National Taiwan University

Team Ranking Contest 2014

An easy problem?	1s
Beautiful substring	5s
Christmas tree	5s
Data generator	1s
Eating cake	1s
Favorite Permuatation	5s
Game	3s
Hyper Tree	10s
Intelligent Path	3s
Just XOR	5s

September 20, 2014

(almost empty page)

An easy problem?

Description

Being a adept ACM competitor, you have solved many easy problems such as **COT**, **COT2**, **COT3** and **COT4**. You may notice that some of those problems are much easier than other problems. For example:

> How may pieces of plane can you form while slicing a plane with N cuts?

This problem may be the third problem a ACM-ICPC competitor solved (after **HelloWorld** and **A+B**).

Most of you will disdain this problem now since it's too easy. However, **how to** cut a plane into as many pieces as possible with N cuts, may be much harder. Can you solve this, too?

Input

There is an integer in the first line, indicating the number of test cases. Every test cases contain one integer N, the number of cuts.

- $1 \leq T \leq 300$
- $1 \leq N \leq 200$

Output

Output N lines for each test case. Each line contains four integers x_1, y_1, x_2, y_2 representing a cut with two points.

The N cuts you output should cut the plane into as many pieces as possible, and the absolute value of every integers should not exceed 500.

Sample Input

```
2
2
3
```

Sample Output

```
0 0 1 1
1 0 0 1
0 0 1 1
1 1 2 0
2 0 0 0
```

Beautiful substring

Description

Fcrh is an expert of pronouncing string studying in the Nation Tomato University. In his world, each string is either beautiful or ugly. Whenever he is asked to pronounce a string, he will check whether the string is beautiful and reject to do the job if it's not. What make things worse is that any substring of a ugly string is also ugly! So, sometimes it's pretty hard to ask Fcrh to pronounce strings for you.

To make things more efficient, Fcrh maintains a list of ugly strings. One can check whether a string is ugly with the list by himself, hence Fcrh won't be bother by those ugly strings.

Shik, an string lover in the ICPC club, find an interesting ICPC task: Given a string S, how many kinds of substring of S is not ugly in Fcrh's opinion according to the list?

Input

There is an integer T in the first line, indicating the number of test cases. Each test case starts with a line containing the string S. The next line contains an integer N, denoting Fcrh has added N strings into the list. Each of the next N lines contains a string F_i, the i-th string Fcrh add into the list.

- $1 \leq T \leq 10$
- $1 \leq |S| \leq 200000$
- $1 \leq \sum |F_i| \leq 1000000$
- Strings are consist of English lowercase letters.

Output

For each F_i, output an integer indicating how many kinds of substring of S is still beautiful after adding F_i into the list. To shorten the output, you don't need to output anything if the number of kinds of beautiful substring is not changed after adding F_i.

Sample Input

```
1
abcabc
4
a
abc
twt
bcab
```

Sample Output

```
14
9
5
```

Christmas tree

Description

The Christmas is coming in three months!

Being the Atnas Claus living in the south pole, you are busy preparing gifts for twins all over the world. You love twins and you have been doing this over years, but you start getting tired of preparing things like chocolates or sweets about ten years ago. So, you decide to give each pair of twin a Christmas tree as a gift this year. Furthermore, to shorten the preparation time, you decide to plant a huge tree and cut it into smaller trees after decorating it.

You have magics to keep a subtree alive even if it has only one node. However, as a gift to twin, decorations on the Christmas tree should be pairable. In another word, the number of each type of decorations on the tree should be even. Now, you have planted and decorated the tree. You wonder how many trees you can cut the huge tree into?

Input

There is an integer T in the first line of file, indicating the number of test cases. Each test case starts with an integer N, the size of the huge Christmas tree. The next line contains N integers, $D_0, D_1, \ldots, D_{N-1}$, the type of decorations you used to decorate the nodes of Christmas tree. Next $N-1$ lines contains two integer A_i, B_i each, denoting there's a edge between node A_i and node B_i.

- $1 \leq T \leq 30$
- $1 \leq N \leq 200000$
- $0 \leq A_i, B_i, D_i < N$

Output

Output an integer for each test case, representing the maximum number of pieces you can cut the tree into.

Sample Input

```
2
4
0 1 1 0
0 1
1 2
2 3
6
0 0 1 1 2 2
0 1
1 2
2 3
2 4
4 5
```

Sample Output

```
1
3
```

(almost empty page)

Data generator

Description

Fcrh is a retired competitor of ACM-ICPC. In order to contribute to his university, he decides to become the teaching assistants of the NTU ACM training course.

The students in NTU work very hard on ACM-ICPC, so most of the problems on the internet are already solved by some of the students. On one hand, it means those students can train themselves without any help. On the other hand, however, it means Fcrh has to prepare many brand new problems for those students if he wants to help them.

One day, while Fcrh is preparing a problem about Burrows-Wheeler transform, he encountered a serious problem: his test data generator won't stop! After some investigate, he find his generator hanging in this function:

```
string rand_a_legal_bwt_result(int len) {
    while (true) {
        string str;
        for (int i = 0; i < len; i++)
            str += char(rand() % 26 + 'a');
        if (is_legal_bwt_result(str))
            return str;
    }
}
```

The function `is_legal_bwt_result` is so simple that Fcrh believe it's bug-free and it runs in linear time. So, obviously, the problem is the probability of a random string to be a legal bwt result is too low.

After knowing this, Fcrh changes the way he generate data and avoid the problem. However, Fcrh is still curious about what the probability of a random string to be a legal bwt result is. Can you help him find out the answer?

In case you don't know what Burrows-Wheeler transform is, we provide pseudo code for it:

```
function bwt(string):
    rotations = all_rotations_of_string(string)
    rotations.sort_by_lexical_order()
    result = empty_string
    for all rotation in rotations:
      result.append(rotation.last_character())
    return result
```

For example: All rotations of "abcab" are ["abcab", "bcaba", "cabab", "ababc", "babca"]. After sorting them in lexical order, we'll get ["ababc", "abcab", "babca", "bcaba", "cabab"]. And, we'll get "cbaab" after collecting the last character of each string. Hence, *bwt("abcab") = "cbaab"*.

Input

There is an integer in the first line, indicating the number of test cases. Each test case contains two integers N, K the length of the random string and the size of charset.

- $1 \leq T \leq 2^{14}$
- $1 \leq N \leq 2^{30}$
- $1 \leq K \leq 2^{30}$

Output

Output the probability of a random string to be a legal bwt result. To be clear, you should output the probability that a random string S can find an string T that $bwt(T) = S$. Since the probability may be very small, output it as an floating point may suffer from rounding error. You should output the result mod 2147483647. Here we define $\frac{a}{b} = a \times inverse(b)$ so that you won't need to deal with fractions.

Sample Input

```
2
2 2
3 1
```

Sample Output

```
1610612736
1
```

Hint

The probability of the first sample is $\frac{3}{4} = 3 \times inverse(4) = 3 \times 536870912 = 1610612736$.

Eating cake

Description

Fcrh is a programmer loves eating cakes. He eats a lot of cakes every day, however, he is too busy to do exercises. To prevent Fcrh from being fatter, his girl friend, Chih-chih, set some rules to make eating cake harder.

Fcrh only eats cakes with a square shape, hence you can regard a cake as a N by N grids. Chih-chih will label each grid of the cake with a type $c_{i,j}$, and Fcrh is permitted to eat some grids of the cake if and only if he obeys these rules:

1. Fcrh should start eating the cake at $(1, 1)$ and end at (N, N)
2. Fcrh can eat a grid if and only if he just ate an adjacent grid (except $(1, 1)$)
3. He should eat exactly $2 \times N - 1$ grids
4. The i-th grid he eat should have the same type with the $(2 \times N - i)$-th grid

Now, Fcrh has evaluated the deliciousness of each grid. He wonder what's the maximum sum of deliciousness he can gain?

Input

There is an integer T in the first line, indicating the number of test case. The first line of each test case contains an integer N, the size of the cake. Each of the next N lines contains N English lowercase letters, the j-th in the i-th line represents the type of the grid (i, j). The next N lines contains N integers each, the j-th integer in the i-th line $d_{i,j}$ denotes the deliciousness of the grid (i, j).

We guarantee that there's at least one way to eat the cake.

- $1 \leq T \leq 20$
- $1 \leq N \leq 250$
- $0 \leq d_{i,j} \leq 100000$

Output

Output the maximum deliciousness Fcrh can gain of each test case.

Sample Input

```
2
3
a a a
b b b
a a a
1 2 3
2 3 4
1 2 3
3
t w t
t w t
t w t
1 2 3
1 2 3
2 4 6
```

Sample Output

```
13
15
```

Favorite Permuatation

Description

Fcrh loves permutation. He wants to find his favorite permutation in sequence $\{a_i\}$. His favorite permutation is the longest consecutive subsequence $a_l, a_{l+1}, \ldots, a_r$ in sequence $\{a_i\}$, which is also a permutation of $1, 2, \ldots, (r-l+1)$.

Input

The first line contains a integer T indicating the total number of test cases. Each test case starts with one line containing an integer n. Then one line contains n integers $a_1, a_2, \ldots, a_n$.

- $1 \leq T \leq 1000$
- $1 \leq n \leq 10^6$
- $1 \leq a_i \leq 10^9$
- There are at most 20 test cases with $n > 1000$

Output

For each test case, output the length of his favorite permutation in sequence $\{a_i\}$.

Sample Input

```
3
6
3 1 4 1 5 9
5
2 1 2 3 2
5
5 3 1 2 4
```

Sample Output

```
1
3
5
```

Game

Description

Seanwu loves playing Touhou games. There is a very hard stage in the limited edition "th5.14", which is a bullet hell game. The player controls a circular machine with radius r and try to dodge the bullets. As the stage starts, some warning strips will show on the screen. Few seconds later, boss Marisa will launch *Final Spark* on the warning strips from outside of the screen, and all things touch the *Final Spark* will be destroyed. The player has enough time to move to anywhere he want. But this stage is incredibly hard, sometimes it is even impossible to pass. Is it possible that Seanwu pass this stage?

Input

The first line contains a integer T indicating the total number of test cases. Each test case starts with one line containing four integers w, h, n, r, denoting the width and height of the screen, the number of warning strips, and the radius of player respectively. Then n lines, each line contains five integers x_1, y_1, x_2, y_2, t denoting that there is a warning strip which pass through $(x_1, y_1), (x_2, y_2)$ with thickness t.

- $1 \leq T \leq 70$
- $3 \leq w \leq 640$
- $3 \leq h \leq 480$
- $1 \leq n \leq 100$
- $1 \leq t \leq 100$
- $1 \leq r < \frac{\min(w,h)}{2}$
- $0 \leq x_1, x_2 \leq w$
- $0 \leq y_1, y_2 \leq h$
- Output would remain unchanged if the radius of player become $r + 10^{-5}$.

Output

For each test case, output the "`OAO`" if possible, "`QAQ`" otherwise.

Sample Input

```
2
100 100 1 1
50 0 50 100 50
640 480 1 1
0 0 640 480 100
```

Sample Output

```
QAQ
OAO
```

Hyper Tree

Description

Fcrh loves spanning tree and bit operations, thus he define the malagaga value of tree T as bitwise OR of weights on each edge. Now he wants to find the hyper tree of a undirected graph G. The hyper tree of a graph G is the spanning tree with minimum malagaga value.

Input

The first line contains a integer T indicating the total number of test cases. Each test case starts with one line containing two integers n, m, denoting the number of nodes and edges in the undirected graph G. Then m lines contains two integers a_i, b_i, c_i, which means there is an edge (a_i, b_i) with weight c_i in G.

- $1 \leq T \leq 1000$
- $1 \leq n \leq 100000$
- $1 \leq m \leq 200000$
- $0 \leq a_i, b_i < n$
- $0 \leq c_i \leq 10^9$
- G is connected
- There are at most 10 test cases with $n + m > 2000$

Output

For each test case, output the malagaga value of the hyper tree.

Sample Input

```
2
2 1
0 1 2
3 3
0 1 5
1 2 6
2 0 8
```

Sample Output

```
2
7
```

Intelligent Path

Description

Fcrh loves crawling(?) on grids. One day he wants to crawl from $(0, 0)$ to (n, n). As an intelligent worm, each step he will crawl from (x, y) to $(x + 1, y)$ or $(x, y + 1)$. There are some delicious food on the diagonal line between $(0, 0)$ and (n, n). If Fcrh takes a path uniform randomly, what's the expected times he meet the food line?

Input

The first line contains a integer T indicating the total number of test cases. Each test case contains one line with one integer n,

- $1 \leq T \leq 10^6$
- $1 \leq n \leq 10^6$

Output

For each test case, output the expected times he meet the food line. The error within 10^{-8} would be accepted.

Sample Input

```
2
1
2
```

Sample Output

```
2.000000000
2.666666667
```

Just XOR

Description

Fcrh loves sequence and bit operations, thus Shik give him a sequence $a_1, a_2, \ldots, a_n$ and m queries about bit operation. In each query, Shik is curious about what is $\max\{a_i \oplus x\}$ where $l \leq i \leq r$?

Input

The first line contains a integer T indicating the total number of test cases. Each test case starts with one line containing two integers n, m, Then one line with n integers $a_1, a_2, \ldots, a_n$. Then m line with 3 integers l, r, x.

- $1 \leq T \leq 500$
- $1 \leq n, m \leq 100000$
- $1 \leq l \leq r \leq n$
- $0 \leq x, a_i \leq 10^9$
- There are at most 5 test cases with $n + m > 2000$

Output

Output the answer of each query in one line.

Sample Input

```
2
3 1
5 1 4
1 2 3
5 1
5 0 2 1 6
2 4 2014
```

Sample Output

```
6
2015
```