt và số chuỗi con nhỏ nhất mà s có thể chia (chuỗi con liên tiếp, mỗi chuỗi con có độ dài chẵn, các bit trong cùng một chuỗi con giống nhau).

1.3.2 Giải thuật:

Gọi $f(i,j)$ ($1 \leq i \leq n$, $0 \leq j \leq 1$) là pair{số bước ít nhất cần thực hiện để chuỗi con $s(1,i-i\%2)$ trở thành chuỗi tốt, số chuỗi con ít nhất}

Gọi $\text{diff}(x,y)$ là hàm trả về sự khác nhau giữa 2 kí tự x và y , nếu x khác y thì $\text{diff}(x,y)$ trả về 1 và ngược lại.

Ta có: $f(i,j) = \{f(i-1,j).first + \text{diff}(s[i],j), f(i-1,j).second\}$ với i chẵn

$$f(i,j) = \min(\{f(i-1,j).first + \text{diff}(s[i],j), f(i-1,j).second\}, \{f(i-1,1-j).first + \text{diff}(s[i],j), f(i-1,1-j).second + 1\})$$

với i lẻ

1.3.3 Source code:

<https://github.com/omlgg/Codeforces-Round-789-Div.2-/blob/main/B2.cpp>

1.4 [Problem C:](#)

1.4.1 Tóm tắt đề bài:

Cho một dãy hoán vị p có độ dài n ($1 \leq n \leq 5000$, $1 \leq p_i \leq n$). Đếm số nhóm 4 phần tử $[a,b,c,d]$ ($1 \leq a < b < c < d \leq n$) sao cho $p_a < p_c$ và $p_b > p_d$.

1.4.2 Giải thuật:

Ta dựng mảng $\text{pre}[i][x]$ để đếm số vị trí $j < i$ sao cho $p_j < p_i$, mảng $\text{suf}[i][x]$ để đếm số vị trí $j > i$ sao cho $p_j < p_i$. Từ đó, ta dễ dàng tính được kết quả bằng cách duyệt vị trí b và c , đếm được số cách chọn a bằng $\text{pre}[b][p[c]]$ và số cách chọn d bằng $\text{suf}[c][p[b]]$. Số cách chọn 4 vị trí a, b, c, d bằng:

$$\sum_{b=2}^{n-2} \sum_{c=b+1}^{n-1} \text{pre}[b][p[c]] * \text{suf}[c][p[b]]$$

1.4.3 Source code:

<https://github.com/omlgg/Codeforces-Round-789-Div.2-/blob/main/C.cpp>

1.5 [Problem D:](#)

1.5.1 Tóm tắt đề bài:

Cho một bảng gồm n hàng và m cột ($1 \leq n, m \leq 1e6$; $1 \leq n*m \leq 1e6$). Có đúng n học sinh tham dự, học sinh được đánh số từ 1 đến $n*m$. Tại thời điểm i , học sinh thứ i tham gia vào bảng và ngồi vào ô $(1,1)$ của bảng, các học sinh còn lại chuyển sang ghế bên phải, nếu học sinh ngồi ở cột ngoài cùng bên phải thì chuyển sang ghế đầu tiên bên trái của hàng tiếp theo.

Chuỗi s có độ dài $n*m$ thể hiện tính cách của $n*m$ học sinh. $s_i = '0'$ nếu học sinh thứ i nghịch ngợm và $s_i = '1'$ nếu học sinh thứ i nghiêm túc. Một hàng hoặc cột được cho là tốt nếu có ít nhất một học sinh nghiêm túc.

Với mỗi thời điểm i , tính tổng số hàng tốt và cột tốt trong bảng.

1.5.2 Giải thuật:

Ta thấy các học sinh có $i\%m$ bằng nhau thì ở cùng cột. Với mỗi i mà học sinh thứ i nghiêm túc, ta xét $i\%m$ đã có học sinh nghiêm túc hay chưa, nếu chưa thì số cột tốt từ thời điểm i đến $n*m$ tăng lên 1

Còn đối với hàng, ta xét lần lượt từng nhóm liên tiếp có độ dài m và các nhóm độ dài bé hơn m bắt đầu từ 1, ta biết rằng các nhóm đó được lặp lại mỗi m bước nên tăng số hàng tốt tại thời điểm $t, t+m, t+2m, \dots$ nếu nhóm đó có học sinh nghiêm túc.

1.5.3 Source code:

<https://github.com/omlgg/Codeforces-Round-789-Div.2-/blob/main/D.cpp>

2. [CodeCraft-22 and Codeforces Round #795 \(Div. 2\)](#) (2986 điểm)

2.1 [Problem A:](#)

2.1.1 Tóm tắt đề bài:

Cho dãy số nguyên a gồm n phần tử ($3 \leq n \leq 1e5$; $1 \leq a_i \leq 1e9$), tìm số phần tử nhỏ nhất cần xóa sao cho tổng hai phần tử kề nhau bất kì trong dãy là chẵn.

2.1.2 Giải thuật:

Để tổng hai phần tử kề nhau là chẵn thì cả hai số cùng chẵn hoặc cùng lẻ, vậy để thỏa điều kiện đề bài ta xóa sao cho tất cả các phần tử còn lại cùng chẵn hoặc cùng lẻ. Từ đó, ta chỉ cần đếm số phần tử chẵn và số phần tử lẻ trong mảng. Kết quả là $\min(\text{số phần tử chẵn}, \text{số phần tử lẻ})$.

2.1.3 Source code:

<https://github.com/omlgg/CodeCraft-22-Codeforces-Round-795-Div.2/blob/main/A.cpp>

2.2 [Problem B:](#)

2.2.1 Tóm tắt đề bài:

Cho dãy số nguyên s không giảm gồm n phần tử ($1 \leq n \leq 1e5$, $1 \leq s_i \leq 1e9$) với s_i là kích cỡ giày của học sinh thứ i . Tìm một hoán vị để đổi giày sao cho mọi học sinh đều mang giày của học sinh khác và kích cỡ giày đó lớn hơn hoặc bằng kích cỡ giày ban đầu.

2.2.2 Giải thuật:

Ta thấy rằng nếu cho một học sinh mang giày kích cỡ lớn hơn kích cỡ ban đầu, vậy cũng sẽ có học sinh phải mang giày có kích cỡ nhỏ hơn ban đầu, điều đó không hợp lệ. Suy ra rằng các học sinh có cùng kích cỡ giày chỉ trao đổi với nhau. Từ đó, kết hợp với điều kiện học sinh phải mang giày của học sinh khác ta có yêu cầu rằng mọi học sinh phải có bạn học có cùng kích cỡ giày. Ta xét từng kích cỡ giày, đổi giày của các bạn học sinh có kích cỡ đó bằng cách: cho bạn thứ hai mang giày bạn thứ nhất, bạn thứ ba mang giày bạn thứ 2, ... bạn k mang giày bạn k-1 và bạn 1 mang giày của k.

2.2.3 Source code:

<https://github.com/omlgg/CodeCraft-22-Codeforces-Round-795-Div.2/blob/main/B.cpp>

2.3 Problem C:

2.3.1 Tóm tắt đề bài:

Cho xâu nhị phân s có độ dài n ($2 \leq n \leq 1e5$), gọi d_i là biểu diễn thập phân của $s_i s_{i+1}$, $f(s)$ là tổng các d_i hợp lệ. Tại một bước ta có thể đổi giá trị của hai phần tử kề nhau, tìm giá trị nhỏ nhất của $f(s)$ sau nhiều nhất k bước ($0 \leq k \leq 1e9$).

2.3.2 Giải thuật:

Gọi $\text{sum}(i, j)$ là tổng các số từ i đến j , ta có $f(s) = \text{sum}(1, n-1) * 10 + \text{sum}(2, n)$, ta thấy rằng trọng số của chữ số đầu tiên là 10, chữ số cuối cùng là 1 và các chữ số còn lại là 11. Vì vậy, ta sẽ ưu tiên đưa số 1 về vị trí có trọng số 1 trước, rồi đến trọng số 10 và cuối cùng là trọng số 11 cũng như ưu tiên đưa chữ số 0 về vị trí có trọng số 11. Đầu tiên ta xét nếu vị trí cuối cùng là chữ số 0, ta sẽ tìm chữ số 1 gần nhất để đổi vị trí với nó. Nếu đủ số bước đổi, lần đổi này sẽ giảm $f(s)$ ban đầu đi 9 nếu vị trí của số 1 nằm ở đầu dãy hoặc 10 trong trường hợp ngược lại. Tiếp đến ta xét nếu chữ số đầu bằng 0, ta tìm chữ số 1 gần nhất trong đoạn $[2, n-1]$. Nếu đủ bước đổi, phép đổi này sẽ giảm $f(s)$ thêm 1.

2.3.3 Source code:

<https://github.com/omlgg/CodeCraft-22-Codeforces-Round-795-Div.2/blob/main/C.cpp>

2.4 Problem D:

2.4.1 Tóm tắt đề bài:

Cho dãy số nguyên a có độ dài n ($1 \leq n \leq 2e5$; $-1e9 \leq a_i \leq 1e9$), yêu cầu kiểm tra điều kiện:

$$\text{Max}(a_i, a_{i+1}, \dots, a_{j-1}, a_j) \geq a_i + a_{i+1} + \dots + a_{j-1} + a_j \text{ Với mọi cặp } (i, j) \text{ mà } 1 \leq i \leq j \leq n.$$

2.4.2 Giải thuật:

Gọi $L[i]$ là vị trí bên trái gần nhất mà $a[L[i]] > a[i]$, $R[i]$ là vị trí bên phải gần nhất mà $a[R[i]] > a[i]$.

Ta có thể xây dựng L và R bằng stack.

Với mỗi a_i , ta xét đoạn con $[j, k]$ mà $a_i = \max(a_j, a_{j+1}, \dots, a_{k-1}, a_k)$ tức là đoạn $[L[i]+1, R[i]-1]$. Ta quản lý bằng cây phân đoạn để tìm tổng đoạn con lớn nhất thuộc đoạn $[L[i]+1, R[i]-1]$, so sánh tổng này với $a[i]$, nếu tổng lớn hơn $a[i]$ vậy suy ra điều kiện đề bài cho không hợp lệ.

2.4.3 Source code:

<https://github.com/omlgg/CodeCraft-22-Codeforces-Round-795-Div.2/blob/main/D.cpp>