

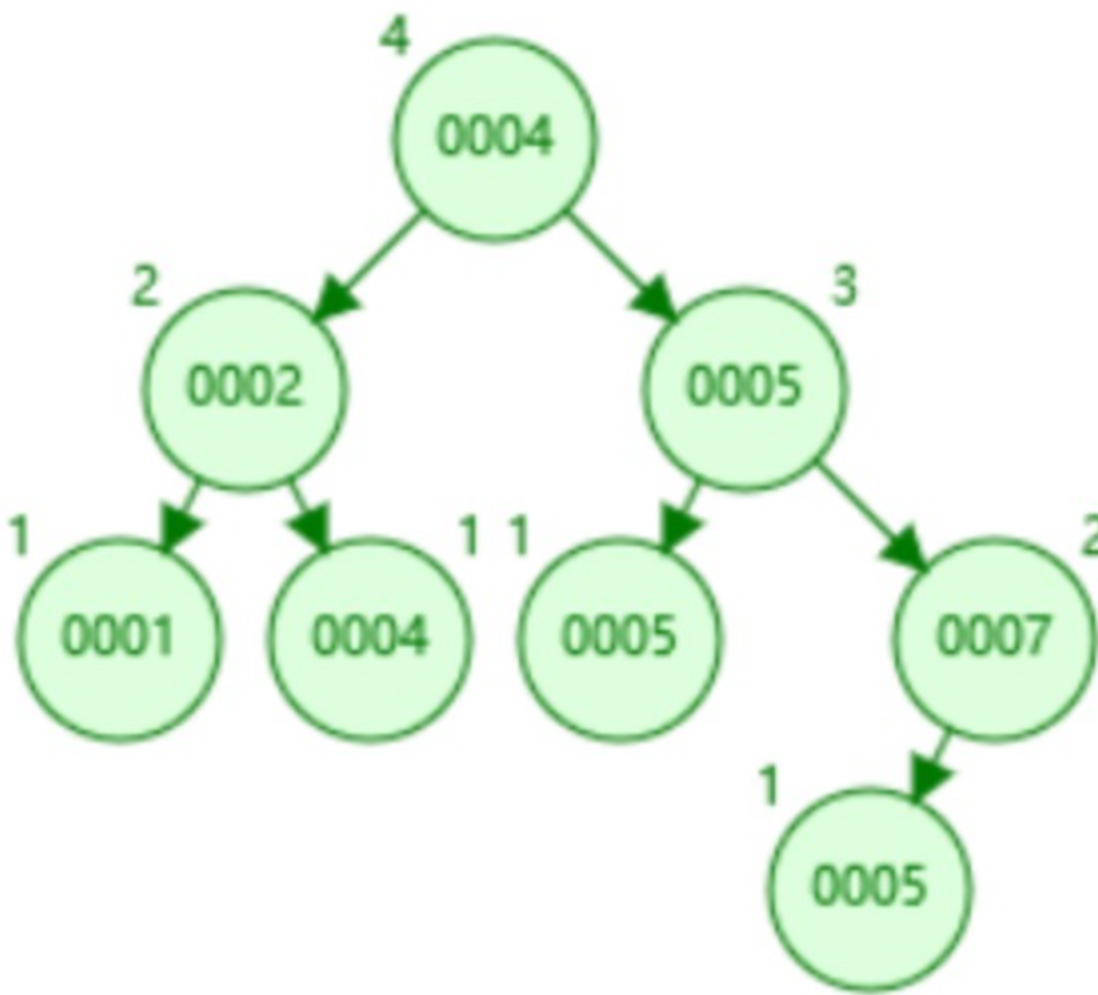
Dynamic tree template

Description

In a dynamic tree, four operations is expected.

- insert a element
- delete a element
- query the rank of a given element .
- query the element by a given rank. (When is does not exist, decrease the rank until it exists.)
- query the size

The 'rank' of an element is defined as the number of elements which are smaller than it. For example, in a tree below



rank(0001)=0, rank(0002)=1, rank(0004)=2, rank(0005)=4, rank(0007)=7

kth(0)=0001, kth(1)=0002, kth(2)=0004, kth(3)=0004, kth(4)=0005, kth(5)=0005, kth(6)=0005,kth(7)=0007

Switch the language below to `c++` to see the template.

Input

The pseudo random generator is used for input instead of stdin in order to reduce input time.

First line five integers $m, lfsr, A, B, C$.

Next 4 lines give a 4x4 matrix, representing the transition probability of a markov chain for generating input.

There are m operations. In each operation, op and imm are given.

if (op == 0) tree.insert(imm);

if (op == 1) tree.remove(tree.kth(imm % tree.size()));

if (op == 2) tans ^= tree.rank(imm);

if (op == 3) tans ^= tree.kth(imm % tree.size());

For 40% cases, $m \leq 10$

For 60% cases, $m \leq 1e5$

For 100% cases, $m \leq 1e6$

It is guarenteed the input is valid. All remove action will not happen on empty trees.

Output

Output $tans$, the xor sum of each operation rank and kth.

Sample Input 1

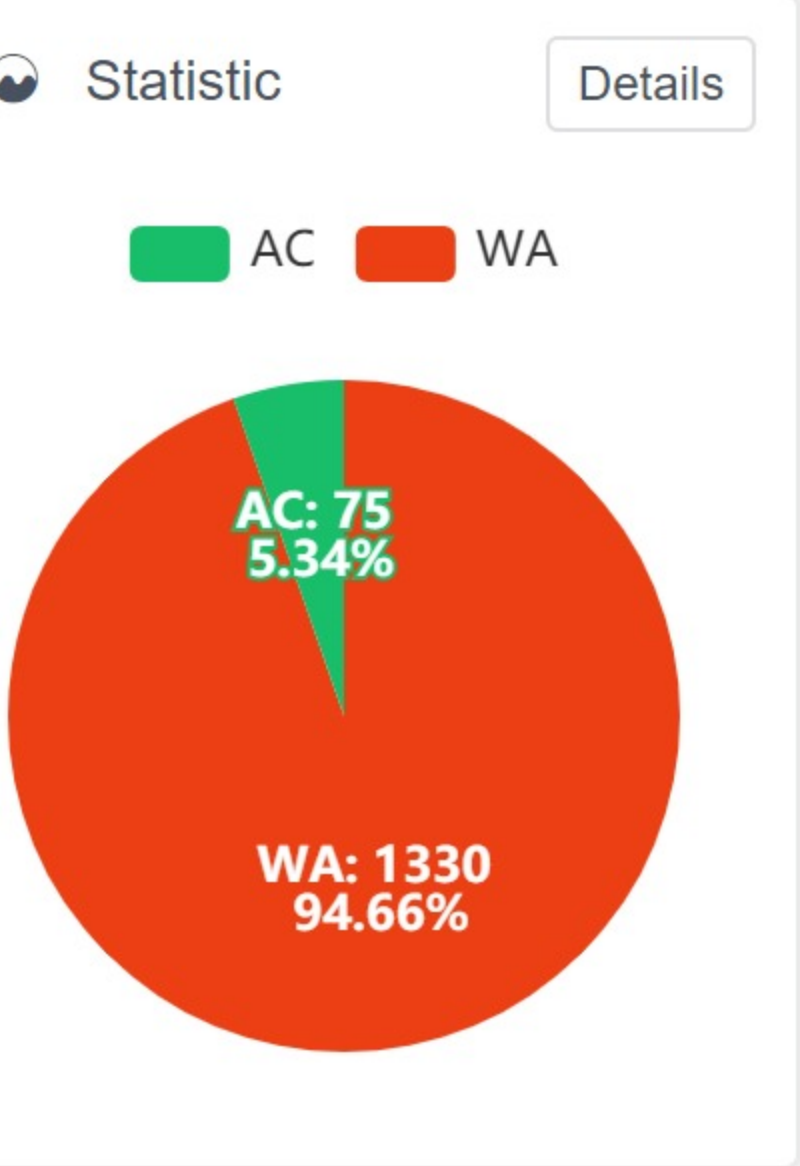
```
10 123 1 2 3
0.25 0.25 0.25 0.25
1 0 0 0
1 0 0 0
1 0 0 0
```

Sample Output 1

```
133
```

Problems
Announcements
Submissions
Rankings
View Contest

Information	
ID	3001
Time Limit	1500MS
Memory Limit	256MB
IO Mode	Standard IO
Created By	root
Level	Low
Score	1000
Tags	Show



Language:

c

Theme:

Solarized Light

1

Submit