2019年非专业级软件能力认证模拟

CCF-CSP-2019

提高级（第二轮） 第二次认证

题目名称	考试	球	树堆
题目类型	传统型	传统型	传统型
目录	test	ball	treap
可执行文件名	test	ball	treap
输入文件名	test.in	ball.in	treap.in
输出文件名	test.out	ball.out	treap.out
每个测试点时限	1.0秒	1.0秒	2.0秒
内存限制	512 MB	512 MB	512 MB
测试点数目	20	10	20
每个测试点分值	5	10	5

提交源程序文件名

对于 C++ 语言	test.cpp	ball.cpp	treap.cpp
对于 C 语言	test.c	ball.c	treap.c
对于 Pascal 语言	test.pas	ball.pas	treap.pas

编译选项

对于 C++ 语言	-lm	-lm	-lm
对于 C 语言	-lm	-lm	-lm
对于 Pascal 语言			

注意事项：

1. 考试时间：3.5小时
2. 文件名（程序名和输入输出文件名）必须使用英文小写。提交文件名为：学校名+本人姓名。
3. 除非特殊说明，结果比较方式均为忽略行末空格及文末回车的全文比较。
4. C/C++中的函数 main()的返回值类型必须是 int，程序正常结束时的返回值必须是0。
5. 需要建子文件夹
6. 只提供 Linux 格式附加样例文件。
7. 评测在 NOI Linux 下进行。
8. 编译时不打开任何优化选项。

考试（test）

【问题描述】

小 S 要参见一场考试，这场考试一共有 k 道题目。每道题目有**分值** a_i，**难度** z_i 和**类型** s_i。这些题目一共有 m 中不同的类型。

由于小 S 偏科严重，所以对不同类型的题目熟练度可能不同，在第 i 种**类型**的题目**熟练度**为 y_i。

为了简化问题，我们认为当小 S 以 y 的熟练度做难度为 z 分值为 a 的题目时，会获得 $a\cdot\left(1-\max\left(0,1-\frac{y}{z}\right)^2\right)$ 的分数。

众所周知，做题不顺利可能影响心态和发挥，在这里我们认为，如果小 S 在某一道题目的得分**低于**总分的 64%，那么接下来做题**熟练度**会**下降**。具体来说，接下来的第 i 道题熟练度会下降 c_i%。每一道题只会受到之前最后一次分数低于 64% 的影响。（就是说如果第 t 道题的分数小于满分的 64%，那么第 t+i 道题的熟练度会下降 Ci%。同时，如果有多道题分数小于满分的 64%，只有最后一次会有影响。比如说第 1 道题分数小于 64%那么第 3（即 1+2）题会下降 C2%，接下来第 4 题又低于 64%那么第 5（即 4+1）题会下降 C1%。）

根据目前的描述，已经可以确定出小 S 在每道题的得分了。但是小 S 有 n 瓶神奇的**饮料**，在做题时喝掉第 i 瓶饮料可以让这道题的熟练度**提高** x_i%。每瓶饮料只能在一道题喝，做一道题时可以喝多瓶饮料。

熟练度的下降和上升是**依次进行**，每次都按当前的百分比计算。简单来说，就是在原来的熟练度上乘 $1\pm u\%$。

现在问小 S 的总分最高是多少。

【输入格式】

从输入文件 ***test.in*** 中读入数据。

第一行三个整数 n,m,k，分别表示饮料的瓶数、题目类型的数量和试题的数量。

第二行 n 个整数 $x_1,x_2,\cdots,x_n$，表示饮料能带来的提升

第三行 m 个整数 $y_1,y_2,\cdots,y_n$，表示每种题目类型的熟练度。

第四行 $k-1$ 个整数 $c_1,c_2,\cdots,c_{k-1}$，表示做题不顺利对之后熟练度的下降程度。

接下来 k 行，每行三个整数 a_i,s_i,z_i，分别表示题目的分值、类型和难度。

【输出格式】

输出到文件 ***test.out*** 中。

输出一行一个实数，表示答案。**四舍五入**后**保留两位小数**输出。

【样例 1 输入】

```
1 1 2
10
100
50
100 1 260
100 1 200
```

【样例 1 输出】

141.72

【样例 1 解释】

一共一瓶饮料，可以选择在做第一题或第二题时喝。

如果在做第二题时喝：

做第一题的能力为 100，得分为 $100 \times \left(1 - \left(1 - \frac{100}{260}\right)^2\right) = 62\frac{22}{169}$。

由于分数低于 64，所以第二题的能力会下降 50%，又由于饮料可以提高 10%，所以实际能力为 $100 \times 50\% \times 110\% = 55$。

于是第二题的得分为 $100 \times \left(1 - \left(1 - \frac{55}{200}\right)^2\right) = 47\frac{7}{16}$。

总得分为 $109\frac{1535}{2704}$ 分。

可以类似计算出在选择在做第一题时喝的得分为 $141\frac{121}{169}$ 分，所以最高分为 $141\frac{121}{169}$ 分。

【样例 2】

见选手目录下的 ***test/test2.in*** 与 ***test/test2.ans***。

【子任务】

<table>
<tr><th>测试点</th><th>n</th><th>m</th><th>k</th><th>约定</th></tr>
<tr><td>1</td><td rowspan="6">≤ 0</td><td rowspan="6">≤ 10</td><td rowspan="2">≤ 10</td><td rowspan="11">无</td></tr>
<tr><td>2</td></tr>
<tr><td>3</td><td rowspan="2">≤ 100000</td></tr>
<tr><td>4</td></tr>
<tr><td>5</td><td rowspan="2">≤ 500000</td></tr>
<tr><td>6</td></tr>
<tr><td>7</td><td rowspan="3">≤ 1</td><td rowspan="2">≤ 5</td><td rowspan="5">≤ 15</td></tr>
<tr><td>8</td></tr>
<tr><td>9</td><td rowspan="7">≤ 10</td></tr>
<tr><td>10</td><td rowspan="4">≤ 6</td></tr>
<tr><td>11</td></tr>
<tr><td>12</td><td rowspan="3">≤ 20</td><td rowspan="4">$c_i = 0$</td></tr>
<tr><td>13</td></tr>
<tr><td>14</td><td rowspan="2">≤ 10</td></tr>
<tr><td>15</td><td rowspan="6">≤ 100</td></tr>
<tr><td>16</td><td rowspan="2">≤ 7</td><td rowspan="3">≤ 1</td><td rowspan="2">无</td></tr>
<tr><td>17</td></tr>
<tr><td>18</td><td rowspan="2">≤ 8</td><td>$x_1 = x_2 = \cdots = x_n$</td></tr>
<tr><td>19</td><td rowspan="2">≤ 10</td><td rowspan="2">无</td></tr>
<tr><td>20</td><td>≤ 10</td></tr>
</table>

对于 100%的测试点，保证$0 \le \mathrm{n} \le 10, 1 \le \mathrm{s_i} \le m \le 10, 1 \le k \le 5 \times 10^5, 0 \le c_{k-1} \le c_{k-2} \le \cdots \le c_1 \le 100, 1 \le a_i, y_i, z_i \le 10^5, 1 \le x_i \le 1000$。

球（ball）

【问题描述】

小 T 有 n 个桶和 $2n-1$ 个球，其中第 i 个桶能装前 $2i-1$ 个球。每个桶只能装一个球。

现在小 T 取了 m 个桶和 m 个球，并将这些球各自放在这些桶里。

问这样的方案有多少。

两种方案不同当且仅当**选择**了**不同的桶或球**或者同一个桶在两种方案**放了不同的球**。

由于方案的数量可能很大，所以只需要求方案数模 998244353 后的结果。

【输入格式】

从输入文件 ***ball.in*** 中读入数据。

第一行一个整数 T，表示数据组数。

接下来 T 行，每行两个整数 n,m，含义见**【问题描述】**。

【输出格式】

输出到文件 ***ball.out*** 中。

输出共 T 行，每行一个整数表示一组数据的答案。

【样例 1 输入】

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

【样例 1 输出】

```
1
4
2
18
```

【样例 1 说明】

对于 $n=m=1$ 的情况，只有选择第一个球和第一个桶，并将第一个球放在第一个桶里这一种方案。

对于 $n=2,m=2$ 的情况，会选择所有桶，第一个桶里放的一定是第一个球，于是第二个桶里可以放第二个或第三个球，共两种方案。

【样例 2 输入】

```
4
1000 1
10000 1
100000 1
1000000 1
```

【样例 2 输出】

```
1000000
100000000
17556470
757402647
```

【子任务】

<table>
<tr><th>测试点编号</th><th>n</th><th>m</th></tr>
<tr><td>1</td><td>≤ 5</td><td>≤ 5</td></tr>
<tr><td>2</td><td>≤ 10</td><td>≤ 10</td></tr>
<tr><td>3</td><td>≤ 15</td><td>≤ 15</td></tr>
<tr><td>4</td><td>≤ 100</td><td rowspan="4">≤ 100</td></tr>
<tr><td>5</td><td rowspan="4">≤ 100000</td></tr>
<tr><td>6</td></tr>
<tr><td>7</td></tr>
<tr><td>8</td><td>≤ 100000</td></tr>
<tr><td>9</td><td rowspan="2">$\le 10^7$</td><td rowspan="2">$\le 10^7$</td></tr>
<tr><td>10</td></tr>
</table>

对于 100% 的测试点，保证 $1 \le T \le 10^5, 1 \le m \le n \le 10^7$。

树堆（treap）

【问题描述】

小 D 有一棵 n 个节点的树，并给了每个节点一个 1 ~ n 的编号。所有节点的编号是<u>互不相同</u>的。

由于小 D 最近对堆十分感兴趣，所以他希望研究这棵树节点的堆性质。由于小 D 比较小，所以他研究的是<u>**小根堆**</u>。

首先，小 D 选择了一个节点作为树的根。接下来对于树的每个节点，如果它的编号是以它为根的<u>**子树中最小的**</u>，那么小 D 就称这个节点满足<u>**堆性质**</u>。

由于小 D 之前分配编号是比较随意，所以可能存在不满足<u>**堆性质**</u>的节点。

设此时树中满足<u>**堆性质**</u>的节点有 k 个，小 D 认为这样整棵树的权值是 p^k，其中 p 是一个小 D 预先设置好的常数。

现在假设小 D 是从 n! 种方案中等概率随机一种编号的方案，然后从 n 种方案中等概率随机一种选择根的方案，那么树的权值的期望是多少？

可以证明期望一定是一个分数 $\frac{x}{y}(\gcd(x,y)=1)$，你只需要求出这个分数模 998244353 的结果：$x \cdot y^{-1} \bmod 998244353 = x \cdot y^{998244351} \bmod 998244353$。

【输入格式】

从输入文件 ***treap.in*** 中读入数据。

第一行两个整数 n, p，分别表示树的大小和小 D 设置的常数。

接下来 n − 1 行，每行两个整数，表示树的一条边。

【输出格式】

输出到文件 ***treap.out*** 中。

输出一行一个整数，表示答案。

【样例 1 输入】

```
18 1
1 2
1 3
1 4
1 16
2 5
3 6
4 7
```

```
4 12
4 17
5 8
6 9
6 11
6 13
7 10
7 14
9 15
15 18
```

【样例 1 输出】

```
1
```

【样例 1 解释】

对任意的树，由于 $p^k = 1^k = 1$，所以权值都是 1，期望权值自然也是 1。

【样例 2 输入】

```
3 2
1 2
1 3
```

【样例 2 输出】

```
776412279
```

【样例 2 解释】

在所有方案中，权值为 2^1 有 4 种，权值为 2^2 有 10 种，权值为 2^3 有 4 种。

所以总权值为 80，期望权值为 $\frac{80}{18} = \frac{40}{9}$，模意义下为 776412279。

【样例 3】

见选手目录下的 ***treap/treap3.in*** 与 ***treap/treap3.ans***。

【子任务】

对于 10% 的测试点，保证 $n \le 10$。
对于 20% 的测试点，保证 $n \le 18$。
对于 30% 的测试点，保证 $n \le 100$。
对于 50% 的测试点，保证 $n \le 1000$。
对于另 10% 的测试点，保证 $p \le 1$。
对于另 10% 的测试点，保证树是一条链。
对于另 10% 的测试点，保证树上存在一个点与其他点均相连。
对于 100% 的测试点，保证 $1 \le n \le 10^6, 0 \le p < 998244353$。