Задача A. Наиболее вкусный торт

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

В ряд расположены n кусочков торта. i-й кусочек имеет вес a_i $(1 \leqslant i \leqslant n)$.

Ценностью торта называется максимальный суммарный вес двух подряд идущих кусочков (т. е. $\max(a_1 + a_2,\ a_2 + a_3,\ \ldots,\ a_{n-1} + a_n)$).

Вы хотите сделать ценность торта как можно больше. Вы можете выполнить следующую операцию не более одного раза (иначе торт будет испорчен):

- Выберите последовательный подотрезок $a[l, r]$ кусочков торта $(1 \leqslant l \leqslant r \leqslant n)$ и разверните его.

Подотрезком $a[l, r]$ массива a называется последовательность $a_l, a_{l+1}, \ldots, a_r$. Если ее развернуть, массив станет равным $a_1, a_2, \ldots, a_{l-2}, a_{l-1}, \underline{a_r}, \underline{a_{r-1}}, \underline{\ldots}, \underline{a_{l+1}}, \underline{a_l}, a_{r+1}, a_{r+2}, \ldots, a_{n-1}, a_n$.

Например, если веса кусочков изначально равны $[5, 2, 1, 4, 7, 3]$, можно развернуть подотрезок $a[2, 5]$, получив $[5, \underline{7}, \underline{4}, \underline{1}, \underline{2}, 3]$. Тогда ценность торта будет равна $5 + 7 = 12$ (а до разворота она была равна $4 + 7 = 11$).

Найдите максимально возможную ценность торта после выполнения не более чем одной операции.

Формат входных данных

Первая строка содержит одно целое число t $(1 \leqslant t \leqslant 50)$ — количество наборов входных данных.

Первая строка каждого набора содержит одно целое число n $(2 \leqslant n \leqslant 1000)$ — количество кусочков торта.

Вторая строка каждого набора содержит n целых чисел $a_1, a_2, \ldots, a_n$ $(1 \leqslant a_i \leqslant 10^9)$, где a_i равняется весу i-го кусочка торта.

Формат выходных данных

Для каждого набора входных данных выведите одно целое число: максимальную ценность торта после выполнения не более чем одной операции.

Пример

стандартный ввод
5 6 5 2 1 4 7 3 3 32 78 78 3 69 54 91 8 999021 999021 999021 999021 999652 999021 999021 999021 2 1000000000 1000000000

стандартный вывод
12 156 160 1998673 2000000000

Замечание

В первом примере после выполнения разворота подотрезка $a[2,5]$ получится торт с весами $[5, \underline{7}, \underline{4}, \underline{1}, \underline{2}, 3]$. Ценность такого торта равна $\max(5+7, 7+4, 4+1, 1+2, 2+3) = 12$. Это максимально возможная ценность, которую можно получить, выполнив операцию не более одного раза.

Во втором тесте оптимально не делать ни одной операции. Ценность равна $78 + 78 = 156$.

В третьем примере после разворота подотрезка $a[1,2]$ получится торт с весами $[\underline{54}, \underline{69}, 91]$. Ценность получившегося торта равна $\max(54+69, 69+91) = 160$. Невозможно получить ценность выше 160, выполнив не более одной операции.

Задача B. Префиксные удаления

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Вам дана строка s, состоящая из строчных букв латинского алфавита. Вы должны выполнить над строкой s следующий алгоритм:

- Обозначим за x длину наибольшего префикса s, который встречается где-то еще в s как подстрока из последовательных символов (это вхождение может пересекаться с самим префиксом). Если $x = 0$, закончите алгоритм. Иначе удалите первые x символов s и повторите.

Префиксом строки называются несколько первых букв данной строки без изменения порядка. Пустая строка тоже является корректным префиксом. Например, у строки «`abcd`» пять префиксов: пустая строка, «`a`», «`ab`», «`abc`» и «`abcd`».

Например, выполним алгоритм над $s =$ «`abcabdc`».

- Изначально «`ab`» является самым длинным префиксом, который встречается где-то еще в s как подстрока, поэтому после 1 операции $s =$ «`cabdc`».
- Затем «`c`» является самым длинным префиксом, который встречается где-то еще в s, поэтому после 2 операций $s =$ «`abdc`».
- После этого $x = 0$ (т. к. нет префиксов «`abdc`», которые встречаются как подстрока где-либо еще в s), поэтому алгоритм завершается.

Найдите финальное состояние строки после завершения алгоритма.

Формат входных данных

Первая строка содержит одно целое число t ($1 \leqslant t \leqslant 10^4$) — количество наборов входных данных.

Далее следуют t строк, каждая из который описывает один набор входных данных. Каждая строка содержит строку s. Данная строка содержит только строчные буквы латинского алфавита и имеет длину от 1 до $2 \cdot 10^5$ включительно.

Гарантируется, что сумма длин s по всем наборам входных данных не превышает $2 \cdot 10^5$.

Формат выходных данных

Для каждого набора входных данных выведите строку s после завершения алгоритма. Можно показать, что эта строка не может быть пустой.

Пример

стандартный ввод	стандартный вывод
6 abcabdc a bbbbbbbbbb codeforces cffcfccffccfcffcfccfcffccffcfccf zyzyzwxxyyxxyyzzyzzxxwzxwywxwzxxyzzw	abdc a b deforces cf xyzzw

Замечание

Первый пример пояснен в условии задачи.

Во втором примере нельзя сделать ни одной операции с s.

В третьем примере,

- Изначально $s =$ «bbbbbbbbbb».

- После 1 операции $s =$ «`b`».

В четвертом примере,

- Изначально $s =$ «`codeforces`».
- После 1 операции $s =$ «`odeforces`».
- После 2 операции $s =$ «`deforces`».

Задача C. Алиса и торт

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

У Алисы есть торт, и она собирается его разрезать. Она выполнит следующую операцию $n-1$ раз: выберет кусочек пирога (изначально весь торт является одним куском) с весом $w \geqslant 2$ и разрежем его на два более маленьких кусочка с весами $\lfloor \frac{w}{2} \rfloor$ и $\lceil \frac{w}{2} \rceil$ ($\lfloor x \rfloor$ и $\lceil x \rceil$ обозначают округление вниз и вверх, соответственно).

После того как Алиса разрежет торт на n кусочков, она расположит эти n кусочков на столе в произвольном порядке. Обозначим за a_i вес i-го кусочка.

Вам дан массив a. Определите, существует ли некоторый начальный вес торта и последовательность операций разрезания такие, чтобы веса кусочков получались равными a.

Формат входных данных

Первая строка содержит одно целое число t ($1 \leqslant t \leqslant 10^4$) — количество наборов входных данных.

Первая строка каждого набора содержит одно целое число n ($1 \leqslant n \leqslant 2 \cdot 10^5$).

Вторая строка содержит n целых чисел $a_1, a_2, \ldots, a_n$ ($1 \leqslant a_i \leqslant 10^9$).

Гарантируется, что сумма n для всех наборов входных данных не превышает $2 \cdot 10^5$.

Формат выходных данных

Для каждого набора входных данных выведите одну строку: выведите `YES`, если массив a может быть получен с помощью действий Алисы, иначе выведите `NO`.

Вы можете выводить буквы в любом регистре (например, `YES`, `Yes`, `yes`, `yEs` будут приняты как положительный ответ).

Пример

стандартный ввод	стандартный вывод
14 1 327 2 869 541 2 985214736 985214737 3 2 3 1 3 2 3 3 6 1 1 1 1 1 1 6 100 100 100 100 100 100 8 100 100 100 100 100 100 100 100 8 2 16 1 8 64 1 4 32 10 1 2 4 7 1 1 1 1 7 2 10 7 1 1 1 3 1 3 3 2 3 10 1 4 4 1 1 1 3 3 3 1 10 2 3 2 2 1 2 2 2 2 2 4 999999999 999999999 999999999 999999999	YES NO YES YES NO YES NO YES YES YES YES NO NO YES

Замечание

В первом примере можно получить массив a, выполнив 0 операций с тортом веса 327.

Во втором примере невозможно получить массив a.

В третьем примере можно получить массив a, выполнив 1 операцию с тортом веса 1 970 429 473:

- Разрезать торт пополам, веса будут равны $[985\,214\,736, 985\,214\,737]$.

Обратите внимание, что начальный вес торта может быть больше 10^9.

В четвертом примере можно получить массив a выполнив 2 операции над тортом веса 6:

- Разрезать торт пополам, веса кусочков будут равны $[3, 3]$.
- Разрезать один из кусочков веса 3, веса кусочков будут равны $[1, 2, 3]$, что совпадает с $[2, 3, 1]$ с точностью до порядка.

Задача D. Урок зельеварения

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

Профессор зельеварения дал следующее задание студентам: сварить зелье, используя n ингредиентов, так, чтобы доля ингредиента i в зелье равнялась бы $r_i > 0$ (при этом $r_1 + r_2 + \cdots + r_n = 1$).

К сожалению, он забыл рецепт зелья, и теперь только помнит набор из $n - 1$ факта вида «ингредиенты i и j должны входить в отношении x к y» (т. е. если за a_i и a_j обозначить объемы ингредиентов i и j в зелье соответственно, то должно выполняться $a_i/a_j = x/y$), где x и y — положительные целые числа. Гарантируется, что данный набор фактов достаточен, чтобы однозначно определить доли ингредиентов в рецепте r_i.

Профессор решил, что он зачтет задание студентам, если они приготовят зелье, удовлетворяющее всем $n - 1$ ограничениям (таких зелий может быть много), и при этом объем каждого ингредиента строго положительный и выражается целым числом.

Найдите минимальный суммарный объем всех ингредиентов, необходимый, чтобы сварить подходящее зелье. Так как ответ может быть большим, выведите его остаток от деления на $998\,244\,353$.

Формат входных данных

Первая строка содержит одно целое число t ($1 \leqslant t \leqslant 10^4$) — количество наборов входных данных.

Первая строка каждого набора содержит одно целое число n ($2 \leqslant n \leqslant 2 \cdot 10^5$).

Каждая из следующих $n - 1$ строк содержит четыре целых числа i, j, x, y ($1 \leqslant i, j \leqslant n$, $i \neq j$, $1 \leqslant x, y \leqslant n$), обозначающие, что ингредиенты i и j должны иметь отношение объемов x к y. Гарантируется, что этот набор фактов достаточен, чтобы однозначно определить исходные доли r_i.

Гарантируется, что сумма значений n по всем наборам входных данных не превосходит $2 \cdot 10^5$.

Формат выходных данных

Для каждого набора входных данных выведите минимальный суммарный объем ингредиентов, необходимый, чтобы сделать зелье, подходящее под требования профессора, по модулю $998\,244\,353$.

Пример

стандартный ввод	стандартный вывод
3 4 3 2 3 4 1 2 4 3 1 4 2 4 8 5 4 2 3 6 4 5 4 1 3 5 2 6 8 2 1 3 5 3 4 3 2 2 5 6 7 4 3 17 8 7 4 16 9 17 4 5 5 14 13 12 11 1 17 14 6 13 8 9 2 11 3 11 4 17 7 2 17 16 8 6 15 5 1 14 16 7 1 10 12 17 13 10 11 16 7 2 10 11 6 4 13 17 14 6 3 11 15 8 15 6 12 8	69 359 573672453

Замечание

В первом наборе входных данных минимальный объем ингредиентов равен 69. Действительно, объемы ингредиентов $1, 2, 3, 4$ должны быть равны $16, 12, 9, 32$, соответственно. Это зелье подходит, потому что

- Ингредиенты 3 и 2 имеют отношение $9 : 12 = 3 : 4$;
- Ингредиенты 1 и 2 имеют отношение $16 : 12 = 4 : 3$;
- Ингредиенты 1 и 4 имеют отношение $16 : 32 = 2 : 4$.

Во втором наборе объемы ингредиентов $1, 2, 3, 4, 5, 6, 7, 8$ в зелье минимального суммарного объема равны $60, 60, 24, 48, 32, 60, 45, 30$.

Задача E. Арифметические операции

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	5 секунд
Ограничение по памяти:	1024 мегабайта

Вам дан массив целых чисел $a_1, a_2, \ldots, a_n$.

Вы можете выполнить следующую операцию любое количество раз (возможно, ноль):

- Выбрать любой индекс i и поменять a_i на любое целое число (положительное, отрицательное или 0).

Какое минимальное количество операций необходимо, чтобы массив a стал арифметической прогрессией? Массив a является арифметической прогрессией, если $a_{i+1} - a_i = a_i - a_{i-1}$ для всех $2 \leqslant i \leqslant n-1$.

Формат входных данных

Первая строка содержит одно целое число n ($1 \leqslant n \leqslant 10^5$).

Вторая строка содержит n целых чисел $a_1, a_2, \ldots, a_n$ ($1 \leqslant a_i \leqslant 10^5$).

Формат выходных данных

Выведите одно целое число: минимальное количество операций, необходимое, чтобы сделать a арифметической прогрессией.

Примеры

стандартный ввод	стандартный вывод
9 3 2 7 8 6 9 5 4 1	6
14 19 2 15 8 9 14 17 13 4 14 4 11 15 7	10
10 100000 1 60000 2 20000 4 8 16 32 64	7
4 10000 20000 10000 1	2

Замечание

В первом примере можно получить массив $a = [11, 10, 9, 8, 7, 6, 5, 4, 3]$, выполнив 6:

- Поменять a_3 на 9: массив становится $[3, 2, 9, 8, 6, 9, 5, 4, 1]$;
- Поменять a_2 на 10: массив становится $[3, 10, 9, 8, 6, 9, 5, 4, 1]$;
- Поменять a_6 на 6: массив становится $[3, 10, 9, 8, 6, 6, 5, 4, 1]$;
- Поменять a_9 на 3: массив становится $[3, 10, 9, 8, 6, 6, 5, 4, 3]$;
- Поменять a_5 на 7: массив становится $[3, 10, 9, 8, 7, 6, 5, 4, 3]$;
- Поменять a_1 на 11: массив становится $[11, 10, 9, 8, 7, 6, 5, 4, 3]$.

Теперь a является арифметической прогрессией: действительно, $a_{i+1} - a_i = a_i - a_{i-1} = -1$ для всех $2 \leqslant i \leqslant n-1$.

Нет последовательности из менее чем 6 операций, которая делает a арифметической прогрессией.

Во втором примере можно получить массив $a = [-1, 2, 5, 8, 11, 14, 17, 20, 23, 26, 29, 32, 35, 38]$, сделав 10 операций.

В третьем примере можно получить массив $a = [100000, 80000, 60000, 40000, 20000, 0, -20000, -40000, -60000$
сделав 7 операций.

Задача F. Минимальное XOR-ирование строки

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	3 секунды
Ограничение по памяти:	512 мегабайт

Вам дано целое число n и строка s, состоящая из 2^n строчных букв латинского алфавита. Обозначим символы строки s как $s_0 s_1 s_2 \cdots s_{2^n-1}$.

Строка t длины 2^n (символы которой будем обозначать как $t_0 t_1 t_2 \cdots t_{2^n-1}$) называется *XOR-ированием* строки s, если существует целое число j ($0 \leqslant j \leqslant 2^n - 1$) такое, что для всех $0 \leqslant i \leqslant 2^n - 1$ выполняется $t_i = s_{i \oplus j}$ (здесь $\oplus$ обозначает операцию побитового исключающего ИЛИ).

Найдите лексикографически минимальное *XOR-ирование* строки s.

Строка a лексикографически меньше строки b, если и только если выполняется один из следующих пунктов:

- a — префикс b, но $a \neq b$;
- в первой позиции, где a и b различны, в строке a находится буква, которая встречается в алфавите раньше, чем соответствующая буква в b.

Формат входных данных

Первая строка содержит одно целое число n ($1 \leqslant n \leqslant 18$).

Вторая строка содержит строку s, состоящую из 2^n строчных букв латинского алфавита.

Формат выходных данных

Выведите одну строку, содержащую лексикографически минимальное XOR-ирование строки s.

Примеры

стандартный ввод	стандартный вывод
2 acba	abca
3 bcbaaabb	aabbbcba
4 bdbcbccdbdbaaccd	abdbdccacbdbdccb
5 ccfcffccccccffcfcfccffffffccccff	cccccffffccccffccfcffcccccfffff
1 zz	zz

Замечание

В первом примере лексикографически минимальное XOR-ирование t строки $s =$«`acba`» является «`abca`». Это XOR-ирование, так как для $j = 3$ получаем:

- $t_0 = s_{0 \oplus j} = s_3 =$ «`a`»;
- $t_1 = s_{1 \oplus j} = s_2 =$ «`b`»;
- $t_2 = s_{2 \oplus j} = s_1 =$ «`c`»;
- $t_3 = s_{3 \oplus j} = s_0 =$ «`a`».

Не существует XOR-ирования строки s лексикографически меньше, чем «`abca`».

Во втором примере лексикографически минимальное XOR-ирование соответствует выбору числа $j = 4$ из определения XOR-ирования.

В третьем примере лексикографически минимальное XOR-ирование соответствует выбору числа $j = 11$ из определения XOR-ирования.

В четвертом примере лексикографически минимальное XOR-ирование соответствует выбору числа $j = 10$ из определения XOR-ирования.

В пятом примере лексикографически минимальное XOR-ирование соответствует выбору числа $j = 0$ или числа $j = 1$ из определения XOR-ирования.

Задача G. Снежная гора

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	5 секунд
Ограничение по памяти:	1024 мегабайта

На горе, покрытой снегом, n отмеченных площадок, пронумерованных от 1 до n, соединенных $n-1$ трассой в форме дерева. Каждая трасса имеет длину 1. Некоторые из площадок являются базами. Высота h_i каждой площадки равна расстоянию до ближайшей базы (все базы имеют высоту 0).

На каждой площадке находится по одному лыжнику, изначально кинетическая энергия каждого лыжника равна 0. Каждый лыжник хочет съехать по наибольшему количеству трасс. Предположим, что лыжник едет по трассе от i-й площадки до j-й. Лыжники не могут двигаться вверх (т. е. невозможно, что $h_i < h_j$). При проезде горизонтального участка (т. е. если $h_i = h_j$) лыжник теряет одну единицу кинетической энергии, а при спуске (т. е. если $h_i > h_j$) лыжник получает одну единицу кинетической энергии. Для каждой площадки определите длину наибольшей последовательность трасс, начинающуюся с данной площадки, которую может проехать лыжник, не уменьшая свою кинетическую энергию ниже нуля. Лыжники могут посещать одну и ту же площадку и проезжать одну и ту же трассу несколько раз.

Формат входных данных

Первая строка содержит одно целое число n ($2 \leqslant n \leqslant 2 \cdot 10^5$).

Вторая строка содержит n целых чисел $l_1, l_2, \ldots, l_n$ ($0 \leqslant l_i \leqslant 1$). Если $l_i = 1$, площадка i является базой; если $l_i = 0$, площадка i не является базой. Гарантируется, что есть хотя бы 1 база.

Каждая из следующих $n-1$ строк содержит два целых числа u, v ($1 \leqslant u, v \leqslant n$, $u \neq v$), означающих, что есть трасса, соединяющая площадки u и v. Гарантируется, что данные трассы образуют дерево.

Формат выходных данных

Выведите n целых чисел: i-е число должно быть равно длине наибольшей последовательности трасс, которую может проехать лыжник, начав на площадке i, не уменьшая кинетическую энергию ниже нуля.

Примеры

стандартный ввод
6 1 1 0 0 0 0 1 3 2 4 3 4 4 5 5 6
стандартный вывод
0 0 1 1 3 5

стандартный ввод
9 0 0 0 0 0 0 1 1 1 1 3 2 3 2 5 3 6 4 5 4 7 5 8 6 9
стандартный вывод
5 3 2 1 1 1 0 0 0

стандартный ввод
14 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 2 2 5 3 4 4 5 3 6 4 8 5 9 7 8 6 11 7 12 8 13 9 14 10 11
стандартный вывод
8 5 4 3 2 2 1 1 1 0 0 0 0 0

стандартный ввод
20 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 1 0 1 17 3 11 12 6 10 18 19 8 14 16 20 5 3 2 11 7 10 2 15 8 3 3 15 9 16 7 13 16 1 19 2 2 16 6 1 4 17
стандартный вывод
2 2 1 5 3 4 8 1 2 6 4 6 10 0 0 0 3 0 1 0

Замечание

В первом примере $h = [0, 0, 1, 1, 2, 3]$. Лыжник, начинающий на площадке 6 может проехать максимум 5 трасс по пути $6 \to 5 \to 4 \to 3 \to 4 \to 2$ (обратите внимание, что лыжник может посещать одну и ту же площадку и проезжать одну и ту же трассу несколько раз):

- на площадке 6 кинетическая энергия равна 0;
- на площадке 5 кинетическая энергия увеличивается на 1 (т. к. $h_5 < h_6$) и становится равной 1;
- на площадке 4 кинетическая энергия увеличивается на 1 (т. к. $h_4 < h_5$) и становится равной 2;
- на площадке 3 кинетическая энергия уменьшается на 1 (т. к. $h_3 = h_4$) и становится равной 1;
- на площадке 4 кинетическая энергия уменьшается на 1 (т. к. $h_4 = h_3$) и становится равной 0;
- на площадке 2 кинетическая энергия увеличивается на 1 (т. к. $h_2 < h_4$) и становится равной 1.

Нет последовательности трасс длиной большей чем 5, на которой кинетическая энергия не становилась бы отрицательной.

Далее,

- оптимальный маршрут для лыжника, стартующего с площадки 1 — это 1 (ноль трасс);
- оптимальный маршрут для лыжника, стартующего с площадки 2 — это 2 (ноль трасс);
- оптимальный маршрут для лыжника, стартующего с площадки 3 — это $3 \to 1$;
- оптимальный маршрут для лыжника, стартующего с площадки 4 — это $4 \to 2$;
- оптимальный маршрут для лыжника, стартующего с площадки 5 — это $5 \to 4 \to 3 \to 1$.

Во втором примере $h = [3, 2, 2, 1, 1, 1, 0, 0, 0]$. Лыжник, начинающий на площадке 1 может проехать максимум 5 трасс по пути $1 \to 3 \to 2 \to 5 \to 4 \to 7$.

- на площадке 1 кинетическая энергия равна 0;
- на площадке 3 кинетическая энергия увеличивается на 1 (т. к. $h_3 < h_1$) и становится равной 1;
- на площадке 2 кинетическая энергия уменьшается на 1 (т. к. $h_2 = h_3$) и становится равной 0;
- на площадке 5 кинетическая энергия увеличивается на 1 (т. к. $h_5 < h_2$) и становится равной 1;
- на площадке 4 кинетическая энергия уменьшается на 1 (т. к. $h_4 = h_5$) и становится равной 0;
- на площадке 7 кинетическая энергия увеличивается на 1 (т. к. $h_7 < h_4$) и становится равной 1.

Нет последовательности трасс длиной большей чем 5, на которой кинетическая энергия не становилась бы отрицательной.

В третьем примере оптимальный маршрут для лыжника, начинающего на площадке 1, такой: $1 \to 2 \to 5 \to 4 \to 3 \to 6 \to 11 \to 10 \to 11$.

Ниже изображены первые три примера, базы отмечены красным: