0_送分題 (Hello World)

(30分)

時間限制: 1 second
記憶體限制: 256 MB

前言

比賽開始了！

趕快驗證一下，

網路是否設定正確?

上傳競賽程式是否順利?

程式解答是否用 STDOUT 輸出？

都沒問題，30分就到手了！ 繼續 ... 衝！衝！衝！

題目敘述

請寫一個程式輸出Hello World!

輸入格式

本題無需輸入值

輸出格式

[A~Z][a~z]、空格，以及常用英文符號。

資料範圍

[A~Z][a~z]、空格，以及驚嘆號 “!”

測試範例

輸入範例 1

(無輸入值)

輸出範例 1

```
Hello World!
```

範例說明

輸入範例1, 無輸入值，簡單而快樂的輸出Hello World!

0_Hello World

(30 points)

Time Limit: 1 second
Memory Limit: 256MB

Introduction

YTP Contest has started!

Let’s verify everything first.

Is the internet setting correct?

Is the source code submission working well?

Do you use STDOUT output for program solutions?

Everything is ready! Go get 30 points now!! Go! Go! Go!

Statement

Please write a program to output Hello World!

Input Format

This problem requires no input.

Output Format

[A~Z][a~z], space, and common English punctuation.

Constraints

[A~Z][a~z], space, and exclamation mark "!".

Test Cases

Input 1

(no input)

Output 1

```
Hello World!
```

Illustrations

Input 1 has no input, simply output Hello World!

1_一級棒 (Top of the Class)

(5 分)

時間限制: 1 second

記憶體限制: 256 MB

題目敘述

給定一個班級所有學生的成績，請你輸出成績最高的同學的名字。

輸入格式

第一行為一個正整數 N，代表有 N 位同學。
接下來 N 行，每行為一個字串 S_i 和一個整數 T_i ，中間由一個空格分開，代表同學的名字和他的分數。

輸出格式

輸出一個字串，代表得分最高的同學的名字。

資料範圍

- $1 \leq N \leq 10^5$
- $1 \leq |S_i| \leq 10$，其中 $|S_i|$ 代表該名字的長度。
- $1 \leq T_i \leq 10^9$
- 名字只由小寫英文字母組成（不包含空白）。
- 保證所有分數相異。

測試範例

範例輸入 1

```
5
adam 5
ben 3
cherry 9
daniel 8
ethreal 4
```

範例輸出 1

```
cherry
```

1_Top of the Class

(5 points)

Time Limit: 1 second

Memory Limit: 256 MB

Statement

Given the grades of students, please output the name of the student with the highest grade.

Input Format

The first line contains a positive integer N, representing the number of classmates.
The next N lines each contain a string S_i and an integer T_i, separated by a space, representing each student's name and their score.

Output Format

Output the name of the student with the highest score.

Constraints

- $1 \leq N \leq 10^5$
- $1 \leq |S_i| \leq 10$, where $|S_i|$ represents the length of the name
- $1 \leq T_i \leq 10^9$
- Name only contains lowercase English letters (does not include white spaces).
- All scores are distinct.

Test Cases

Input 1

```
5
adam 5
ben 3
cherry 9
daniel 8
ethreal 4
```

Output 1

```
cherry
```

2_魔杖 (Wand)

(2 分/8 分)

時間限制: 1.5 seconds

記憶體限制: 512MB

題目敘述

在奧利凡德魔杖商店中，有 n 種不同的樹枝，第 i 種樹枝有個相容值 a_i，並且每種樹枝都有無限個。

一根魔杖由兩根樹枝所組成，並且這兩根樹枝可以是同一種，也可以是不同種。

魔杖會有個穩定值。一根由第 i 種和第 j 種的樹枝所製作而成的魔杖，其穩定值會是 $a_i \times a_j$。

太穩定不好、太不穩定也不好。根據研究，穩定值越接近 k 越好。

你，身為一個魔法學院的新生，現在正在奧利凡德魔杖商店內準備購買自己未來三年所使用的魔杖。

請問你能夠組成多好的魔杖；換言之，請你找到 $\min_{1 \le i,j \le n} |k - a_i \times a_j|$。

請注意，在奧利凡德魔杖商店中，樹枝是按照一種特殊方式來排序的。保證存在某個數字 p 滿足 $1 \le p \le n$，使得 $a_1 \le a_2 \le \cdots \le a_p$ 並且 $a_p \ge a_{p+1} \ge \cdots \ge a_n$。也就是說 $a[1 \ldots p]$ 是非嚴格遞增、而 $a[p \ldots n]$ 是非嚴格遞減的。

輸入格式

輸入的第一行包含兩個正整數 n, k，代表共有幾種不同的樹枝、還有最好的穩定值。

輸入的第二行包含 n 個非負整數 $a_1, a_2, \ldots, a_n$，其中 a_i 代表第 i 種樹枝的相容值。

保證存在某個數字 p 滿足 $1 \le p \le n$，使得$a[1 \ldots p]$ 是非嚴格遞增、而 $a[p \ldots n]$ 是非嚴格遞減的。

輸出格式

請輸出一個整數，代表 $\min_{1 \le i,j \le n} |k - a_i \times a_j|$。

資料範圍

- $1 \le n \le 10^6$。
- $0 \le k \le 10^{18}$。
- $0 \le a_i \le 10^9$ for $i = 1, 2, \ldots, n$。

子任務

- 子任務 1 滿足 $n \le 1000$。
- 子任務 2 無額外限制。

測試範例

輸入範例 1

```
5 5
1 2 3 2 1
```

輸出範例 1

```
1
```

輸入範例 2

```
5 4
1 2 3 2 1
```

輸出範例 2

```
0
```

輸入範例 3

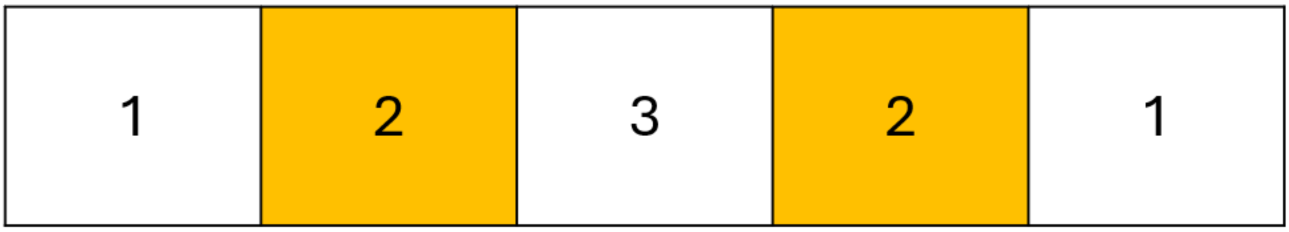
```
4 5
1 2 3 4
```

輸出範例 3

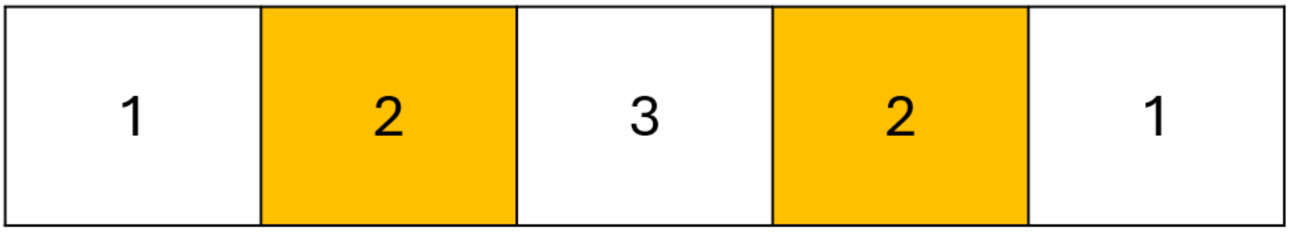
```
1
```

範例說明

在範例 1 中，你可以使用第 2 種和第 4 種的樹枝組成一根穩定值為 $2 \times 2 = 4$ 的魔杖，$|k-4| = 1$ 是最佳的。

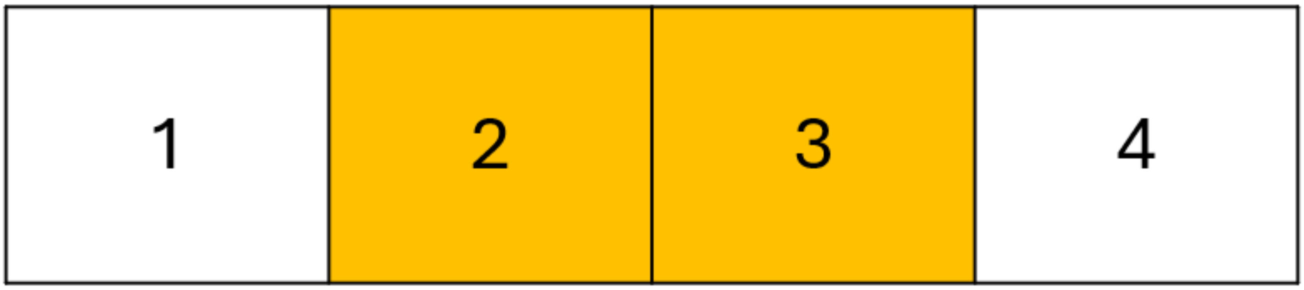


在範例 2 中，你可以使用第 2 種和第 4 種的樹枝組成一根穩定值為 $2 \times 2 = 4$ 的魔杖，$|k-4| = 0$ 是最佳的。



1	2	3	2	1

在範例 3 中，你可以使用第 2 種和第 3 種的樹枝組成一根穩定值為 $2 \times 3 = 6$ 的魔杖，$|k - 5| = 1$ 是最佳的。



1	2	3	4

2_Wand

(2 points/8 points)

Time Limit: 1.5 seconds

Memory Limit: 512 MB

Statement

In the Wand Emporium of Ollivander, there are n different types of branches. The i-th type of branch has a compatibility value a_i, and each type of branch has an infinite supply.

A wand is composed of two branches, which can be of the same type or different types.

Each wand has a stability value. The stability value of a wand made from the i-th and j-th types of branches is $a_i \times a_j$.

Too stable is not good, and too unstable is also not good. According to research, the closer the stability value is to k, the better.

You, as a new student at the School of Magic, are currently in Ollivander's Wand Emporium preparing to purchase wands for your next three years of use.

What is the best wand you can assemble? In other words, find $\min_{1 \le i,j \le n} |k - a_i \times a_j|$.

Please note that in Ollivander's Wand Emporium, branches are sorted in a special way. It is guaranteed that there exists a number p satisfying $1 \le p \le n$, such that $a_1 \le a_2 \le \cdots \le a_p$ and $a_p \ge a_{p+1} \ge \cdots \ge a_n$. In other words, $a[1 \ldots p]$ is non-descending, and $a[p \ldots n]$ is non-increasing.

Input Format

The first line of input contains two positive integers n and k, representing the number of different types of branches and the desired stability value.

The second line contains n non-negative integers $a_1, a_2, \ldots, a_n$, where a_i represents the compatibility value of the i-th type of branch.

It is guaranteed there exists a number p satisfying $1 \le p \le n$, such that $a[1 \ldots p]$ is non-descending, and $a[p \ldots n]$ is non-increasing.

Output Format

Output an integer, representing $\min_{1 \le i,j \le n} |k - a_i \times a_j|$.

Constraints

- $1 \le n \le \times 10^6$.
- $0 \le k \le 10^{18}$.
- $0 \le a_i \le 10^9$ for $i = 1, 2, \ldots, n$.

Subtasks

- Subtask 1 satisfies that $n \leq 1000$.
- Subtask 2 has no extra constraints.

Test Cases

Input 1

```
5 5
1 2 3 2 1
```

Output 1

```
1
```

Input 2

```
5 4
1 2 3 2 1
```

Output 2

```
0
```

Input 2

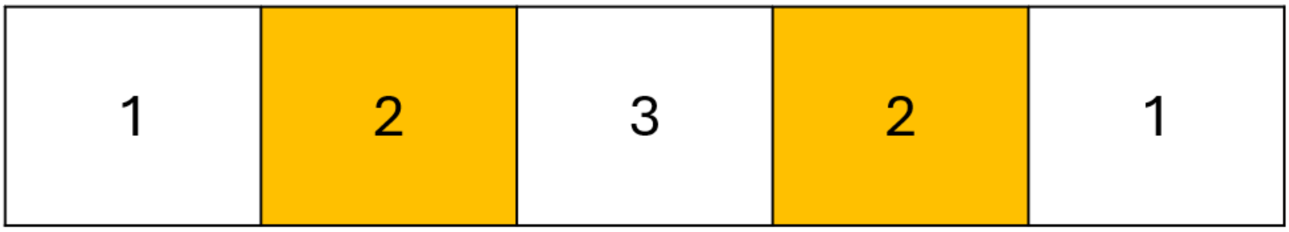
```
4 5
1 2 3 4
```

Output 2

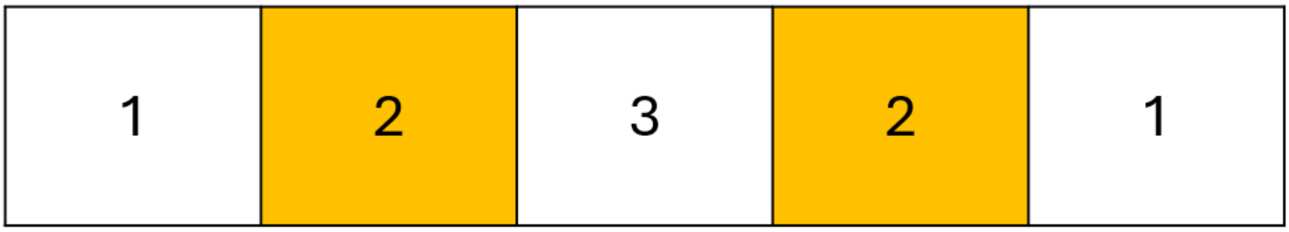
```
1
```

Illustrations

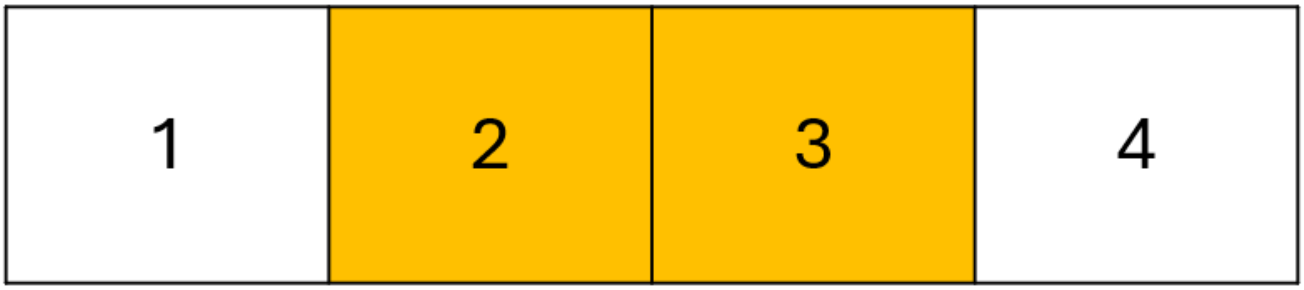
In Example 1, you can use the 2nd and 4th types of branches to make a wand with a stability value of $2 \times 2 = 4$, where $|k - 4| = 1$ is the best.



In Example 2, you can use the 2nd and 4th types of branches to make a wand with a stability value of $2 \times 2 = 4$, where $|k - 4| = 0$ is the best.



In Example 3, you can use the 2nd and 3rd types of branches to make a wand with a stability value of $2 \times 3 = 6$, where $|k - 5| = 1$ is the best.



3_電車難題(The Trolley Dilemma)

(10分)

時間限制: 1 second
記憶體限制: 256 MB

題目敘述

YTP 國有一個電車軌道系統，我們可以用一個 N 個點的有根樹來形容這個系統，樹上的每個節點代表著一個中繼站或是終點站，而節點和節點之間的邊代表連接兩者的軌道。節點 1 是這個有根樹的根節點，每個節點會有若干個子節點，如果一個節點沒有子節點，則稱它為一個終點站，否則稱它為一個中繼站。

每個中繼站必須選擇恰好一條連接它和子節點的軌道作為**主要軌道**，一台電車會從根節點一路行駛主要軌道直到抵達某個終點站為止。一開始每個中繼站已經選擇了它們目前的主要軌道。

YTP 國的氣球倫理學家 Joylintp 提出了以下的 K-電車難題：現在每條軌道上各綁了若干顆氣球，當電車行駛過一個軌道時，軌道上的氣球會全部破掉。如果能改變最多 K 個中繼站所選的主要軌道，該怎麼做比較好？

因為氣球倫理道德實在是太複雜了，所以這裡我們考慮一個簡單的目標：最少化破掉的氣球個數。

對於所有 $K = 0, 1, 2, \ldots, N-1$，請回答在 K-電車難題中，破掉的氣球個數最少能是多少？

輸入格式

第一行輸入一個正整數 N，代表電車軌道系統的節點數。

接下來輸入 $N-1$ 行，每一行會輸入四個整數 u, v, w，代表節點 u 有子節點 v，它們之間的軌道上有 w 顆氣球。

最後輸入一行，這行有 N 個整數 $m_1, m_2, \ldots, m_N$。如果節點 i 為終點站，則 $m_i = 0$，否則節點 i 為中繼站且它的主要軌道為節點 i, m_i 之間的軌道。

輸出格式

輸出 N 行，第 i 行輸出一個整數，代表當 $K = i - 1$ 時，在 K-電車難題中，破掉的氣球個數最小值。

資料範圍

- $2 \leq N \leq 2 \times 10^5$
- $1 \leq u, v \leq N$
- $u \neq v$
- $0 \leq w \leq 10^9$
- $0 \leq m_i \leq N$
- 保證輸入的所有軌道形成一個以節點 1 為根節點的有根樹
- 如果 $m_i = 0$，則節點 i 一定是終點站
- 如果 $m_i > 0$，則節點 i, m_i 之間的軌道一定存在
- 保證每個中繼點恰有一個主要軌道

測試範例

輸入範例 1

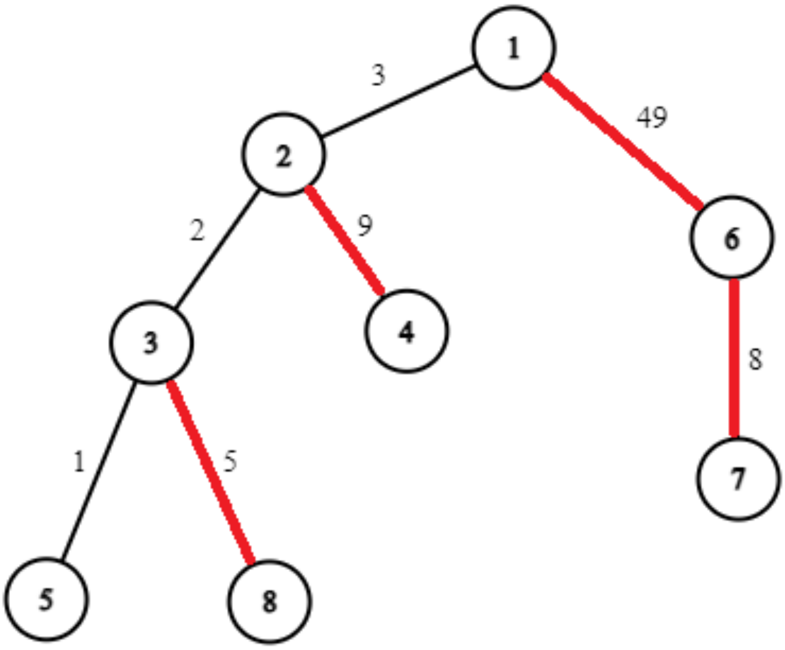
```
8
1 2 3
2 3 2
2 4 9
3 5 1
1 6 49
6 7 8
3 8 5
6 4 8 0 0 7 0 0
```

輸出範例 1

```
57
12
10
6
6
6
6
6
```

範例說明

以下是測試範例的樹狀結構，其中紅色的邊為主要軌道。



$K = 0$：電車一定是經過節點 $1 \to 6 \to 7$。

$K = 1$：將中繼站 1 的主要軌道換成和 2 之間的軌道後，電車會經過節點 $1 \to 2 \to 4$。

$K = 2$：將中繼站 $1, 2$ 的主要軌道都換成另一個軌道後，電車會經過節點 $1 \to 2 \to 3 \to 8$。

$K \geq 3$：將中繼站 $1, 2, 3$ 的主要軌道都換成另一個軌道後，電車會經過節點 $1 \to 2 \to 3 \to 5$。

3_The Trolley Dilemma

(10 points)

Time Limit: 1 second
Memory Limit: 256MB

Statement

YTP Nation has a trolley track system that can be represented as a rooted tree with N nodes. Each node in the tree represents either a relay station or a terminal station, and the edges between the nodes represent the tracks connecting them. Node 1 is the root of this tree. Each node may have several child nodes. If a node has no child nodes, it is called a terminal station, otherwise, it is called a relay station.

Each relay station must select exactly one track connecting it to one of its child nodes as the **main track**. A trolley will travel from the root node along the main tracks until it reaches a terminal station. Initially, each relay station has already selected its current main track.

Joylintp, a balloon ethicist from YTP Nation, proposed the following K-trolley dilemma: each track has a certain number of balloons tied to it. If a trolley travels along a track, all balloons on that track will be popped. If it is possible to change the main track of up to K relay stations, what is the best way to do so?

Since balloon ethics are too complex, we consider a simple objective here: minimize the number of popped balloons.

For all $K = 0, 1, 2, \ldots, N-1$, determine the minimum number of popped balloons in the K-trolley dilemma.

Input Format

The first line contains a positive integer N, representing the number of nodes in the trolley track system.

The next $N-1$ lines each contain four integers u, v, w, indicating that node u has a child node v, and there are w balloons tied on the track between them.

The final line contains N integers $m_1, m_2, \ldots, m_N$. If node i is a terminal station, then $m_i = 0$; otherwise, node i is a relay station and its main track is the track between node i and node m_i.

Output Format

Output N lines. The i-th line should contain a single integer, representing the minimum number of popped balloons when $K = i - 1$ in the K-trolley dilemma.

Constraints

- $2 \leq N \leq 2 \times 10^5$
- $1 \leq u, v \leq N$
- $u \neq v$
- $0 \leq w \leq 10^9$

- $0 \leq m_i \leq N$
- It is guaranteed that the input tracks form a rooted tree with node 1 as the root node.
- If $m_i = 0$, then node i is guaranteed to be a terminal station.
- If $m_i > 0$, then the track between node i and node m_i is guaranteed to exist.
- Each relay station has exactly one main track.

Test Cases

Input 1

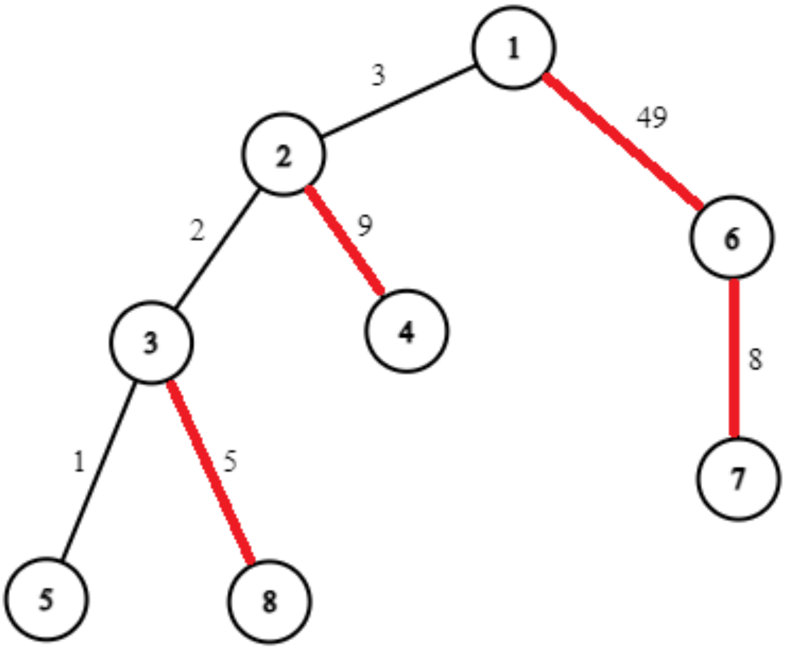
```
8
1 2 3
2 3 2
2 4 9
3 5 1
1 6 49
6 7 8
3 8 5
6 4 8 0 0 7 0 0
```

Output 1

```
57
12
10
6
6
6
6
6
```

Illustrations

Below is the tree structure of the sample testcase. The initial main tracks are marked in red.



$K = 0$: The trolley must travel through nodes $1 \to 6 \to 7$.

$K = 1$: After changing the main track of relay station 1 to the track between 1 and 2, the trolley will travel through nodes $1 \to 2 \to 4$.

$K = 2$: After changing the main tracks of relay stations 1 and 2 to other tracks, the trolley will travel through nodes $1 \to 2 \to 3 \to 8$.

$K \geq 3$: After changing the main tracks of relay stations 1, 2, and 3 to other tracks, the trolley will travel through nodes $1 \to 2 \to 3 \to 5$.

4_火氣很大的競賽人(Fiery Contestants)

(15分)

時間限制: 1 second
記憶體限制: 256 MB

題目敘述

有 N 個火氣很大的競賽人，編號 $1, 2, \ldots, N$，由左而右排成一排。

這 N 個人都有在玩鳴神和原潮兩款遊戲，競賽人 i 在鳴神的遊戲等級是 a_i，在原潮的遊戲等級是 b_i。

這些競賽人有一個討厭的毛病，那就是他們會看遊戲等級嘲諷人。一個競賽人的鳴神和原潮遊戲等級如果都**大於等於**他向右看第一個競賽人的鳴神和原潮遊戲等級，那他會嘲諷向右看的第一個競賽人。如果一個競賽人右邊沒有其他人，那他不會嘲諷任何人。注意到每個競賽人只會在乎在他右邊的第一個競賽人，不會去嘲諷更右邊的人。

你的任務是把一些人從排隊隊伍中移除，使得留在隊伍中的人不會嘲諷別人，並最大化留下來的人數。

輸入格式

第一行輸入一個正整數 T，代表子測試資料的個數。

每一筆子測試資料會輸入三行。

第一行輸入一個正整數 N，代表競賽人人數。

第二行輸入 N 個正整數 $a_1, a_2, \ldots, a_N$，代表 N 個競賽人在鳴神的遊戲等級。

第三行輸入 N 個正整數 $b_1, b_2, \ldots, b_N$，代表 N 個競賽人在原潮的遊戲等級。

輸出格式

資料範圍

- $1 \leq T \leq 10000$
- $1 \leq N \leq 3 \times 10^5$
- 保證 N 的總和不超過 3×10^5
- $1 \leq a_i \leq 10^9$
- $1 \leq b_i \leq 10^9$

測試範例

輸入範例 1

```
3
3
1 2 3
6 5 4
5
1 3 1 2 3
2 2 1 3 1
5
1 1 1 1 1
5 4 3 2 1
```

輸出範例 1

```
3
4
1
```

範例說明

第一個子測試資料中，所有人都能留下來。

第二個子測試資料中，可以只移除競賽人 3。

第三個子測試資料中，不可能留下超過一個人，但可以留下任何一個人。

4_Fiery Contestants

(15 points)

Time Limit: 1 second
Memory Limit: 256MB

Statement

There are N fiery contestants, numbered $1, 2, \ldots, N$, standing in a line from left to right.

Each of these N contestants plays two games: Wuthering Shin and Gen Waves. Contestant i is level a_i in Wuthering Shin and level b_i in Gen Waves.

These contestants have an annoying habit of taunting others based on their game levels. If a contestant's levels in both Wuthering Shin and Gen Waves are **greater than or equal to** the levels of the first contestant to his right, he will taunt that first contestant to his right. If a contestant has no one to his right, he will not taunt anyone. Note that each contestant only cares about the first contestant to his right and will not taunt anyone further to the right.

Your task is to remove some contestants from the line so that no one remaining will taunt anyone else, while maximizing the number of contestants left in the line.

Input Format

The first line contains a positive integer T, representing the number of test cases.

Each test case consists of three lines:

The first line contains a positive integer N, representing the number of contestants.

The second line contains N positive integers $a_1, a_2, \ldots, a_N$, representing the Wuthering Shin levels of the contestants.

The third line contains N positive integers $b_1, b_2, \ldots, b_N$, representing the Gen Waves levels of the contestants.

Output Format

For each test case, output a single line containing the maximum number of contestants that can remain without any taunting occurring.

Constraints

- $1 \leq T \leq 10000$
- $1 \leq N \leq 3 \times 10^5$
- The sum of N over all test cases does not exceed 3×10^5
- $1 \leq a_i \leq 10^9$
- $1 \leq b_i \leq 10^9$

Test Cases

Input 1

```
3
3
1 2 3
6 5 4
5
1 3 1 2 3
2 2 1 3 1
5
1 1 1 1 1
5 4 3 2 1
```

Output 1

```
3
4
1
```

Illustrations

For the first test case, all contestants can stay without any taunting.

For the second test case, only contestant 3 needs to be removed.

For the third test case, it's impossible to keep more than one contestant, but any one contestant can be kept.

5_賭博遊戲 (Gamble)

(15 分)

時間限制: 1 second

記憶體限制: 256 MB

題目敘述

在一個熱鬧的賭場裡，Alice 和 Bob 兩位勇敢的賭徒坐下來進行一場刺激的賭博。他們面對著賭桌上的籌碼，心跳加速，期待著下一局的結果。 Alice 目光炯炯，手中的籌碼穩穩地擺放在賭桌上，而 Bob 則神情凝重，帶著一絲不安地等待著。這場賭局的結果將決定誰能笑到最後，誰將空著口袋離開這裡。

現在，Alice 和 Bob 各自持有一定數量的賭注。Alice 手中擁有 m 元，而 Bob 則擁有 n 元。他們決定將這些賭注全部拿出來進行一場刺激的對決。然而，這並不僅僅是一場賭博，更是一場關於勇氣、智慧和運氣的角逐。

賭局開始了。Alice 和 Bob 交替進行著他們的下注，每一輪都緊張刺激。已知 Alice 在每一場贏的機率為 $p\%$，Bob 贏的機率則為 $q\%$，其中 $q = 100 - p$。在每一次的賭注中，如果 Alice 贏了，Bob 就得支付給 Alice 一元；反之，Alice 則支付給 Bob 一元。這樣的規則讓整個賭場都充滿了緊張與刺激的氣氛。

當雙方其中一個輸光所有賭注，則賭博結束。舉例來說，Alice 贏光了 Bob，也就是 Alice 拿到 $m + n$ 元，Bob 則剩下 0 元，該場賭博結束。

現在已知 Alice 和 Bob 分別有多少賭注，請問 Alice 把 Bob 贏光的機率是多少？可以證明答案可以用最簡分數 $\frac{a}{b}$ 表示，請以 $a \times b^{-1} \mod 998244353$ 的格式輸出，其中 b^{-1} 為 b 在 998244353 的模逆元。

輸入格式

輸入僅一行，包含三個正整數 m、n、p，分別代表 Alice 和 Bob 的賭注以及 Alice 獲勝的機率。

輸出格式

輸出一個數字如題目所述。

資料範圍

- $1 \le m, n \le 3 \times 10^5$。
- $0 \le p \le 100$。

測試範例

輸入範例 1

```
1 1 40
```

輸出範例 1

```
199648871
```

輸入範例 2

```
5 5 0
```

輸出範例 2

```
0
```

輸入範例 3

```
5 5 40
```

輸出範例 3

```
747775770
```

範例說明

在範例 1 中，很明顯可以得到 Alice 贏光 Bob 的機率是 40%（贏一場即贏光且 Alice 在一場中獲勝的機率為 40%），也就是 $\frac{2}{5}$。所以在 998244353 的模數中，$2 \times 5^{-1} \mod 998244353$ 的結果為 199648871。

在範例 2 中，Alice 每一場的獲勝機率為 $0\% = 0$，因此贏光 Bob 的機率也就是 $0\% = 0$。在 998244353 的模數中，結果是 0。

5_Gamble

(15 points)

Time Limit: 1 second

Memory Limit: 256 MB

Statement

In a bustling casino, Alice and Bob, two brave gamblers, sit down for an exciting gamble. Facing the chips on the gambling table, their hearts beat faster, anticipating the outcome of the next round. Alice's gaze is sharp, her chips placed steadily on the table, while Bob looks serious, waiting with a hint of anxiety. The outcome of this gamble will determine who can laugh last and who will leave with empty pockets.

Now, Alice and Bob each hold a certain amount of money. Alice has m dollars in hand, while Bob has n dollars. They decide to put all these money on the line for an exciting showdown. However, this is not just a gamble, but a contest of courage, intelligence, and luck.

The gamble begins. Alice and Bob take turns placing their money, each round nerve-wracking. It is known that Alice's probability of winning each round is $p\%$, and Bob's probability of winning is $q\%$, where $q = 100 - p$. In each bet, if Alice wins, Bob pays Alice one dollar; conversely, if Bob wins, Alice pays Bob one dollar. This rule fills the entire casino with tension and excitement.

When one side loses all their money, the gamble ends. For example, if Alice wins all of Bob's money, Alice gets $m + n$ dollars, and Bob is left with 0 dollars, and the gamble ends.

Now, knowing the number of money Alice and Bob have respectively, what is the probability that Alice will win all of Bob's money? It can be proven that the answer can be represented by the irreducible fraction $\frac{a}{b}$, and please output it in the format $a \times b^{-1} \mod 998244353$, where b^{-1} is the modular inverse of b modulo 998244353.

Input Format

The input consists of a single line containing three positive integers m, n, and p, representing the bets of Alice and Bob, and the probability of Alice winning, respectively.

Output Format

Output a number as described in the problem statement.

Constraints

- $1 \le m, n \le 3 \times 10^5$.
- $0 \le p \le 100$.

Test Cases

Input 1

```
1 1 40
```

Output 1

```
199648871
```

Input 2

```
5 5 0
```

Output 2

```
0
```

Input 3

```
5 5 40
```

Output 3

```
747775770
```

Illustrations

In Example 1, it is obvious that the probability of Alice winning all of Bob's money is 40% (winning a game means winning all, and Alice's probability of winning a game is 40%), which is $\frac{2}{5}$. So, the result of $2 \times 5^{-1} \mod 998244353$ is 199648871 modulo 998244353.

In Example 2, Alice's probability of winning each round is $0\% = 0$, so the probability of winning all of Bob's money is also $0\% = 0$. The result modulo 998244353 is 0.

6_植樹問題 (Tree Planting Problem)

(4 分/4 分/12 分)

時間限制: 6 seconds

記憶體限制: 1024 MB

題目敘述

最近一位資工系的學生 Chyen 愛上了植樹，於是他上網買了一批種子，希望能種出一棵美麗的樹。

根據說明書，這些種子能種出一棵有著 N 個節點的樹，節點編號為 1 到 N ，有 $N-1$ 條邊連接著這 N 個點，使得 N 個點是連通的，且第 i 個點上有一個固定的點權 C_i。

Chyen 用買來的種子嘗試種植了很多次，但每次都不能種出和說明書上一模一樣的樹，只能種植出說明書上的樹的一部分，更具體的說，Chyen 種出的樹會包含 1 到 N 的某些節點（至少一個點），並且這些節點在說明書上的樹上是連通的（也就是說明書上的樹的一個連通塊）。

Chyen 是一個熱愛數字的人，他非常在意每棵樹上的點權總和。並且他很好奇，在所有他可能種出的相異的樹中，點權總和第 K 小的樹點權總和為多少？

請你寫一個程式幫助 Chyen 來解決這個問題。如果無法種出 K 種不一樣的樹，請輸出 "-1" 來告訴 Chyen。

注意：Chyen 可能可以種出多棵點權一樣的樹，但只要這棵樹包含的節點（1 到 N 的某個非空子集i）和另一棵樹不同，這樣就視為相異的樹。

輸入格式

輸入的第一行包含兩個正整數 N、K，代表樹的節點數量、Chyen 想找出第幾小的點權總和。

第二行包含 N 正整數 $C_1, C_2, \ldots, C_N$，代表每個點的點權。

接下來的 $N-1$ 行，每行包含兩個正整數 u, v 代表樹上的一條邊。

輸出格式

輸出一個整數，代表第 K 小的樹點權總和。如果無法種出 K 種不一樣的樹，請輸出 "-1"。

資料範圍

- $1 \leq N \leq 2 \times 10^5$
- $1 \leq K \leq 2 \times 10^5$
- $1 \leq C_i \leq 10^9$ $(\forall i \in [1, N])$

子任務

- 子任務 1 滿足 $1 \leq N, K \leq 200$。
- 子任務 2 滿足 給定的樹是一條鏈。
- 子任務 3 無額外限制。

測試範例

輸入範例 1

```
5 10
1 1 1 1 1
1 2
2 3
3 4
4 5
```

輸出範例 1

```
3
```

輸入範例 2

```
7 20
8 2 1 7 1 10 2
3 7
7 6
6 1
3 5
1 2
1 4
```

輸出範例 2

```
21
```

輸入範例 3

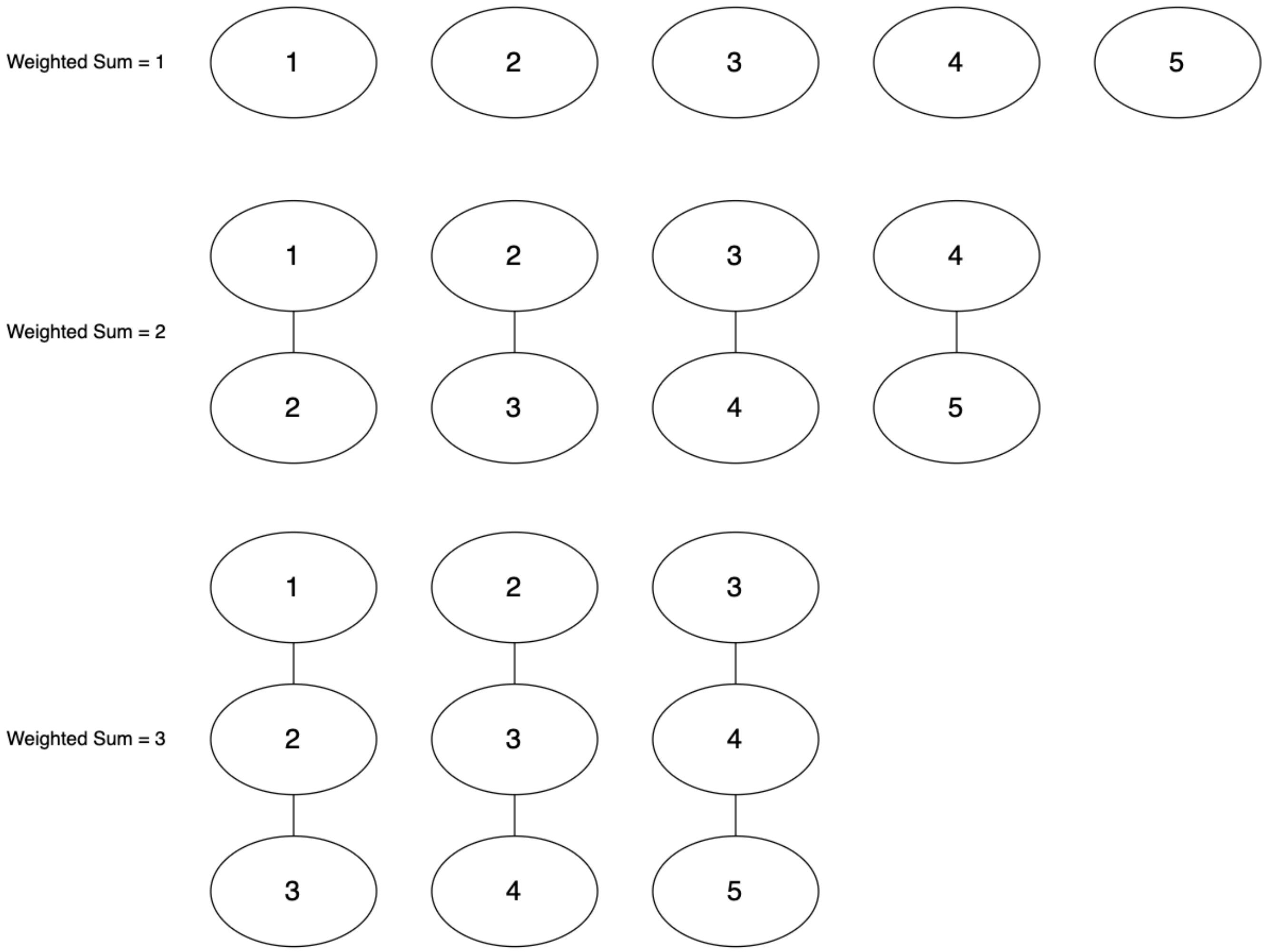
```
2 100
1 1
1 2
```

輸出範例 3

```
-1
```

範例說明

在範例 1 中，前 10 小的點權總和依序為：$[1, 1, 1, 1, 1, 2, 2, 2, 2, 3]$，下圖列出點權總和 ≤ 3 的連通塊。


Weighted Sum = 1
1
2
3
4
5
Weighted Sum = 2
1
2
3
4
2
3
4
5
Weighted Sum = 3
1
2
3
2
3
4
3
4
5

6_Tree Planting Problem

(4 points/4 points/12 points)

Time Limit: 6 seconds

Memory Limit: 1024 MB

Statement

Recently, a computer science student named Chyen has developed a passion for planting trees. He ordered a batch of seeds online, hoping to grow a beautiful trees.

According to the manual, these seeds can grow into a tree with N nodes, numbered from 1 to N, with $N-1$ edges connecting these nodes to make them connected. Each node i has a fixed point weight C_i.

Chyen has tried planting multiple times but has only been able to grow part of the tree described in the manual. More specifically, the trees that Chyen grows will contain some of the nodes from 1 to N (at least one node), and these nodes are connected as they are in the tree described in the manual (a connected subgraph).

Chyen, being a lover of numbers, is very interested in the total sum of the point weights on each tree. He is curious about the K-th smallest sum of point weights among all the distinct trees he can grow.

Your task is to write a program to help Chyen solve this problem. If it's not possible to grow K different trees, output "-1" to inform Chyen.

Note: Chyen may grow multiple trees with the same total point weights, but as long as the set of nodes (a non-empty subset of 1 to N) included in one tree is different from that of another, they are considered distinct trees.

Input Format

The first line of the input contains two positive integers N, K, representing the number of nodes in the tree and the Kth smallest sum of point weights Chyen wants to find, respectively.

The second line contains N positive integers $C_1, C_2, \ldots, C_N$, representing the point weight of each node.

The following $N-1$ lines, each contain two positive integers u, v representing an edge in the tree.

Output Format

Output an integer representing the K-th smallest sum of point weights of the trees. If it is not possible to grow K different trees, output "-1".

Constraints

- $1 \leq N \leq 2 \times 10^5$
- $1 \leq K \leq 2 \times 10^5$
- $1 \leq C_i \leq 10^9 \; \forall i \in [1, N]$

Subtasks

- Subtask 1 satisfies that $1 \leq N, K \leq 200$.
- Subtask 2 satisfies that the given tree is a chain.
- Subtask 3 has no additional constraints.

Test Cases

Input 1

```
5 10
1 1 1 1 1
1 2
2 3
3 4
4 5
```

Output 1

```
3
```

Input 2

```
7 20
8 2 1 7 1 10 2
3 7
7 6
6 1
3 5
1 2
1 4
```

Output 2

```
21
```

Input 3

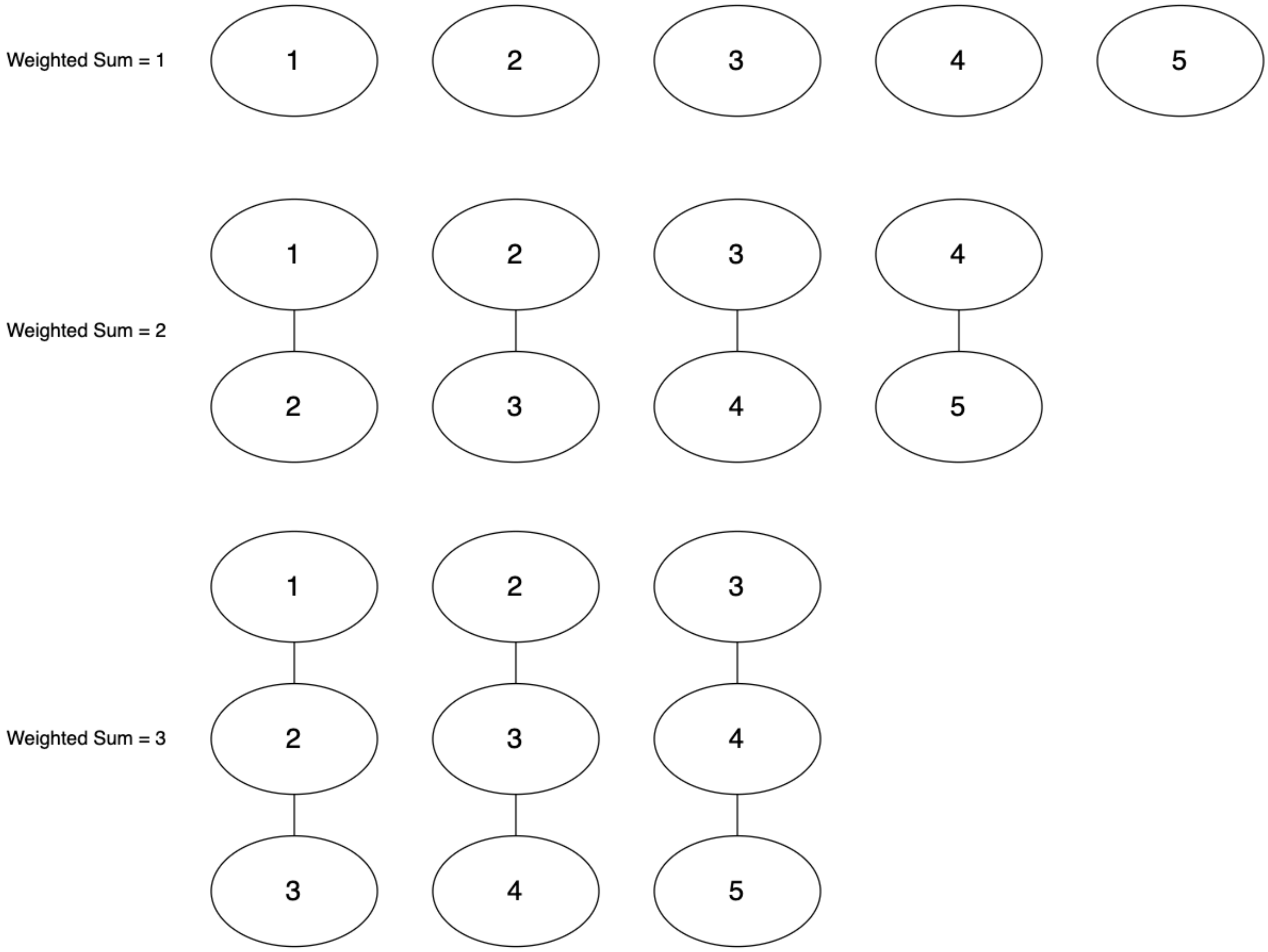
```
2 100
1 1
1 2
```

Output 3

```
-1
```

Illustrations

In Example 1, the first 10 smallest sums of point weights in order are: $[1, 1, 1, 1, 1, 2, 2, 2, 2, 3]$. The graph below lists all the connected subgraphs whose point weights are ≤ 3.



7_禁忌之島 (Forbidden Island)

(3 分/17 分)

時間限制: 5 seconds

記憶體限制: 1024 MB

題目敘述

在禁忌之島中，一共有 N 個城市。除了禁忌之島的首都以外，每個城市皆被一個上屬城市管轄，且一個城市可以有多個直屬下屬城市。首都位於編號 1 的城市。

禁忌之島時常舉辦嘉年華，當某座城市作為嘉年華的舉辦方時，該城市所有下屬們（包含直屬下屬的下屬，以此類推）也將會一起舉辦嘉年華。

在每座城市在年初有 A_i 元的經費。為了準備嘉年華，舉辦的城市們會需要進行大量的購物。每座城市將會用自己的經費大量購買相同的物品，直到經費不足。

在某些時候，某座城市可能覺得某個上屬表現不佳，轉而跑去找另外一個城市做為上屬。

在西元 998244353 年中，一共發生了 Q 筆事件。事件的種類一共有三種，描述如下：

1. 城市 u 舉辦嘉年華（他的所有下屬也將一同舉辦），並購買的物品單價為 x 元。
2. 城市 u 的上屬更改成 v。
3. 詢問今年以城市 u 與他下屬們（包含直屬下屬的下屬，以此類推）一共買了多少個物品，並對答案 $\pmod{998244353}$。

輸入格式

輸入的第一行為一個正整數 N，代表一共有 N 個城市。

第二行一共有 $N-1$ 個正整數，第 i 個整數代表編號為 $i+1$ 的上屬城市的編號 p_i。

第三行一共有 N 個正整數 A_i，代表城市 i 在年初的經費為 A_i 元。

第四行為一個正整數 Q，代表在第 998244353 年，一共發生了 Q 筆事件。

接下來 Q 行，每行包含若干個整數，這 Q 行中的第 j 行的第一個整數 k 代表了第 j 個事件的種類：

- 當 $k=1$，代表發生了第一種事件，接下來會有兩個整數 u, x，意義與題目敘述相同。
- 當 $k=2$，代表發生了第二種事件，接下來會有兩個整數 u, v，意義與題目敘述相同。
- 當 $k=3$，代表發生了第三種事件，接下來會有一個整數 u，意義與題目敘述相同。

輸出格式

對於每個第 3 種的事件，輸出一個正整數，代表以城市 u 做為舉辦方的嘉年華之中，購買物品的數量，並對答案 $\bmod 998244353$。

資料範圍

- $2 \le N, Q \le 3 \cdot 10^5$

- $1 \leq u, v \leq N$
- $p_i < i, 1 \leq i \leq N$
- $0 \leq A_i < 998244353$
- $1 \leq x < 998244353$

子任務

- 子任務 1 滿足 $N, Q \leq 5000$。
- 子任務 2 無額外限制。

測試範例

範例輸入 1

```
3
1 1
10 10 10
10
3 1
1 1 3
3 1
3 2
3 3
2 2 3
1 3 1
3 1
3 2
3 3
```

範例輸出 1

```
0
9
3
3
11
4
8
```

範例輸入 2

```
8
1 1 3 4 3 5 1
544420781 729609648 110923301 240695139 305509796 177030453 789149504 251576103
11
3 6
1 4 100
```

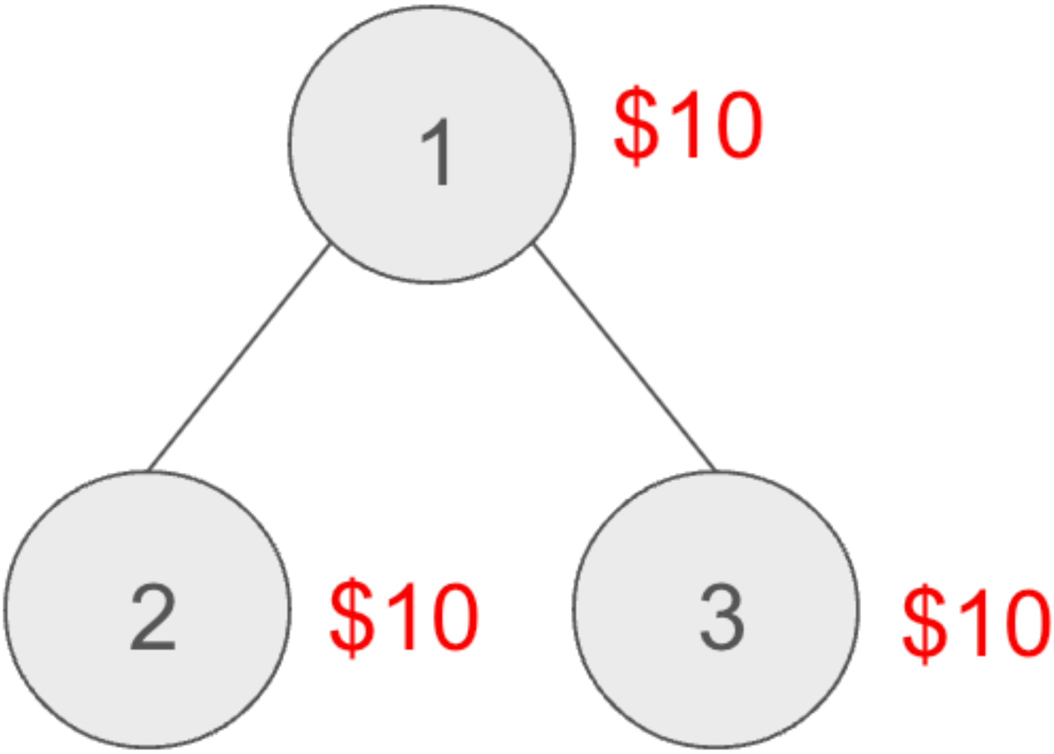
```
2 7 2
1 6 48763
3 5
2 2 6
1 2 114514
2 5 7
3 3
2 2 6
3 1
```

範例輸出 2

```
0
3055097
13363544
13363544
```

範例說明

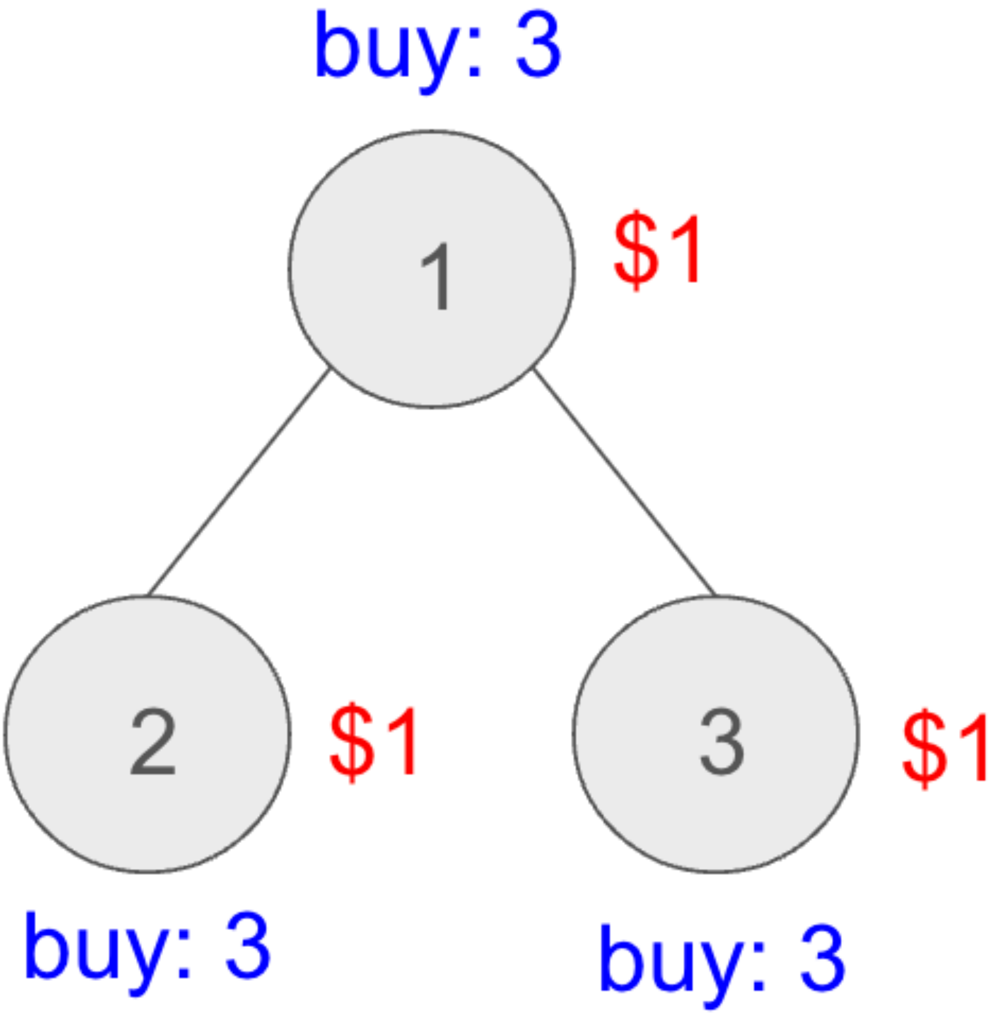
在第一筆範例測資中，一開始的城市圖如下：



第一次事件時，因為還沒有舉辦任何的嘉年華，因此輸出 0。

第二次事件是由城市 1 舉辦嘉年華，並購買價錢為 3 的商品，因此：

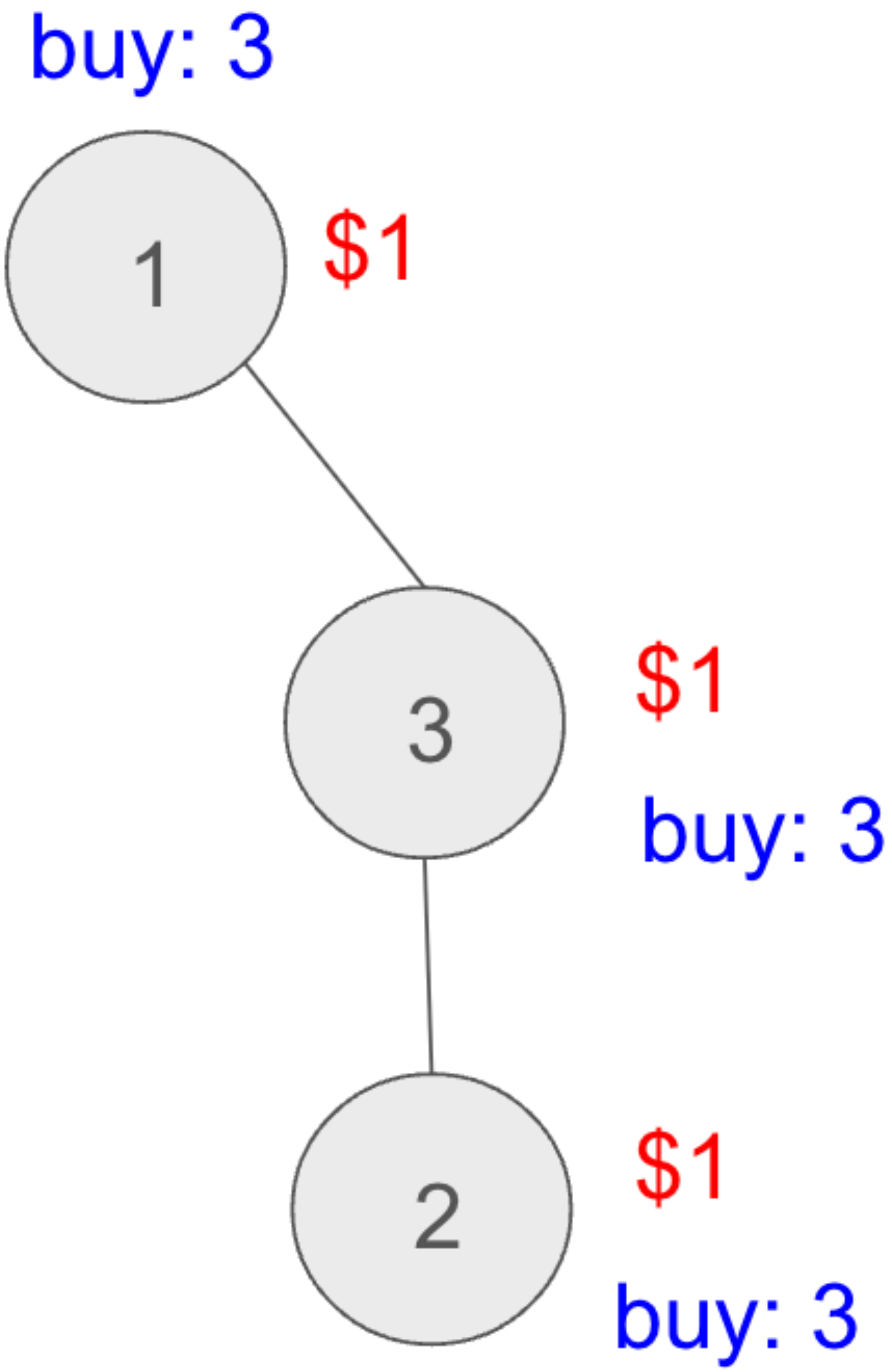
- 主辦城市 1 購買了 3 個商品，經費剩餘 1。
- 城市 2 是 1 的下轄城市，因此也要一起舉辦嘉年華，並購買了 3 個商品，經費剩餘 1。
- 城市 3 是 1 的下轄城市，因此也要一起舉辦嘉年華，並購買了 3 個商品，經費剩餘 1。



第三到五次事件就是詢問各個城市與其下轄城市購買的商品總和，對於 1 來說，他有兩個下轄城市 2，3，一共購買了 9 個商品。

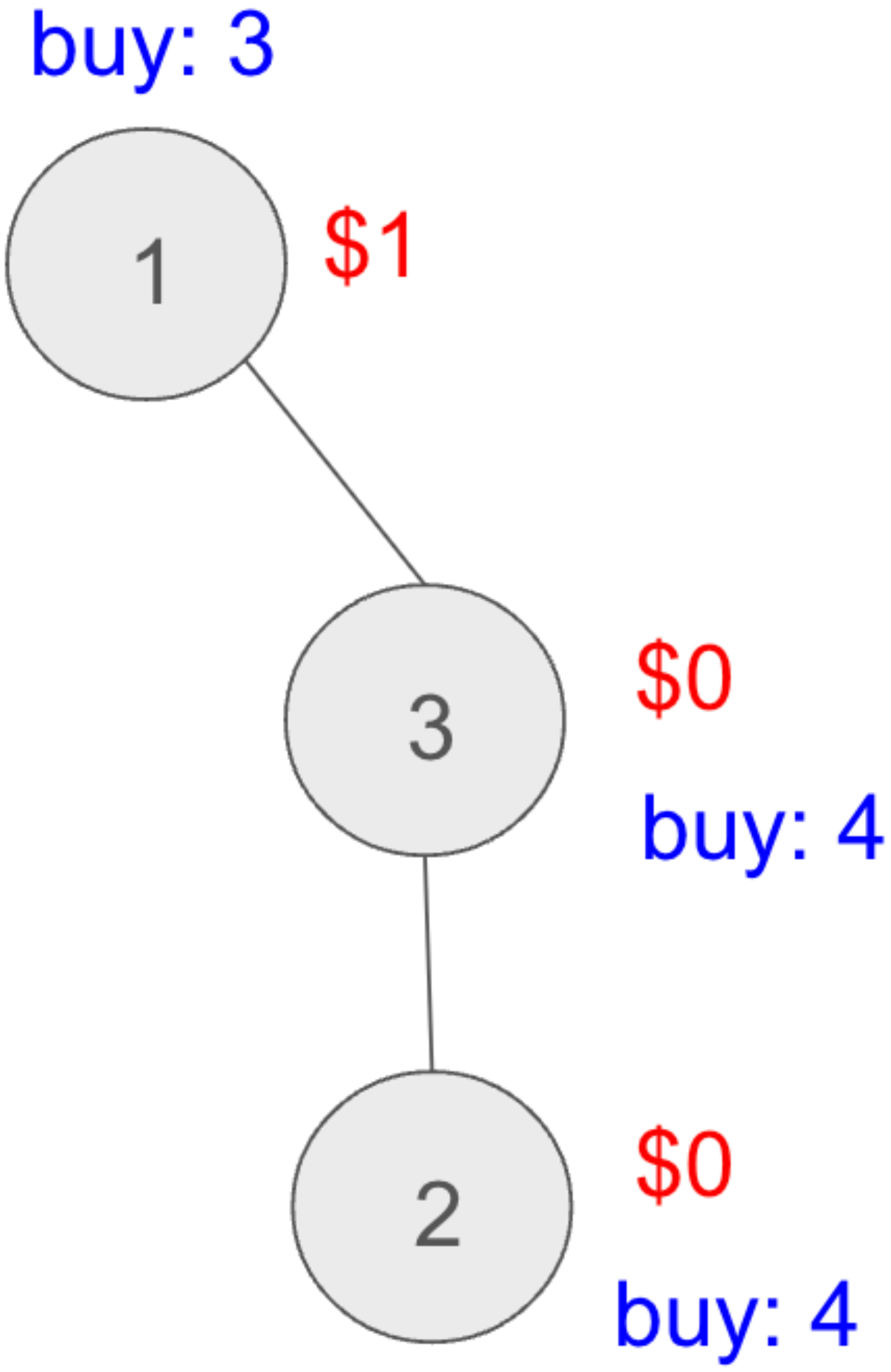
對於 $2, 3$ 來說，由於他們都沒有下轄城市，因此只要輸出自己購買的商品數量 3 即可。

接下來的第六次事件中，城市 2 變成城市 3 的下轄城市，此時城市圖為：



第七次事件中，由城市 3 舉辦嘉年華，購買價值為 1 的商品，因此：

- 主辦城市 3 購買了 1 個商品，經費剩餘 0。
- 城市 2 是 3 的下轄城市，因此也要一起舉辦嘉年華，並購買了 1 個商品，經費剩餘 0。



剩下的事件就是詢問各個城市與其下轄城市購買的商品總和，對於 1 來說，他有兩個下轄城市 2，3，一共購買了 11 個商品。

對於 3 來說，他有一個下轄城市 2，因此總共購買了 8 個商品。

對於 2 來說，由於他沒有下轄城市，因此只要輸出自己購買的商品數量 4 即可。

7_Forbidden Island

(3 points/17 points)

Time Limit: 5 seconds

Memory Limit: 1024 MB

Statement

There are N cities on the Forbidden Island. Except for the capital, every city is governed by a parent city. A parent city may have multiple child cities. The capital is always numbered as city 1.

The residents of the Forbidden Island frequently hold carnivals. When a city hosts a carnival, every offspring of that city will participate in the carnival together.

Initially, each city has A_i dollars as the budget for the carnival. To hold a carnival, every participating city will buy items until they run out of budget.

Occasionally, a city may decide that its parent city is unsuitable and choose a different city as its new parent.

In B.C. 998244353, there are Q events happening on the Forbidden Island. Here is a description of the three types of events:

1. The u-th city hosts a carnival (along with its offspring cities, if any), and they buy items priced at x dollars each.
2. The u-th city designates the v-th city as its new parent.
3. Query the total number of items bought by the u-th city and its offspring. The answer should be modulo 998244353.

Input Format

The first line contains a single integer N, representing the number of cities.

The second line contains $N-1$ integers, where the i-th integer represents the parent city of the $i+1$-th city (the 1-st city is always the capital and have no parent city).

The third line contains N integers, representing the initial budget A_i for each cities.

The fourth line contains a single integer Q, representing the number of events.

For following Q lines, each line represent one event. For each line, the first integer represent the event type k.

- If $k = 1$, it represent the first event, and the line contains three integers: 1, u, and x. This means the u-th city hosts a carnival and buys items priced at x dollars each.
- If $k = 2$, it represent the second event, and the line contains three integers: 2, u, and v. This means the u-th city changes its parent to the v-th city.
- If $k = 3$, it represent the third event, and the line contains two integers: 33 and uu. This means you should output the total number of items bought by the uu-th city and its offspring.

Output Format

For every third event, output a single integer on a new line representing the total number of items bought by the u-th city and its offspring. The result should be modulo 998244353.

Constraints

- $2 \leq N, Q \leq 3 \cdot 10^5$
- $1 \leq u, v \leq N$
- $p_i < i, 1 \leq i \leq N$
- $0 \leq A_i < 998244353$
- $1 \leq x < 998244353$

Subtasks

- Subtask 1 satisfies $N, Q \leq 5000$.
- Subtask 2 has no additional constraints.

Test Cases

Input 1

```
3
1 1
10 10 10
10
3 1
1 1 3
3 1
3 2
3 3
2 2 3
1 3 1
3 1
3 2
3 3
```

Output 1

```
0
9
3
3
11
4
8
```

Input 2

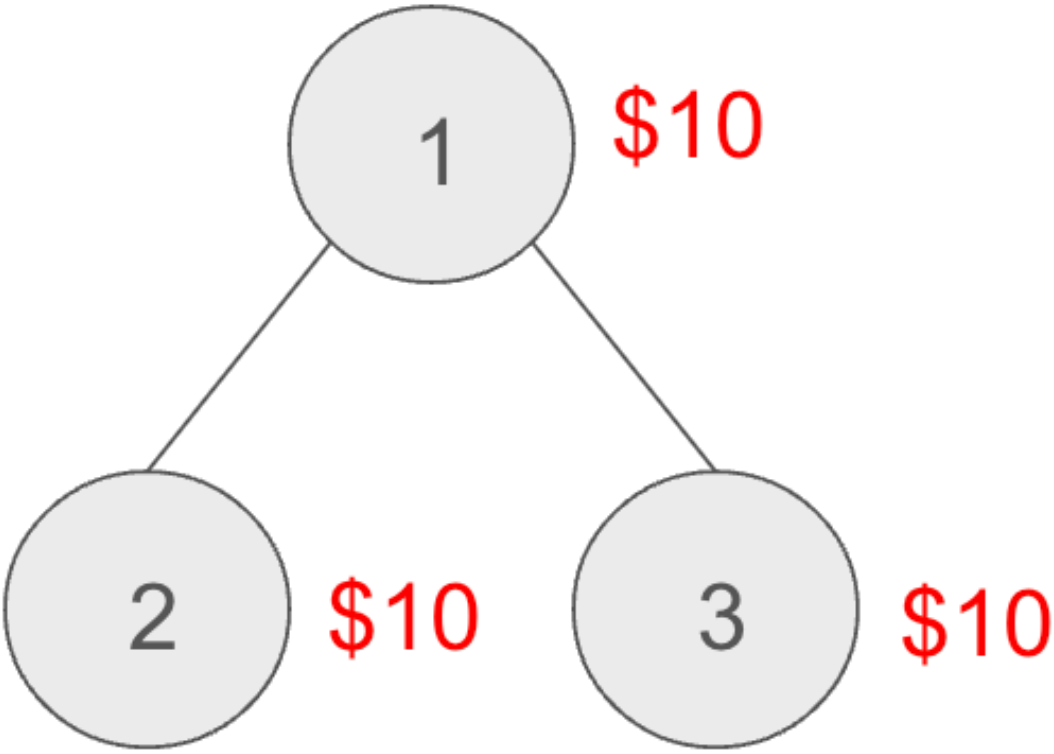
```
8
1 1 3 4 3 5 1
544420781 729609648 110923301 240695139 305509796 177030453 789149504 251576103
11
3 6
1 4 100
2 7 2
1 6 48763
3 5
2 2 6
1 2 114514
2 5 7
3 3
2 2 6
3 1
```

Output 2

```
0
3055097
13363544
13363544
```

Illustrations

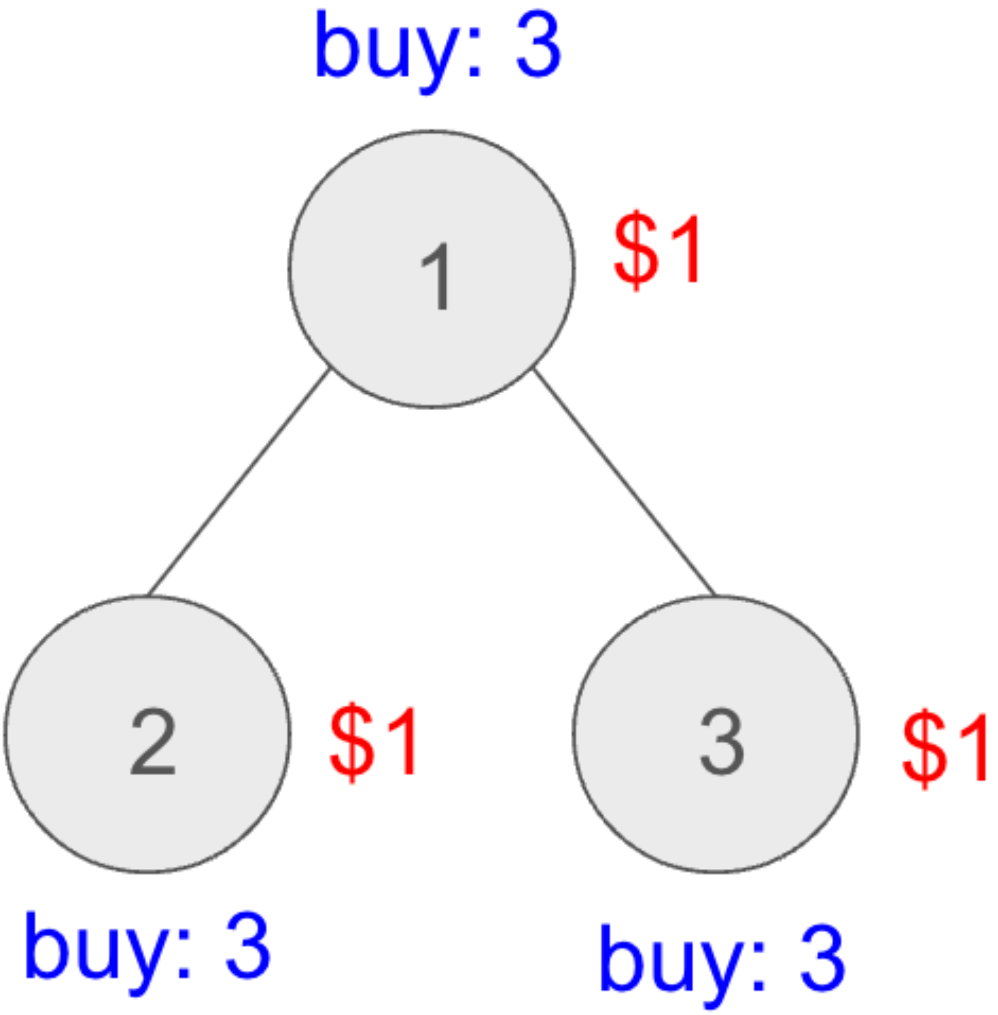
In the first example test case, the initial city graph is as follows:



During the first event, since no carnivals have been held yet, the output is 0.

For the second event, city 1 hosts a carnival and buys items priced at 3. Therefore:

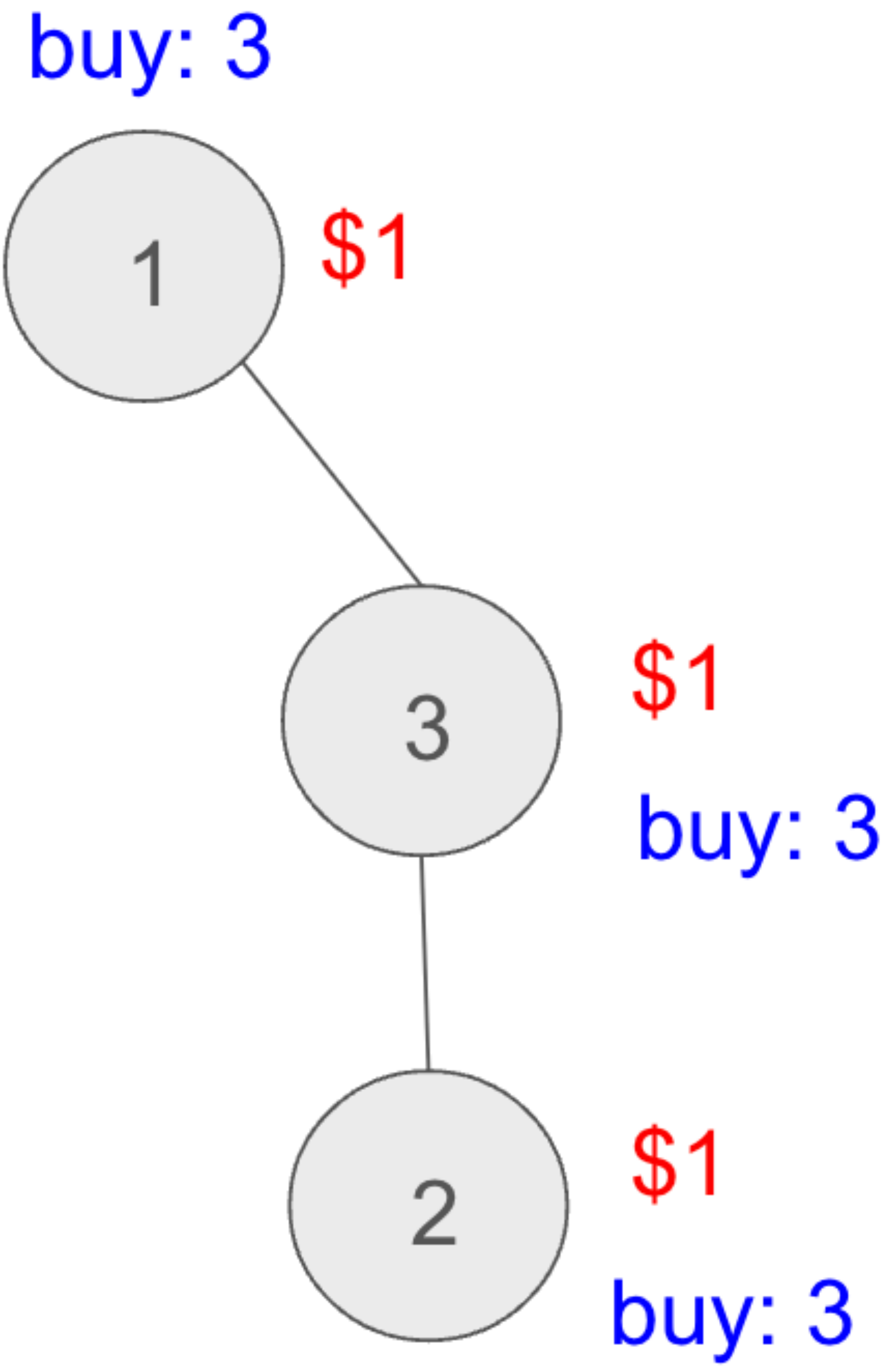
- The host city 1 buys 3 items, leaving a budget of 1.
- City 2 is offspring city of city 1, so it also participates in the carnival and buys 3 items, leaving a budget of 1.
- City 3 is offspring city of city 1, so it also participates in the carnival and buys 3 items, leaving a budget of 1.



Events three to five are queries about the total number of items bought by each city and its offsprings. For city 1, which has two offspring cities (2 and 3), the total number of items bought is 9.

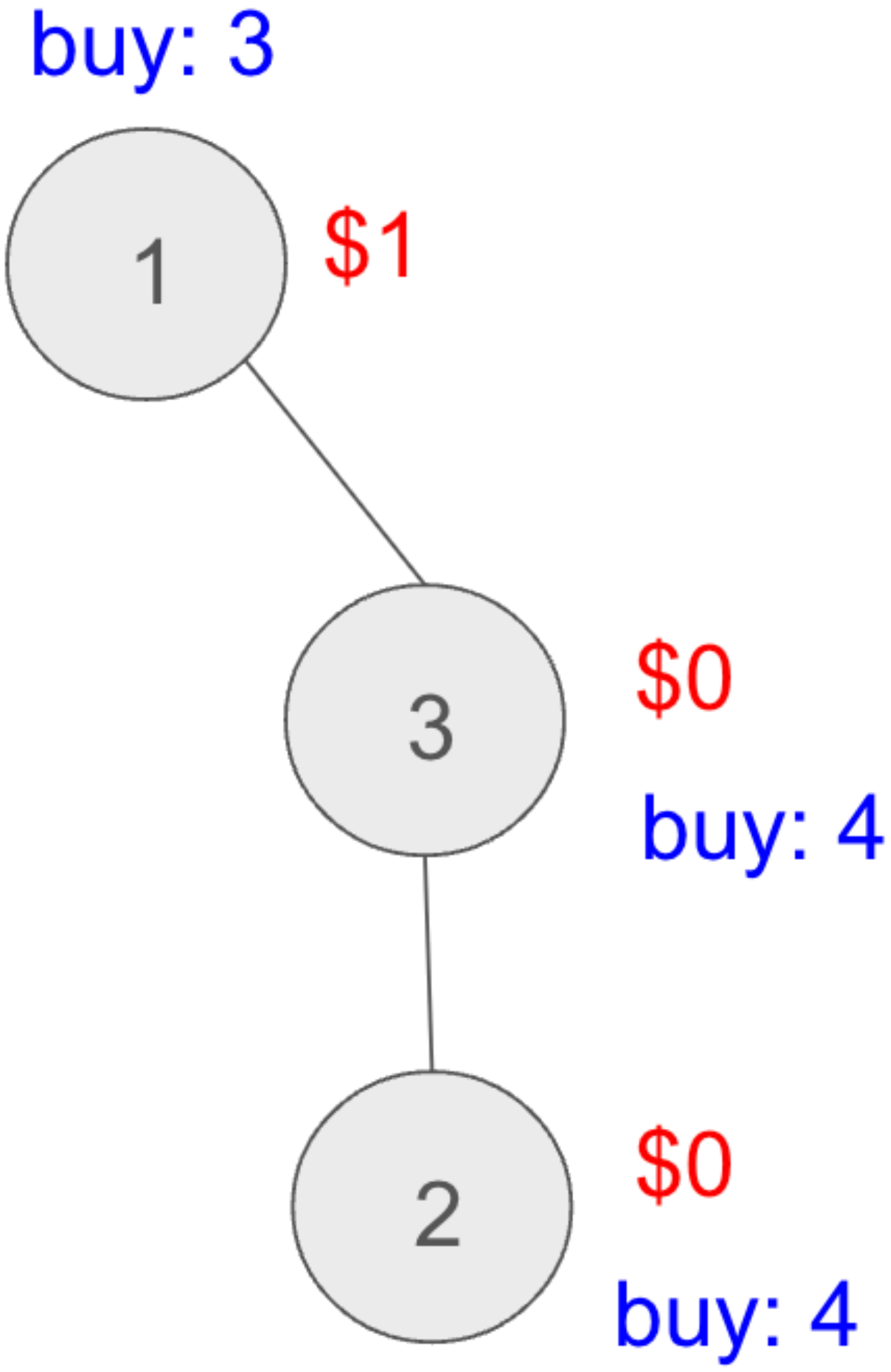
For cities 2 and 3, since they have no offspring, the output is simply the number of items each bought, which is 3.

In the sixth event, city 2 becomes a subordinate of city 3, and now the city graph is as follows:



In the seventh event, city 3 hosts a carnival and buys items priced at 1. Therefore:

- The host city 3 buys 1 item, leaving a budget of 0.
- City 2 is offspring city of city 3, so it also participates in the carnival and buys 1 item, leaving a budget of 0.



The remaining events are queries about the total number of items bought by each city and its offsprings. For city 1, which has two subordinates (2 and 3), the total number of items bought is 11.

For city 3, which has one offspring (2), the total number of items bought is 8.

For city 2, since it has no offspring, the output is simply the number of items it bought, which is 4.

8_興建道路 (Building Roads)

(2 分/2 分/6 分/10 分)

時間限制: 2.5 second

記憶體限制: 1024 MB

題目敘述

YTP 城市一共有 N 棟高樓大廈。這些高樓大廈之間被 M 條雙向道路連接。高樓大廈之間彼此互相連通，也就是說對於每對高樓大廈 U 跟 V，皆存在一些高樓大廈 $s_1, s_2, \ldots, s_k$ 使得 U 和 s_1 之間有道路、s_i 和 s_{i+1} 之間有道路、s_k 和 V 之間有道路。每條道路各自被一個集團所控制，經過第 i 條道路要支付 W_i 元的過路費。為了讓城市美觀，YTP 政府打算拆掉一些道路，但是維持高樓大廈之間能夠互相抵達。這個消息一公布就引發各大集團的恐慌！特別聰明的你想要知道如果在保留第 i 條道路的情況下，總過路費至少要多少？

輸入格式

輸入的第一行包含兩個正整數 N 和 M，代表高樓大廈和道路的數量。

接下來一共有 M 行，每行有三個正整數 U_i、V_i、W_i，代表這條道路雙向連接 U_i 和 V_i 這兩棟高樓大廈，且過路費為 W_i。

輸出格式

輸出 M 行，第 i 行為保留第 i 條道路至少需要的總過路費。

資料範圍

- $2 \leq N \leq 5 \times 10^5$。
- $N - 1 \leq M \leq \min\left(\frac{N(N-1)}{2}, 10^6\right)$。
- $1 \leq U_i, V_i \leq N$（$1 \leq i \leq M$）。
- $U_i \neq V_i$（$1 \leq i \leq M$）。
- $1 \leq W_i \leq 10^9$（$1 \leq i \leq M$）。
- 保證一開始所有高樓大廈之間互相是連通的，且不會有兩條道路連接同一對高樓大廈。

子任務

- 子任務 1 滿足 $M \leq N + 20$。
- 子任務 2 滿足 $N \leq 2000$。
- 子任務 3 滿足 $N \leq 50\,000$。
- 子任務 4 無額外限制。

測試範例

輸入範例 1

```
4 6
4 1 4
1 2 6
4 2 3
4 3 1
2 3 5
3 1 5
```

輸出範例 1

```
8
10
8
8
10
9
```

輸入範例 2

```
2 1
2 1 48763
```

輸出範例 2

```
48763
```

輸入範例 3

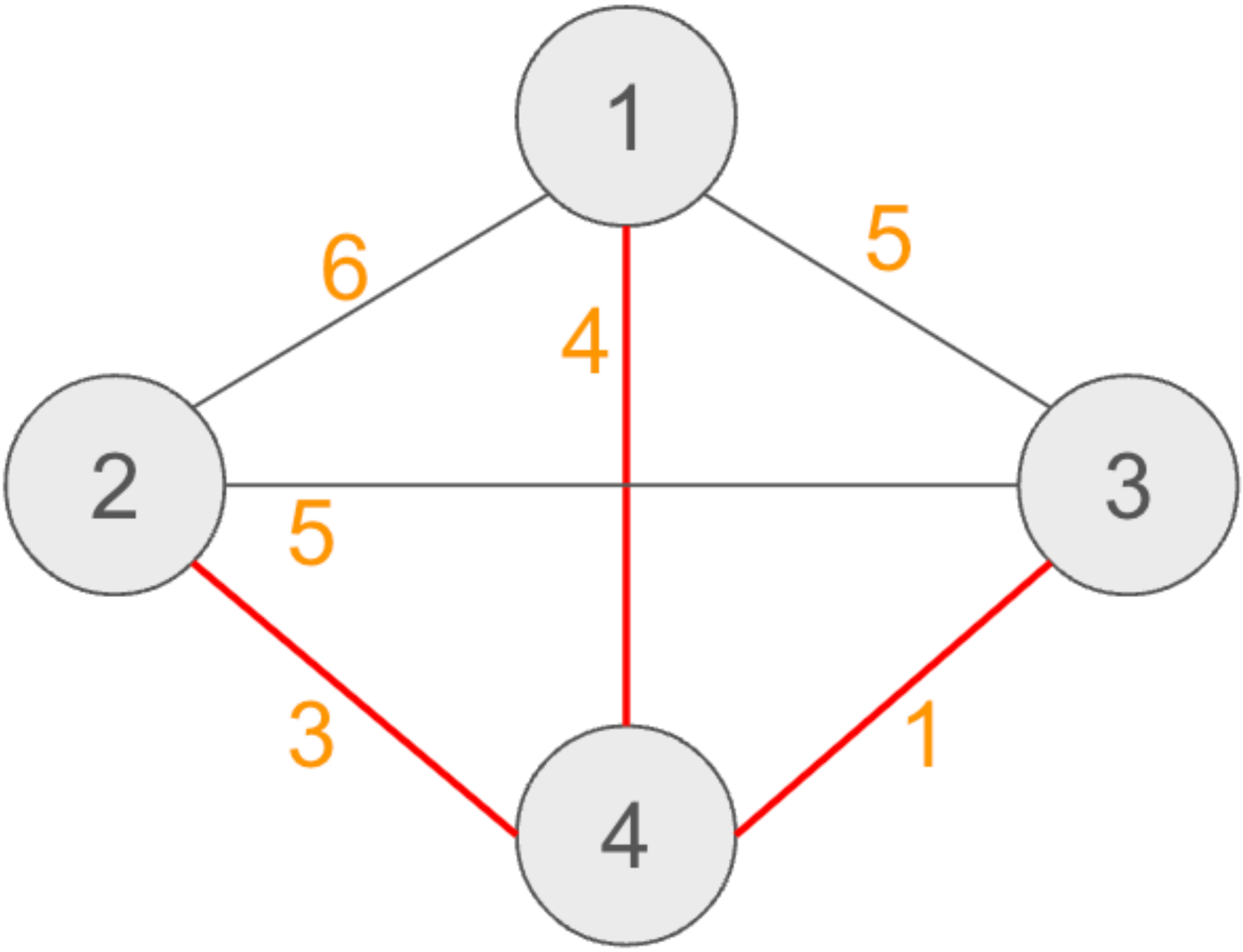
```
6 15
1 3 10
5 3 5
1 6 9
4 2 1
6 2 8
1 2 9
1 5 10
3 2 7
6 4 7
1 4 2
6 5 7
3 6 1
2 5 8
4 3 7
4 5 2
```

輸出範例 3

```
16
11
15
11
14
18
19
13
13
11
13
11
17
13
11
```

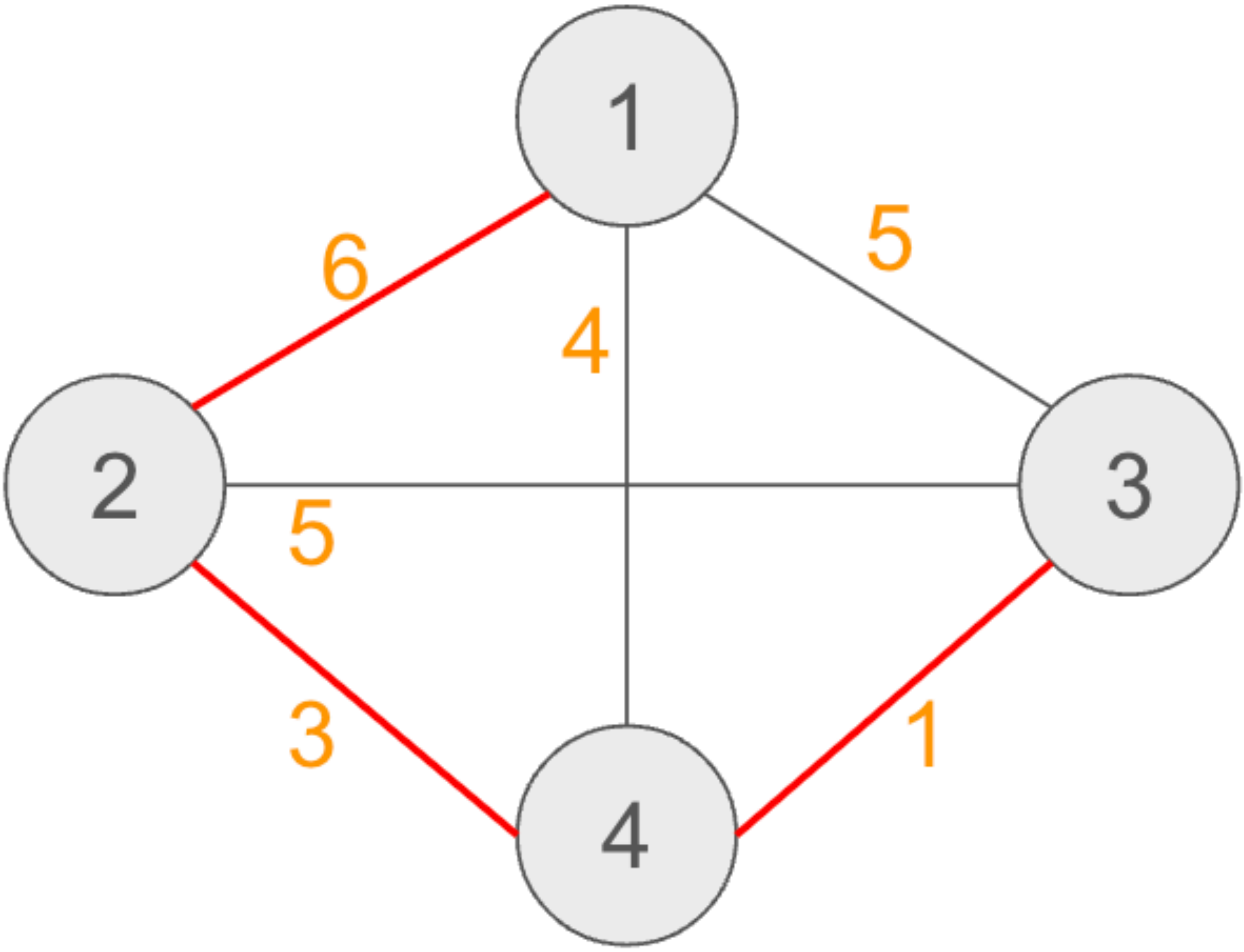
範例說明

在範例一中：



如果要保留第 1 條道路，一個可行的方案為同時保留第 3 和 4 條道路可以讓所有高樓大廈之間互相連通，且總過路費為 8 元。沒有方案能達到比 8 元更少的總過路費。

以下為 $i = 2$ 時所使用的道路，可以證明沒有其他更好的作法可以達到更低的過路費。



在範例二中，只有一條道路，因此一定要保留。

8_Building Roads

(2 points/2 points/6 points/10 points)

Time Limit: 2.5 seconds

Memory Limit: 1024 MB

Statement

The city of YTP has a total of N tall buildings, connected by M bidirectional roads. Buildings are reachable from each other initially. In other words, for any pair of buildings U and V, there exists some buildings $s_1, s_2, \dots, s_k$ such that there is a road between U and s_1, s_i and s_{i+1}, s_k and V. Each road is controlled by a different group, and passing through road i requires paying a toll of W_i dollars. To beautify the city, the YTP government plans to demolish some of these roads while maintaining the ability for buildings to reach each other. This announcement has caused panic among the groups. As a particularly clever person, you want to know the minimum total toll fee required to preserve road i.

Input Format

The first line of the input consists of two integers N and M, representing the number of buildings and the number of bidirectional roads.

The next M lines describe the roads, with each line consisting of three integers, U_i, V_i, W_i, indicating that road i connects city U_i and V_i bidirectionally with a toll of W_i dollars.

Output Format

Output M lines. The i-th line is the minimum total toll fee required to preserve road i.

Constraints

- $2 \le N \le 5 \times 10^5$.
- $N - 1 \le M \le \min\left(\frac{N(N-1)}{2}, 10^6\right)$.
- $1 \le U_i, V_i \le N$ ($1 \le i \le M$).
- $U_i \ne V_i$ ($1 \le i \le M$).
- $1 \le W_i \le 10^9$ ($1 \le i \le M$).
- It is guaranteed that the buildings are reachable from each other initially, and no two different roads connect the same pair of buildings.

Subtasks

- Subtask 1 satisfies that $M \le N + 20$.
- Subtask 2 satisfies that $N \le 2000$.
- Subtask 3 satisfies that $N \le 50\,000$.
- Subtask 4 has no additional constraints.

Test Cases

Input 1

```
4 6
4 1 4
1 2 6
4 2 3
4 3 1
2 3 5
3 1 5
```

Output 1

```
8
10
8
8
10
9
```

Input 2

```
2 1
2 1 48763
```

Output 2

```
48763
```

Input 3

```
6 15
1 3 10
5 3 5
1 6 9
4 2 1
6 2 8
1 2 9
1 5 10
3 2 7
6 4 7
1 4 2
6 5 7
3 6 1
2 5 8
4 3 7
```

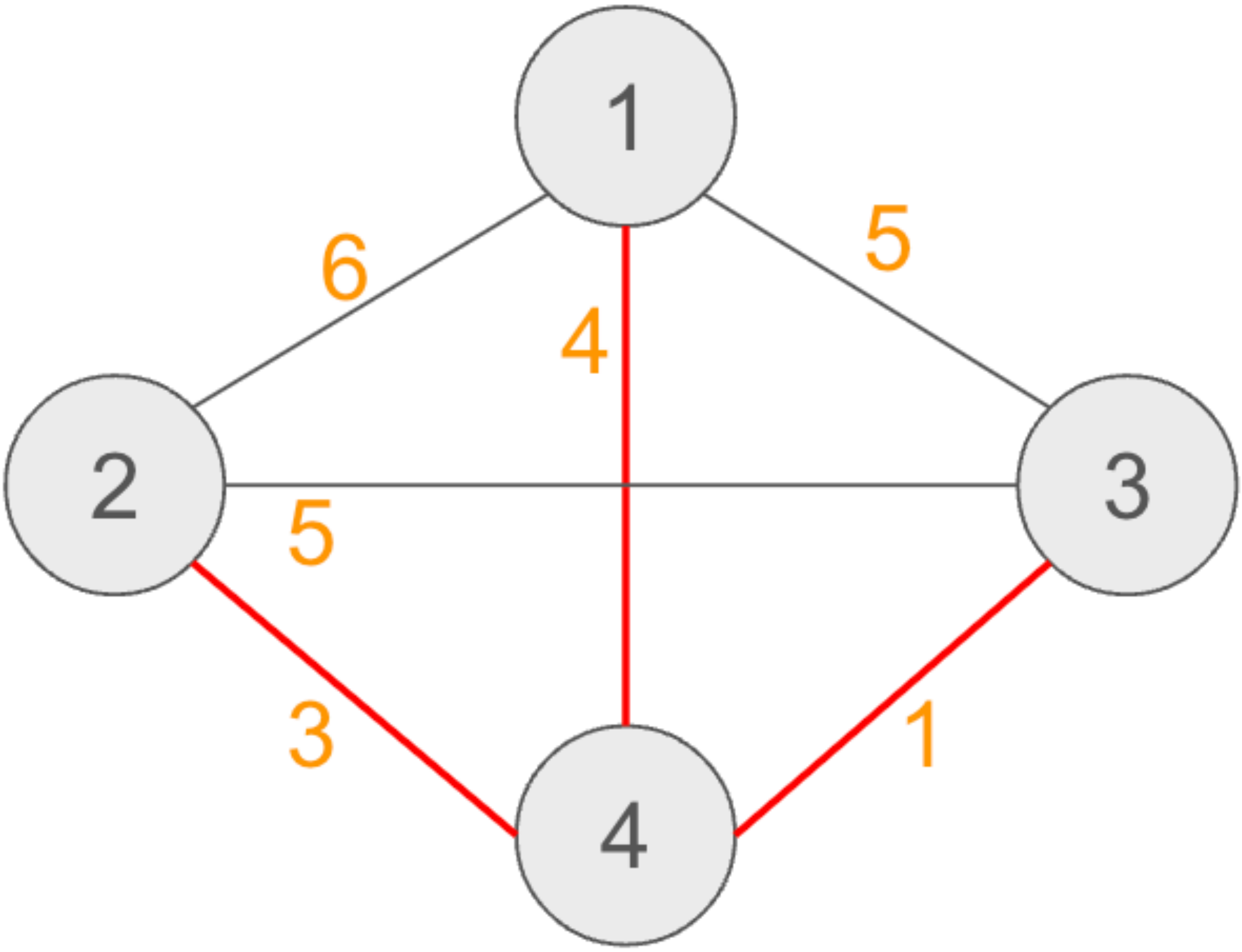
```
4 5 2
```

Output 3

```
16
11
15
11
14
18
19
13
13
11
13
11
17
13
11
```

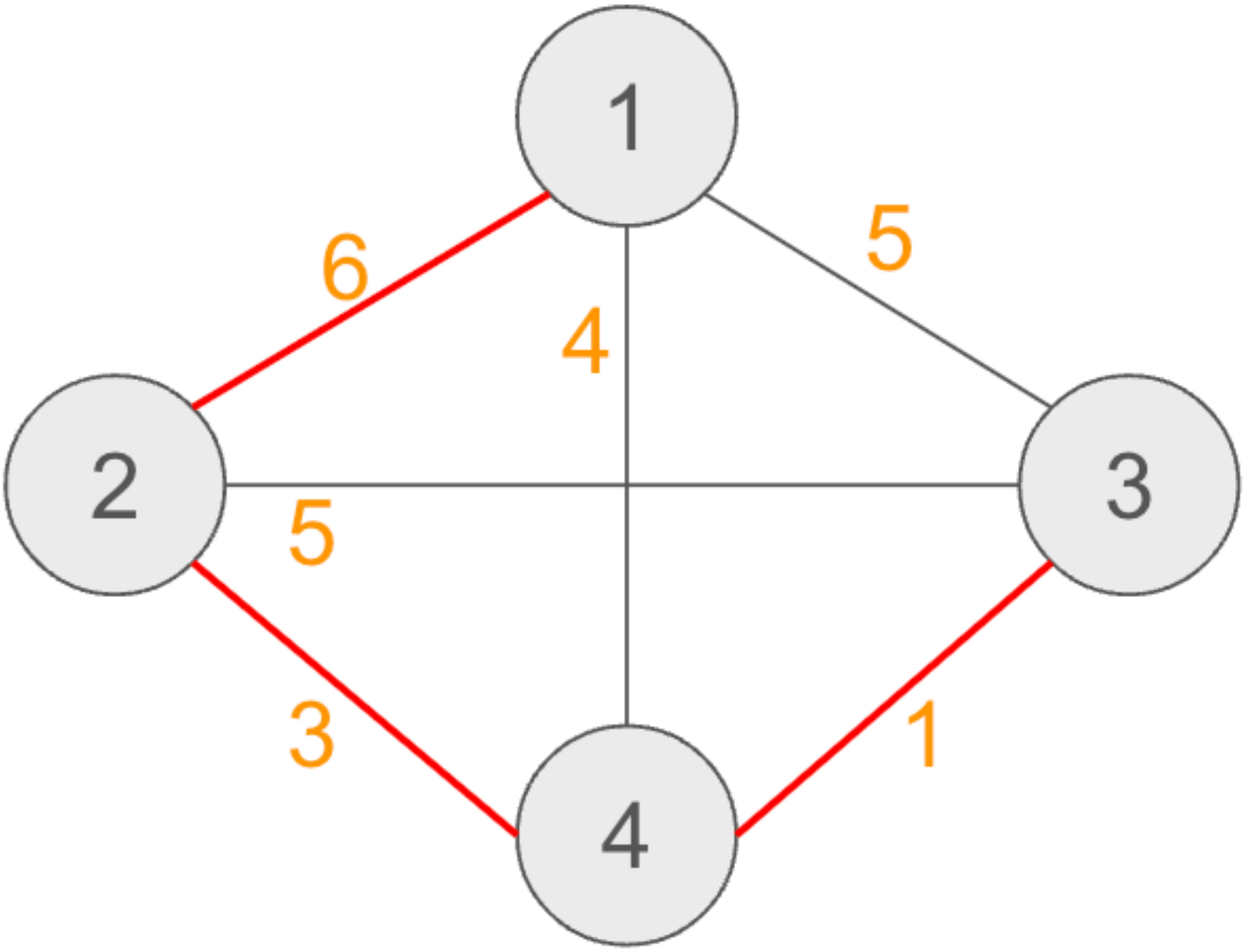
Illustrations

In example 1:



If we preserve road 1, one possible solution is to preserve road 3 and road 4, then all buildings are able to reach each other. The toll of this solution is 8 dollars, which is the minimum among all solutions.

For $i = 2$, the following road is used. It can be proven that there is no better way to achieve a lower toll fee.



In example 2, there is only one road, so it must be kept.