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

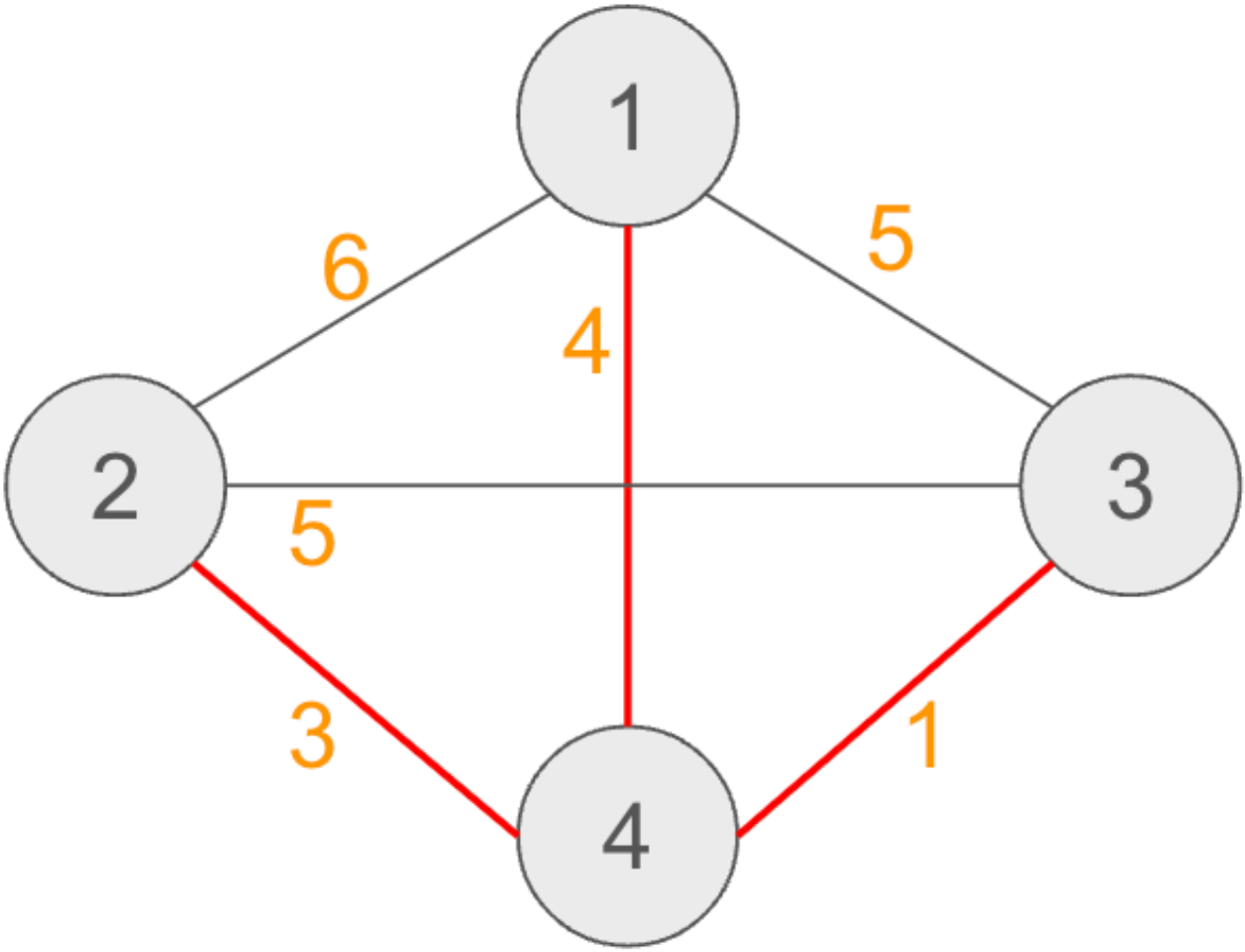
```
6 15
1 3 10
5 3 5
1 6 9
4 2 1
6 2 8
1 2 9
1 5 10
3 2 7
6 4 7
1 4 2
6 5 7
3 6 1
2 5 8
4 3 7
4 5 2
```

輸出範例 3

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

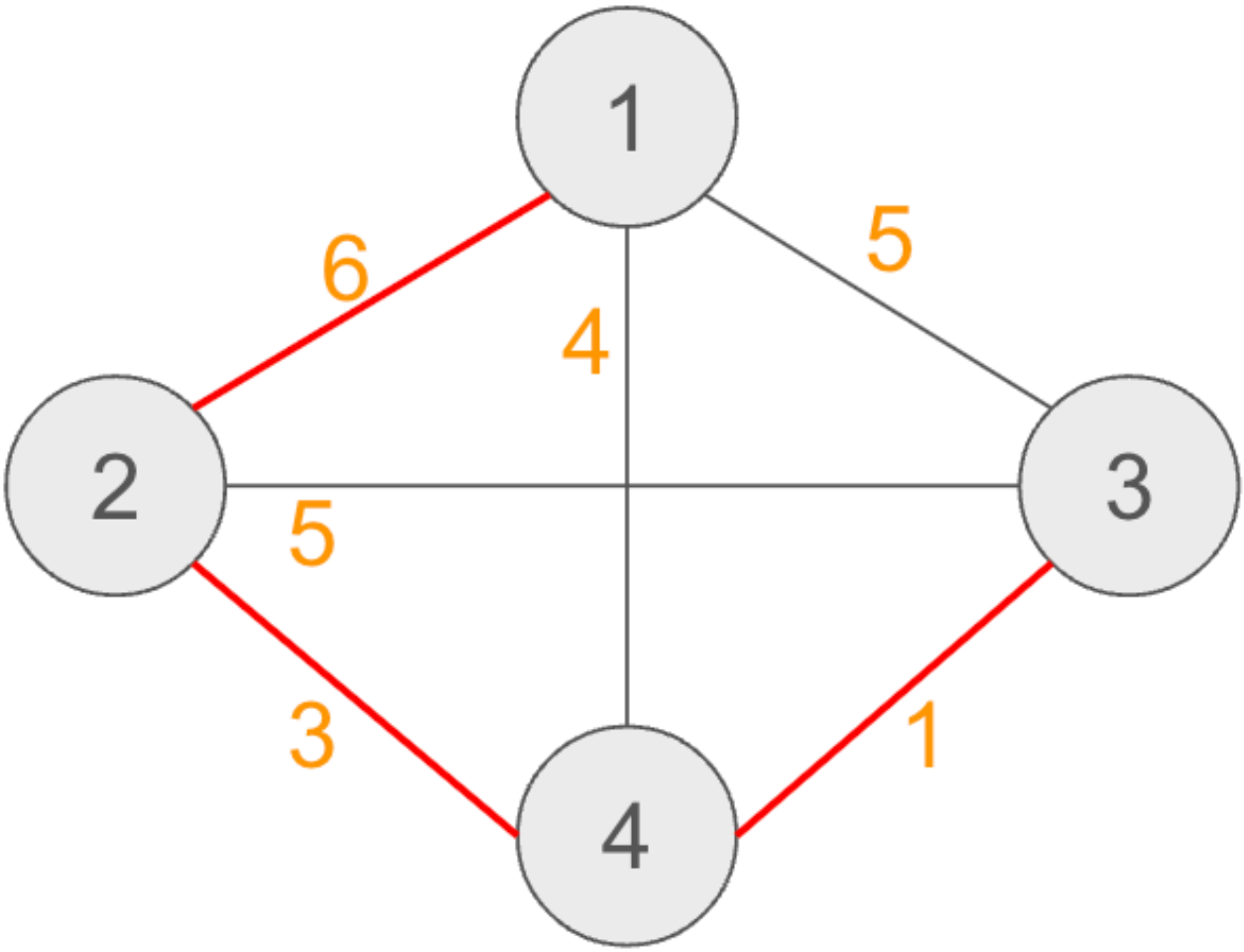
範例說明

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 $s_1, s_2, \ldots, s_k$ 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

- $2 \leq N \leq 5 \times 10^5$.
- $N - 1 \leq M \leq \min\left(\frac{N(N-1)}{2}, 10^6\right)$.
- $1 \leq U_i, V_i \leq N$ ($1 \leq i \leq M$).
- $U_i \neq V_i$ ($1 \leq i \leq M$).
- $1 \leq W_i \leq 10^9$ ($1 \leq i \leq M$).
- 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 $M \leq N + 20$.
- Subtask 2 satisfies that $N \leq 2000$.
- Subtask 3 satisfies that $N \leq 50\,000$.
- 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
```

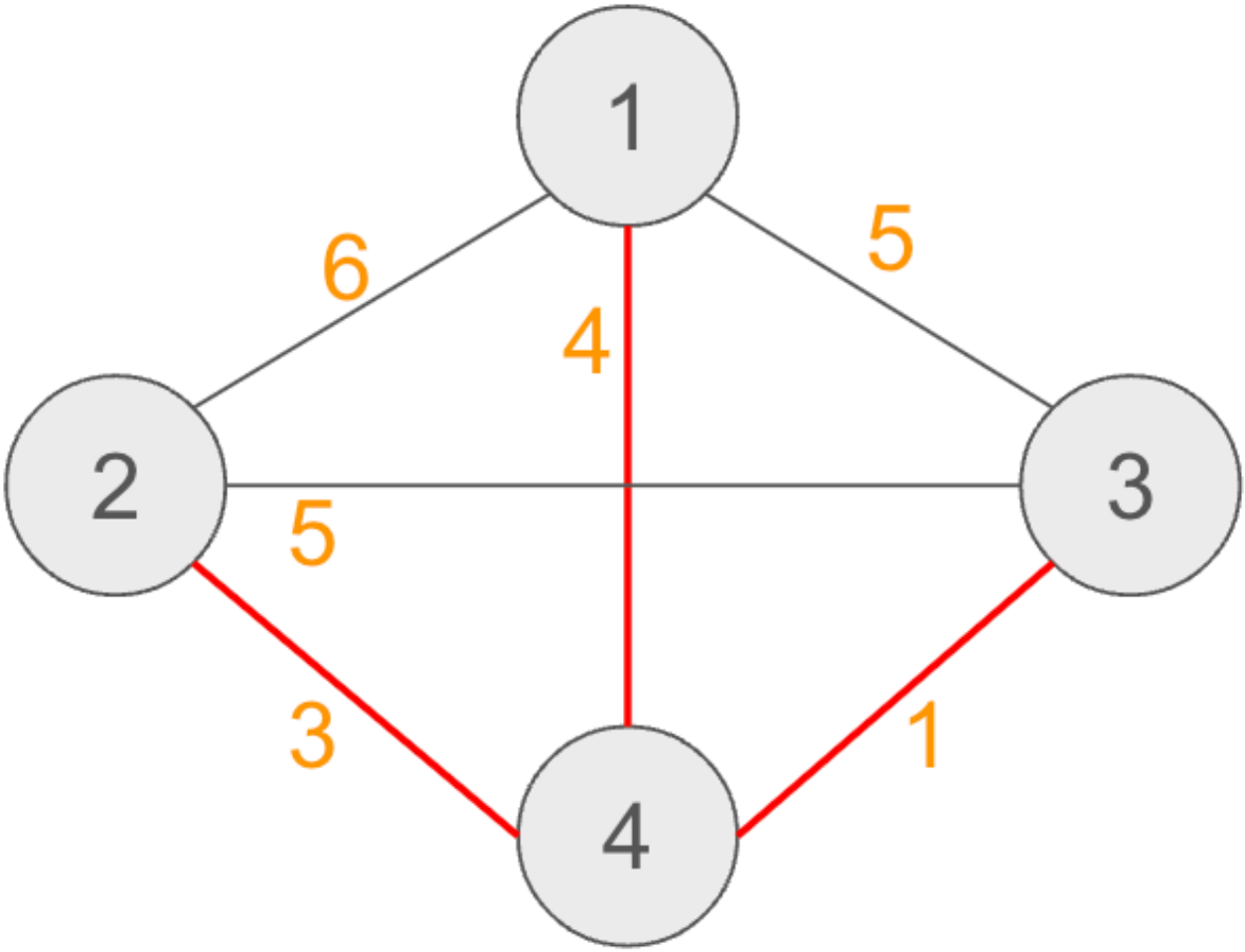
```
4 5 2
```

Output 3

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

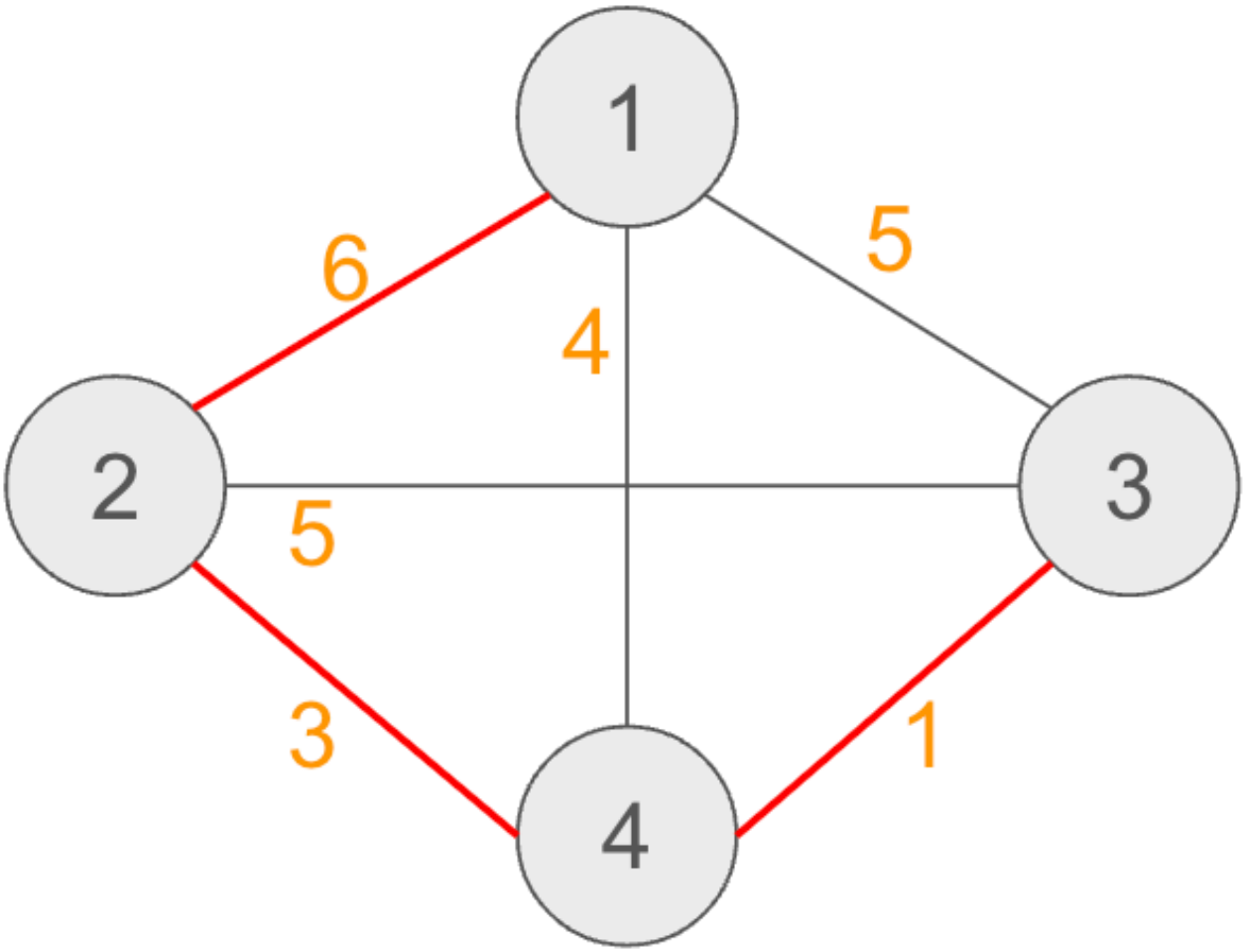
Illustrations

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.