

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA KHOA HỌC MÁY TÍNH



BÁO CÁO ĐỒ ÁN CUỐI KÌ
MÔN: CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT
CHỦ ĐỀ: HAI KÌ THI CODEFORCES (DIV. 2)

Giảng viên: **Nguyễn Thanh Sơn**
Sinh viên thực hiện: Hà Văn Hoàng
MSSV: 21520033
Lớp sinh hoạt: KHTN2021
Lớp học: IT003.M21.KHTN2021

Thành phố Hồ Chí Minh, ngày 01 tháng 06 năm 2022

MỤC LỤC

MỤC LỤC	2
LỜI MỞ ĐẦU	3
PHẦN I. KÌ THI THỨ NHẤT	4
1. Mở đầu	4
2. Về bài A. Prof. Slim	4
3. Về bài B. Dorms War	4
4. Về bài C. Where is the Pizza?	5
5. Về bài D. Very Suspicious	5
6. Minh chứng	6
PHẦN II. KÌ THI THỨ HAI	8
1. Mở đầu	8
2. Về bài A. Tokitsukaze and All Zero Sequence	8
3. Về bài B1. Tokitsukaze and Good 01-String (easy version)	8
4. Về bài C. Tokitsukaze and Strange Inequality	9
5. Về bài D. Tokitsukaze and Strange Inequality	9
6. Minh chứng	10

LỜI MỞ ĐẦU

Dưới đây là các phân tóm tắt đề, lời giải, bài giải (code) của những bài đã làm được, minh chứng trong hai kì thi có thời gian diễn ra trong khoảng thời gian cho phép thực hiện đồ án với chủ đề hai kì thi Codeforces (Div. 2).

Kì thi thứ nhất là kì thi Codeforces Round #788 (Div. 2), diễn ra vào lúc 21 giờ 35 phút ngày 06 tháng 05 năm 2022.

Kì thi thứ hai là kì thi Codeforces Round #789 (Div. 2), diễn ra vào lúc 21 giờ 35 phút ngày 08 tháng 05 năm 2022.

Tài khoản để dùng cho kỳ thi là tài khoản: [hhoangcpascal](#)

Lưu ý: Tất cả các code nằm trong đường link:

<https://github.com/hhoangcpascal/Codeforces> và cách viết code, cách gọi biến trong code có thể khác so với cách gọi trong lời giải, vì cách gọi trong lời giải giúp dễ hiểu hơn, nhưng về mặt tư tưởng giải thuật thì giống nhau.

PHẦN I. KÌ THI THỨ NHẤT

1. Mở đầu:

Kì thi thứ nhất là kì thi [Codeforces Round #788 \(Div. 2\)](#), với những bài giải được là:

- A. Prof. Slim
- B. Dorms War
- C. Where is the Pizza?
- D. Very Suspicious

Số điểm đạt được: 4331

2. Về bài A. Prof. Slim

2.1. Tóm tắt đề

Cho một dãy số có n số nguyên khác 0 a_1, a_2, \dots, a_n . Hãy đưa về dãy đã được sắp xếp không giảm với không hoặc một hoặc một và thao tác như sau:

- Chọn hai vị trí i và j ($1 \leq i < j \leq n$) sao cho hai số a_i, a_j khác dấu.
- Hoán đổi dấu của hai số a_i và a_j . Ví dụ với $a_i = 3, a_j = -2$ thì sau khi hoán đổi thì $a_i = -3, a_j = 2$.

Nhiệm vụ đề ra là cho biết với các thao tác trên thì có thể đưa về dãy được sắp xếp không giảm được không?

2.2. Lời giải, bài giải

Thuật toán: Duyệt, hai con trỏ

Với bài này, sử dụng hai con trỏ i, j , i duyệt từ 1 tới n , j duyệt ngược từ n và 1.

Với i sao cho a_i dương, thì ta sẽ chạy j cho tới khi $j \leq i$ hoặc a_j âm. Với $j \leq i$ thì ta thoát do mục đích là đưa dấu âm hết về bên trái dấu dương, còn với a_j dương thì ra sẽ sắp lại. Sau khi thực hiện xong, ta sẽ kiểm tra xem dãy có không giảm hay không.

Về phần code thì có thể xem tại:

[https://github.com/hhoangcpascal/Codeforces/blob/main/Round%20788%20\(Div.%202\)/A.cpp](https://github.com/hhoangcpascal/Codeforces/blob/main/Round%20788%20(Div.%202)/A.cpp)

3. Về bài B. Dorms War

3.1. Tóm tắt đề

Cho một chuỗi s có độ dài n chỉ gồm các kí tự in thường. Cho k kí tự đặc biệt

c_1, c_2, \dots, c_k .

Có một chương trình. Chương trình này mỗi lần chạy sẽ thực hiện các bước sau:

- Liệt kê tất cả các vị trí i sao cho kí tự $s[i + 1]$ là một kí tự đặc biệt.
- Xóa hết tất cả các kí tự ở các vị trí i đã liệt kê ngay cả khi $s[i]$ là một ký tự đặc biệt. Trong trường hợp không có vị trí i nào chương trình sẽ phát ra âm thanh rất khó chịu.
- Những kí tự chưa xóa sẽ được dồn lại và không thay đổi thứ tự.

Ví dụ cho chuỗi “abdec” và có hai kí tự đặc biệt ‘d’ và ‘c’. Sau khi chạy lần một, thì ta còn chuỗi “adc”, sau khi chạy lần hai thì còn chuỗi “c”. Nếu chạy lần ba thì chương trình sẽ phát ra tiếng kêu khó chịu.

Nhiệm vụ đề ra là tìm số lần chạy tối đa của chương trình sao cho trong suốt quá trình chạy chương trình không phát ra tiếng kêu khó chịu.

3.2. Lời giải, bài giải

Thuật toán: Duyệt hoặc quy hoạch động.

Với bài này, ta có thể sử dụng quy hoạch động như sau: Gọi $dp[i]$ là số lần chạy tối đa để ký tự $s[i]$ bị xóa. Công thức như sau: $dp[i] = 1$ nếu $s[i + 1]$ là ký tự đặc biệt, còn không thì $dp[i] = dp[i + 1] + 1$. Với xâu s bắt đầu từ vị trí 0, ta có thể khởi tạo $dp[n - 1] = -n$ để phòng trường hợp những ký tự không bao giờ xóa có $dp \geq 0$.

Kết quả là $\max(0, \text{all}(dp))$.

Về phần code, có thể xem tại:

[https://github.com/hhoangcpascal/Codeforces/blob/main/Round%20788%20\(Div.%202\)/B.cpp](https://github.com/hhoangcpascal/Codeforces/blob/main/Round%20788%20(Div.%202)/B.cpp)

4. Về bài C. Where is the Pizza?

4.1. Tóm tắt đề

Cho ba dãy a, b, d có độ dài n . Hai dãy a, b đều là các hoán vị từ 1 tới n và d_i bằng một trong ba số: 0, a_i, b_i .

Cho dãy c thỏa mãn các điều kiện sau:

- Dãy c là một hoán vị từ 1 tới n .
- Khi $d_i = 0$, c_i có thể mang giá trị tùy ý. Ngược lại, $c_i = d_i$.

Nhiệm vụ đề ra là tìm số lượng dãy c thỏa mãn.

4.2. Lời giải, bài giải

Thuật toán: DFS, BFS, đồ thị.

Với bài toán này, trước tiên ta hãy coi các số từ 1 tới n là các đỉnh, a_i có cạnh với b_i . Ta sẽ duyệt nếu $d_i > 0$ thì sẽ có bao nhiêu đỉnh liên thông với đỉnh d_i , từ đó loại tất cả đỉnh đó ra vì những đỉnh đó sẽ cố định trong dãy c . Sau khi loại xong, ta sẽ đếm số thành phần liên thông với những đỉnh chưa bị loại. Với mỗi thành phần như vậy sẽ có 2 cách điền, nên kết quả bài toán sẽ là:

$2^{\text{số thành phần liên thông với các đỉnh chưa bị loại}}$.

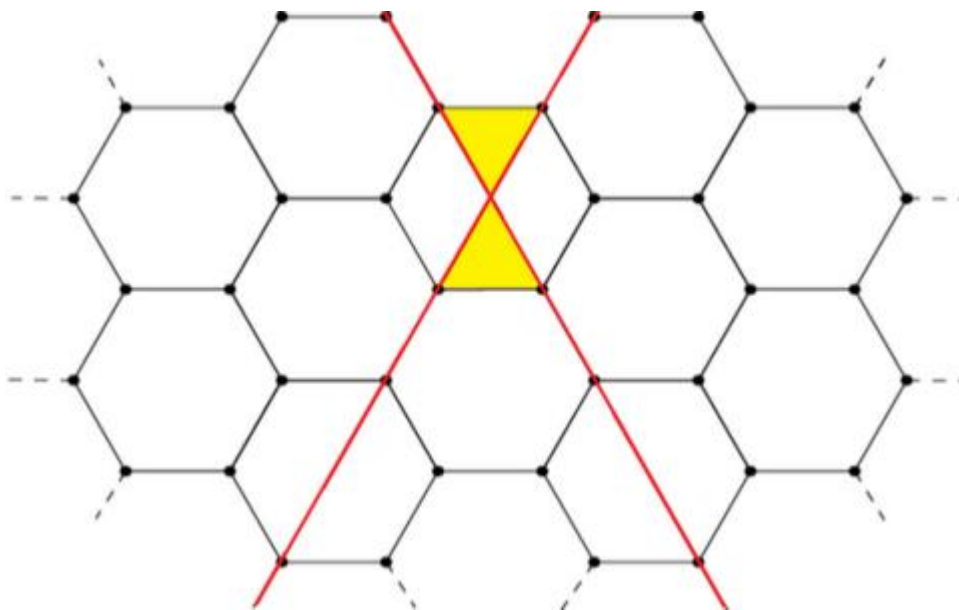
Về phần code, có thể đọc tại:

[https://github.com/hhoangcpascal/Codeforces/blob/main/Round%20788%20\(Div.%202\)/C.cpp](https://github.com/hhoangcpascal/Codeforces/blob/main/Round%20788%20(Div.%202)/C.cpp)

5. Về bài D. Very Suspicious

5.1. Tóm tắt đề

Cho một cái lưới lục giác vô hạn. Ta có thể vẽ lên lưới các đường thẳng để tạo ra các tam giác đều đặc biệt sao cho không có đường thẳng nào hoặc cạnh lục giác nào cắt qua các tam giác đều đặc biệt này, và các đường thẳng phải song song với 1 trong các cạnh của lục giác đều. Ví dụ ta có thể vẽ như sau:



Cho t câu hỏi, mỗi câu hỏi cho một số n . Nhiệm vụ đề ra là tìm số lượng ít nhất các đường thẳng sao cho khi vẽ sẽ tạo ít nhất n tam giác đều.

5.2. Lời giải, bài giải

Thuật toán: Toán

Với bài này, nhận thấy rằng số cặp đường thẳng giao nhau tại tâm của lục giác đều nhân đôi sẽ bằng số tam giác đều đặc biệt được tạo ra. Và có ba góc độ để vẽ các đường thẳng. Nếu muốn tạo nhiều tam giác đều nhất, phải vẽ các đường thẳng sao cho số cặp đường thẳng giao nhau tại tâm của lục giác đều nhiều nhất. Từ đó ta suy ra phải vẽ các đường thẳng sao cho số lượng các đường thẳng được vẽ ở ba góc độ đó càng gần nhau càng tốt, nghĩa là chênh lệch số đường thẳng của hai góc bất kì trong ba góc nhỏ hơn hoặc bằng 1. Chứng minh như sau:

Gọi số đường thẳng ở ba góc lần lượt là a, b, c . Không mất tính tổng quát, giả sử $a \leq b \leq c$. Công thức tính số tam giác đều đặc biệt như sau: $num = 2(ab + bc + ac)$. Nếu thêm đường ở góc có số đường thẳng là c thì số tam giác đều đặc biệt sẽ tăng thêm $2(a + b)$, khi ta thêm đường ở góc có số đường thẳng là b , số tam giác đều đặc biệt tăng thêm $2(a + c)$ nhưng ta thêm đường ở góc có số đường thẳng là a thì sẽ tăng lên $2(b + c)$. Và rõ ràng $a + b \leq a + c \leq b + c$. Hãy gọi các góc bằng các số tương ứng 1, 2, 3. Khi vẽ lần được các đường thẳng với các góc lần lượt là 1, 2, 3, 1, 2, 3, ... thì số cặp đường thẳng giao nhau tại tâm lục giác đều sẽ tăng lên, cụ thể là: $+0, +1, +2, +2, +3, +4, +4, +5, \dots$. Khi đó, ta thấy số tam giác đều đặc biệt sẽ tăng lên là: $+0, +2, +4, +4, +6, +8, +8, +10, \dots$. Và từ đó ta tính toán dựa trên dãy tăng lên này, với $n = 1$ thì số tam giác đều sẽ bằng $+0 = 0$, với $n = 2$ thì số tam giác đều sẽ bằng $+0 + 2 = 2$, và cứ tiếp tục. Về phần code, có thể đọc tại:

[https://github.com/hhoangcpascal/Codeforces/blob/main/Round%20788%20\(Div.%202\)/D.cpp](https://github.com/hhoangcpascal/Codeforces/blob/main/Round%20788%20(Div.%202)/D.cpp)

6. Minh chứng:

Tài khoản: [hhoangcpascal](#)

Bảng điểm có thể xem tại: [Standings - Codeforces Round #788 \(Div. 2\) - Codeforces](#)

Dưới đây là hình ảnh bảng rank được chụp lại:

424	yswlike	4345	478	916	1484	1467
425	SUZUKO_DAZE	4344	482	872	1379	1611
425	cocodev0113	4344	414	792	1554	1584
427	Koning	4343	482	876	1428	1557
428	DimonKrutoi	4342	490	956	1398	1498
429	9.86960440109	4341	478	924	1400	1539
429	bbmy	4341	490	884	1365	1602
431	raideva	4338	492	944	1498	1404
432	SmarthMehtalsBoring	4334	442	920	1442	1530
432	nambansal013	4334	486	874	1435	1539
432	Runtime-Terr0r	4334	486	924	1385	1539
435	adarsh1130	4332	492	952	1322	1566
436	panslava	4331	488	928	1421	1494
436	hhoangcpascal	4331	492	952	1456	1431
436	victor.gao	4331	352	752	1265	1962
439	Evil-BunKing	4329	494	932	1526	1377
440	debajyoti123	4327	494	948	1463	1422
441	Zemrith	4326	488	880	1428	1530
441	V4V	4326	496	672	1498	1660
441	Shirost	4326	478	848	1659	1341
444	Leokkk17	4324	492	968	1379	1485
445	atulrathore	4322	492	932	1463	1435
445	O0o0Oo0O0O00	4322	492	916	1393	1521



[hhoangcpascal](#) | [Logout](#)

HOME TOP CATALOG **CONTESTS** GYM PROBLEMSET GROUPS RATING EDU API CALENDAR HELP

PROBLEMS | COMMON STANDINGS **FRIENDS STANDINGS** | RATING CHANGES FRIENDS RATING CHANGES

☐ show unofficial

Codeforces Round #788 (Div. 2)

Final standings

You may double click into cells (or ctrl+click) to view the submissions history or hack the solution

#	Who	=	*	A 500	B 1000	C 1750	D 2250	E 2750	F 3000
1 (436)	hhoangcpascal	4331		492 00:04	952 00:12	1456 00:42	1431 01:01		
2 (1121)	Dukkha	3114				1260 01:15	1854 00:44		
3 (1206)	nghethuat102	2898		492 00:04	936 00:16	1470 00:40			
	Accepted Tried			7336 8565	4885 6140	2691 3373	1231 1374	143 150	32 63

Codeforces (c) Copyright 2010-2022 Mike Mirzayanov
The only programming contests Web 2.0 platform
Server time: Jun/01/2022 23:18:01^{UTC+7} (k3).
Desktop version, switch to [mobile version](#).
[Privacy Policy](#)

Supported by



PHẦN II. KÌ THI THỨ HAI

1. Mở đầu:

Kì thi thứ hai là kì thi [Codeforces Round #789 \(Div. 2\)](#), với những bài giải được là:

- A. Tokitsukaze and All Zero Sequence
- B1. Tokitsukaze and Good 01-String (easy version)
- C. Tokitsukaze and Strange Inequality
- D. Tokitsukaze and Meeting

Số điểm đạt được: 3507

2. Về bài A. Tokitsukaze and All Zero Sequence

2.1. Tóm tắt đề

Cho một dãy gồm n số nguyên không âm $a_1, a_2, a_3, \dots, a_n$. Với mỗi thao tác, chọn hai vị trí i, j ($1 \leq i \leq j$) và thực hiện như sau:

- Nếu $a_i = a_j$, sẽ chuyển $a_i = 0$ hoặc $a_j = 0$.
- Còn không, $a_i = a_j = \min(a_i, a_j)$.

Nhiệm vụ đề ra là đếm số thao tác tối thiểu để tất cả các $a_i = 0$.

2.2. Lời giải

Thuật toán: Duyệt

Ta nhận thấy rằng, nếu trong dãy ban đầu tồn tại ít nhất một số 0, thì đáp án sẽ là $n - \text{số lượng số 0}$. Còn không, bắt buộc phải có ít nhất một bước để dãy tồn tại một số 0, sau đó ra đáp án bình thường. Có hai trường hợp có thể xảy ra:

- Khi tồn tại một cặp số bằng nhau, đáp án sẽ là n , do phải tốn một bước để tạo một số 0, và thêm $n - 1$ bước để biến tất cả các $a_i = 0$.
- Còn không, đáp án sẽ là $n + 1$ do phải tốn thêm bước tạo ra một cặp số bằng nhau.

Về phần code, có thể đọc tại:

[https://github.com/hhoangcpascal/Codeforces/blob/main/Round%20789%20\(Div.%202\)/A.cpp](https://github.com/hhoangcpascal/Codeforces/blob/main/Round%20789%20(Div.%202)/A.cpp)

3. Về bài B1. Tokitsukaze and Good 01-String (easy version)

3.1. Tóm tắt đề

Cho một chuỗi s độ dài n chỉ chứa các kí tự 0, 1. Đảm bảo rằng n chẵn. Đề sẽ chia n ký tự thành ít nhất các đoạn liên tiếp, sao cho với mỗi đoạn chỉ gồm các kí tự 0 hoặc chỉ gồm các kí tự 1. Một chuỗi s gọi là tốt nếu với mỗi đoạn trên có độ dài chẵn.

Ví dụ: 00111100 là chuỗi tốt vì khi chia ra thành số đoạn ít nhất là 3: 00, 1111, 00, thì đoạn 00 có độ dài 2, đoạn 1111 có độ dài 4, 00 có độ dài 2, và tất cả đều chẵn. Có thể thực hiện không hoặc một hoặc một vài thao tác như sau:

- Chọn một vị trí i
- Nếu $s[i] = 1$, gán $s[i] = 0$, còn không, gán $s[i] = 1$.

Nhiệm vụ đề ra là tìm số thao tác ít nhất để biến chuỗi s thành chuỗi tốt.

3.2. Lời giải, bài giải

Thuật toán: Tham lam

Với bài này, ta sẽ tách ra thành các đoạn ít nhất. Gọi a_i là độ dài đoạn thứ i sau khi

tách. Nếu a_i lẻ, cộng vào a_{i+1} một lượng là 1 rồi xét tiếp. Lí do là khi a_i lẻ, ta sẽ thao tác ký tự cuối của đoạn thứ i , khi đó ký tự đó sẽ được sát nhập vào đoạn a_{i+1} . Đáp án sẽ là số lượng phải cộng, tức là số lượng thao tác trong khi duyệt.

Về phần code, có thể xem tại:

[https://github.com/hhoangcpascal/Codeforces/blob/main/Round%20789%20\(Div.%202\)/B1.cpp](https://github.com/hhoangcpascal/Codeforces/blob/main/Round%20789%20(Div.%202)/B1.cpp)

4. Về bài C. Tokitsukaze and Strange Inequality

4.1. Tóm tắt đề

Cho một dãy gồm n phần tử p_1, p_2, \dots, p_n . Đếm số bộ 4 (a, b, c, d) sao cho:

- $1 \leq a < b < c < d \leq n$.
- $p_a < p_c, p_b > p_d$

4.2. Lời giải, bài giải

Thuật toán: Duyệt, cấu trúc dữ liệu dữ liệu đặc biệt.

Với bài này, ta sẽ duyệt hai vòng for với mục đích là chọn hai chốt giữa b và c . Với mỗi bộ (b, c) ta sẽ đếm số bộ (a, d) thỏa mãn. Để đếm được, ta sẽ dùng hai cây chỉ số nhị phân (Binary Indexed Tree), một cây để đếm số a thỏa mãn $a < b$ và $p_a < p_c$, một cây để đếm số d sao cho $d > c$ và $p_d < p_b$.

Về phần code, có thể xem tại:

[https://github.com/hhoangcpascal/Codeforces/blob/main/Round%20789%20\(Div.%202\)/C.cpp](https://github.com/hhoangcpascal/Codeforces/blob/main/Round%20789%20(Div.%202)/C.cpp)

5. Về bài D. Tokitsukaze and Strange Inequality

5.1. Tóm tắt đề

Cho một lớp có kích thước $n \times m$, và có $n \cdot m$ chỗ ngồi. Có $n \cdot m$ học sinh, được đánh số từ 1 tới $n \cdot m$, mỗi học sinh được gán số 0 là học sinh nghịch ngợm hoặc số 1 là học sinh nghiêm túc.

Các học sinh lần lượt vào lớp từ 1 tới $n \cdot m$. Với mỗi lượt vào lớp, học sinh vào sau nhất sẽ được ngồi ở vị trí $(1, 1)$, còn các học sinh khác khi đang ở vị trí (i, j) sẽ chuyển chỗ tới vị trí $(i, j + 1)$, trong trường hợp $j = m$ thì chuyển tới chỗ $(i + 1, 1)$.

Nhiệm vụ đề ra là sau mỗi lượt vào lớp thì hãy đếm xem có bao nhiêu hàng và cột trong đó có ít nhất một học sinh nghiêm túc.

5.2. Lời giải, bài giải

Thuật toán: Quy hoạch động

Với bài này, ta sẽ tính số hàng riêng và số cột riêng, sau đó cộng lại.

- Với tính số hàng, với lượt thứ i , ta phải kiểm tra các đoạn $[i - m + 1, i]$, $[i - 2m + 1, i - m]$, ... xem có bao nhiêu đoạn chứa ít nhất một số 1. Ta có thể gọi $row[i]$ là số lượng hàng thỏa mãn khi đến lượt học sinh i bước vào. Công thức như sau: Nếu đoạn $[\max(1, i - m + 1), i]$ có ít nhất một số 1 thì $row[i] = row[\max(0, i - m)] + 1$, còn không, $row[i] = row[\max(0, i - m)]$.
- Với tính số cột, biết rằng những học sinh cùng cột là những học sinh có cùng số dư khi chia cho m . Ta có thể gọi $column[i]$ là số lượng cột thỏa mãn khi đến lượt học sinh i bước vào. Ta sẽ có dãy $column[i]$ không giảm, lí do khi học sinh i bước vào, các học sinh cùng số dư khi chia cho m sẽ cùng dịch chuyển

theo cột, nên khi đã có học sinh nghiêm túc thì khi dịch chuyển thì vẫn có học sinh nghiêm túc. Và ta sẽ duyệt học sinh tới modulo m , với mỗi số dư thì tìm học sinh i , i nhỏ nhất cùng số dư và là học sinh nghiêm túc. Khi tìm được sẽ tăng $column[i], column[i + 1], \dots, column[n \cdot m]$ lên 1.

Về phần code, có thể đọc tại:

[https://github.com/hhoangcpascal/Codeforces/blob/main/Round%20789%20\(Div.%202\)/D.cpp](https://github.com/hhoangcpascal/Codeforces/blob/main/Round%20789%20(Div.%202)/D.cpp)

6. Minh chứng

Tài khoản: [hhoangcpascal](#)

Bảng điểm có thể xem tại: [Standings - Codeforces Round #789 \(Div. 2\) - Codeforces](#)

Dưới đây là hình ảnh bảng rank được chụp lại:

#	Who	=	*	A 500	B1 750	B2 1000	C 1250	D 2000	E 2000	F 2750
1 (142)	Dukkha	3869					925 00:55	1656 00:43	1288 01:29	
2 (182)	hhoangcpascal	3507		438 00:56	717 00:11	-2	960 00:58	1392 01:16	-1	
3 (623)	ngethystuat102	2260		484 00:58	711 00:13	-1	1065 00:37			

Codeforces (c) Copyright 2010-2022 Mike Mirzayanov
The only programming contests Web 2.0 platform
Server time: Jun/01/2022 23:11:18+7 (k3).
Desktop version, switch to [mobile version](#).
[Privacy Policy](#)

Supported by



173	DKSH_2022	3558	494 00:59	699 00:11	-1	1005 00:43	1360 01:01	-8	
174	gohant23	3556	494 00:59	717 00:11	-1	1035 00:33	1310 01:20	-1	
175	kaylous	3532	496 00:52	729 00:07	-2	1115 00:27	1192 01:41		
176	vrobot	3531	436 00:32	732 00:06	-7	1115 00:27	1248 01:34		
177	Agaric	3524	406 00:47	627 00:41	598 01:03	845 01:25	1048 01:59		
178	Son	3522	-1	613 00:59	864 00:54	955 00:43		1090 01:35	
179	The_Hallak	3520	496 00:52	729 00:07	-1	1015 00:43	1280 01:05		
180	Tdolaberidze	3518	484 00:58	705 00:15	-4	1065 00:37		1264 01:32	
181	Yoks1729	3516	494 00:53	729 00:07	-3	1045 00:41	1248 01:34		
182	hhoangcpascal	3507	438 00:56	717 00:11	-2	960 00:58	1392 01:16	-1	
183	tfr	3499	490 00:55	654 00:31		1115 00:27	1240 01:35		
184	Jessc_wan	3497	400 00:51	585 00:55	-4	880 01:14	1632 00:46		
185	lovyamypm	3477	440 00:55	696 00:18		1085 00:33	1256 01:33	-2	
186	behti_hawa	3476	494 00:53	723 00:09	-2	995 00:51	1264 01:32	-1	
187	viethq_sad13	3471	494 00:53	720 00:10		1025 00:45	1232 01:36		
188	shjohw12	3470	496 00:52	726 00:08	-3	1160 00:19	1088 01:54		
189	Jtf_shine	3446	480 00:51	699 00:21		1075 00:53	1192 01:41		
190	snowy2002	3441	-1	672 00:25	856 00:36	785 01:03	1128 01:49		
191	th-loading	3439	484 00:58	714 00:12		1065 00:37	1176 01:43		
192	mirbek	3437	492 00:54	711 00:13	-2	970 00:56	1264 01:32		
193	mofumofu	3436	476 00:52	699 00:17	-3	1005 00:49		1256 01:33	
194	Abdulrahman27	3433	488 00:56	714 00:12	-2	975 00:53	1256 01:33		
195	Amano_Sei	3431	488 00:56	711 00:13	920 00:30	-1	1312 01:26		