水题(water)

Time Limit:1000ms Memory Limit:128MB

题目描述

LYK出了道水题。

这个水题是这样的：有两副牌，每副牌都有n张。

对于第一副牌的每张牌长和宽分别是xi和yi。对于第二副牌的每张牌长和宽分别是aj和bj。第一副牌的第i张牌能覆盖第二副牌的第j张牌当且仅当xi>=aj并且yi>=bj。（注意牌不能翻转）当然一张牌只能去覆盖最多一张牌，而不能覆盖好多张。

LYK想让两副牌的各n张一一对应叠起来。它想知道第二副牌最多有几张能被第一副牌所覆盖。

输入格式(water.in)

第一行一个数n。

接下来n行，每行两个数xi,yi。

接下来n行，每行两个数aj,bj。

输出格式(water.out)

输出一个数表示答案。

输入样例

3

2 3

5 7

6 8

4 1

2 5

3 4

输出样例

2

数据范围

对于50%的数据n<=10。

对于80%的数据n<=1000。

对于100%的数据1<=n<=100000，1<=xi,yi,aj,bj<=10^9。

梦境(dream)

Time Limit:1000ms Memory Limit:128MB

题目描述

LYK做了一个梦。

这个梦是这样的，LYK是一个财主，有一个仆人在为LYK打工。

不幸的是，又到了月末，到了给仆人发工资的时间。但这个仆人很奇怪，它可能想要至少x块钱，并且当LYK凑不出恰好x块钱时，它不会找零钱给LYK。

LYK知道这个x一定是1~n之间的正整数。当然抠门的LYK只想付给它的仆人恰好x块钱。但LYK只有若干的金币，每个金币都价值一定数量的钱（注意任意两枚金币所代表的钱一定是不同的，且这个钱的个数一定是正整数）。LYK想带最少的金币，使得对于任意x，都能恰好拼出这么多钱。并且LYK想知道有多少携带金币的方案总数。

具体可以看样例。

输入格式(dream.in)

第一行一个数n，如题意所示。

输出格式(dream.out)

输出两个数，第一个数表示LYK至少携带的金币个数，第二数表示方案总数。

输入样例

6

输出样例

3 2

样例解释

LYK需要至少带3枚金币，有两种方案，分别是{1,2,3}，{1,2,4}来恰好得到任意的1~n之间的x。

输入样例2

10

输出样例2

4 8

数据范围

对于30%的数据n<=10。

对于60%的数据n<=100。

对于100%的数据n<=1000。

动态规划(dp)

Time Limit:1000ms Memory Limit:128MB

题目描述

LYK在学习dp，有一天它看到了一道关于dp的题目。

这个题目是这个样子的：一开始有n个数，一段区间的价值为这段区间相同的数的对数。我们想把这n个数切成恰好k段区间。之后这n个数的价值为这k段区间的价值和。我们想让最终这n个数的价值和尽可能少。

例如6个数1,1,2,2,3,3要切成3段，一个好方法是切成[1],[1,2],[2,3,3]，这样只有第三个区间有1的价值。因此这6个数的价值为1。

LYK并不会做，丢给了你。

输入格式(dp.in)

第一行两个数n,k。

接下来一行n个数ai表示这n个数。

输出格式(dp.out)

一个数表示答案。

输入样例

10 2

1 2 1 2 1 2 1 2 1 2

输出样例

8

数据范围

对于30%的数据n<=10。

对于60%的数据n<=1000。

对于100%的数据1<=n<=100000,1<=k<=min(n,20),1<=ai<=n。

其中有30%的数据满足ai完全相同均匀分布在所有数据中。