

N01 2025 模拟赛 day2

2025 年 6 月 23 日

题目名称	种树	分布式计算	药丸称重
题目类型	传统型	传统型	传统型
目录	plant	compute	pills
可执行文件名	plant	compute	pills
输入文件名	plant.in	compute.in	pills.in
输出文件名	plant.out	compute.out	pills.out
每个测试点时限	1.5 秒	2.0 秒	4.0 秒
内存限制	512 MB	512 MB	1024 MB
测试点数目	20	20	25
测试点是否等分	是	是	是

提交源程序文件名

对于 C++ 语言	plant.cpp	compute.cpp	pills.cpp
-----------	-----------	-------------	-----------

编译选项

对于 C++ 语言	-O2 -std=c++14
-----------	----------------

注意事项（请仔细阅读）

1. 选手提交的源程序请**直接放在个人目录下**，无需建立子文件夹；
2. 文件名（包括程序名和输入输出文件名）必须使用英文小写。
3. C++ 中函数 `main()` 的返回值类型必须是 `int`，值必须为 `0`。
4. 对于因未遵守以上规则对成绩造成的影响，相关申诉不予受理。
5. 若无特殊说明，结果比较方式为忽略行末空格、文末回车后的全文比较。
6. 若无特殊说明，每道题的**代码大小限制为 100KB**。
7. 程序可使用的栈空间内存限制与题目的内存限制一致。

种树 (plant)

【题目描述】

昨天小 A 把全世界所有的树都砍了，于是今天小 B 决定重新种一棵树。小 B 要种一棵 **最小生成树**。

现在有一张 n 个点的无向图，其中任意两个不同点之间都有可能有一条权值在 $1 \sim m$ 之间的边，每条边是否出现以及边的权值是独立随机的。具体的，对于每条边 (u, v) ，其有 $\frac{p_0}{100}$ 的概率不出现，出现且权值为 1 的概率是 $\frac{p_1}{100}$ ， \dots ，出现且权值为 m 的概率是 $\frac{p_m}{100}$ 。

对于这样一张随机生成的图，小 B 想知道这张图的最小生成树边权和是什么。请你对 s 从 $n-1$ 到 $m(n-1)$ 分别求出，存在最小生成树且其边权和为 s 的概率是多少？答案对 $10^9 + 7$ 取模。

【输入格式】

从文件 *plant.in* 中读入数据。

本题每个测试点有多组测试数据：

第一行一个整数 T 表示测试数据组数。

对于每组测试数据：

第一行两个整数 n, m ，表示图中节点数和最大边权。

接下来一行 $m+1$ 个正整数 p_0, p_1, \dots, p_m ， $\sum_{i=0}^m p_i = 100$ ，表示每条边不出现和对应权值的概率。

【输出格式】

输出到文件 *plant.out* 中。

对于每组测试数据，输出一行 $(m-1)(n-1)+1$ 个整数，第 i 个整数表示最小生成树存在且边权和为 $i+n-2$ 的概率，对 $10^9 + 7$ 取模。

【样例 1 输入】

```
3
3 2
0 50 50
2 1
100 0
```

5 3

25 25 25 25

【样例 1 输出】

500000004 375000003 125000001

0

784950262 273168566 557370190 190968515 146759988 943528182

888933188 870250708 589744572

【样例 2 输入/输出】

见下发文件中 *plant2.in/ans*。

【测试点约束】

对于所有测试点， $1 \leq T \leq 5$ ， $1 \leq n \leq 50$ ， $1 \leq k \leq 10$ ， $0 \leq p_i \leq 100$ ， $\sum_{i=0}^m p_i = 100$ 。

测试点	$n \leq$	$m \leq$
1 ~ 2	5	5
3 ~ 6	20	10
7 ~ 10	50	1
11 ~ 20	50	10

分布式计算 (compute)

【题目描述】

小 A 有一个由 n 个计算节点组成的网络，有 $n - 1$ 条通信线路，每条通信线路连接两个节点，并且图中任意两个节点间都可以通过若干条通信线路互相通信，简单地说，所有节点和通信线路组成一棵树。

每一个节点初始时有一个负载值 a_i 和一个数据量 l_i 。第 0 秒时， i 号节点存储的数据量恰为 l_i ，之后每一秒开始时，它会新产生一些计算结果，存储的数据量会增大 a_i 。有时候小 A 希望获取到目前为止所有的计算结果，他会选择一个计算节点 u ，要求所有节点将目前存储的所有数据都通过通信线路发送到节点 u ，代价是 $f(u) = \sum_v l_v \times \text{dis}(u, v)$ ，其中 $\text{dis}(u, v)$ 表示 u 到 v 路径上的边数。小 A 总是会选择代价最小的 u 来进行这一操作。

接下来会依次发生 Q 个事件，第 i 个事件有两种可能的类型：

- 1 $x_i \ u_i \ v_i \ w_i$ ：第 x_i 秒结束时，对于两个相邻的节点 u_i, v_i ，将 u_i 的一部分工作量分给 v_i ，使 a_{u_i} 减小 w_i ， a_{v_i} 增大 w_i 。保证 u_i, v_i 之间有边，且操作后 $a_{u_i} \geq 0$ 。
- 2 x_i ：第 x_i 秒结束时，小 A 想获取所有数据，请你求出最小代价 $\min_u f(u)$ 。

【输入格式】

从文件 `compute.in` 中读入数据。

第一行两个整数 n, Q ，分别表示节点数和事件数。

第二行 n 个整数 a_1, \dots, a_n ，表示初始时每个节点的负载值每秒增大的量。

第三行 n 个整数 l_1, \dots, l_n ，表示初始时每个节点的负载值。

接下来 $n - 1$ 行，每行两个整数 u, v 表示一条通信线路。

接下来 Q 行，每行描述一个事件，格式见题目描述，保证事件按发生顺序给出，即给出的 x_i 严格单调递增。

【输出格式】

输出到文件 `compute.out` 中。

对于每次查询 (2 类事件)，输出一行一个整数表示当前的 $\min_u f(u)$ 。

【样例 1 输入】

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

【样例 1 输出】

```
44
83
116
134
146
158
```

【样例 2 输入/输出】

见下发文件中 *compute2.in/ans*。

【样例 3 输入/输出】

见下发文件中 *compute3.in/ans*。

【测试点约束】

对于所有测试点， $1 \leq n, Q \leq 5 \times 10^5$ ， $0 \leq a_i, l_i \leq 1000$ ， $1 \leq x_i \leq 10^9$ ，1 类事件给出的 u_i, v_i 之间有边，所有事件给出的 x_i 严格单调递增。

测试点	$n, Q \leq$	特殊性质
1 ~ 4	5000	无
5 ~ 8	10^5	所有 1 类事件都发生在所有 2 类事件之前
9 ~ 10	5×10^5	$\forall 1 \leq i < n$ ，点 i 和点 $i + 1$ 之间有边
11 ~ 14	10^5	无
15 ~ 20	5×10^5	无

药丸称重 (pills)

【题目描述】

小 A 在医院工作。他面前有 n 个装药丸的罐子，第 i 个罐子中有 a_i 颗药丸。在所有的药丸中恰好有一颗是次品，它的重量比其它的药丸重，其它所有正常药丸的重量都是一样的。

小 A 想要找出那颗超重的药丸。他有一个电子秤，并且他知道一颗正常药丸的重量是多少，每次他可以选择其中一罐药丸，取出这罐药丸中任意一个药丸的子集来称重，就可以确定超重药丸是否在这个子集里面。注意，他一次只能选择一罐药丸，不能从多罐中每罐取出一些放在一起称重。

小 A 有 Q 次询问，每次小 A 假设超重的药丸一定位于第 l 罐到第 r 罐中的某一罐药丸中，请你帮小 A 确定，此时在最优策略下，最多需要多少次称重就一定能够找出超重药丸。

【输入格式】

从文件 *pills.in* 中读入数据。

第一行包含三个整数 n, m, q ，分别表示有多少罐药丸， a_i 的二进制位数，以及询问次数。

接下来 n 行，第 i 行一个长度为 m 的 01 串为 a_i 的二进制表示（从高位到低位）。

接下来 Q 行，每行两个整数 l, r 表示一次询问。

【输出格式】

输出到文件 *pills.out* 中。

对于每次询问，输出一行一个整数表示在最优策略下，最多需要多少次称重就一定能够找出超重药丸。

【样例 1 输入】

```
2 2 1
11
01
1 2
```

【样例 1 输出】

```
2
```

【样例 1 解释】

第一罐药丸有 3 颗，第二罐药丸只有 1 颗。

小 A 可以从第一罐药丸中取出 2 颗称重，如果超重药丸在这 2 颗之中就只需要再称一次就可以找出，如果不在这 2 颗中就在第一罐剩下那一颗或第二罐的那一颗之中，再称一次也可以找出，因此最多需要称 2 次。

【样例 2 输入】

```
5 3 5
100
100
101
001
111
1 3
1 3
3 3
4 4
5 5
```

【样例 2 输出】

```
4
4
3
0
3
```

【样例 3 输入/输出】

见下发文件中 *pills3.in/ans*。

【测试点约束】

对于所有测试点, $1 \leq n, Q \leq 5 \times 10^4, 1 \leq m \leq 1000, 1 \leq a_i \leq 2^m - 1, 1 \leq l \leq r \leq n$ 。

测试点	特殊性质
1 ~ 2	$n = 1$
3 ~ 6	$m = 2$
7 ~ 9	$n = 2$
10 ~ 14	$n, Q \leq 100$
15 ~ 25	无