

江南大学ACM算法俱乐部2024级定选招新 赛

题数: 8题

时长: 3小时

2025年02月22日

A. 修建掩体

时限: 1s 内存限制: 128MB

在一条长为 n 的战线上，最初所有位置的掩体高度均为 0（平地）。现在有 k 次掩体修筑操作，每次操作会在指定的区间内提升掩体高度。具体而言，第 i 次操作会在区间 $[l_i, r_i]$ 内的所有位置基础上额外修建高为 h_i 的掩体。

经过所有操作后，我们定义**有风险的掩体**如下：

- 位置 $i (i > 1)$ 的掩体被认为是**有风险的**，当且仅当在它左侧（即位置 1 到 $i - 1$ ）没有比它更高的掩体。
- 位置 1 的掩体总被认为是**有风险的**。

你的任务是计算最终战线上有多少个**有风险的掩体**。

输入

第一行包含两个整数 n 和 $k (1 \leq n, k \leq 2 \times 10^5)$ ，分别表示战线的长度和操作的次数。

接下来的 k 行，每行包含三个整数 $l_i, r_i, h_i (1 \leq l_i \leq r_i \leq n, 1 \leq h_i \leq 10^9)$ ，表示在区间 $[l_i, r_i]$ 上，每个位置的掩体高度增加 h_i 。

输出

输出一个整数，表示最终战线上**有风险的掩体**数量。

样例

序	输入样例	输出样例
#0	7 4	4
	1 7 1	
	2 6 1	
	3 5 1	
	4 4 1	
#1	5 1	5
	1 5 1	

提示

样例 #0: 掩体建造完毕后，7 个位置的高度分别为 1, 2, 3, 4, 3, 2, 1，其中前 4 个掩体是有风险的。

样例 #1: 掩体建造完毕后，5 个位置的高度分别为 1, 1, 1, 1, 1。

B. 天空之城

时限: 2s 内存限制: 256MB

小 K 和小 W 来到了神秘的天空之城。

天空之城被毁灭咒语バルス所破坏成 n 个小岛，这些小岛之间被构建起了 m 条双向通道，通过这些通道的时候会被收缴一定的金币，每一条道路所要收取的税金都是不一样的。

这 n 座小岛上存在 k 个特殊的小岛，这些小岛上残余着天空石毁灭天空之城时的魔力，小 W 对这些魔力非常感兴趣，但是由于她们来到这个世界时被天空石的魔力干扰，会随机降落到一个岛屿上，小 K 是个守财奴，所以你需要求出所有的岛屿到这些特殊岛屿中任意一个的最短路径。特别的，如果她们已经在特殊的岛屿上了，那么就不需要花费任何金币。

输入

第一行三个整数 n, m, k 。 ($1 \leq k \leq n \leq m \leq 2 \times 10^5$)

第二行 k 个整数，表示这些特殊岛屿的标号。

接下来 m 行，每一行三个整数 u, v, w ，表示 (u, v) 之间的通道需要收缴 w 枚金币。
($1 \leq u, v \leq n, 1 \leq w \leq 10^9$)

保证任意两个小岛间至少有一条路径。也就是说，任意的两个小岛都可以相互到达。

输出

输出 n 个整数，第 i 个数表示第 i 个岛屿到特殊岛屿所需要的最小花费。

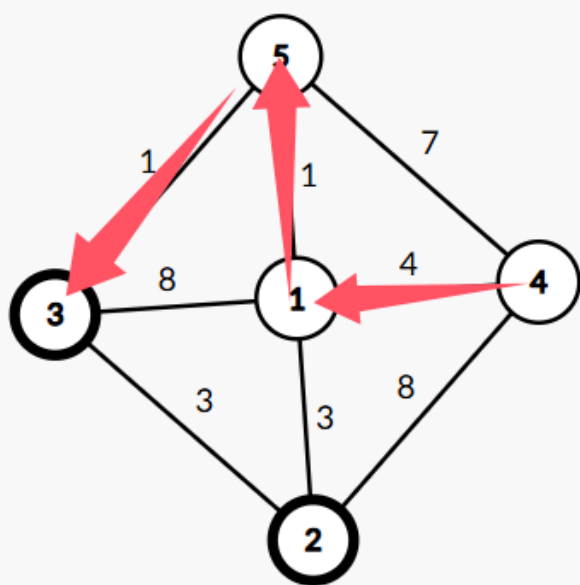
样例

序	输入样例	输出样例
#0	5 8 2	2 0 0 6 1
	2 3	
	1 2 3	
	1 3 8	
	1 4 4	
	1 5 1	
	2 4 8	
	2 3 3	
	3 5 1	
	4 5 7	

提示

路线如下：

$1 \rightarrow 5 \rightarrow 3; 4 \rightarrow 1 \rightarrow 5 \rightarrow 3; 5 \rightarrow 3。$



C. 三的倍数

时限: 1s 内存限制: 128MB

STB6 有一个长度为 n 的数字字符串 s ，他可以选择两个整数 l 和 r ，满足：

- $0 \leq l < n$
- $0 \leq r < n$
- $0 \leq l + r < n$

然后他从字符串 s 中删去前 l 个字符和后 r 个字符，得到一个新的字符串。STB6 希望这个新的字符串表示的数是 3 的倍数，请计算他有多少种不同的 (l, r) 方案可以选择。

两种方案被认为是不同的，当且仅当两者的 (l, r) 不完全相同。

输入

第一行一个整数 $n(1 \leq n \leq 2 \times 10^5)$ ，表示字符串的长度。

第二行一个长度为 n 的数字字符串，由 0-9 组成。

输出

输出一个整数，表示满足条件的 (l, r) 方案数。

样例

序	输入样例	输出样例
#0	4 3030	10

D. 数（三声）数（四声）

时限: 2s 内存限制: 256MB

给定 n, m, k , 和一个长度为 $m + 1$ 的正整数数组 v_0, v_1, \dots, v_m 。

对于一个长度为 n , 下标从 1 开始且每个元素均不超过 m 的非负整数 $\{a_i\}$, 我们定义它的权值为 $v_{a_1} \times v_{a_2} \times v_{a_3} \times \dots \times v_{a_n}$ 。

当这样的序列 $\{a_i\}$ 满足 $S = 2^{a_1} + 2^{a_2} + \dots + 2^{a_n}$ 的二进制表示中 1 的个数不超过 k 时, 我们认为 $\{a_i\}$ 是一个合法序列。

计算所有合法序列 $\{a_i\}$ 的权值和对 998244353 取模的结果。

输入

第一行三个整数 n, m, k , 意义如题。 ($1 \leq k \leq n \leq 5, 0 \leq m \leq 30$)

第二行 $m + 1$ 个整数, 分别是 v_0, v_1, \dots, v_m 。 ($1 \leq v_i \leq 998244352$)

输出

仅一行一个整数, 表示所有合法序列的权值和对 998244353 取模的结果。

样例

序	输入样例	输出样例
#0	5 1 1 2 1	40

提示

由于 $k = 1$, 而且由 $n \leq S \leq n \times 2^m$ 知道 $5 \leq S \leq 10$, 合法的 S 只有一种可能: $S = 8$, 这要求 a 中必须有 2 个 0 和 3 个 1 , 于是由 $\binom{5}{2} = 10$ 种可能的序列, 每种序列的贡献都是 $v_0^2 v_1^3 = 4$, 权值为 $10 \times 4 = 40$ 。

E. 格雷码

时限: 1s 内存限制: 128MB

通常，人们习惯将所有 n 位二进制串按照字典序排列，例如所有 2 位二进制串按字典序从小到大排列为：00, 01, 10, 11。

格雷码 (Gray Code) 是一种特殊的 n 位二进制串排列法，它要求相邻的两个二进制串间**恰好**有一位不同，特别地，第一个串与最后一个串也算作相邻。

所有 2 位二进制串按格雷码排列的一个例子为：00, 01, 11, 10。

n 位格雷码不止一种，下面给出其中一种格雷码的生成算法：

- 1 位格雷码由两个 1 位二进制串组成，顺序为：0, 1。
- $n + 1$ 位格雷码的前 2^n 个二进制串，可以由依此算法生成的 n 位格雷码（总共 2^n 个 n 位二进制串）按**顺序**排列，再在每个串前加一个前缀 0 构成。
- $n + 1$ 位格雷码的后 2^n 个二进制串，可以由依此算法生成的 n 位格雷码（总共 2^n 个 n 位二进制串）按**逆序**排列，再在每个串前加一个前缀 1 构成。

综上， $n + 1$ 位格雷码，由 n 位格雷码的 2^n 个二进制串按顺序排列再加前缀 0，和按逆序排列再加前缀 1 构成，共 2^{n+1} 个二进制串。

按该算法，2 位格雷码可以这样推出：

- 已知 1 位格雷码为 0, 1。
- 前两个格雷码为 00, 01。后两个格雷码为 11, 10。合并得到 00, 01, 11, 10。

同理，3 位格雷码可以这样推出：

- 已知 2 位格雷码为：00, 01, 11, 10。
- 前四个格雷码为：000, 001, 011, 010。后四个格雷码为：110, 111, 101, 100。合并得到：000, 001, 011, 010, 110, 111, 101, 100。

另外，我们对所有的格雷码按照顺序进行编号，编号从 0 开始，长度为 i 的格雷码接在长度为 $i - 1$ 的格雷码的后面。例如编号 0 - 10 的格雷码分别为：0, 1, 00, 01, 11, 10, 000, 001, 011, 010, 110。

现在给出一个格雷码的编号 x ，请你求出按上述算法生成的该编号格雷码所对应的二进制串。

输入

一行一个整数 $x(0 \leq x \leq 10^{18})$ ，表示格雷码的编号。

输出

一行一个二进制串表示对应的格雷码。

样例

序	输入样例	输出样例
#0	5	10
#1	11	111
#2	1145141919810	000011111101000000100100100000000110011

F. 小 C 想要玩棋盘

时限: 2s 内存限制: 256MB

小 K 和小 W 认识了一个新的朋友小 C。

这一天小 C 拿过来了一个棋盘，这是一个 $n \times m$ 的棋盘，棋盘的每一个格子里要么是黑色块，要么是白色块。

小 C 想知道如果使用黑色块能够组成多少个字母 C 的形状。特别的，构成字母 C 的块必须满足以下要求：

1. 存在坐标 (x, y) 作为整个字母块 C 的最左上角，存在坐标 (z, y) 作为字母 C 的最左下角
2. 要求从 $(x, y) \rightarrow (z, y)$ 之间的字母块都是黑色，包括两个端点，且 $x + 2 \leq z$ 。
3. 存在 $1 \leq t_1, t_2 \leq m$ ，使得 $(x, y) \rightarrow (x, y + t_1), (z, y) \rightarrow (z, y + t_2)$ 都是黑色的方块。

对于每一个 C 而言，当且仅当 x, y, z, t_1, t_2 存在一个不同时，两个 C 被称为不同的。

小 C 将这个问题抛给了小 W，小 W 觉得太简单了，所以问题被抛给了小 K，小 K 头顶尖尖的，所以她来询问你怎么解决这个问题。

输入

第一行两个整数 n, m 表示网格图的大小。 ($1 \leq n, m \leq 10^3$)

接下来 n 行长度为 m 的字符串，第 i 行表示第 i 行网格图的状态，其中 . 表示白色块，# 表示黑色块。

输出

一行一个整数，表示能够组成的不同的 C 的数量。

样例

序	输入样例	输出样例
#0	5 5 ###.. ##..# ##### ##.## ..##.#	18

G. 小 W 的书籍管理系统

时限: 2s 内存限制: 256MB

小 W 有很多书，但是她很少去收拾这些书。

小 W 家的书一共有 n 堆，每一堆都有若干本书（也可能没有），这些书参差不齐，显得十分难看。

小 K 是一个爱整齐的女孩子，她去小 W 家玩的时候发现了这堆乱七八糟的书，于是想要帮小 W 整理这些书。为了方便整理，她定义这些书的混乱程度为任意相邻两堆书的书本数量的差之和。需要注意的是，第一堆书和最后一堆书都与地面相邻（地面的高度为 0）。

例如，这几堆书的数量分别为 3, 7, 2，那么这几堆书的混乱程度为 $|0 - 3| + |3 - 7| + |7 - 2| + |2 - 0| = 14$ 。

由于小 W 不允许小 K 随意搬走她的书（因为她有一套自己的书籍管理系统），所以小 K 只能往这几堆书上面垒自己的书。

小 K 想知道最后她至少需要垒多少本书才能够让总的混乱程度小于等于 k 。

输入

第一行包含两个正整数 $n, k (n \leq 5 \times 10^5, k \leq 10^{18})$ 。

第二行包含 n 个正整数，其中第 i 个正整数表示第 i 堆书的数量 $a_i (a_i \leq 10^9)$ 。

输出

输出一行一个正整数，表示至少需要垒的书，如果无论垒多少书都无法使得总的混乱程度小于等于 k ，则输出 -1。

样例

序	输入样例	输出样例
#0	3 14	2
	3 1 7	
#1	3 13	-1
	3 7 2	

H. 妙妙数组

时限: 1s 内存限制: 128MB

STB6 有一个长为 n 的整数数组 a ，他想要选择两个连续子数组，记为 $[l_1, r_1]$ 和 $[l_2, r_2]$ ，满足 $l_1 < r_1$ 和 $l_2 < r_2$ ，且 l_1, r_1, l_2, r_2 互不相等。一个数组的权值记为这个数组的所有元素之和，STB6 希望他选择的这两个连续子数组的权值之和最大，请你帮他计算可能得到的最大值。

输入

第一行一个整数 $n(4 \leq n \leq 2 \times 10^5)$ ，表示数组大小。

第二行 n 个整数 a_1, a_2, \dots, a_n ，保证 $|a_i| \leq 10^9$ 。

输出

一行一个整数，表示 STB6 通过合理选取 l_1, r_1, l_2, r_2 ，可以获取的最大权值和。

样例

序	输入样例	输出样例
#0	6 1 1 4 5 1 4	27
#1	6 1 -2 3 4 -5 6	14

提示

样例 #0: 可以选择区间 $[1, 6]$ 和区间 $[2, 5]$ 。

样例 #1: 可以选择区间 $[1, 4]$ 和区间 $[3, 6]$ 。

两个区间的选择不一定是唯一的，但是最终的最大值是确定的。