2 胖

2.1 题目描述

Cedyks 是九条可怜的好朋友（可能这场比赛公开以后就不是了），也是这题的主人公。

Cedyks 是一个富有的男孩子。他住在著名的 The Place（宫殿）中。

Cedyks 是一个努力的男孩子。他每天都做着不一样的题来锻炼他的 The Salt（灵魂）。

这天，他打算在他的宫殿外围修筑一道城墙，城墙上有 n 座瞭望塔。你可以把城墙看做一条线段，瞭望塔是线段上的 n 个点，其中 1 和 n 分别为城墙的两个端点。其中第 i 座瞭望塔和第 $i+1$ 座瞭望塔的距离为 w_i，他们之间的道路是双向的。

城墙很快就修建好了，现在 Cedyks 开始计划修筑他的宫殿到城墙的道路。因为这题的题目名称，Cedyks 打算用他的宫殿到每一个瞭望塔的最短道路之和来衡量一个修建计划。

现在 Cedyks 手上有 m 个设计方案，第 k 个设计方案会在宫殿和瞭望塔之间修建 T_k 条双向道路，第 i 条道路连接着瞭望塔 a_i，长度为 l_i。

计算到每一个瞭望塔的最短路之和是一个繁重的工程，本来 Cedyks 想用广为流传的 SPFA 算法来求解，但是因为他的 butter（缓冲区）实在是太小了，他只能转而用原始的贝尔福特曼算法来计算，算法的流程大概如下：

1. 定义宫殿是 0 号点，第 i 个瞭望塔是 i 号点，双向边 (u_i, v_i, l_i) 为一条连接 u_i 和 v_i 的双向道路。令 d 为距离数组，最开始 $d_0 = 0, d_i = 10^{18} (i \in [1, n])$。
2. 令辅助数组 $c = d$。依次对于每一条边 (u_i, v_i, w_i) 进行增广，$c_{u_i} = \min(c_{u_i}, d_{v_i} + w_i)$，$c_{v_i} = \min(c_{v_i}, d_{u_i} + w_i)$。
3. 令 t 为 c 和 d 中不一样的位置个数，即令 $S = \{i | c_i \neq d_i\}$，则 $t = |S|$。若 $t = 0$，说明 d 就是最终的最短路，算法结束。否则令 $d = c$，回到第二步。

因为需要计算的设计方案实在是太多了，所以 Cedyks 雇佣了一些人来帮他进行计算。为了避免这些人用捏造出来的数据偷懒，他定义一个设计方案的校验值为在这个方案上运行贝尔福特曼算法每一次进入第三步 t 的和。他会让好几个雇佣来的人计算同样的设计方案，并比对每一个人给出的校验值。

你是 Cedyks 雇佣来的苦力之一，聪明的你发现在这个情形下计算最短路的长度的和是一件非常简单的事情。但是寄人篱下不得不低头，你不得不再计算出每一个方案的校验值来交差。

2.2 输入格式

第一行输入两个整数 n, m，表示瞭望塔个数和设计方案个数。

接下来一行 $n-1$ 个数 w_i，表示瞭望塔 i 和 $i+1$ 之间道路的长度。

接下来 m 行，每行描述一个设计方案。第一个整数 K 表示设计方案中的道路数量，接下来 K 个数对 (a_i, l_i) 为一条宫殿到瞭望塔的边。

2.3 输出格式

对于每一个设计方案，输出一行一个整数表示校验值。

2.4 样例输入

```
5 5
2 3 1 4
1 2 2
2 1 1 4 10
3 1 1 3 1 5 1
3 1 10 2 100 5 1
5 1 1 2 1 3 1 4 1 5 1
```

2.5 样例输出

```
5
8
5
8
5
```

2.6 样例解释

- 对于第一个设计方案，每一个阶段 d 的变化为：
  - $[0, 10^{18}, 10^{18}, 10^{18}, 10^{18}, 10^{18}] \to [0, 10^{18}, 2, 10^{18}, 10^{18}, 10^{18}] \to$
  - $[0, 4, 2, 5, 10^{18}, 10^{18}] \to [0, 4, 2, 5, 6, 10^{18}] \to [0, 4, 2, 5, 6, 10]$。

  因此校验值为 $1 + 2 + 1 + 1 = 5$。
- 对于第二个设计方案，每一个阶段 d 的变化为：
  - $[0, 10^{18}, 10^{18}, 10^{18}, 10^{18}, 10^{18}] \to [0, 1, 10^{18}, 10^{18}, 10, 10^{18}] \to$
  - $[0, 1, 3, 11, 10, 14] \to [0, 1, 3, 6, 10, 14] \to [0, 1, 3, 6, 7, 14] \to [0, 1, 3, 6, 7, 11]$。

  因此校验值为 $2 + 3 + 1 + 1 + 1 = 8$。
- 对于第三个设计方案，每一个阶段 d 的变化为：
  - $[0, 10^{18}, 10^{18}, 10^{18}, 10^{18}, 10^{18}] \to [0, 1, 10^{18}, 1, 10^{18}, 1] \to [0, 1, 3, 1, 2, 1]$。

  因此校验值为 $3 + 1 + 1 = 5$。
- 对于第四个设计方案，每一个阶段 d 的变化为：
  - $[0, 10^{18}, 10^{18}, 10^{18}, 10^{18}, 10^{18}] \to [0, 10, 100, 10^{18}, 10^{18}, 1] \to$
  - $[0, 10, 12, 103, 5, 1] \to [0, 10, 12, 6, 5, 1] \to [0, 10, 9, 5, 1]$。

  因此校验值为 $3 + 3 + 1 + 1 = 8$。

- 对于第五个设计方案，每一个阶段 d 的变化为：
  - $[0, 10^{18}, 10^{18}, 10^{18}, 10^{18}, 10^{18}] \to [0, 1, 1, 1, 1, 1]$

  因此校验值为 5。

2.7 数据范围与约定

测试点	n	m	K	其他约定
1	≤ 1000	≤ 1000	≤ 100	无
2				
3	$\leq 2 \times 10^5$	$\leq 2 \times 10^5$		
4				
5			$\leq 2 \times 10^5$	$1 \leq w_i, l_i \leq 50$
6				
7				无
8				
9				
10				

对于 100% 的数据，保证每个设计方案 a_i 两两不同且 $1 \leq a_i \leq n$。

对于 100% 的数据，保证 $1 \leq w_i, l_i \leq 10^9, 1 \leq \sum K \leq 2 \times 10^5$。