

模拟赛

题目名称	FFT	SORT	SSSP
题目类型	传统型	传统型	传统型
目录	fft	sort	sssp
可执行文件名	fft	sort	sssp
输入文件名	fft.in	sort.in	sssp.in
输出文件名	fft.out	sort.out	sssp.out
每个测试点时限	1s	2s	4s
内存限制	512MB	512MB	512MB

编译选项: -O2 -std=c++11

FFT

小 ω 最近学习了分治 FFT，她想计算一类特殊的分治 FFT 的最小代价。

分治的过程大致如下：

```
solve(l, r):
    if r - l == 1:
        process(l)
        return
    在开区间 (l, r) 中选取一个整数 mid
    solve(l, mid)
    fft(r - l)
    solve(mid, r)
    return
```

为了方便，其中 process 函数执行时代价为 1。只有函数执行时计入代价，其余语句均**不计入**代价。

其中 fft(n) 函数的代价计算方式如下：

- 设 p 为最小的满足 $2^p \geq n$ 的整数，则代价为 $p \cdot 2^p$ 。

小 ω 希望知道在**合适地选取** mid 的值的的情况下调用 solve(0, n) 的**最小**代价（对 998244353 取模）。

小 ω 在此基础上还要对 n 进行修改，具体方法是对 n 加上或减去 2^b ，你需要在**每一次修改后**输出你的答案。

但是由于这个 n 特别大（由于**在得便机上运行**所以不用考虑使用 FFT 的合理性），将会以一种特殊的方式读入。

存在唯一一组 $(l_1, r_1, l_2, r_2, l_3, r_3, \dots, l_k, r_k)$ ，满足 $l_i \leq r_i$ ， $r_i + 1 < l_{i+1}$ ，且 $n = \sum_i \sum_{l_i \leq j \leq r_i} 2^j$ 。在输入时只输入 k 和 l_i, r_i 。

输入格式

第一行两个非负整数 k, m ，分别表示题目描述中的 k 和操作次数。

接下来 k 行每行两个非负整数 l_i, r_i 。

接下来 m 行每行两个非负整数 opt_i, b_i 表示一次操作。

$opt_i = 0$ 表示令 n 加上 2^{b_i} ； $opt_i = 1$ 表示减去。

输出格式

在每次操作**过后**输出最小代价对 998244353 取模的结果。

样例一

input

```
1 3
2 4
1 0
1 1
0 3
```

output

469
457
897

explanation

初始时 $n = 28$, 三次操作过后 n 分别变为了 27, 25, 33。

对于 $n = 27$ 的情况:

在 $\text{solve}(0, 27)$ 时选取 mid 为 15 是一种最优的方案。

在 $\text{solve}(0, 15)$ 时选取 mid 为 7 是一种最优的方案。

在 $\text{solve}(15, 27)$ 时选取 mid 为 19 是一种最优的方案。

样例二

input

5 1
2 2
5 5
7 7
9 9
15 15
1 13

output

3251728

explanation

初始时 n 的值为 33444。

限制与约定

对于所有数据, 保证 $0 \leq k, m \leq 1.5 \times 10^5$, $0 \leq b_i, l_i, r_i \leq 10^9$, $opt_i \in \{0, 1\}$, 保证 n 时刻为正。

Subtask 1(10 pts): 保证 n 时刻小于 2^{13} 。

Subtask 2(30 pts): 保证 n 时刻小于 2^{60} 。

Subtask 3(30 pts): 保证 $r_k, m, b_i \leq 10^4$ 。

Subtask 4(30 pts): 没有特殊限制。

SORT

小 ω 是一个喜欢随机的人，所以他又开始随机了。

他有一个长度为 n 的排列，他想将它排序。于是他写了一个冒泡排序。其中冒泡一次的伪代码如下：

```
for j = 2 to n do
    if a[j] < a[j - 1] then
        swap(a[j], a[j - 1])
```

这当然是对的了，于是小 ω 愉快的使用了它。

但是，他发现出现了问题！第 2 行的比较可能会返回错误的结果！

具体的来说，它有 $233/12345679$ 的概率会返回错误结果，否则它将会返回正确的结果。

小 ω 发现这样排序就会错误，于是，他只好做了 m 遍冒泡，希望提高正确率。

现在，钱菜鸡想知道冒泡 m 次后第 i 个数的期望大小，你能帮帮他吗。

输入格式

第一行两个正整数 n, m ,表示排列长度，冒泡次数。

下面一行 n 个正整数表示该排列。

输出格式

一行 n 个整数表示答案，对 12345678 取模。

样例一

input

2 1
1 2

output

234 12345447

explanation

第1个数的期望大小为 $\frac{12345912}{12345679}$ ，模意义下为234。

样例2

input

5 5
3 4 2 5 1

output

1576195 9219702 655961 7980372 5259141

限制与约定

测试点编号	分值	n =	m =
1	7	1	500
2	7	2	500
3	17	5	5
4	11	7	500
5	15	9	500
6	17	12	500
7	26	15	500

SSSP

小 ω 是一个活泼的女孩子。

外面的世界可以用一张有向图表示（有单行道），每条边有一个边权代表它的长度，而现在，小 ω 安排了 q 次出行计划，第 i 次将会从 s_i 出发，到达 t_i ，走**恰好** a_i 条边，小 ω 则想知道，这样的出行计划中，长度之和第 k_i 短的计划要走多少路。

输入格式

第一行三个正整数 n, m, q 表示有向图的点数、边数、九条可怜的出行计划个数。

下面 m 行，每行三个正整数 x_i, y_i, z_i 表示 x_i 到 y_i 有一条长度为 z_i 的边，保证没有自环，但是可能有重边。

下面 q 行，每行四个正整数 s_i, t_i, a_i, k_i ，表示一次出行计划。

输出格式

一共 q 行，每行输出一个正整数表示路径长度，如果不存在路径，那么输出-1。

样例一

input

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

output

```
6
```

样例二

input

```
2 2 1
1 2 1
1 2 1
1 2 1 2
```

output

```
1
```

限制与约定

对于所有数据，保证 $1 \leq s_i, t_i \leq n$ ， $1 \leq k \leq 15$ ， $1 \leq n, q \leq 166$ ， $1 \leq m \leq 1000000$ ， $1 \leq z_i \leq 500$ ， $1 \leq a_i \leq 10^6$ 。

Subtask 1(25)： $1 \leq n, m, q, a_i \leq 10$ 。

Subtask 2(25)： $1 \leq a_i \leq 100$ 。

Subtask 3(25): $1 \leq m \leq 100000$, $q \leq 10$.

Subtask 4(25): 没有特殊限制。