distance (1s 512MB)

给定两个可重集A和B。对于一个操作，有以下两种情况：
1.让一个A集合的元素a_i加1，即$a_i = a_i + 1$。
2.让一个B集合的元素b_i加1，即$b_i = b_i + 1$。
定义$C(A, B)$为使得A集合和B集合相同的最小操作数。如果没有办法让A集合和B集合相同，那么$C(A, B) = 0$。
现在你有两个可重集S和T，请计算$\Sigma_{A \subseteq S, B \subseteq T} C(A, B)$。答案对$998, 244, 353$取模。

注意，在可重集定义下的子集同样允许可重的元素。

输入描述

第一行一个整数n，代表S集合和T集合的元素大小。
第二行有n个整数s_i，代表可重集S的元素。
第三行有n个整数t_i，代表可重集T的元素。
注意输入的序列不一定保证有序。

输出描述

一个整数，代表答案。注意对$998, 244, 353$取模。

输入样例

3
1 2 2
1 1 3

输出样例

22

数据范围

对于30%的数据，满足$n \leq 10$。
对于60%的数据，满足$n \leq 100$。
对于100%的数据，满足$n \leq 2000, 1 \leq s_i \leq 10^5, 1 \leq t_i \leq 10^5$。

tree (1s 256MB)

给定一个n个节点和$n - 1$条边的树，每条边都有权值。一开始每个点都有颜色（白色或者黑色）。
你可以选择去反转一些点的颜色，但是会有一定花费。

我们规定价值为$\Sigma_{x\in V_1,y\in V_2}val(x,y)$，其中$V_1$是白色节点集，$V_2$是黑色节点集，$val(x,y)$是$x$点到$y$点所经过边的最大权值。
你需要反转一些节点的颜色（或者也可以不反转），从而使得价值减去花费的值最大。请输出这个答案。

输入描述

第一行一个整数n，代表这个树的节点数。
第二行有n个整数a_i，代表第i个节点的初始颜色(0为白色，1为黑色)。
第三行有n个整数$cost_i$，代表反转i号节点颜色的代价。
接下来有$n-1$行，每行都有三个正整数u_j,v_j,w_j，分别代表这条无向边的两端以及这条边的权值。
保证输入是一棵树。

输出描述

一个整数，代表最大的价值减去花费。

输入样例

3
0 0 0
1 2 3
1 2 1
2 3 2

输出样例

2

样例解释

反转第一个点的颜色。

数据范围

对于30%的数据，满足$n\leq 10$。
对于60%的数据，满足$n\leq 100$。
对于100%的数据，满足$n\leq 3000,0\leq a_i\leq 1,1\leq cost_i\leq 10^9,1\leq u_j,v_j,w_j\leq n$。

string (3s 256MB)

给定n个字符串s_i,s_i中只有$01?$三种字符。$?$代表这个位置可以同时匹配0或者1。

你需要找出有多少只含01的字符串至少匹配这些字符串中的一种。答案对$998,244,353$取模。

输入描述

第一行一个整数n。
接下来n行，每行都有一个字符串s_i。

输出描述

一个整数，代表答案，对$998,244,353$取模。

输入样例

3
10?
?0?
?11

输出样例

6

数据范围

对于前30%的数据，满足s_i只会有01两种字符。
对于接下来30%的数据，满足$n \leq 10$。
对于100%的数据，满足$n \leq 400, 1 \leq \Sigma|s_i| \leq 400$。

even (2s 512MB)

你有一个n个元素的集合$a_1, a_2, ...a_n$。

你需要回答q个询问（每个询问互不影响）。每个询问会给你三个整数l, r, k，并且会执行下面的代码：

```
a[0] = 0;
bool have_even = 0;
for(int i = l; i <= r; i++)if (a[i] % 2 == 0 && a[i] > 0 ) have_even = 1;
if (have_even)
{
          index = 0;
          for(int i = l; i <= r; i++)
          {
                    if (a[index] < a[i] && a[i] % 2 == 0) index = i;
          }
          a[index] /= 2;
}else
{
          index = l;
          for(int i = l; i <= r; i++)
          {
                    if (a[index] < a[i]) index = i;
          }
          if (a[index] > 0) a[index] = (a[index] - 1) / 2;
}
```

你现在想知道k次如上操作后，区间$[l,r]$的最大值。

输入描述

第一行两个整数n,q。
第二行有n个整数a_i，代表a_i。
接下来有q行，每行三个整数l,r,k。
这些符号的含义如题面所示。

输出描述

q行，每行一个整数。代表答案（操作后区间的最大值）。

输入样例

5 5
33 15 22 9 7
3 5 5
2 4 7
1 1 6
2 4 10
1 5 5

输出样例

7
5
0
1
15

数据范围

对于30%的数据，满足$n \leq 10, q \leq 30$。
对于60%的数据，满足$n \leq 100$。
对于100%的数据，满足$1 \leq n \leq 10^4, 1 \leq q \leq 10^5, 1 \leq a_i, k \leq 10^9, 1 \leq l, r \leq n$。