Задача A. Абсолютно неадекватні слова

Ліміт часу: 1 second
Ліміт використання пам'яті: 256 megabytes

Деякі абсолютно неадекватні люди люблять використовувати в словах великі латинські літери `V` та `Z`.

Назвемо слово **абсолютно неадекватним**, якщо в ньому є велика латинська літера `V` чи велика латинська літера `Z`.

Для кожного слова з вхідних даних визначте, чи є воно **абсолютно неадекватним**.

Формат вхідних даних

Перший рядок містить єдине ціле число t ($1 \leqslant t \leqslant 10^4$) — кількість наборів вхідних даних. Далі слідує опис наборів вхідних даних.

Єдиний рядок кожного набору вхідних даних містить слово s (довжина s не перевищує 100), що складається лише з маленьких латинських літер, та, можливо, великих латинських літер `V` та `Z`.

Формат вихідних даних

Для кожного слова з вхідних даних, якщо воно є **абсолютно неадекватним**, виведіть `YES`. Інакше виведіть `NO`.

Виводити `YES` та `NO` можна у будь-якому регістрі (наприклад, рядки `yEs`, `yes`, `Yes` будуть сприйняті як позитивна відповідь).

Приклад

standard input	standard output
6 peremoga mogiliZazia Vodka oZVerenie zvzv zVZv	NO YES YES YES NO YES

Зауваження

Слова з другого, третього, четвертого, та шостого наборів вхідних даних містять літери `V` чи `Z`, а з інших ні.

Задача B. Мінус сума

Ліміт часу: 1 second
Ліміт використання пам'яті: 256 megabytes

Дано масив з n цілих чисел $[a_1, a_2, \ldots, a_n]$. За одну операцію можна зробити наступне:

- Виберіть довільне i від 1 до n. Після цього, замініть a_i на $-(a_1 + a_2 + \ldots + a_n)$.

Ви можете застосовувати дану операцію до масиву довільну кількість разів. Скільки різних масивів ви можете отримати? Якщо ви можете отримати нескінченну кількість масивів, виведіть -1, інакше виведіть число цих масивів по модулю 998244353.

Два масиви вважаються різними, якщо вони відрізняються принаймні в одній позиції.

Формат вхідних даних

Перший рядок містить єдине ціле число t ($1 \leqslant t \leqslant 10^4$) — кількість наборів вхідних даних. Далі слідує опис наборів вхідних даних.

Перший рядок кожного набору вхідних даних містить єдине ціле число n ($2 \leqslant n \leqslant 2 \cdot 10^5$) — довжину масиву.

Другий рядок кожного набору вхідних даних містить n цілих чисел $a_1, a_2, \ldots, a_n$ ($-10^9 \leqslant a_i \leqslant 10^9$) — елементи масиву.

Гарантується, що сума n по всім наборам вхідних даних не перевищує $2 \cdot 10^5$.

Формат вихідних даних

Для кожного набору вхідних даних, якщо ви можете отримати нескінченну кількість масивів, виведіть -1, інакше виведіть число цих масивів по модулю 998244353.

Приклад

standard input	standard output
3 1 -100 3 0 0 0 5 -1 1 -1 1 -1	2 1 20

Зауваження

В першому прикладі, з [100] ми можемо отримати лише [−100], а з [−100] лише [100].

В другому прикладі, масив завжди міститиме лише нулі.

Задача C. Найближчі точки

Ліміт часу: 2 seconds
Ліміт використання пам'яті: 256 megabytes

Вам дано n попарно різних точок $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$, кожна з яких лежить на одній з осей Ox та Oy (тобто, для кожного i принаймні одне з чисел x_i та $y_i = 0$).

Для кожної точки знайдіть відстань до найближчої точки до неї. Іншими словами, для кожного i знайдіть мінімальне значення $\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$ по всім $j \neq i$.

Формат вхідних даних

Перший рядок містить єдине ціле число t ($1 \leqslant t \leqslant 10^4$) — кількість наборів вхідних даних. Далі слідує опис наборів вхідних даних.

Перший рядок кожного набору вхідних даних містить одне ціле число n ($2 \leqslant n \leqslant 2 \cdot 10^5$) — кількість точок.

i-ий з наступних n рядків містить два цілих числа x_i, y_i ($-10^9 \leqslant x_i, y_i \leqslant 10^9$, принаймні одне з чисел x_i та y_i рівне нулю) — координати i-ої точки.

Гарантується, що всі точки попарно різні (тобто для кожних $i \neq j$ виконується принаймні одне з $x_i \neq x_j$, $y_i \neq y_j$).

Гарантується, що сума n по всім наборам вхідних даних не перевищує $2 \cdot 10^5$.

Формат вихідних даних

Для кожного набору вхідних даних виведіть n чисел, i-те з яких рівне відстані від i-ої точки до найближчої точки до неї.

Ваша відповідь буде вважатися правильною, якщо абсолютна або відносна помилка кожної відстані не перевищує 10^{-6}.

Формально, нехай ваша відповідь дорівнює a, а відповідь журі дорівнює b. Ваша відповідь буде зарахована, якщо і тільки якщо $\frac{|a-b|}{\max(1,|b|)} \leqslant 10^{-6}$.

Приклад

standard input	standard output
4 2 0 1000000000 0 -1000000000 4 0 10 0 -10 10 0 -10 0 4 0 4 0 -5 8 0 -8 0 5 0 -5 0 -2 0 0 0 2 0 5	2000000000.0 2000000000.0 14.142136 14.142136 14.142136 14.142136 8.944272 9.0 8.944272 8.944272 3.0 2.0 2.0 2.0 3.0

Зауваження

В першому наборі вхідних даних, найближча точка до точки 1 — точка 2, а до точки 2 — точка 1.

В другому наборі вхідних даних, точки 1, 3, 2, 4 (в цьому порядку) формують квадрат зі стороною $10\sqrt{2}$, тому для кожної точки найближча до неї точка знаходиться на відстані $10\sqrt{2}$.

В третьому наборі вхідних даних, точка 1 найближча до точок $2, 3, 4$. Для точки 1 же однаково близько точки 3 та 4.

Задача D. Запити

Ліміт часу: 1 second
Ліміт використання пам'яті: 256 megabytes

Вам дано n чисел $a_1, a_2, \ldots, a_n$. Ви маєте обробити q запитів наступних двох видів:

- 1 x: Додайте до кожного числа x
- 2 k: Знайдіть k-те за величиною число (наприклад, якщо $k = 1$, знайдіть найбільше за величиною число)

Формат вхідних даних

Перший рядок містить два цілих числа n, q $(1 \leqslant n, q \leqslant 2 \cdot 10^5)$ — кількість чисел та кількість запитів відповідно.

Другий рядок містить n чисел $a_1, a_2, \ldots, a_n$ $(1 \leqslant a_i \leqslant 10^9)$ — початкові значення чисел. Далі йде q рядків, що описують запити.

Якщо запит першого типу, то відповідний рядок містить число 1 та число x $(1 \leqslant x \leqslant 10^9)$. Інакше відповідний рядок містить число 2 та число k $(1 \leqslant k \leqslant n)$.

Гарантується, що серед запитів принаймні один буде другого типу.

Формат вихідних даних

Для кожного запиту другого типу виведіть k-те за величиною число.

Приклад

standard input	standard output
3 5 42 1 2 2 1 1 1000000 2 2 1 2022 2 3	42 1000002 1002023

Зауваження

- **Запит 1:** 1-ше число в масиві за величиною — 42.
- **Запит 2:** Ми додаємо 1000000 до всіх чисел. Тепер вони рівні $1000042, 1000001, 1000002$.
- **Запит 3:** 2-ге число в масиві за величиною — 1000002.
- **Запит 4:** Ми додаємо 2022 до всіх чисел. Тепер вони рівні $1002064, 1002023, 1002024$.
- **Запит 5:** 3-тє число в масиві за величиною — 1002023.

Задача E. Закусити

Ліміт часу: 1 second
Ліміт використання пам'яті: 256 megabytes

Масив $a_1, a_2, \ldots, a_m$ цілих чисел називається **непарним**, якщо в ньому непарна кількість інверсій, і **парним** в іншому випадку. Нагадаємо, що інверсія — це пара (i, j) з $1 \leqslant i < j \leqslant m$ така, що $a_i > a_j$. Наприклад, в масиві $[2, 4, 1, 3]$ 3 інверсії: $(1, 3), (2, 3), (2, 4)$ (оскільки $a_1 > a_3, a_2 > a_3, a_2 > a_4$), тому він **непарний**.

Для перестановки $p_1, p_2, \ldots, p_n$ чисел від 1 до n назвемо її красою довжину її найдовшої **непарної** підпослідовності, якщо така існує, а інакше -1. Наприклад, краса перестановки $(1, 2, 3)$ рівна -1, адже кожна її підпослідовність **парна**, краса $(4, 1, 2, 3)$ рівна 4, адже вся перестановка є **непарною**, а краса $(4, 1, 3, 2)$ рівна 3, адже вся перестановка є **парною**, а підпослідовність $(4, 3, 2)$ є **непарною**.

Нам дана початкова перестановка $p_1, p_2, \ldots, p_n$. До неї буде q запитів оновлення. Після i-го запиту ми матимемо обміняти місцями p_{u_i} та p_{v_i}.

Знайдіть красу перестановки після кожного запиту.

Нагадаємо, що масив b є підпослідовністю c, якщо b можна отримати з c видаленням кількох (можливо, жодного або всіх) елементів.

Нагадаємо, що перестановка чисел від 1 до n — це масив довжини n, що містить кожне число від 1 до n рівно один раз.

Формат вхідних даних

Перший рядок містить два цілих числа n, q $(1 \leqslant n, q \leqslant 2 \cdot 10^5)$ — довжину перестановки та кількість запитів відповідно.

Наступний рядок містить n цілих чисел $p_1, p_2, \ldots, p_n$ $(1 \leqslant p_i \leqslant n$, всі p_i попарно різні$)$ — початкову перестановку p.

i-ий з наступних q рядків містить два цілих числа u_i, v_i $(1 \leqslant u_i, v_i \leqslant n, u_i \neq v_i)$, позначаючі, що після i-го запиту ви маєте обміняти місцями p_{u_i} та p_{v_i}.

Формат вихідних даних

Виведіть q чисел — красу перестановки після кожного запиту оновлення.

Приклад

standard input	standard output
5 6 2 1 3 4 5 1 2 1 2 1 4 2 1 3 5 1 3	-1 5 4 5 3 5

Зауваження

- Після першого запиту перестановка має вигляд $(1, 2, 3, 4, 5)$. В ній нема жодної **непарної** підпослідовності.
- Після другого запиту перестановка має вигляд $(2, 1, 3, 4, 5)$. Вся перестановка **непарна**, адже в ній рівно одна інверсія.

- Після третього запиту перестановка має вигляд $(4, 1, 3, 2, 5)$. Вся перестановка **парна**, але її підпослідовність $(4, 3, 2, 5)$ **непарна**.
- Після четвертого запиту перестановка має вигляд $(1, 4, 3, 2, 5)$. Вся перестановка **непарна**.
- Після п'ятого запиту перестановка має вигляд $(1, 4, 5, 2, 3)$. Вся перестановка **парна**, і всі її підпослідовності довжини 4 **парні**, але підпослідовність $(1, 5, 2)$ **непарна**.
- Після шостого запиту перестановка має вигляд $(5, 4, 1, 2, 3)$. Вся перестановка **непарна**.

Задача F. Універсальний обмінювач

Ліміт часу: 1 second
Ліміт використання пам'яті: 256 megabytes

Вам в руки потрапив пристрій для роботи з перестановками $p_1, p_2, \ldots, p_n$ чисел від 1 до n. Він може робити m операцій. i-та операція описується двома числами a_i, b_i ($1 \leqslant a_i, b_i \leqslant n, a_i \neq b_i$). Вона полягає в наступному: якщо $p_{a_i} > p_{b_i}$, то пристрій обміняє місцями p_{a_i}, p_{b_i}. Інакше він нічого не зробить.

Ви можете застосовувати ці операції будь-яку кількість разів, в будь-якому порядку (в тому числі, одну операцію ви можете використати більше одного разу).

Вас цікавить, чи можна з допомогою даного пристрою з будь-якої перестановки отримати будь-яку іншу. Іншими словами, визначте, чи для кожних двох перестановок $p_1, p_2, \ldots, p_n$ і $q_1, q_2, \ldots, q_n$ чисел від 1 до n існує послідовність операцій, яку можна застосувати до p, щоб отримати q.

Нагадаємо, що перестановка чисел від 1 до n — це масив довжини n, що містить кожне число від 1 до n рівно один раз.

Формат вхідних даних

Перший рядок містить єдине ціле число t ($1 \leqslant t \leqslant 10^4$) — кількість наборів вхідних даних. Далі слідує опис наборів вхідних даних.

Перший рядок кожного набору вхідних даних містить два числа n, m ($1 \leqslant n \leqslant 2 \cdot 10^5, 0 \leqslant m \leqslant 2 \cdot 10^5$) — довжину перестановок, з якими працює ваш пристрій, та кількість операцій, що він вміє робити.

i-ий з m наступних рядків містить два числа a_i, b_i ($1 \leqslant a_i, b_i \leqslant n, a_i \neq b_i$), що описують i-ту операцію: якщо $p_{a_i} > p_{b_i}$, пристрій може обміняти місцями елементи p_{a_i}, p_{b_i} перестановки.

Гарантується, що всі пари (a_i, b_i) попарно різні, але зверніть увагу, що можуть зустрічатись водночас пари (x, y) та (y, x) для деяких x, y.

Гарантується, що сума n по всім наборам вхідних даних не перевищує $2 \cdot 10^5$, а також сума m по всім наборам вхідних даних не перевищує $2 \cdot 10^5$.

Формат вихідних даних

Для кожного набору вхідних даних, виведіть `YES`, якщо з допомогою даного пристрою з будь-якої перестановки можна тримати будь-яку іншу, і `NO` інакше.

Виводити `YES` та `NO` можна у будь-якому регістрі (наприклад, рядки `yEs`, `yes`, `Yes` будуть сприйняті як позитивна відповідь).

Приклад

standard input	standard output
5 2 1 1 2 3 4 1 2 2 1 1 3 3 1 5 4 1 2 2 3 3 4 4 5 5 6 3 5 5 1 1 2 4 3 2 4 4 1 4 6 3 1 2 3 2 1 3 2 1 2 1 3	NO YES NO YES NO

Зауваження

В першому прикладі, ми можемо обміняти p_1 та p_2 місцями, якщо $p_1 > p_2$. Таким чином, ми можемо отримати перестановку $(2, 1)$ з перестановки $(1, 2)$, але не зможемо отримати перестановку $(1, 2)$ з перестановки $(2, 1)$, тому відповідь `NO`.

В другому прикладі, ми можемо обміняти місцями p_1, p_2 незалежно від того, яке з них більше (адже в нас є обидві операції обміну: і з $a_i = 1, b_i = 2$, і з $a_i = 2, b_i = 1$). Також, ми можемо обміняти місцями p_1, p_3 незалежно від того, яке з них більше. Нескладно показати, що в такому разі ми можемо отримати з будь-якої перестановки будь-яку іншу, тому відповідь `YES`.

Задача G. Мінімізуйте суму

Ліміт часу: 1 second
Ліміт використання пам'яті: 256 megabytes

Табличка $n \times m$ заповнена невід'ємними числами. Андрійко з'ясував, що XOR всіх чисел в i-му рядку рівний row_i (для i від 1 до n), а XOR всіх чисел в i-му стовпчику рівний col_i (для i від 1 до m).

Чи міг Андрійко сказати правду? Якщо так, то якою найменшою могла бути сума всіх чисел в табличці?

Нагадаємо, що XOR позначає операцію **побітового виключного АБО**. Наприклад, 13 XOR 6 = 11, адже в двійковому записі 13 = `1101`, а 6 = `0110`, тому їх XOR має бути рівним `1011` = 11.

Формат вхідних даних

Перший рядок містить єдине ціле число t ($1 \leqslant t \leqslant 10^4$) — кількість наборів вхідних даних. Далі слідує опис наборів вхідних даних.

Перший рядок кожного набору вхідних даних містить два цілих числа n, m ($1 \leqslant n, m \leqslant 2 \cdot 10^5$) — кількості рядків та стовпчиків відповідно.

Другий рядок кожного набору вхідних даних містить n цілих чисел $row_1, row_2, \ldots, row_n$ ($0 \leqslant row_i < 2^{20}$).

Третій рядок кожного набору вхідних даних містить m цілих чисел $col_1, col_2, \ldots, col_m$ ($0 \leqslant col_i < 2^{20}$).

Гарантується, що сума n по всім наборам вхідних даних не перевищує $2 \cdot 10^5$, а також що сума m по всім наборам вхідних даних не перевищує $2 \cdot 10^5$.

Формат вихідних даних

Для кожного набору вхідних даних, якщо такої таблички не могло існувати, виведіть -1. Інакше, виведіть мінімальну можливу суму всіх чисел в табличці, в якій XOR всіх чисел в i-му рядку рівний row_i (для i від 1 до n), а XOR всіх чисел в i-му стовпчику рівний col_i (для i від 1 до m).

Приклад

standard input	standard output
3 2 3 0 0 0 0 0 3 3 1 2 3 2 3 4 4 5 32 64 96 128 160 96 32 224 128	0 -1 640

Зауваження

В першому наборі вхідних даних можна просто заповнити всю таблицю нулями. XOR в кожному рядочку та стовпчику буде рівний 0. Очевидно, що отримати меншу суму неможливо.

В другому наборі вхідних даних можна показати, що такої таблиці не може існувати.

В третьому наборі вхідних даних один з прикладів заповнення таблиці такий:

160	32	0	32	128
0	64	0	0	0
0	0	32	64	0
0	0	0	128	0

Задача H. Фарбування кола

Ліміт часу: 1 second
Ліміт використання пам'яті: 256 megabytes

Розглянемо n точок, розташованих на колі. Ми хочемо пофарбувати їх в три кольори таким чином, щоб жодні дві сусідні точки не мали однаковий колір.

Але є додаткова умова: ми не можемо фарбувати точку i в колір b_i. Чи можна пофарбувати всі точки?

Іншими словами, визначте, чи існують такі $a_1, a_2, \ldots, a_n$, що для кожного i:

- $1 \leqslant a_i \leqslant 3$
- $a_i \neq b_i$
- $a_i \neq a_{i+1}$ (тут $a_{n+1} = a_1$)

Якщо так, то знайдіть якісь такі $a_1, a_2, \ldots, a_n$.

Формат вхідних даних

Перший рядок містить єдине ціле число t ($1 \leqslant t \leqslant 10^4$) — кількість наборів вхідних даних. Далі слідує опис наборів вхідних даних.

Перший рядок кожного набору вхідних даних містить єдине ціле число n ($3 \leqslant n \leqslant 2 \cdot 10^5$) — кількість точок.

Другий рядок кожного набору вхідних даних містить n цілих чисел $b_1, b_2, \ldots, b_n$ ($1 \leqslant b_i \leqslant 3$) — заборонені кольори.

Гарантується, що сума n по всіх наборах вхідних даних не перевищує $2 \cdot 10^5$.

Формат вихідних даних

Для кожного набору вхідних даних, якщо такого розфарбування не існує, виведіть `NO`.

Інакше, виведіть `YES`. В наступному рядку виведіть n чисел $a_1, a_2, \ldots, a_n$, що задовольняють всім вимогам з умови.

Виводити `YES` та `NO` можна у будь-якому регістрі (наприклад, рядки `yEs`, `yes`, `Yes` будуть сприйняті як позитивна відповідь).

Приклад

standard input	standard output
4 4 1 2 1 2 3 3 3 3 5 1 2 3 1 2 6 1 1 2 2 3 3	YES 2 1 2 1 NO YES 2 1 2 3 1 YES 2 3 1 3 2 1

Задача I. Проста умова

Ліміт часу: 1.5 seconds
Ліміт використання пам'яті: 256 megabytes

Вам дано неорієнтований граф на n вершинах. Для кожної пари вершин (i, j) $(i \neq j)$, визначте, чи існує Гамільтонів шлях, що починається в i, а закінчується в j.

Нагадаємо, що Гамільтонів шлях — це шлях, що складається з $n - 1$ ребер, що проходить по всім вершинам рівно по одному разу.

Формат вхідних даних

Перший рядок містить одне ціле число n $(2 \leqslant n \leqslant 24)$ — кількість вершин в графі.

i-ий з наступних n рядків містить бінарний рядок s_i довжини n. Його i-ий символ завжди рівний `0`, а для $j \neq i$ його j-ий символ рівний 1, якщо між вершинами i та j є ребро, і `0`, якщо немає.

Гарантується, що для довільних $i \neq j$ i-ий символ j-го рядка співпадає з j-им символом i-го рядка.

Формат вихідних даних

Виведіть n рядків. В i-му з них виведіть бінарний рядок довжини n. Його i-ий символ має бути рівним `0`, а j-ий при $j \neq i$ має бути рівним `1`, якщо між вершинами i та j є Гамільтонів шлях, і `0`, якщо немає.

Приклади

standard input	standard output
4 0110 1010 1101 0010	0001 0001 0000 1100
6 010001 101000 010100 001010 000101 100010	010001 101000 010100 001010 000101 100010
4 0111 1011 1101 1110	0111 1011 1101 1110

Зауваження

В першому прикладі, Гамільтонів шлях є між парами $(1, 4)$ та $(2, 4)$.

В другому прикладі, граф це цикл довжини 6. Гамільтонів шлях тут є лише між парами сусідніх вершин.

В третьому прикладі ми маємо повний граф на 4 вершинах. Гамільтонів шлях тут є між кожною парою вершин.

Задача J. Зростаюча таблиця

Ліміт часу: 1 second
Ліміт використання пам'яті: 256 megabytes

Є таблиця $n \times m$, яку ми хочемо заповнити цілими числами. Позначатимемо число в клітині на перетині i-го рядка та j-го стовпця, як $a_{i,j}$.

Ми хочемо, щоб виконувались наступні умови:

- $1 \leqslant a_{i,j} \leqslant n + m$ для всіх $1 \leqslant i \leqslant n$, $1 \leqslant j \leqslant m$.
- $a_{i-1,j} < a_{i,j}$ для всіх $1 \leqslant i \leqslant n - 1$, $1 \leqslant j \leqslant m$.
- $a_{i,j-1} < a_{i,j}$ для всіх $1 \leqslant i \leqslant n$, $1 \leqslant j \leqslant m - 1$.

Іншими словами, ми хочемо заповнити таблицю числами від 1 до $n + m$, причому в кожному рядку та в кожному стовпці числа мають зростаюти.

Також, нам відомі значення чисел в деяких клітинках таблиці. Знайдіть кількість способів підібрати значення чисел, що нам не відомі, щоб всі умови вище виконувались. Оскільки ця кількість може бути дуже великою, виведіть її по модулю 998244353.

Формат вхідних даних

Перший рядок містить єдине ціле число t ($1 \leqslant t \leqslant 10^4$) — кількість наборів вхідних даних. Далі слідує опис наборів вхідних даних.

Перший рядок кожного набору вхідних даних містить два цілих числа n, m ($1 \leqslant n, m \leqslant 2 \cdot 10^5, 1 \leqslant nm \leqslant 2 \cdot 10^5$) — розміри таблиці. Далі йдуть n рядків.

i-ий з наступних n рядків містить m цілих чисел $a_{i,1}, a_{i,2}, \ldots, a_{i,m}$ ($1 \leqslant a_{i,j} \leqslant n+m$ чи $a_{i,j} = -1$). Якщо $a_{i,j} \neq -1$, то число на перетині i-го рядка та j-го стовпця нам відоме (і рівне $a_{i,j}$), інакше його потрібно підібрати.

Гарантується, що сума nm по всім наборам вхідних даних не перевищує $2 \cdot 10^5$.

Формат вихідних даних

Для кожного набору вхідних даних виведіть єдине ціле число — кількість заповнити невідомі значення таблиці таким чином, щоб всі умови виконувались.

Приклад

standard input	standard output
4 2 3 1 2 -1 -1 4 5 4 4 -1 -1 -1 -1 -1 -1 -1 -1 -1 4 4 -1 -1 -1 -1 -1 4 4 -1 -1 -1 -1 -1 -1 -1 -1 -1 4 5 -1 -1 -1 -1 -1 3 5 1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1	4 0 17 55

Зауваження

В першому наборі вхідних даних, нам лишилось лише підібрати значення $a_{2,1}$ та $a_{1,3}$. $a_{2,1}$ може приймати значення $2, 3$, а $a_{1,3}$ — $3, 4$. Всього 4 варіанти.

В другому наборі вхідних даних, нема жодної такої таблиці, оскільки вже порушена умова $a_{3,2} < a_{3,3}$.

Задача K. Різнобарвний кістяк

Ліміт часу: 2 seconds
Ліміт використання пам'яті: 256 megabytes

В нас є граф на n вершинах. i-та вершина має колір c_i і значення a_i. Якщо вершини i та j різного кольору, то між ними є ребро ваги $|a_i - a_j|$, інакше між ними немає ребра. Гарантується, що є хоча б два різні кольори.

Знайдіть вагу мінімального кістяка в цьому графі.

Нагадаємо, що кістяк графу — це підмножина з $n-1$ ребер цього графу, що формують дерево. Мінімальний кістяк — це кістяк з найменшою сумою ваг ребер в ньому.

Формат вхідних даних

Перший рядок вхідних даних містить єдине ціле число n ($2 \leqslant n \leqslant 2 \cdot 10^5$) — кількість вершин в графі.

Другий рядок вхідних даних містить n цілих чисел $c_1, c_2, \ldots, c_n$ ($1 \leqslant c_i \leqslant n$) — кольори вершин графу. Гарантується, що серед $c_1, c_2, \ldots, c_n$ є принаймні два різні кольори.

Третій рядок вхідних даних містить n цілих чисел $a_1, a_2, \ldots, a_n$ ($0 \leqslant a_i \leqslant 10^9$).

Формат вихідних даних

Виведіть єдине число — вагу мінімального кістяку цього графу.

Приклади

standard input	standard output
4 1 2 3 4 10 40 20 30	30
6 1 2 1 2 1 2 0 100 1 99 2 98	486

Зауваження

В першому прикладі, мінімальний кістяк формують ребра $(1,3), (3,4), (4,2)$, кожне з вагою 20.

В другому прикладі, один з мінімальних кістяків формують ребра $(5,6), (5,4), (5,2), (6,3), (6,1)$ з вагами $96, 97, 98, 97, 98$ відповідно. Вага всього кістяка рівна $96 + 97 + 98 + 97 + 98 = 486$.

Задача L. Ненеспадаючі масиви

Ліміт часу: 1 second
Ліміт використання пам'яті: 256 megabytes

Масив $b_1, b_2, \ldots, b_m$ називається неспадаючим, якщо $b_1 \leqslant b_2 \ldots \leqslant b_m$. Інакше масив називається **ненеспадаючим**.

Вам дано **ненеспадаючий** масив $a_1, a_2, \ldots, a_n$. Ви хочете розбити його на максимальну кількість послідовних підмасивів, кожен з яких є **ненеспадаючим**. На яку максимальну кількість підмасивів ви можете розбити a?

Формат вхідних даних

Перший рядок містить єдине ціле число t ($1 \leqslant t \leqslant 10^4$) — кількість наборів вхідних даних. Далі слідує опис наборів вхідних даних.

Перший рядок кожного набору вхідних даних містить одне ціле число n ($2 \leqslant n \leqslant 2 \cdot 10^5$) — довжину масиву.

Другий рядок кожного набору вхідних даних містить n цілих чисел $a_1, a_2, \ldots, a_n$ ($0 \leqslant a_i \leqslant 10^6$).

Гарантується, що масив a є **ненеспадаючим**.

Гарантується, що сума n по всім наборам вхідних даних не перевищує $2 \cdot 10^5$.

Формат вихідних даних

Для кожного набору вхідних даних виведіть єдине ціле число — максимальну можливу кількість **ненеспадаючих** підмасивів, на яку можна розбити a.

Приклад

standard input	standard output
3 6 1 0 1 0 1 0 5 3 2 1 4 5 9 100 10 1 100 10 1 100 10 1	3 1 3

Зауваження

В першому наборі вхідних даних, можна розбити масив на три підмасиви $[1, 0], [1, 0], [1, 0]$.

В другому наборі вхідних даних, не можна розбити масив на більше ніж один ненеспадаючий підмасив.

В третьому наборі вхідних даних, можна розбити масив на три підмасиви $[100, 10], [1, 100, 10, 1, 100], [10, 1]$.