Note: Some problems have been weakened.

A

There are at most 3 contest. We consider the hardest case that $k = 3$. For each account, we compute the set of contests it participated in.

Let cnt_S $(S \subseteq \{1, 2, 3\})$ be the number of accounts whose participated contest is set S.

It is easy to see for an account that participated in the contests $\{1, 2, 3\}$, it can't participate other contests.

For an account that participated in the contest $\{1, 2\}$, it can match with an account that only participated in the contest $\{3\}$. It's the same for $\{1, 3\}$ and $\{2, 3\}$.

For the remaining account, we need $\max(cnt_{\{1\}}, cnt_{\{2\}}, cnt_{\{3\}})$ peoples.

B

First, let's see how to compute the size of S_u. When two sets S_u and S_v merge, the size becomes $|S_u| + |S_v| - |S_u \cap S_v|$. We can see $S_u \cap S_v$ is exactly the set when these two sets merged last time. So we can compute it in $O(n + m)$.

For the original problem, we can solve the problem backward. The meaning of the set S_u becomes which vertices that u can reach.

C

For a query $[l, r]$ and a vertex $[L, R]$ on the segment tree.

- If $[L, R]$ intersects with $[l, r]$ and $[L, R]$ is not included in $[l, r]$, it will contribute to the number of visited vertices by 1.
- If $[L, R]$ is included in $[l, r]$ and $L \neq R$, it will contribute to the number of visited vertices by -1.
- If $[L, R]$ is included in $[l, r]$ and $L = R$, it will contribute to the number of visited vertices by 1.

Thus the contribution of a vertex only depends on the vertex ifself now, instead of the structure of the tree (whether its father is included in $[l, r]$).

We can use the partial sum technique to compute the contribute of each pair $[L, R]$, denoted as $w_{L,R}$.

Let $dp_{L,R}$ be the minimum total contributition of a subtree with the root $[L, R]$, $dp_{L,R} = \min(dp_{L,k} + dpk + 1, R + w_{L,R})$.

D

Recall the circle union algorithm, we need to find the border of the circle and find the answer.

In this problem, we can easily solve it in a similar way. For each piece of arcs on the circle, we just simply compute the probability that it's on the border and contribute it to the answer.

E

We can enumerate first 3 elements in A and calculate the fourth element in $O(1)$.

F

Let $dp_{i,j}$ be the number of permutations with i elements and j inverse pairs, $dp_{i,j} = \sum_{k=0}^{i-1} dp_{i-1,j-k}$.

The answer $f_i = \sum_{j=0}^{i(i-1)/2} dp_{i,j} j^k$. We can compute first $O(k)$ terms in $O(k^3)$.

We can prove sequence $\{f_n/(n-2k)!\}$ is a polynomial, thus we can compute first terms and find n-th term by interpolation.

However, CaiDui said $\{f_n/n!\}$ is a linear recurrence sequence, we can solve it by Berlekamp-Massey algorithm.

G

We can regard the modulo inverse as a random sequence. If we compute the first $\Theta(\sqrt{p})$ term and the minimum value can become $O(\sqrt{p})$. Thus for the remaining term, we can enumerate the value and compute the position of the value to see if it is the minimum. The expected time complexity is $O(\sqrt{p})$.

H

First, let's see how to compute the distance d from O to plane ABC. We can compute the volume V of $OABC$ by the determinant. Then we have $\frac{Sd}{3} = V$, where S is the area of triangle ABC, which can be computed stably.

Then we can try some constructions and test locally. One intuitive way is that find an equilateral triangle and make some numerical perturbations.

```
cout << "999999 1000000 0" << endl;
cout << "-999997 0 999999" << endl;
cout << "0 -999996 -999997" << endl;
```

I

n is small, we can compute the entire matrix in $O(n^2)$. Then find the maximum by 2d segment tree.

J

We need to find the rank of binary vectors. It can be solved in $O(2^{6k}/w)$ by Gaussian elimination and bitset.