

0911 BDFZ-NOIP模拟赛

题目名称	子段排序	伪快速排序	没有上司的涨薪舞会	冒泡排序
输入/输出文件名	subarray.in/out	qsort.in/out	salary.in/out	bubble.in/.out
测试点时限	2s	2s	2s	3s
内存限制	256MB	256MB	256MB	512MB
分值	100	100	100	100
测试点/子任务个数	4	10	10	10
题目类型	传统型	传统型	传统型	传统型

注意事项

1. 选手提交的源文件【**不需要建立子文件夹**】。
2. 若无特殊说明，输入文件中同一行内的多个整数、浮点数、字符串等均使用一个 空格进行分隔。
3. 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。
4. 程序可使用的栈空间大小与该题内存空间限制一致。
5. 评测时采用的机器配置为：Intel(R) Core(TM) i5-6500 CPU @ 3.20GHz，内存 8GB。上述时限以此配置为准。
6. 编译选项： `-std=c++14 -w1,--stack=536870912`

子段排序

题目描述

给定长度为 n 的数组 a 和 b 。你每次可以选择数组 a 的一个子段 $a[l \dots r]$ ，将其中的元素原地从小到大排序。例如 $a = [3, 2, 1, 5, 4]$ ，选择子段 $[2, 5]$ 之后变为 $[3, 1, 2, 4, 5]$ 。

请问是否可以利用上述操作将 a 变为 b ，若可以输出 `0`，然后给出一个操作方案；否则输出 `-1`。

输入格式

第一行1个整数 T ，代表有 T 组数据

每组数据第一行1个整数 n

第二行 n 个整数 $a[1 \dots n]$

第三行 n 个整数 $b[1 \dots n]$

保证 a, b 中所有数字出现次数相同。

输出格式

对于每组数据，首先输出一行 `0/-1`。

若答案为 `0`，第二行输出一个整数 m ，代表操作次数（你需要保证 $m \leq n^2$ ），接下来 m 行每行2个整数 (l, r) 代表一次操作，注意一个操作方案合法需要同时满足以下条件：

- $m \leq n^2$
- 每次操作 $1 \leq l \leq r \leq n$
- 所有操作之后 a, b 相等

若有多个合法方案，输出任何一个均可。

样例

样例1输入

```
1 4
2 7
3 1 7 1 4 4 5 6
4 1 1 4 4 5 7 6
5 5
6 1 3 1 5 3
7 1 1 3 3 5
8 2
9 2 1
10 1 2
11 3
12 1 2 3
13 2 1 3
```

样例1输出

1	0
2	1
3	2 6
4	0
5	2
6	2 3
7	4 5
8	0
9	1
10	1 2
11	-1

样例2输入

1	7
2	8
3	2 3 4 5 6 5 1 4
4	1 2 3 4 5 6 5 4
5	7
6	1 7 4 3 2 5 6
7	1 5 2 3 4 7 6
8	7
9	4 6 5 1 2 7 3
10	4 3 1 2 5 6 7
11	7
12	4 6 7 5 2 3 1
13	1 3 4 2 6 5 7
14	7
15	6 4 1 5 3 2 7
16	1 5 2 4 6 3 7
17	7
18	6 4 7 3 1 2 5
19	4 6 1 5 2 3 7
20	7
21	6 5 7 4 3 1 2
22	6 5 4 2 3 1 7

样例2输出

1	0
2	6
3	6 7
4	5 6
5	4 5
6	3 4
7	2 3
8	1 2
9	-1
10	-1
11	-1
12	-1
13	-1
14	-1

数据范围

对于 100% 的数据, $T \leq 10, \sum n \leq 1000, 1 \leq a[i], b[i] \leq n$, 保证 a, b 中所有数字出现次数相同。

子任务1 (20分) , $\sum n \leq 10$

子任务2 (20分) , $\sum n \leq 50$

子任务3 (30分) , $\sum n \leq 100$

子任务4 (30分) , $\sum n \leq 1000$

注意你需要保证输出方案中 $m \leq n^2$ 。

伪快速排序

题目描述

203机房电脑是老古董，经常会递归爆栈，就是递归层数太多就RE了。所以在203机房电脑上不能运行正经的快速排序程序，而只能运行【伪快速排序】程序。

伪快速排序的代码如下：

```
1  int a[MAXN];
2  int partition(int l, int r){//这部分和快排相同
3      int x = a[r];
4      int i = l;
5      for(int j=l;j<r;j++){
6          if(a[j] < x){
7              swap(a[i], a[j]);
8              ++i;
9          }
10     }
11     swap(a[i], a[r]);
12     return i;
13 }
14
15 void Qsort(int l, int r, int h){
16     if(l<r && h>1){
17         int m = partition(l,r);
18         Qsort(l,m-1,h-1);
19         Qsort(m+1,r,h-1);
20     }
21 }
```

给定正整数 n, k ，你想知道有多少个 $1, 2, \dots, n$ 的排列 $a[1\dots n]$ 在调用 `Qsort(1, n, k)` 之后能变为有序。

输入格式

输入包含多组测试数据。

第一行包含两个整数 T, q ，依次表示测试数据的数量和一个用于输出的模数。

接下来 T 行，每行描述一组测试数据，包含两个正整数 n, k ，依次表示排列的长度以及递归的最大深度。

输出格式

对于每组测试数据，输出一行一个整数表示这组测试数据的答案对 q 取模的值。

样例

样例输入1

```
1 4 998244353
2 4 1
3 4 2
4 4 3
5 4 4
```

样例输出1

1	1
2	8
3	20
4	24

样例输入2

1	5	10007
2	10	4
3	40	3
4	30	2
5	20	10
6	300	20

样例输出2

1	6401
2	2733
3	5369
4	3539
5	8445

数据范围

对于测试点1-2, $n, k \leq 5$

对于测试点3, $n \leq 50, k = 1$

对于测试点4, $n \leq 50, k = 2$

对于测试点5-6, $n \leq 13$

对于测试点7-10, $n, k \leq 300$

对于100%的数据, $T \leq 500, 1 \leq n, k \leq 300, 10^8 \leq q \leq 10^9$

没有上司的涨薪舞会

题目描述

一个公司有 n 名员工，1 没有直接上司，其余员工都有一个直接上司。这样的直接上司关系构成以 1 为根的有根树。已知这颗树。一个员工的直接下属，指以他为直接上司的所有员工。一个员工的下属，指他子树内除他以外的所有员工。

这些员工都十分渴望涨薪，所以有些人会要求涨薪。记 $s[u]$ 为 u 是否要求涨薪，“是”为 1，“否”为 0。不知道序列 $s[1\dots n]$ 。

现在所有员工都被邀请去了一个舞会。但是并不是所有人都十分想去，而且他们是否参加会以自己的直接下属参加情况为参考。每个人 u 都会等待自己所有下属都决定完，然后：

- 若不存在直接下属决定参加舞会，则 u 以 $p[u]$ 的概率参加舞会；
- 否则， u 一定不参加舞会。

已知序列 $p[1\dots n]$ 。

这个公司里，每位员工都有权力和义务给自己的下属涨薪/降薪。在这个舞会上，每位参加舞会的员工 u 会对每个要求涨薪并且参加舞会的下属 v 涨薪 $a[u]$ 。这个值当然可以是负的，因为有些人心情很差，专门给要求涨薪的人降薪。已知序列 $a[1\dots n]$ 。

求使公司所有员工涨薪量之和期望最大的序列 $s[1\dots n]$ 。不用输出序列 s ，只用输出期望的大小。保证答案在 10^{11} 以内。

输入格式

第一行一个整数 n 表示员工数目。

第二行 $n - 1$ 个整数 $fa[u]$ ，表示 $2 - n$ 员工每个员工的直接上司。直接上司编号小于自己的编号。

第三行 n 个小数 $p[u]$ ，表示每个员工的概率， $0 < p[u] < 1$ 。

第四行 n 个整数 $a[u]$ ，表示每个员工对于自己每个有要求的下属的涨薪幅度。

输出格式

第一行1个小数 ans 表示答案，输出到小数点后第6位。

样例

样例输入1

```
1 | 5
2 | 1 1 3 3
3 | 0.20 0.60 0.60 0.40 0.40
4 | 3 2 0 4 -3
```

样例输出1

```
1 | 0.192000
```

样例输入2

1	10
2	1 2 1 4 2 4 7 4 4
3	0.60 0.80 0.10 0.20 0.60 0.40 0.80 0.30 0.70 0.70
4	0 10 8 4 -5 4 8 4 -5 9

样例输出2

1	0.008640
---	----------

样例3-4

见下发样例。

数据范围

对于测试点1, $n \leq 20$

对于测试点2-4, $n \leq 10^3$

对于测试点5, $n \leq 10^5$, 树是以1为根的完全二叉树

对于测试点6-7, $n \leq 10^5$

对于测试点8-10, $n \leq 5 \times 10^5$

对于 100% 的数据, $1 \leq n \leq 5 \times 10^5, 0 < p[u] < 1, |a[u]| \leq 10^4$, 保证答案 $0 < ans \leq 10^{11}$ 。

冒泡排序

题目描述

对于一个排列 $a[1\dots n]$ ，进行一趟冒泡排序的代码为：

```
1  for(int i=1;i<n;++i){
2      if(a[i]>a[i+1]) swap(a[i], a[i+1]);
3  }
```

在进行 $n - 1$ 趟冒泡排序之后，数组变为有序。

给一个长度为 n 的排列 $a[1\dots n]$ 和 q 次询问，每次询问形如 (k, x) ：代表询问 x 这个数在 k 趟排序之后的位置下标。

输入格式

- 第一行1个整数 n
- 第二行 n 个整数 $a[1\dots n]$ ，保证是一个排列
- 第三行1个整数 q
- 接下来 q 行，每行2个整数 (k, x) 代表一次询问

输出格式

输出 q 行，每行1个整数代表答案

样例

样例输入1

```
1  5
2  4 3 5 1 2
3  3
4  2 1
5  4 4
6  1 5
```

样例输出1

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

样例输入2

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

样例输出2

1	2
2	2
3	3
4	4
5	5

样例3

见下发样例。

数据范围

对于20%的数据, $n, q \leq 2000$ 。

对于另20%的数据, $n, q \leq 10^5$, 不同的 k 取值不超过20种。

对于另20%的数据, $n, q \leq 10^5$, 不同的 x 取值不超过20种。

对于另20%的数据, $n, q \leq 10^5$ 。

对于100%的数据, $1 \leq n, q \leq 5 \times 10^5, 1 \leq k < n, 1 \leq x \leq n$, $a[1...n]$ 保证是一个排列。

注意输入输出量较大, 请使用快速的读写方式。