C. Equal Values

time limit per test: 2 seconds
memory limit per test: 512 megabytes

You are given an array $a_1, a_2, \ldots, a_n$, consisting of n integers.

In one operation, you are allowed to perform one of the following actions:

- Choose a position i ($1 < i \leq n$) and make all elements to the left of i equal to a_i. That is, assign the value a_i to all a_j ($1 \leq j < i$). The cost of such an operation is $(i - 1) \cdot a_i$.
- Choose a position i ($1 \leq i < n$) and make all elements to the right of i equal to a_i. That is, assign the value a_i to all a_j ($i < j \leq n$). The cost of such an operation is $(n - i) \cdot a_i$.

Note that the elements affected by an operation may already be equal to a_i, but that doesn't change the cost.

You are allowed to perform any number of operations (including zero). What is the minimum total cost to make all elements of the array equal?

Input

The first line contains one integer t ($1 \leq t \leq 10^4$) — the number of test cases.

The first line of each test case contains a single integer n ($2 \leq n \leq 5 \cdot 10^5$).

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \leq a_i \leq n$).

The sum of n over all test cases does not exceed $5 \cdot 10^5$.

Output

For each test case, print a single integer — the minimum total cost of operations to make all elements of the array equal.

Example

input Copy

```
3
4
2 4 1 3
3
1 1 1
10
7 5 5 5 10 9 9 4 6 10
```

output Copy

```
3
0
35
```

Note

In the first test case, you can perform the operation twice:

- choose $i = 3$ and make all elements to the left of it equal to it, the cost will be $2 \cdot 1 = 2$;
- choose $i = 3$ and make all elements to the right of it equal to it, the cost will be $1 \cdot 1 = 1$.

The total cost is $2 + 1 = 3$.

In the second test case, all elements are already equal, so no operations need to be performed.

Supported by