



Задача В'язниця

 2 sec.  1024 MB

Алісу та Боба несправедливо засудили до в'язниці суворого режиму. Тепер вони повинні спланувати свою втечу. Для цього їм потрібно мати можливість спілкуватися якомога ефективніше (зокрема, Алісі потрібно щодня надсилати Бобу інформацію). Однак вони не можуть зустрічатися і можуть обмінюватися інформацією лише через записки, написані на серветках. Щодня Аліса хоче надсилати Бобу новий фрагмент інформації – число від 0 до $N - 1$. На кожному обіді Аліса отримує три серветки та пише на кожній серветці число від 0 до $M - 1$ (можливі повторення) та залишає їх на своєму місці. Потім їхній ворог, Чарлі, знищує одну із серветок і змішує дві інші. Зрештою, Боб знаходить дві серветки, що залишилися, і зчитує числа на них. Він повинен точно розшифрувати початкове число, яке Аліса хотіла йому надіслати. На серветках обмежений простір, тому M фіксоване. Однак мета Аліси та Боба – максимізувати пропускну здатність інформації, тому вони можуть вільно вибирати N якомога більшим. Допоможіть Алісі та Бобу, застосувавши для кожного з них стратегію, намагаючись максимізувати значення N .



Деталі реалізації

Оскільки це проблема зв'язку, ваша програма буде виконуватися у двох окремих виконаннях (одне для Аліси та одне для Боба), які не можуть обмінюватися даними або спілкуватися будь-яким способом, окрім описаного тут. Вам потрібно реалізувати три функції:

```
int setup(int M);
```

Вона буде викликано один раз на початку виконання вашої програми Алісою та один раз на початку виконання Бобом. Їй надається M , і вона повинна повернути бажане N . Обидва виклики `setup` повинні повернути одне і те саме N .

```
std::vector<int> encode(int A);
```

Вона реалізує стратегію Аліси. Вона буде викликана з числом для кодування A ($0 \leq A < N$) і повинна повернути три числа W_1, W_2, W_3 ($0 \leq W_i < M$), які кодують A . Ця функція буде викликана загалом T разів – один раз на день (значення A можуть повторюватися між днями).

```
int decode(int X, int Y);
```

Вона реалізує стратегію Боба. Вона буде викликана з двома з трьох чисел, повернутих `encode`, у певному порядку. Вона повинна повернути те саме значення A , яке отримала `encode`. Ця функція також буде викликана T разів – відповідно до T ви-



кликів encode; вони будуть у тому ж порядку. Усі виклики encode відбуватимуться перед усіма викликами decode.



Обмеження

- $M \leq 4300$
- $T = 5000$



Підрахунок балів

Для конкретного підзавдання частка S отриманих вами балів залежить від найменшого значення N , яке повертає `setup` у будь-якому тесті в цьому підзавданні. Вона також залежить від N^* , яке є цільовим значенням N , необхідним для отримання повної кількості балів за підзадачу:

- Якщо ваше рішення не пройде хоча б один тест, тоді $S = 0$.
- Якщо $N \geq N^*$, тоді $S = 1.0$.
- Якщо $N < N^*$, тоді $S = \max \left(0.35 \max \left(\frac{\log(N) - 0.985 \log(M)}{\log(N^*) - 0.985 \log(M)}, 0.0 \right)^{0.3} + 0.65 \left(\frac{N}{N^*} \right)^{2.4}, 0.01 \right)$.



Підзадачі

Підзадача	Бали	M	N^*
1	10	700	82017
2	10	1100	202217
3	10	1500	375751
4	10	1900	602617
5	10	2300	882817
6	10	2700	1216351
7	10	3100	1603217
8	10	3500	2043417
9	10	3900	2536951
10	10	4300	3083817



Приклад

Розглянемо наступний приклад з $T = 5$. Тут ми маємо схему кодування, де Аліса надсилає три однакові числа для кодування 0 або три різні числа для кодування



1. Зверніть увагу, що Боб може декодувати вихідне число з будь-яких двох із трьох чисел, надісланих Алісою.

Виконання	Виклик функції	Повернене значення
Аліса	<code>setup(10)</code>	2
Боб	<code>setup(10)</code>	2
Аліса	<code>encode(0)</code>	{5, 5, 5}
Аліса	<code>encode(1)</code>	{8, 3, 7}
Аліса	<code>encode(1)</code>	{0, 3, 1}
Аліса	<code>encode(0)</code>	{7, 7, 7}
Аліса	<code>encode(1)</code>	{6, 2, 0}
Боб	<code>decode(5, 5)</code>	0
Боб	<code>decode(8, 7)</code>	1
Боб	<code>decode(3, 0)</code>	1
Боб	<code>decode(7, 7)</code>	0
Боб	<code>decode(2, 0)</code>	1



Приклад градера

Для зразка оцінювання всі виклики `encode` та `decode` будуть виконуватися в одному і тому ж виконанні вашої програми. Крім того, `setup` буде викликатися тільки один раз (на відміну від двох разів, по одному разу на кожне виконання, як в системі оцінювання).

Вхідні дані складаються з одного цілого числа M . Потім він виведе число N , яке повернув ваш `setup`. Потім він викличе `encode` та `decode` T з випадково згенерованими числами від 0 до $N - 1$. Серед трьох чисел, які поверне `encode` він вибере 2 випадковим чином, в випадковому порядку і передасть їх в `decode`. Якщо ваше рішення спрацювало некоректно то він виведе повідомлення про помилку.