

[Кар6502](#) | [Выйти](#)[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

D. Расстояние в дереве

ограничение по времени на тест: 3 seconds

ограничение по памяти на тест: 512 megabytes

Деревом называется связный граф, не содержащий циклов.

Расстоянием между двумя вершинами дерева называется длина (в ребрах) кратчайшего пути между этими вершинами.

Дано дерево из n вершин и положительное число k . Посчитайте количество различных пар вершин дерева, расстояние между которыми равно k . Обратите внимание, что пары (v, u) и (u, v) считаются одной и той же парой.

Входные данные

В первой строке записаны два целых числа n и k ($1 \leq n \leq 50000$, $1 \leq k \leq 500$) — количество вершин дерева и требуемое расстояние между вершинами.

В следующих $n - 1$ строках записаны ребра дерева в формате « $a_i b_i$ » (без кавычек) ($1 \leq a_i, b_i \leq n$, $a_i \neq b_i$), где a_i и b_i — вершины дерева, соединенные i -м ребром. Все заданные ребра различны.

Выходные данные

Выведите единственное целое число — количество различных пар вершин дерева, расстояние между которыми равно k .

Пожалуйста, не используйте спецификатор `%lld` для чтения или записи 64-х битовых чисел на C++. Рекомендуется использовать потоки `cin`, `cout` или спецификатор `%I64d`.

Примеры

входные данные	Скопировать
5 2 1 2 2 3 3 4 2 5	
выходные данные	Скопировать
4	

входные данные	Скопировать
5 3 1 2 2 3 3 4 4 5	
выходные данные	Скопировать
2	

Примечание

В первом примере парами вершин, расстояние между которыми равно 2, будут пары (1, 3), (1, 5), (3, 5) и (2, 4).

[Codeforces](#) (c) Copyright 2010-2025 Михаил Мирзаянов
Соревнования по программированию 2.0
Время на сервере: 09.04.2025 11:18:35^{UTC+5} (к3).
Мобильная версия, переключиться на [десктопную](#).
[Privacy Policy](#) | [Terms and Conditions](#)

При поддержке

**ИТМО**

[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

С. Юра и работа

ограничение по времени на тест: 1 секунда
ограничение по памяти на тест: 256 мегабайт

Совсем недавно вышел новый ITone 6, и Юра очень захотел себе его купить. К сожалению, денег у него не хватало, поэтому Юра устроился работать программистом. На работе Юра столкнулся со следующей задачей:

Задана последовательность из n чисел p_1, p_2, \dots, p_n . Нужно выбрать k пар целых чисел:

$[l_1, r_1], [l_2, r_2], \dots, [l_k, r_k]$ ($1 \leq l_1 \leq r_1 < l_2 \leq r_2 < \dots < l_k \leq r_k \leq n$; $r_i - l_i + 1 = m$),
так чтобы сумма $\sum_{i=1}^k \sum_{j=l_i}^{r_i} p_j$ была как можно больше. Помогите Юре справиться с этим заданием.

Входные данные

В первой строке содержится три целых числа n, m и k ($1 \leq (m \times k) \leq n \leq 5000$). Во второй строке содержится n целых чисел p_1, p_2, \dots, p_n ($0 \leq p_i \leq 10^9$).

Выходные данные

В единственной строке выведите целое число — максимальное значение суммы.

Примеры

входные данные	Скопировать
5 2 1 1 2 3 4 5	
выходные данные	Скопировать
9	

входные данные	Скопировать
7 1 3 2 10 7 18 5 33 0	
выходные данные	Скопировать
61	



[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

В. Прогулка по массиву

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт

Задан массив a_1, a_2, \dots, a_n , состоящий из n **положительных** целых чисел.
Изначально вы находитесь в позиции 1, и ваше количество очков равно a_1 . Можно совершать два типа шагов:

- 1. шаг вправо — перейти из текущей позиции x в $x + 1$ и получить a_{x+1} очков. Этот шаг можно делать только если $x < n$.
 - 2. шаг влево — перейти из текущей позиции x в $x - 1$ и получить a_{x-1} очков. Этот шаг можно делать только если $x > 1$.
- Также нельзя совершать два или более шагов влево подряд.

Вам требуется совершить **ровно** k шагов. Не более z из них могут быть шагами влево.
Какое наибольшее количество очков можно получить?

Входные данные

В первой строке записано одно целое число t ($1 \leq t \leq 10^4$) — количество наборов входных данных.
В первой строке каждого набора входных данных записаны три целых числа n, k и z ($2 \leq n \leq 10^5, 1 \leq k \leq n - 1, 0 \leq z \leq \min(5, k)$) — количество элементов в массиве, суммарное количество шагов, которое вы должны сделать, и максимальное количество шагов влево, которое вы можете сделать.
Во второй строке каждого набора входных данных записаны n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^4$) — данный массив.
Сумма n по всем наборам входных данных не превосходит $3 \cdot 10^5$.

Выходные данные

Выведите t целых чисел — для каждого набора входных данных выведите наибольшее количество очков, которое можно получить, если требуется сделать ровно k шагов, не более z из них могут быть шагами влево и не должно быть двух шагов влево подряд.

Пример

входные данные	Скопировать
4 5 4 0 1 5 4 3 2 5 4 1 1 5 4 3 2 5 4 4 10 20 30 40 50 10 7 3 4 6 8 2 9 9 7 4 10 9	
выходные данные	Скопировать
15 19 150 56	

Примечание

В первом наборе входных данных не разрешается ходить влево вообще. Поэтому делаем четыре шага вправо и получаем $a_1 + a_2 + a_3 + a_4 + a_5$ очков.
Во втором наборе входных данных можно сделать один ход влево. Тогда сделаем такие шаги: вправо, вправо, влево, вправо. Получится $a_1 + a_2 + a_3 + a_2 + a_3$ очков.
В третьем наборе входных данных можно ходить влево до четырех раз, но это все равно не оптимально, можем просто сделать четыре шага вправо и получить $a_1 + a_2 + a_3 + a_4 + a_5$ очков.

[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

A. Огромное дерево Parsa

ограничение по времени на тест: 1 секунда
ограничение по памяти на тест: 256 мегабайт

У Parsa есть огромное дерево на n вершинах.

На каждой вершине v он записал два целых числа l_v и r_v .

Чтобы дерево Parsa выглядело еще более величественным, Nima хочет назначить число a_v ($l_v \leq a_v \leq r_v$) для каждой вершины v таким образом, чтобы красота дерева Parsa была максимальной.

Восприятие красоты Nima довольно причудливо. Он определяет красоту дерева как сумму $|a_u - a_v|$ по всем ребрам (u, v) дерева.

Поскольку дерево Parsa слишком велико, Nima не может самостоятельно максимизировать его красоту. Ваша задача — найти **максимальную** возможную красоту для дерева Parsa.

Входные данные

Первая строка содержит целое число t ($1 \leq t \leq 250$) — количество наборов входных данных. Далее следует описание наборов входных данных.

Первая строка каждого набора входных данных содержит одно целое число n ($2 \leq n \leq 10^5$) — количество вершин в дереве Parsa.

В i -й из следующих n строк содержится два целых числа l_i и r_i ($1 \leq l_i \leq r_i \leq 10^9$).

Каждая из следующих $n - 1$ строк содержит по два целых числа u и v ($1 \leq u, v \leq n, u \neq v$), что обозначает наличие ребра между вершинами u и v в дереве Parsa.

Гарантируется, что данный граф является деревом.

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

Выходные данные

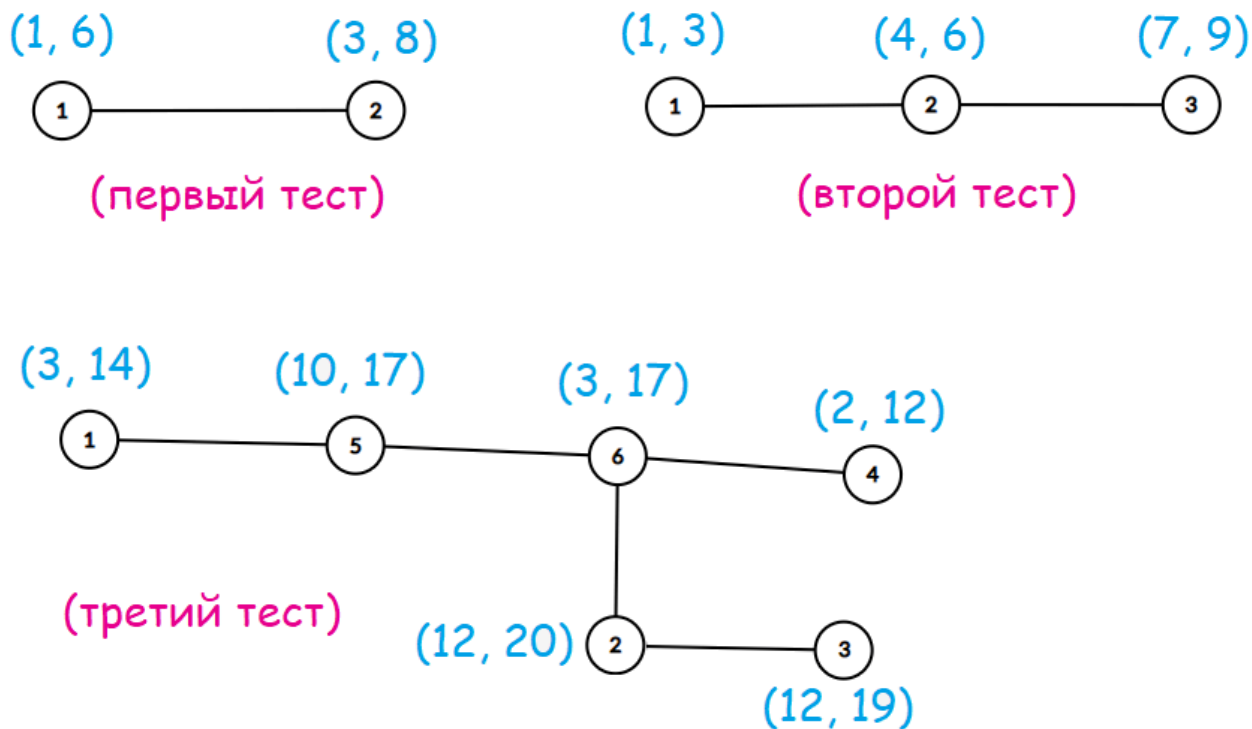
Для каждого набора входных данных выведите **максимальную** возможную красоту для дерева Парса.

Пример

входные данные	Скопировать
3 2 1 6 3 8 1 2 3 1 3 4 6 7 9 1 2 2 3 6 3 14 12 20 12 19 2 12 10 17 3 17 3 2 6 5 1 5 2 6 4 6	
выходные данные	Скопировать
7 8 62	

Примечание

Деревья в примере:



В первом наборе входных данных одно из возможных назначений — $a = \{1, 8\}$, что приводит к $|1 - 8| = 7$.

Во втором наборе входных данных одно из возможных назначений — $a = \{1, 5, 9\}$, что приводит к красоте $|1 - 5| + |5 - 9| = 8$.

[Codeforces](#) (c) Copyright 2010-2025 Михаил Мирзаянов
Соревнования по программированию 2.0
Время на сервере: 09.04.2025 11:18:43^{UTC+5} (к3).
Мобильная версия, переключиться на [десктопную](#).
[Privacy Policy](#) | [Terms and Conditions](#)

При поддержке



ІТМО

[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

D. Сделай равными

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт

У вас есть массив целых чисел a размера n . Изначально все элементы массива равны 1. Вы можете выполнять операцию следующего вида: выбрать два целых числа i ($1 \leq i \leq n$) и x ($x > 0$), а затем увеличить значение a_i на $\lfloor \frac{a_i}{x} \rfloor$ (т.е. сделать $a_i = a_i + \lfloor \frac{a_i}{x} \rfloor$).

После выполнения всех операций вы получите c_i монет для тех i , в которых $a_i = b_i$.

Ваша задача — определить максимальное количество монет, которое вы можете получить выполнив не более k операций.

Входные данные

Первая строка содержит одно целое число t ($1 \leq t \leq 100$) — количество наборов входных данных.

Первая строка каждого набора содержит два целых числа n и k ($1 \leq n \leq 10^3; 0 \leq k \leq 10^6$) — размер массива и максимальное количество операций, соответственно.

Вторая строка содержит n целых чисел b_1, b_2, \dots, b_n ($1 \leq b_i \leq 10^3$).

Третья строка содержит n целых чисел c_1, c_2, \dots, c_n ($1 \leq c_i \leq 10^6$).

Сумма n по всем наборам входных данных не превосходит 10^3 .

Выходные данные

Для каждого набора входных данных выведите одно целое число — максимальное количество монет, которое вы можете получить выполнив не более k операций.

Пример

входные данные	Скопировать
4 4 4 1 7 5 2 2 6 5 2 3 0 3 5 2 5 4 7 5 9 5 2 5 6 3 5 9 1 9 7 6 14 11 4 6 2 8 16 43 45 9 41 15 38	
выходные данные	Скопировать
9 0 30 167	

