A. Mex in the Grid

```
Input file:     standard input
Output file:    standard output
Time limit:     2 seconds
Memory limit:   256 megabytes
```

You are given n^2 cards with values from 0 to $n^2 - 1$. You are to arrange them in a n by n grid such that there is **exactly** one card in each cell.

The MEX (minimum excluded value) of a subgrid* is defined as the smallest non-negative integer that does not appear in the subgrid.

Your task is to arrange the cards such that the sum of MEX values over all $\left(\frac{n(n+1)}{2}\right)^2$ subgrids is maximized.

*A subgrid of a n by n grid is specified by four numbers l_1, r_1, l_2, r_2 satisfying $1 \le l_1 \le r_1 \le n$ and $1 \le l_2 \le r_2 \le n$. The element in the i-th row and the j-th column of the grid is part of the subgrid if and only if $l_1 \le i \le r_1$ and $l_2 \le j \le r_2$.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 100$). The description of the test cases follows.

The first line of each test case contains a single integer n ($1 \le n \le 500$) — the side length of the grid.

It is guaranteed that the sum of n over all test cases does not exceed 1000.

Output

For each test case, output n lines, each containing n integers representing the elements of the grid.

If there are multiple answers, you can output any of them.

Standard Input	Standard Output
2 2 3	0 1 2 3 8 4 5 6 0 1 7 2 3

Note

In the first test case, one valid arrangement is:

0	1
2	3

There are 9 subgrids in total, and the 4 of them with non-zero MEX are shown below:

0

values: $[0]$ — MEX: 1

0	1

values:$[0, 1]$ — MEX: 2

0
2

values:$[0, 2]$ — MEX: 1

0	1
2	3

values:$[0, 1, 2, 3]$ — MEX: 4

The sum of MEX over all subgrids would be $1 + 2 + 1 + 4 = 8$. It can be proven that no other arrangements have a larger sum of MEX values.

,

B. Quartet Swapping

```
Input file:     standard input
Output file:    standard output
Time limit:     2 seconds
Memory limit:   256 megabytes
```

You are given a permutation a of length n^*. You are allowed to do the following operation any number of times (possibly zero):

- Choose an index $1 \le i \le n-3$. Then, swap a_i with a_{i+2}, and a_{i+1} with a_{i+3} simultaneously. In other words, permutation a will be transformed from $[\ldots, a_i, a_{i+1}, a_{i+2}, a_{i+3}, \ldots]$ to $[\ldots, a_{i+2}, a_{i+3}, a_i, a_{i+1}, \ldots]$.

Determine the lexicographically smallest permutation[†] that can be obtained by applying the above operation any number of times.

---

[*] A permutation of length n is an array consisting of n distinct integers from 1 to n in arbitrary order. For example, $[2,3,1,5,4]$ is a permutation, but $[1,2,2]$ is not a permutation (2 appears twice in the array), and $[1,3,4]$ is also not a permutation ($n=3$ but there is 4 in the array).

[†] An array x is lexicographically smaller than an array y of the same size if and only if the following holds:

- in the first position where x and y differ, the array x has a smaller element than the corresponding element in y.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 1000$). The description of the test cases follows.

The first line of each test case contains a single integer n ($4 \le n \le 2 \cdot 10^5$) — the length of permutation a.

The second line contains n integers $a_1, a_2, \ldots, a_n$ $(1 \le a_i \le n)$ — the elements of permutation a.

It is guaranteed that the sum of n over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, output the lexicographically smallest permutation that can be obtained by applying the above operation any number of times.

Standard Input	Standard Output
3 4 3 4 1 2 5 5 4 3 1 2 10 10 9 8 7 6 5 4 3 2 1	1 2 3 4 2 1 3 4 5 2 1 4 3 6 5 8 7 10 9

Note

In the first test case, an operation can be done on index $i=1$, and the permutation will become $[1,2,3,4]$, which is the lexicographically smallest permutation achievable.

In the second test case, we can do the following sequence of operations:

- Do an operation on index $i = 2$. The permutation becomes $[5, 1, 2, 4, 3]$.
- Do an operation on index $i = 1$. The permutation becomes $[2, 4, 5, 1, 3]$.
- Do an operation on index $i = 2$. The permutation becomes $[2, 1, 3, 4, 5]$.

,

C. 23 Kingdom

```
Input file:     standard input
Output file:    standard output
Time limit:     4 seconds
Memory limit:   256 megabytes
```

The *distance* of a value x in an array c, denoted as $d_x(c)$, is defined as the largest gap between any two occurrences of x in c.

Formally, $d_x(c) = \max(j - i)$ over all pairs $i < j$ where $c_i = c_j = x$. If x appears only once or not at all in c, then $d_x(c) = 0$.

The *beauty* of an array is the sum of the distances of each distinct value in the array. Formally, the beauty of an array c is equal to $\sum\limits_{1 \le x \le n} d_x(c)$.

Given an array a of length n, an array b is *nice* if it also has length n and its elements satisfy $1 \le b_i \le a_i$ for all $1 \le i \le n$. Your task is to find the maximum possible beauty of any nice array.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 10^4$). The description of the test cases follows.

The first line of each test case contains a single integer n ($1 \le n \le 2 \cdot 10^5$) — the length of array a.

The second line of each test case contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le n$) — the elements of array a.

It is guaranteed that the sum of n over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, output a single integer representing the maximum possible beauty among all nice arrays.

Standard Input	Standard Output
4 4 1 2 1 2 2 2 2 10 1 2 1 5 1 2 2 1 1 2 8 1 5 2 8 4 1 4 2	4 1 16 16

Note

In the first test case, if $b = [1, 2, 1, 2]$, then $d_1(b) = 3 - 1 = 2$ and $d_2(b) = 4 - 2 = 2$, resulting in a beauty of $2 + 2 = 4$. It can be proven that there are no nice arrays with a beauty greater than 4.

In the second test case, both $b = [1, 1]$ and $b = [2, 2]$ are valid solutions with a beauty of 1.

In the third test case, if $b = [1, 2, 1, 4, 1, 2, 1, 1, 1, 2]$ with $d_1(b) = 9 - 1 = 8$, $d_2(b) = 10 - 2 = 8$, and $d_4(b) = 0$, resulting in a beauty of 16.

,

D. Mani and Segments

```
Input file:     standard input
Output file:    standard output
Time limit:     3 seconds
Memory limit:   256 megabytes
```

An array b of length $|b|$ is *cute* if the sum of the length of its Longest Increasing Subsequence (LIS) and the length of its Longest Decreasing Subsequence (LDS)* is **exactly** one more than the length of the array. More formally, the array b is cute if $\operatorname{LIS}(b) + \operatorname{LDS}(b) = |b| + 1$.

You are given a permutation a of length $n^{\dagger}$. Your task is to count the number of non-empty subarrays‡ of permutation a that are cute.

---

*A sequence x is a subsequence of a sequence y if x can be obtained from y by the deletion of several (possibly, zero or all) element from arbitrary positions.

The longest increasing (decreasing) subsequence of an array is the longest subsequence such that its elements are ordered in strictly increasing (decreasing) order.

†A permutation of length n is an array consisting of n distinct integers from 1 to n in arbitrary order. For example, $[2, 3, 1, 5, 4]$ is a permutation, but $[1, 2, 2]$ is not a permutation (2 appears twice in the array), and $[1, 3, 4]$ is also not a permutation ($n = 3$ but there is 4 in the array).

‡An array x is a subarray of an array y if x can be obtained from y by the deletion of several (possibly, zero or all) elements from the beginning and several (possibly, zero or all) elements from the end.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 10^4$). The description of the test cases follows.

The first line of each test case contains a single integer n ($1 \le n \le 2 \cdot 10^5$) — the length of permutation a.

The second line of each test case contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le n$) — the elements of permutation a.

It is guaranteed that the sum of n over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, output the number of cute non-empty subarrays of permutation a.

Standard Input	Standard Output
5 3 3 1 2 5 2 3 4 5 1 4 3 4 1 2 7 1 2 3 4 5 6 7 10 7 8 2 4 5 10 1 3 6 9	6 15 9 28 36

Note

In the first test case, all of the 6 non-empty subarrays are cute:

- $[3]$: $\mathrm{LIS}([3]) + \mathrm{LDS}([3]) = 1 + 1 = 2$.
- $[1]$: $\mathrm{LIS}([1]) + \mathrm{LDS}([1]) = 1 + 1 = 2$.
- $[2]$: $\mathrm{LIS}([2]) + \mathrm{LDS}([2]) = 1 + 1 = 2$.
- $[3, 1]$: $\mathrm{LIS}([3, 1]) + \mathrm{LDS}([3, 1]) = 1 + 2 = 3$.
- $[1, 2]$: $\mathrm{LIS}([1, 2]) + \mathrm{LDS}([1, 2]) = 2 + 1 = 3$.
- $[3, 1, 2]$: $\mathrm{LIS}([3, 1, 2]) + \mathrm{LDS}([3, 1, 2]) = 2 + 2 = 4$.

In the second test case, one of the cute subarrays is $[2, 3, 4, 5, 1]$ as $\mathrm{LIS}([2, 3, 4, 5, 1]) = 4$ and $\mathrm{LDS}([2, 3, 4, 5, 1]) = 2$, which satisfies $4 + 2 = 5 + 1$.

,

E. Kia Bakes a Cake

```
Input file:     standard input
Output file:    standard output
Time limit:     6 seconds
Memory limit:   512 megabytes
```

You are given a binary string s of length n and a tree T with n vertices. Let k be the number of 1s in s. We will construct a complete undirected weighted graph with k vertices as follows:

- For each $1 \le i \le n$ with $s_i = 1$, create a vertex labeled i.
- For any two vertices labeled u and v that are created in the above step, define the edge weight between them $w(u, v)$ as the distance* between vertex u and vertex v in the tree T.

A **simple** path† that visits vertices labeled $v_1, v_2, \ldots, v_m$ in this order is *nice* if for all $1 \le i \le m-2$, the condition $2 \cdot w(v_i, v_{i+1}) \le w(v_{i+1}, v_{i+2})$ holds. In other words, the weight of each edge in the path must be at least twice the weight of the previous edge. Note that $s_{v_i} = 1$ has to be satisfied for all $1 \le i \le m$, as otherwise, there would be no vertex with the corresponding label.

For each vertex labeled i ($1 \le i \le n$ and $s_i = 1$) in the complete undirected weighted graph, determine the maximum number of vertices in any nice simple path starting from the vertex labeled i.

---

*The distance between two vertices a and b in a tree is equal to the number of edges on the unique simple path between vertex a and vertex b.

†A path is a sequence of vertices $v_1, v_2, \ldots, v_m$ such that there is an edge between v_i and v_{i+1} for all $1 \le i \le m-1$. A simple path is a path with no repeated vertices, i.e., $v_i \ne v_j$ for all $1 \le i < j \le m$.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 10^4$). The description of the test cases follows.

The first line of each test case contains a single integer n ($1 \le n \le 7 \cdot 10^4$) — the length of the binary string s and the number of vertices in the tree T.

The second line of each test case contains a binary string with n characters $s_1 s_2 \ldots s_n$ ($s_i \in \{0, 1\}$) — the string representing the vertices to be constructed in the complete undirected weighted graph.

Each of the next $n-1$ lines contains two integers u and v ($1 \le u, v \le n$) — the endpoints of the edges of the tree T.

It is guaranteed that the given edges form a tree.

It is guaranteed that the sum of n over all test cases does not exceed $7 \cdot 10^4$.

Output

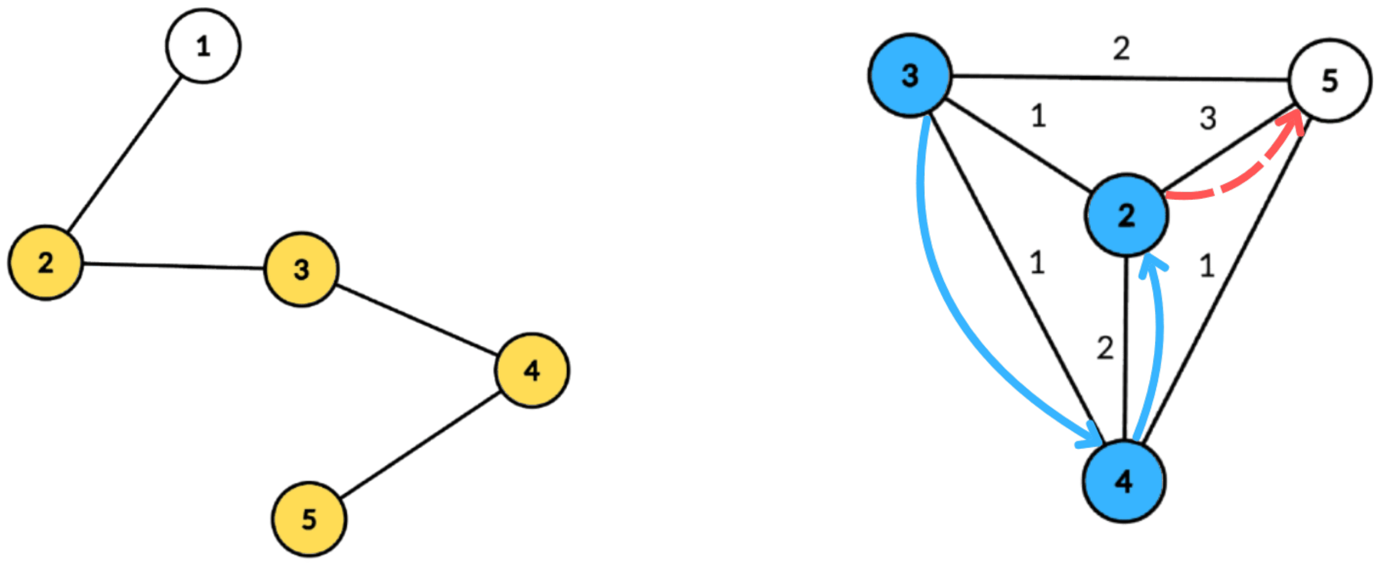
For each test case, output n integers, the i-th integer representing the maximum number of vertices in any nice simple path starting from the vertex labeled i. If there is no vertex labeled i, i.e., $s_i = 0$, output -1 instead.

Standard Input	Standard Output
3 5 01111	-1 3 3 3 3 -1 5 4 -1 4 -1 5 5 5 5 -1 4 -1 5 5 -1 3 -1 1

```
1 2
2 3
3 4
4 5
17
01101011110101101
1 2
2 3
3 4
4 5
5 6
6 7
7 8
8 9
9 10
10 11
11 12
12 13
13 14
14 15
15 16
16 17
2
01
1 2
```

Note

In the first test case, the tree T and the constructed graph are as follows:



Left side is the tree T with selected nodes colored yellow. The right side is the constructed complete graph.

The nice path shown in the diagram is $3 \to 4 \to 2$. The path is nice as $w(4, 2) = 2$ is at least twice of $w(3, 4) = 1$. Extending the path using $2 \to 5$ is not possible as $w(2, 5) = 3$ is less than twice of $w(4, 2) = 2$.

In the second test case, the tree T is a simple path of length 17. An example of a nice path starting from the vertex labeled 2 is $2 \to 3 \to 5 \to 9 \to 17$, which has edge weights of $1, 2, 4, 8$ doubling each time.

,

F. Shoo Shatters the Sunshine

Input file:	standard input
Output file:	standard output
Time limit:	7 seconds
Memory limit:	256 megabytes

You are given a tree with n vertices, where each vertex can be colored red, blue, or white. The *coolness* of a coloring is defined as the maximum distance* between a red and a blue vertex.

Formally, if we denote the color of the i-th vertex as c_i, the coolness of a coloring is $\max d(u, v)$ over all pairs of vertices $1 \le u, v \le n$ where c_u is `red` and c_v is `blue`. If there are no red or no blue vertices, the coolness is zero.

Your task is to calculate the sum of coolness over all 3^n possible colorings of the tree, modulo $998\,244\,353$.

---

* The distance between two vertices a and b in a tree is equal to the number of edges on the unique simple path between vertex a and vertex b.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 50$). The description of the test cases follows.

The first line of each test case contains a single integer n ($2 \le n \le 3000$) — the number of vertices in the tree.

Each of the next $n - 1$ lines contains two integers u and v ($1 \le u, v \le n$) — the endpoints of the edges of the tree.

It is guaranteed that the given edges form a tree.

It is guaranteed that the sum of n over all test cases does not exceed 3000.

Output

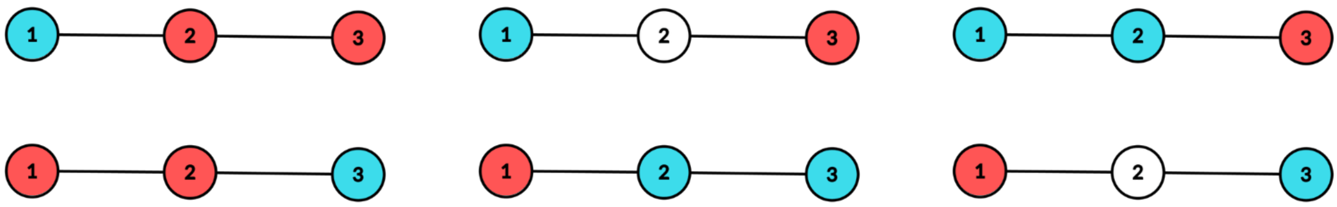
For each test case, output the sum of coolness over all 3^n possible colorings of the tree, modulo $998\,244\,353$.

Standard Input	Standard Output
3 3 1 2 2 3 6 1 2 1 3 1 4 3 5 5 6 17 1 2 1 3 1 4 1 5	18 1920 78555509

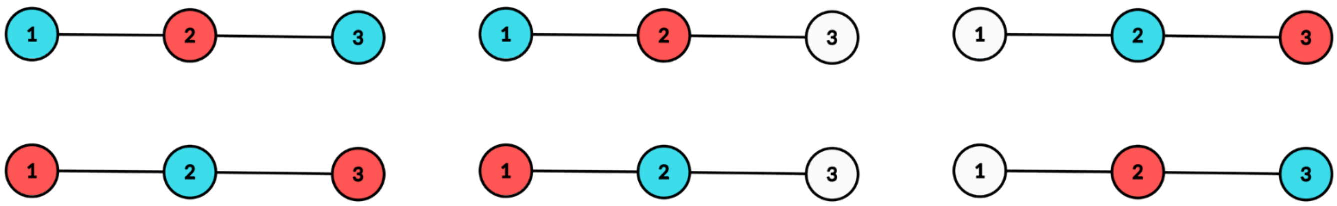
```
2 6
2 7
2 8
3 9
3 10
7 11
7 12
11 13
13 14
14 15
10 16
16 17
```

Note

In the first test case, there are 12 colorings that have at least one blue and one red node. The following pictures show their coloring and their coolness:



All these colorings have coolness 2



All these colorings have coolness 1

Therefore, the sum of coolness over all possible colorings is $6 \cdot 2 + 6 \cdot 1 = 18$.

In the second test case, the following are some examples of colorings with a coolness of 3: