2019 年 NOIP 模拟

第二试

题目名称	极好的问题	背包问题	子树问题
题目类型	传统型	传统型	传统型
目录	awesome	bag	subtree
可执行文件名	awesome	bag	subtree
输入文件名	awesome.in	bag.in	subtree.in
输出文件名	awesome.out	bag.out	subtree.out
每个测试点时限	1.0 秒	1.0 秒	1.0 秒
内存限制	512 MB	710 MD	710 MD
1.3.4.4.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.	312 MD	512 MB	512 MB
测试点/包数目	20	10	25

提交源程序文件名

对于 C++ 语言	awesome.cpp	bag.cpp	subtree.cpp
对于 C 语言	awesome.c	bag.c	subtree.c
对于 Pascal 语言	awesome.pas	bag.pas	subtree.pas

编译选项

对于 C++ 语言	-02
对于 C 语言	-02
对于 Pascal 语言	-02

极好的问题 (awesome)

【题目描述】

Yazid 有一个序列 A_i ($1 \le i \le n$) 和一个**质数** P。对于一个三元组 (x,y,z),如果满足 $x \le y \le z$ 、 $x \times y \times z \mod P = 1$,且存在三个**互不相同**的下标 i,j,k 使得 $A_i = x$, $A_i = y$, $A_k = z$,那么我们就说这个三元组是极好的。

现在,请你求出本质不同的极好的三元组的数目。两个三元组 (x_1,y_1,z_1) $,(x_2,y_2,z_2)$ 被认为是本质不同的,当且仅当 $x_1 \neq x_2$ 或 $y_1 \neq y_2$ 或 $z_1 \neq z_2$ 。

如果你能自如地运用你学习的算法解决这个极好的问题,那么你就会获得"Wow"的赞美,这将是你独享的时刻。

【输入格式】

从文件 awesome.in 中读入数据。

第一行 2 个用空格隔开的整数 n, P。

第二行 n 个用空格隔开的整数 A_1, \ldots, A_n 。

【输出格式】

输出到文件 awesome.out 中。

输出一行一个整数,表示极好的三元组的数目。

【样例 1 输入】

5 5

1 1 1 2 3

【样例 1 输出】

2

【样例1解释】

唯二的极好的三元组为 (1,1,1) 和 (1,2,3)。

【样例 2 输入】

4 5

7 4 2 2

【样例 2 输出】

2

【样例2解释】

(2,2,4) 和 (2,4,7) 都是极好的三元组。

【样例 3】

见选手目录下的 awesome/awesome3.in 与 awesome/awesome3.ans。

【样例 4】

见选手目录下的 awesome/awesome4.in 与 awesome/awesome4.ans。

【子任务】

测试点编号	n =	是否保证 $A_i < P$
1	3	
2,3,4,5,6	100	是
7,8,9,10,11,12,13,14	2,333	
15,16,17,18,19,20	∠, ააა	否

对于所有测试点,保证 $3 \le n \le 2,333$, $0 \le A_i, P < 2^{30}$, P 为质数。

背包问题(bag)

【题目背景】

Yazid 喜欢背包问题,所以就有了这道题

【题目描述】

Yazid 有 n 个物品,编号从 $1 \subseteq n$ 。其中编号为 i 的物品的重量为 w_i ,其价值为 v_i 。 Yazid 还有 m 个背包,编号从 $1 \subseteq m$ 。其中编号为 i 的背包承重上限为 t_i 。

Yazid 希望你选出若干个物品,同时选出**同等数量**的背包,并在每个背包中放入**恰好一个**物品。对于每个背包,你都需要保证放入其中的物品重量不超过其承重上限。

接下来,你需要将这些背包从左到右排成一排。如果任意相邻的两个背包,都满足左边背包中物品的重量**不超过**右边背包中物品的重量,且左边背包中物品的价值**不超过**右边背包中物品的价值,那么我们说这个背包排列是**优秀的**。

你需要挑选出尽可能多的物品,使得你可以顺利挑选背包、将这些物品按规则放 入、并构造出一个优秀的背包排列。

方便起见,你只需要输出最多能够挑选出的物品数量即可。

【输入格式】

从文件 baq.in 中读入数据。

单个测试点中包含多组数据,输入第一行为一个非负整数 T,描述数据组数。接下来依次描述每组数据,对于每组数据:

第一行一个非负整数n,描述物品数量。

第 2 行至第 n+1 行,每行两个用空格隔开的正整数,其中第 i+1 行的两个数依次为 w_i, v_i ,分别描述第 i 个物品的重量和价值。

接下来一行一个非负整数m,描述背包数量。

接下来一行 m 个用空格隔开的正整数 t_1, \ldots, t_m ,依次描述各背包的承重上限。

【输出格式】

输出到文件 baq.out 中。

一行一个整数,表示能够选出的物品数量的最大值。

【样例 1 输入】

2

4

- 1 200
- 2 300
- 5 400
- 3 100

4

1 10 5 1

4

- 1 200
- 2 300
- 5 400
- 10 500

5

1 2 3 4 5

【样例1输出】

3

3

【样例1解释】

对于第一组数据:选出前3个物品,并分别把它们放入前3个背包,再按1,2,3的顺序排列三个物品即可。

对于第二组数据,没有背包能放置编号为4的物品。

【样例 2】

见选手目录下的 bag/bag2.in 与 bag/bag2.ans。

【子任务】

测试点编号	<i>n</i> ≤	<i>m</i> ≤
1	5	5
2	10	10
3	20	20
4	100	100
5	500	500
6	1000	1000
7	5000	5000
8,9,10	10^{5}	10^{5}

对于所有测试点,保证 $T \le 5$ 。

对于所有测试点中的每组数据,保证 $n \le 10^5$, $m \le 10^5$,所有 $1 \le w_i, v_i, t_i \le 10^9$ 。

子树问题 (subtree)

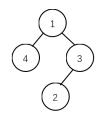
【题目描述】

对于一棵**有根树**(设其节点数为 n,则节点编号从 $1 \subseteq n$),如果它满足所有非根节点的编号均比起父亲**更大**,我们就说它是 Yazid 树。

此外,Yazid 给出了 k 个整数 a_1, \ldots, a_k ,并规定,只要一棵有根树存在一个子树包含的节点数恰好为 a_1, \ldots, a_k 中的某一个值,那么它就是坏的;相对地,不坏的有根树就是好的。

现给定 n、k, a_1, \ldots, a_k ,并额外给定整数 L, R,请你对于 d = L, $L+1, \ldots, R$,分别 求出 n 个节点的深度为 d 的**好的 Yazid 树**的数量。

有根树的深度定义的是所有节点到达根所需经过的节点数(包含起点和终点)。例如,下图的有根树(根节点为1号节点)深度即为3,且恰好它不是一棵 Yazid 树:



两棵有根树被认为是不同的,当且仅当存在一个节点编号 x,该节点编号在两棵树中对应的父亲是不同的。

由于答案可能很大,请输出它们对998,244,353取模的结果。

【输入格式】

从文件 subtree.in 中读入数据。

第一行 2 个用空格隔开的整数 n.k。

接下来 k 行,每行一个整数,依次为 a_1,\ldots,a_k 。

接下来一行 2 个用空格隔开的整数 L,R。

保证 $1 \le L \le R \le n$, $2 \le a_1, \ldots, a_k \le n$ 且互不相同。

【输出格式】

输出到文件 subtree.out 中。

输出用单个空格隔开的 R-L+1 个整数,依次表示深度为 L,L+1,...,R 的好的 Yazid 树数目对 998,244,353 取模的结果。

【样例1输入】

- 3 0
- 1 3

【样例1输出】

0 1 1

【样例1解释】

显然不存在 3 个节点的深度为 1 的有根树;唯二节点数目为 3 的 Yazid 有根树深度分别为 2 和 3。

【样例 2 输入】

- 4 1
- 2
- 2 4

【样例 2 输出】

1 1 0

【样例3输入】

- 4 1
- 4
- 1 4

【样例3输出】

0000

【样例3解释】

节点个数为4的有根树必定存在大小为4的子树,因此答案均为0。

【样例 4 输入】

10 3

5

7

9

1 9

【样例4输出】

0 1 19183 94549 48006 6930 315 0 0

【样例 5】

见选手目录下的 *subtree/subtree5.in* 与 *subtree/subtree5.ans*。

【子任务】

测试点编号	$n \leq$	k	特殊限制
1,2,3,4,5,6	10	< n	无
7,8,9,10,11			R=3
12,13,14,15,16	100	=0	L = n - 2
17,18,19	100		
20,21,22		< n	无
23,24,25	500	< n	

对于所有测试点,保证 $0 \le k < n \le 500$ 。