

全国青少年信息学奥林匹克（NOI2017）

山东省队第二轮选拔赛

第二试

考试时间：2017 年5 月18 日8：00—13：00

对于C/C++程序，最终测试将开启 -O2 优化

Problem A. 天才黑客 (hack.c/cpp/pas)

Input file: hack.in
 Output file: hack.out
 Time limit: 3 seconds
 Memory limit: 512 megabytes

SD0062号选手小Q同学为了偷到SDOI7012的试题，利用高超的黑客技术潜入了SDOI出题组的内联网的中央控制系统，然而这个内联网除了配备有中央控制系统，还为内联网中的每条单向网线设定了特殊的通信口令，这里通信口令是一个字符串，不同网线的口令可能不同。这让小Q同学感觉有些棘手，不过这根本难不倒他，很快他就分析出了整个内联网的结构。

内联网中有 n 个节点（从1到 n 标号）和 m 条单向网线，中央控制系统在第1个节点上，每条网线单向连接内联网中的某两个节点，从1号节点出发经过若干条网线总能到达其他任意一个节点。每个节点都可以运行任意的应用程序，应用程序会携带一条通信口令，当且仅当程序的口令与网线的口令相同时，程序才能通过这条网线到达另一端的节点继续运行，并且通过每条网线都需要花费一定的时间。

每个应用程序可以在任意一个节点修改通信口令，修改通信口令花费的时间可以忽略不计，但是为了减小修改量，需要先调用一个子程序来计算当前程序的口令和网线的口令的最长公共前缀（记其长度为 len ），由于获取网线的口令的某个字符会比较耗时，调用一次这个子程序需要花费 len 个单位时间。

除此之外，小Q同学还在中央控制系统中发现了一个字典，每条网线的口令都是字典中的某个字符串。具体来说，这个字典是一棵 k 个节点（从1到 k 标号）的有根树，其中根是第1个节点，每条边上有一个字符，字符串 S 在字典中当且仅当存在某个点 u 使得从根节点出发往下走到 u 的这条路径上的字符顺次拼接构成 S 。

现在小Q同学在1号节点同时开启了 $n - 1$ 个应用程序，这些应用程序同时运行且互不干扰，每个程序的通信口令都为空，他希望用最短的时间把这些程序分别发送到其他节点上，你需要帮小Q同学分别计算出发送到第 i ($= 2, 3, \dots, n$)个节点的程序完成任务的最短时间。

Input

第一行是一个正整数 T ，表示测试数据的组数，

对于每组测试数据，

第一行是三个整数 n, m, k ，分别表示内联网的节点数、内联网的网线条数、字典树的节点数，

接下来 m 行，每行包含四个整数 a_i, b_i, c_i, d_i ($1 \leq a_i, b_i \leq n, 0 \leq c_i \leq 20000, 1 \leq d_i \leq k$)，表示沿着这条网线可以从第 a_i 个节点花费 c_i 个单位时间到达第 b_i 个节点，网线的口令是由从字典树的根到 d_i 这个点的路径上的字符顺次拼接构成的字符串（可能为空），需要注意的是这个内联网可能有自环和重边，

接下来 $k - 1$ 行，每行包含三个整数 u_i, v_i, w_i ($1 \leq u_i, v_i \leq k, 1 \leq w_i \leq 20000$)，表示字典树上有一条 $u_i \rightarrow v_i$ 的边，边上有字符 w_i ，保证给出的边构成一棵以1为根的有根树，并且每个点连出去的边上的字符互不相同。

Output

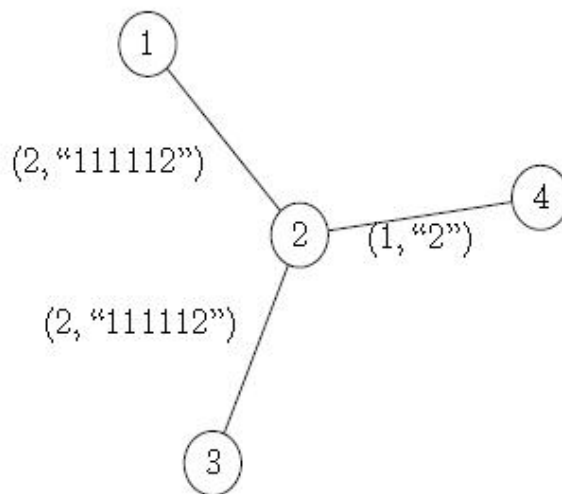
对于每组测试数据，输出 $n - 1$ 行，第 i 行表示发送到第 $i + 1$ 个节点的程序完成任务的最短时间。

Examples

hack.in	hack.out
1	2
4 4 6	7
1 2 2 5	3
2 3 2 5	
2 4 1 6	
4 2 1 6	
1 2 1	
2 3 1	
3 4 1	
4 5 2	
1 6 2	

Notes

对于样例，从1到3的一条可行路径是 $1 \rightarrow 2 \rightarrow 3$ ，所需时间是 $(2 + lcp("", "1112")) + (2 + lcp("1112", "1112")) = 8$ ，但这条路径不是最优的，最优路径是 $1 \rightarrow 2 \rightarrow 4 \rightarrow 2 \rightarrow 3$ ，



样例中内联网的结构

对于100%的数据， $T \leq 10$ ， $2 \leq n \leq 50000$ ， $1 \leq m \leq 50000$ ， $1 \leq k \leq 20000$ ，保证满足 $n > 5000$ 或 $m > 5000$ 的数据不超过2组。

测试点编号	n	m	k	
1,2,3,4,5	≤ 5000	≤ 5000	≤ 20000	
6,7,8,9,10,11,12,13,14	≤ 50000	≤ 50000	≤ 20000	$nk \leq 200000$
15,16,17,18,19,20	≤ 50000	≤ 50000	≤ 20000	

Problem B. 遗忘的集合 (set.c/cpp/pas)

Input file: set.in
 Output file: set.out
 Time limit: 5 seconds
 Memory limit: 512 megabytes

小Q在他的个人主页上放出了一个悬赏：征集只含正整数的非空集合 S ，其中的每个元素都不超过 n ，并且满足一些附加条件。

众所周知，我们可以很轻松地对于任意不超过 n 的正整数 x ，计算出把 x 表示成 S 中元素之和的方案数 $f(x)$ ，在这里我们约定，在任意方案中每个数字可以出现多次，但是不考虑数字出现的顺序。

例如，当 $S = \{1, 2, 3, 4, 5\}$ 时，我们可以计算出 $f(2) = 2, f(3) = 3, f(4) = 5, f(5) = 7$ 。

再例如，当 $S = \{1, 2, 5\}$ 时，我们可以计算出 $f(4) = 3, f(5) = 4, f(6) = 5, f(7) = 6$ 。

麻烦地是现在小Q忘记了 S 里有哪些元素，幸运地是他用存储设备记录下了所有 $f(i) \bmod p$ 的值，小Q希望你能利用这些信息帮他恢复出 S 原来的样子。

具体来说，他希望你找到这样一个正整数的非空集合 S ，其中的每个元素都不超过 n ，并且对于任意的 $i = 1, 2, \dots, n$ ，满足把 i 表示成 S 中元素之和的方案数在模 p 意义下等于 $f(i)$ ，其中 p 是记录在存储设备中的一个质数。他向你保证：一定存在这样的集合 S 。

然而，小Q觉得他存储的信息并不足以恢复出唯一的 S ，也就是说，可能会存在多个这样的集合 S ，所以小Q希望你能给出所有解中字典序最小的解。

对于满足条件的两个不同的集合 S_1 和 S_2 ，我们认为 S_1 的字典序比 S_2 的字典序小，当且仅当存在非负整数 k ，使得 S_1 的前 k 小元素与 S_2 的前 k 小元素完全相等，并且，要么 S_1 的元素个数为 k ，且 S_2 的元素个数至少为 $(k+1)$ ，要么 S_1 和 S_2 都有至少 $(k+1)$ 个元素，且 S_1 的第 $(k+1)$ 小元素比 S_2 的第 $(k+1)$ 小元素小。

Input

第一行包含两个整数 n 和 p ，满足 p 是质数。

第二行包含 n 个整数 $f(1), f(2), \dots, f(n)$ ，含义见题目描述。

保证每一行中相邻的整数由恰好一个空格隔开，并且末尾没有多余的空格。

Output

你需要输出两行信息来描述字典序最小的解，其中第一行包含一个整数 m ($m > 0$)，表示 S 的元素个数，第二行包含 m 个正整数 s_1, s_2, \dots, s_m ，表示将 S 的所有元素按升序排序后得到的序列。

你需要保证输出的每一行中相邻的整数由恰好一个空格隔开，并且每一行的末尾没有多余的空格。

Examples

set.in	set.out
5 1000003	5
1 2 3 5 7	1 2 3 4 5

set.in	set.out
9 1000003	3
1 2 2 3 4 5 6 7 8	1 2 5

Notes

对于 100% 的数据，有 $1 \leq n < 2^{18}$, $10^6 \leq p < 2^{30}$, $0 \leq f(i) < p$ ($i = 1, 2, \dots, n$)。

测试点编号	n	p	特殊约定
1	$n = 5$	$p = 1000003$	无特殊约定
2	$n \leq 20$	同最大限制	
3	$n \leq 25$		
4			
5	$n \leq 5000$		
6			
7	$n \leq 8000$	$p = 1000003$	无特殊约定
8		$p = 1000000007$	
9		同最大限制	
10			$m = s_m$
11		同最大限制	
12			
13			
14			
15		$p = 998244353$	无特殊约定
16	$p = 991668907$		
17	$p = 1000000007$		
18	同最大限制		
19			
20			

Problem C. 文本校正 (fix.c/cpp/pas)

Input file: fix.in
 Output file: fix.out
 Time limit: 1 seconds
 Memory limit: 512 megabytes
 Feedback: **Special Judge**

小Q在研发一种数据混淆的算法时不慎将重要的文档都给混淆了。幸运的是，将这些文档校正对于他来说并不是难事。他凭借着敏锐的观察力成功地用肉眼完成了校正。

为了防止这种情况再次发生，小Q希望开发一种文本校正工具，他的目标是将一个文本串 T 分成连续的3段，要求每段都不能为空，然后按一定顺序将这3段从左往右拼接起来，将其还原为初始文本串 S 。

在进行了大量肉眼校正工作之后，小Q需要休息一下，因此他把这个任务交给你。请写一个程序，判断是否可以还原，并给出一个合法的还原方案。

Input

第一行包含一个正整数 $Case$ ，表示需要进行的校正次数。

接下来 $Case$ 个部分依次表示每次校正工作，每个部分第一行包含两个正整数 n, m ，分别表示文本串的长度以及字符集的大小。

每个部分第二行包含 n 个正整数 S_1, S_2, \dots, S_n ，表示 S 串。

每个部分第三行包含 n 个正整数 T_1, T_2, \dots, T_n ，表示 T 串。

Output

对于每次校正工作，若无解，则仅输出一行“NO”(不含引号)，否则第一行输出“YES”(不含引号)，接下来三行每行两个正整数 l_i, r_i ，按拼接顺序依次表示 T 的3个子串。

若存在多种还原方案，请输出任意一种。

Examples

fix.in	fix.out
3	YES
5 3	5 5
2 1 1 1 1	1 3
1 1 1 1 2	4 4
5 5	NO
5 2 3 3 4	YES
2 5 3 4 3	2 2
5 5	1 1
4 5 2 1 4	3 5
5 4 2 1 4	

Notes

对于100%的数据, $3 \leq n \leq 1000000, 1 \leq S_i, T_i \leq m \leq 1000000$ 。

测试点编号	n, m	$Case$	$\sum n, \sum m$
1,2	≤ 6	≤ 50000	≤ 300000
3,4,5,6	≤ 2000	≤ 10	≤ 20000
7,8,9,10,11,12	≤ 200000	≤ 30	≤ 200000
13,14,15,16,17,18,19,20	≤ 1000000	≤ 30	≤ 1000000