杂题选讲

2021.08.20

Hanyuan Shi

subset

给出n个数，要求求出两个不相交的subset，使得他们的和一样。

n, a[i] <= 1e5

subset

考虑22个数字就可以啦！

2^22 = 4194304 >= 22 * 100000

根据鸽笼原理，前 22 个数中一定有两个 subset 的和一样大！

把公共部分剪掉就可以啦！

City Brain(ICPC EC-Final 2020 D)

Prof. Pang works for the City Brain program of Capital Grancel. The road network of Grancel can be represented by an undirected graph. Initially, the speed limit on each road is 1m/s. Prof. Pang can increase the speed limit on a road by 1 m/s with the cost of 1 dollar. Prof. Pang has k dollars. He can spend any nonnegative integral amount of money on each road. If the speed limit on some road is am/s, it takes $1/a$ seconds for anyone to go through the road in either direction.

After Prof. Pang spent his money, Prof. Du starts to travel from city s_1 to city t_1 and Prof. Wo starts to travel from city s_2 to city t_2. Help Prof. Pang to spend his money wisely to minimize the sum of minimum time of Prof. Du's travel and Prof. Wo's travel. It is guaranteed that s_1 and t_1 are connected by at least one path and that s_2 and t_2 are connected by at least one path.

Input

The first line contains three integers n, m, k $(1 \leq n \leq 5000, 0 \leq m \leq 5000, 0 \leq k \leq 10^9)$ separated by single spaces denoting the number of vertices, the number of edges in the graph and the number of dollars Prof. Pang has.

Each of the following m lines contains two integers a, b $(1 \leq a, b \leq n, a \neq b)$ separated by a single space denoting the two endpoints of one road. There can be multiple roads between the same pair of cities.

The following line contains four integers s_1, t_1, s_2, t_2 $(1 \leq s_1, t_1, s_2, t_2 \leq n)$ separated by single spaces denoting the starting vertices and ending vertices of Prof. Du and Prof. Wo's travels.

Output

Output one decimal in the only line – the minimum sum of Prof. Du's travel time and Prof. Wo's travel time. The answer will be considered correct if its absolute or relative error does not exceed 10^{-9}.

City Brain(ICPC EC-Final 2020 D)

1. 两条路径的公共部分一定是连续的一段！ 为什么呢？
2. 枚举公共部分的起点和终点，我们可以知道这种情况下公共部分的长度和个人部分的长度！
3. 可以通过二分算出钱怎么分配！
4. 这样复杂度是 O(n^2logn) 的，会 TLE！
5. 在公共部分长度确定的情况下，个人部分长度越短越好！
6. 所以我们只要二分 O(n) 次！

Mr. Panda and Typewriter(CCPC Final 2019 C)

Mr. Panda recently got a brand-new typewriter as a birthday gift from Mr. Champion. Mr. Panda likes the typewriter so much. He wants to use it to type a thank you letter S and mail it to Mr. Champion.

To type the thank you letter, Mr. Panda starts with an empty string on a white-paper, and the following operations are allowed to perform by using the typewriter:

- Spend X units of time to add any single character to the end of Mr. Panda's string.
- Spend Y units of time to copy any substring of Mr. Panda's string (that is, all of the sequential characters between some start point and some end point in Mr. Panda's string) to the clipboard. Doing this overwrites whatever was in the clipboard before. The clipboard starts off empty.
- Spend Z units of time to add the entire contents of the clipboard to the end of Mr. Panda's string. (The contents of the clipboard do not change.)

Mr. Panda needs to make his string exactly the same as the contents in the thank you letter S. Note that Mr. Panda must create exactly the thank you letter with no additional character.

Mr. Panda wants to find a way to type the thank you letter with the minimum amount of spent time. Because Mr. Panda is too lazy, he asks for your help.

Could you please help Mr. Panda find an optimized way to type the thank you letter so that the amount of time spent is minimized? Note that you just need to tell Mr. Panda the minimum number of time units that are needed.

Input

The first line of the input gives the number of test cases T $(1 \leq T \leq 100)$. T test cases follow.

Each test case starts with a line consisting of four integers n $(1 \leq n \leq 5000)$, the length of Mr. Panda's thank you letter, X, Y and Z ($1 \leq X, Y, Z \leq 10^9$). X, Y and Z are the time cost of operations that can be performed by the typewriter.

Then, a line consisting of n integers S_0, S_1, ..., S_{n-1} follows, denoting the contents of Mr. Panda's thank you letter. Each integer S_i ($1 \leq S_i \leq 10^9$) represents a single character in the letter.

It is guaranteed that $n \leq 1000$ in at least 80% of test cases.

Output

For each test case, output one line containing "`Case #x: y`", where `x` is the test case number (starting from 1) and `y` is the minimum number of time units that are needed to type the thank you letter.

Mr. Panda and Typewriter(CCPC Final 2019 C)

1. f[i][j] 表示我们已经打完了前 i 个字符，buffer 里是**最后** j 个字符时的最小代价；

2. 转移
   a. f[i][j] -> f[i+1][0] 操作 1，新打一个字符；
   b. f[i][j] -> f[k][j] 把当前 buffer 里的子串粘贴到下一次出现的地方，k 是子串 (i-j+1, i) 下一次出现的位置的末尾，中间都用操作 1 填满；
   c. f[i][j] -> f[i+k][k] 如果前面存在子串 (i+1, i+k)，粘贴过来；

1. 复杂度O(n^2)；

Bookcase Solidity United(Grand Prix of Belarus K)

You work in *Bookcase Solidity United* (BSU), a company that tests furniture under various load and measures its reliability. Right now, they are testing a bookcase with n shelves, placed from top to bottom.

The bookcase will be tested by putting heavy iridium balls on some of the shelves and observing them break. We assume that all balls are the same and BSU has infinitely many of them.

The engineers measured that the i-th shelf from the top can endure strictly less than a_i balls. If there are $x \geq a_i$ balls on the shelf, it breaks and the balls fall. If there are no unbroken shelves, all the balls fall on the floor. Otherwise, $\lfloor \frac{x}{2} \rfloor$ balls fall on the nearest unbroken shelf j below, and the rest of the balls fall on the floor. (Don't worry, the floor is solid enough to hold all the balls.) If, after this operation, the number of balls on the j-th shelf is not less than a_j, then the j-th shelf breaks, the balls fall from the shelf in the same way as described above, and so on. The process terminates either when all the balls are on the floor or the next shelf is solid enough to hold the balls which fell on it.

To measure the reliability, the employees of BSU put the balls one by one on some of the shelves. The goal is to break k upmost shelves using the minimum possible number of balls. As trying various placement variants is costly, time-consuming, and produces much noise from heavy falling balls, the management of the company has ordered you to calculate the least number of balls to break k upmost shelves, for each k from 1 to n.

Input

The first line contains an integer n, the number of shelves in the bookcase ($1 \leq n \leq 70$).

The second line contains n integers a_i, where a_i is the minimal number of balls that break the i-th shelf ($1 \leq a_i \leq 150$). The shelves are numbered from top to bottom.

Output

Print n integers. The k-th integer equals the minimal number of balls needed to break k upmost shelves.

Bookcase Solidity United(Grand Prix of Belarus K)

1. dp[l][r][k] 表示我们把第 l 层到第 r 层全击碎（假设其他层都不存在），最后还留下 k 个球时的最小花费；

2. 现在有两种情况
   a. 这些球用来击碎第 r+1 层 $dp[l][r+1][\max\{k, a_{r+1}\}/2] = dp[l][r][k] + \max\{0, a_{r+1} - k\}$
   b. 这些球漏到了第 p 层（第 r+1 到第 p 层事先被其他球击碎了）

   $$dp[l][p][k+k_1] = dp[l][r][k] + dp[r+1][p][k_1]$$

3. k 最大是 max(ai)；

4. 区间 dp 之！

Infimum of Paths(CCPC Final 2019 B)

On a directed graph, we use $lex(p)$ to denote the lexical weight of a path p, where the path p can be regarded as a sequence of consecutive edges. The lexical weight is defined by the recurrence relation

$$lex([]) = 0, lex([e_1, e_2, \ldots, e_n]) = \frac{w(e_1) + lex([e_2, e_3, \ldots, e_n])}{10},$$

where $w(e_1)$ is the weight of edge e_1, which is an integer between 0 and 9, inclusive.

Given a directed graph, find the infimum of the lexical weights of all paths from node 0 to node 1. The infimum of a set of rational numbers is the greatest rational number that, if exists, is less than or equal to all elements in this set.

Input

The first line of the input gives the number of test cases, T $(1 \leq T \leq 100)$. T test cases follow.

For each case, the first line contains two integers, n $(2 \leq n \leq 2000$, $\sum n \leq 20000)$ and m ($1 \leq m \leq 4000$, $\sum m \leq 40000$), where n is the number of nodes and m is the number of edges.

Then m lines follow, each of which contains three integers u, v, w $(0 \leq u, v < n, 0 \leq w \leq 9)$, indicating an edge from u to v of weight w.

It is guaranteed that there exists at least one path from node 0 to node 1 for each test case.

Output

For each test case, output one line containing "`Case #x: y`", where `x` is the test case number (starting from 1), and `y` is the answer modulo $(10^9 + 7)$. More specifically, if the answer can be formed as an irreducible fraction $\frac{A}{B}$, then `y` will be $(A \cdot B^{-1}) \bmod (10^9 + 7)$.

Infimum of Paths(CCPC Final 2019 B)

1. 从 0 到 1 的路径有两种情况：中间有环的和中间没环的；
2. 删掉所有不能到 1 号点的点（这些点没什么用）；
3. 如果我们知道了从 0 开始走 k 步能到哪里的集合，我们很容易能得到 k+1 步能到哪里的集合；对于每一个点，我们维护一下权值最小的路是从哪里来的；
4. 一共走 n+1 步，如果中间的过程中到了 1 号点，我们就把它的值记下来；
5. 走完 n+1 步后所有中间没环的情况就考虑完啦！
6. 再考虑中间有环的情况，我们把到每个点的环的值都算出来（走完 n+1 步以后每个点一定在值最小的环中）；
7. 最后比下大小；

Bytelandia States Union(Grand Prix of Belarus H)

There are so many natural wonders in the Bytelandia States Union (BSU)! But the most mysterious wonder is, undoubtedly, the Murbeda Rectangle. Here time and space behave in a rather unusual way. Bytelandian scientists still haven't found a reason why such anomalies occur, even after many years of research. Luckily, they managed to understand how physics works at the Murbeda Rectangle.

Consider the Murbeda Rectangle as a large rectangle. The scientists have divided the rectangle into a grid of $2 \cdot 10^9 \times 2 \cdot 10^9$ small squares. Each square has coordinates (x, y), where the x-axis goes from north to south, and the y-axis goes from west to east. So, the northwestern square is at $(1, 1)$, and the southeastern square is at $(2 \cdot 10^9, 2 \cdot 10^9)$. There is a portal in the square (x_2, y_2) which is the only way to connect the Murbeda Rectangle to the outer world.

Suppose you are in the square (x, y) of the rectangle. You can move in one of four directions (north, south, east, or west), thus increasing or decreasing one of the coordinates by one. You cannot go out of the rectangle: for instance, you cannot go south from the square $(2 \cdot 10^9, 42)$ or go west from the square $(42, 1)$. If you do, you may fall into a deep canyon filled with poisonous snakes.

The most fascinating thing is the amount of time you spend on moving in some direction. If you are in the square (x, y), then:

- going south (increasing the x-coordinate by one) takes $f_s(x, y) = 2xy^2 + 2y^2 + x^2$ seconds;
- going north (decreasing the x-coordinate by one) takes $f_n(x, y) = -2xy^2 + 2y^2 + x^2$ seconds;
- going east (increasing the y-coordinate by one) takes $f_e(x, y) = 2x^2y + 2x^2 + y^2$ seconds;
- going west (decreasing the y-coordinate by one) takes $f_w(x, y) = -2x^2y + 2x^2 + y^2$ seconds.

The amount of time spent on moving between squares may even be negative! The place is *really* special.
The scientists intend to rescue n people from the Murbeda Rectangle. For a person who stands in the square (x_1, y_1), they need to determine the minimum time needed to reach the portal. One can prove that such a minimal amount of time exists, so no one can reach an infinitely small moment by walking around.

Since the place is extremely unusual, each of the n persons may require a different portal.

Input

The first line contains an integer n, the number of people to rescue ($1 \le n \le 5 \cdot 10^4$).

Each of the following n lines contains four integers x_1, y_1, x_2, y_2, denoting the location of the i-th person and the location of their rescue portal ($1 \le x_1, y_1, x_2, y_2 \le 10^9$).

Output

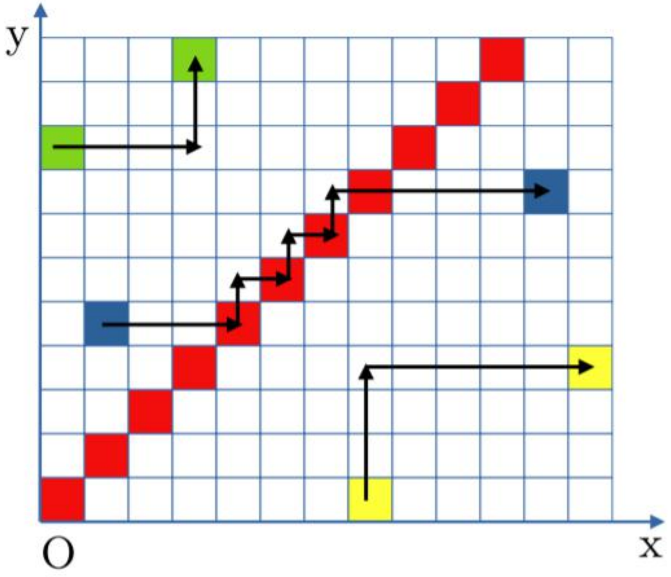
Print n lines. The i-th line should contain the minimal amount of time in seconds for the i-th person to reach their rescue portal. Since this amount can be pretty large, print it modulo $998\,244\,353$.

Bytelandia States Union(Grand Prix of Belarus H)

1. 单独走一步 (xs, ys) 到 (xt, yt) 的代价是 $x_t^2 y_t^2 - x_s^2 y_s^2 + x_s^2 + y_s^2$
2. 路径 (x1,y1)….(xk,yk) 的花费长这样 $T = x_k^2 y_k^2 - x_1^2 y_1^2 + \sum_{i=1}^{k-1} x_i^2 + y_i^2.$
3. 我们把和起点和终点有关的项 $x_k^2 y_k^2 - x_1^2 y_1^2 - x_k^2 - y_k^2$ 拿掉，相当于我们要最小化

$$\sum_{i=1}^{k} x_i^2 + y_i^2$$

4. 假设 x1<=x2 && y1<=y2 or x1>=x2 && y1>=y2，情况长这样



5. Otherwise, 先往下再往右。

Mr. Panda and Cactus(CCPC Final 2018 E)

Mr. Panda is exploring in a desert. He finds that there are many cactuses growing in the oasis in the center of the desert. The scenery inspires him to come up with a cactus problem.

As we know, a cactus is a connected undirected graph with each edge belonging to at most one simple cycle. Given a weighted graph with each connected component being a cactus with no self loops, you are requested to color each vertex with one of the given K colors. Each color is required to be used at least once.

Mr. Panda wants to minimize the sum of weight of edges which connects vertices of different color. Could you help him?

Input

The first line of the input gives the number of test cases, T $(1 \le T \le 100)$. T test cases follow.

For each test case, the first line contains three integers N, M and K $(1 \le N \le 10^5, 0 \le M \le 2 \times 10^5$, $1 \le K \le \min\{N, 1000\})$, where N is the number of vertices, M is the number of edges, and K is the number of colors. We ensure the sum of N in all cases is not greater than 5×10^5.

The following M lines describe the edges between the vertices. Each line contains 3 integers x, y, w $(1 \le x \ne y \le N$, $1 \le w \le 10^9)$, representing an edge with weight w between vertex x and vertex y.

Output

For each test case, output one line containing "`Case x: y`" first, where `x` is the test case number (starting from 1) and `y` is the minimum sum of weight of edges which connects vertices of different color.

Then output a line consists of N integers, the i^{th} integer is the color of i^{th} vertex, any valid way is acceptable. Make sure each color is used at least once.

Mr. Panda and Cactus(CCPC Final 2018 E)

1. 最后等价于分成 k 个联通块，每个联通块染一种颜色；
2. 边分为两种，一种在环上，一种不在环上（树边）；
3. 对于每条树边，删掉会加一个联通块；
4. 对于每个环，权值从小到大删边，删掉 i 条边会加 i-1 个联通块；
5. 直接dp，f[i][j] 表示前 i 个组成部分（环或者树边），有 j 个联通块的情况下的最小花费！

Query On A Tree 17(Grand Prix of Korea I)

On *Baekjoon Online Judge*, there are a series of problems related to processing queries on a tree. We KAISTians are proud to offer the seventeenth edition in this series.

You are given a tree with N vertices. Each vertex is numbered from 1 to N. The tree is rooted at vertex 1.

People can live in each vertex. Let $A[i]$ be the number of people living in vertex i. Initially, $A[i] = 0$ for all $1 \leq i \leq N$.

Write a program that processes the Q following queries:

- `1 u`: Add 1 to $A[i]$ for all vertices i in the subtree rooted at vertex u.
- `2 u v`: Add 1 to $A[i]$ for all vertices i on the unique shortest path between the two vertices u and v. Note that u and v might be equal.

After each query, print the vertex x that minimizes the quantity $\sum_{y=1}^{N} A[y] \times dist(x, y)$, where $dist(x, y)$ is the number of edges on the path between x and y. If there is more than one such vertex, print the vertex with minimum distance from the root (vertex 1). It can be proven that such vertex is unique. In other words, we should find a vertex that minimizes the total distance needed for everyone to gather.

Input

The first line contains the single integer N ($2 \leq N \leq 100\,000$).

The next $N - 1$ lines describe all edges of the tree. Each line contains two space-separated integers u, v denoting that there is an edge connecting vertices u and v. ($1 \leq u, v \leq N, u \neq v$)

The next line contains the single integer Q ($1 \leq Q \leq 100\,000$).

The next Q lines contains several integers in one of the following forms:

- `1 u` ($1 \leq u \leq N$)
- `2 u v` ($1 \leq u, v \leq N$)

Output

Print Q lines, denoting the answer after each update.

Query On A Tree 17(Grand Prix of Korea I)

1. 最后答案是带权重的树的重心（否则往重心走更优），现在我们相当于要找深度最浅的重心；

2. 对于一个点 v，假如它是答案，把它作为根，那么 root 所在的子树大小 <= 一半（根据重心的定义），也就是说剩下的大小（root 为根时子树 v 的大小） >= 一半，也就是说权值的中位数 x 一定在 v 所包含的子树里；

3. dfs序 + 轻重链剖分 + 线段树，可以找出权值的中位数 x 在哪里；

4. 最后可以在 root 到 x 的路径上二分出答案！

背包

n 个物体，第 i 个物体权值为 a[i]，要求选出一些物体，使得权值和 <= c 并且最大。

n, a[i] <= 2e4 c <= 1e9

背包

1. 贪心出一组 <= c 的解，那么正解 x 和贪心出的解 y 的 delta <= 2e4；

2. 把贪心拿的物体的权值标成负数，其他物体的权值标为正数；

3. 把正数放上面，把负数放下面，用 f[i][j][k] 表示用了上面前 i 个数，下面前 j 个数，能不能表达 k； abs(k) <= 2e4

4. j 不用记，用 g[i][k] 表示上面用了 i 个数，为了表达 k，下面最少用多少数；

5. 两种转移，第一种看 i+1 取不取，另一种看 g[i][k] 后面哪一个取（g[i][k] + 1 ~ g[i–1][k] 就可以啦）！

Beautiful Sequence Unraveling(Grand Prix of Belarus B)

You are a happy possessor of the powerful tool called *Beautiful Sequence Unraveler* (BSU). This tool works with beautiful sequences. A *beautiful sequence* is an array $a_1, a_2, \ldots, a_n$ of n integers for which the following statement holds: there are no integers i such that $1 \le i < n$ and $\max\{a_1, \ldots, a_i\} = \min\{a_{i+1}, \ldots a_n\}$.

BSU deals with beautiful sequences pretty well, but you do not know how frequently such sequences occur. So you want to calculate the number of beautiful sequences among all the arrays of length n which consist of integers between 1 and k, inclusively. Since this number may be large, you are required to calculate it modulo prime number p.

Input

The only line contains three integers n, k, p ($1 \le n \le 400$, $1 \le k \le 10^8$, $998\,244\,353 \le p \le 10^9 + 9$).

It is guaranteed that p is prime.

Output

Print the answer to the problem modulo p.

Beautiful Sequence Unraveling(Grand Prix of Belarus B)

dp[x][y] 表示长度为 x 的包含 1…y 的美丽序列的方案数（y <= x）；

美丽序列方案数 = 总方案数（y^x）- 不美丽序列方案数

令 t 为最大的 i 满足 $\max\{a_1,\ldots,a_i\} = \min\{a_{i+1},\ldots a_n\}$
令 m 为此时的 $\min\{a_{i+1},\ldots a_n\}$

此时 t 右边的元素都大于等于 t 左边的元素；

此时 t 右边必须也是个美丽序列（不然 t 不会是最大的）；

m,t 对 dp[x][y] 的贡献是 $(dp[x-t][y-m+1]-dp[x-t][y-m])\cdot(m^t-(m-1)^t).$

Beautiful Sequence Unraveling(Grand Prix of Belarus B)

继续优化

$$dp[x-t][y-m+1]\cdot m^t = m^x \cdot \left(\frac{dp[x-t][y-m+1]}{m^{x-t}}\right)$$

当 m, y 固定的时候，这个可以用前缀和 O(n) 求；

其他几块也一样；

于是 dp 数组可以 O(n^3) 整出来；

用 dp[x][y] 数组我们可以算出恰好使用 y 个数字的美丽序列的方案数；

接下来我们整个组合数就行啦！

GCD Land(CCPC Final 2018 C)

GCD land consists of N cities initially labeled from 1 to N. Two cities are connected by a highway if GCD (greatest common divisor) of their labels is greater than 1. For example, initially city 4 is connected with city 6, and there is no direct highway between city 4 and city 7.

As the president of GCD land, Mr. Panda doesn't want his nation to be divided into disconnected pieces. Mr. Ang, the wise prime minister of the nation, suggests to increase the label of each city by a non-negative magic number X, such that after reconnecting the highways according to cities' new labels, the whole nation is connected. For example, if Mr. Panda increases all cities' labels by 8, city 4 will be connected with both city 6 and city 7, because $\gcd(12, 14) > 1$ and $\gcd(12, 15) > 1$.

Can you help Mr. Panda find the magic number X when it should be lower than 10^N? If no such X exists, output -1 instead.

Input

The first line of input gives the number of test cases T ($1 \le T \le 20$). T test cases follow. Each test case contains one integer N ($1 \le N \le 10^5$), the number of cities in GCD land. For at least 75% test cases, it is guaranteed that $N \le 5 \times 10^4$.

Output

For each test case, output one line containing "`Case x: y`", where `x` is the test case number (starting from 1), `y` is the non-negative magic number if the solution exists, otherwise output -1 instead. Any valid solution will be accepted.

GCD Land(CCPC Final 2018 C)

1. 考虑 1…n/2 之间所有的素数，用它们朝中间的点 x 连边；连完以后除了最靠近 x 的两个点 x-1, x+1 都被连起来啦；

2. 我们拿 1…n/2 之间最大的两个素数 p1,p2 把最靠近 x 的两个点连进去，那么除了 x-p1, x+p1, x-p2, x+p2 都被连进去啦；

3. 我们再用大于 n/2 的最小的 4 个素数把它们连进去；