Псевдослучайность и хеширование

Никита Гаевой (102, 106)
Иван Казменко (101, 103)
Лиана Хазалия (104, 105)

Санкт-Петербургский Государственный Университет

Понедельник, 14 февраля 2022 года

Содержание

Содержание

Мотивирующая задача

Задача:

- У Алисы есть секретная строка из $2n$ букв.
- Известно, что в ней n букв «a» и n букв «b».
- Боб может спросить: «какая буква на позиции i?» и получить ответ.
- Цель Боба — получить ответ «a», задав $\leqslant 100$ вопросов.

Мотивирующая задача

Задача:

- У Алисы есть секретная строка из $2n$ букв.
- Известно, что в ней n букв «a» и n букв «b».
- Боб может спросить: «какая буква на позиции i?» и получить ответ.
- Цель Боба — получить ответ «a», задав $\leqslant 100$ вопросов.

Вариации:

- Что, если Боб обманывает?

Мотивирующая задача

Задача:

- У Алисы есть секретная строка из $2n$ букв.
- Известно, что в ней n букв «a» и n букв «b».
- Боб может спросить: «какая буква на позиции i?» и получить ответ.
- Цель Боба — получить ответ «a», задав $\leqslant 100$ вопросов.

Вариации:

- Что, если Боб обманывает?
- Например, если он как-то узнал всю секретную строку.

Мотивирующая задача

Задача:

- У Алисы есть секретная строка из $2n$ букв.
- Известно, что в ней n букв «a» и n букв «b».
- Боб может спросить: «какая буква на позиции i?» и получить ответ.
- Цель Боба — получить ответ «a», задав $\leqslant 100$ вопросов.

Вариации:

- Что, если Боб обманывает?
- Например, если он как-то узнал всю секретную строку.
- Тогда он может победить за 1 вопрос.

Мотивирующая задача

Задача:

- У Алисы есть секретная строка из $2n$ букв.
- Известно, что в ней n букв «a» и n букв «b».
- Боб может спросить: «какая буква на позиции i?» и получить ответ.
- Цель Боба — получить ответ «a», задав $\leqslant 100$ вопросов.

Вариации:

- Что, если Алиса обманывает?

Мотивирующая задача

Задача:

- У Алисы есть секретная строка из $2n$ букв.
- Известно, что в ней n букв «a» и n букв «b».
- Боб может спросить: «какая буква на позиции i?» и получить ответ.
- Цель Боба — получить ответ «a», задав $\leqslant 100$ вопросов.

Вариации:

- Что, если Алиса обманывает?
- Например, если она подменяет секретную строку после вопроса.
- Но так, чтобы предыдущие ответы оставались верными.

Мотивирующая задача

Задача:

- У Алисы есть секретная строка из $2n$ букв.
- Известно, что в ней n букв «a» и n букв «b».
- Боб может спросить: «какая буква на позиции i?» и получить ответ.
- Цель Боба — получить ответ «a», задав $\leqslant 100$ вопросов.

Вариации:

- Что, если Алиса обманывает?
- Например, если она подменяет секретную строку после вопроса.
- Но так, чтобы предыдущие ответы оставались верными.
- Тогда она может заставить Боба задать $n + 1$ вопрос.

Мотивирующая задача

Задача:

- У Алисы есть секретная строка из $2n$ букв.
- Известно, что в ней n букв «a» и n букв «b».
- Боб может спросить: «какая буква на позиции i?» и получить ответ.
- Цель Боба — получить ответ «a», задав $\leqslant 100$ вопросов.

Вариации:

- Как Алисе и Бобу играть честно?

Мотивирующая задача

Задача:

- У Алисы есть секретная строка из $2n$ букв.
- Известно, что в ней n букв «a» и n букв «b».
- Боб может спросить: «какая буква на позиции i?» и получить ответ.
- Цель Боба — получить ответ «a», задав $\leqslant 100$ вопросов.

Вариации:

- Как Алисе и Бобу играть честно?
- Алиса загадывает строку, записывает её и отдаёт запись Карлу.
- После игры Алиса предъявляет строку.
- Боб убеждается, что все ответы верные.
- Карл показывает запись, и Боб убеждается, что строка та же.

Мотивирующая задача

Задача:

- У Алисы есть секретная строка из $2n$ букв.
- Известно, что в ней n букв «a» и n букв «b».
- Боб может спросить: «какая буква на позиции i?» и получить ответ.
- Цель Боба — получить ответ «a», задав $\leqslant 100$ вопросов.

Вариации:

- Как Алисе и Бобу играть честно?
- Алиса загадывает строку, вычисляет *трудно обратимую функцию* от неё и показывает ответ Бобу.
- После игры Алиса предъявляет строку.
- Боб убеждается, что все ответы верные.
- Боб вычисляет функцию от строки Алисы и убеждается, что строка та же.

Мотивирующая задача

Задача:

- У Алисы есть секретная строка из $2n$ букв.
- Известно, что в ней n букв «a» и n букв «b».
- Боб может спросить: «какая буква на позиции i?» и получить ответ.
- Цель Боба — получить ответ «a», задав $\leqslant 100$ вопросов.

Как решать?

- При любом решении при $n > 100$ может оказаться, что 100 вопросов недостаточно.

Мотивирующая задача

Задача:

- У Алисы есть секретная строка из $2n$ букв.
- Известно, что в ней n букв «a» и n букв «b».
- Боб может спросить: «какая буква на позиции i?» и получить ответ.
- Цель Боба — получить ответ «a», задав $\leqslant 100$ вопросов.

Как решать?

- При любом решении при $n > 100$ может оказаться, что 100 вопросов недостаточно.
- Нас устроит *вероятностное* решение задачи.

Мотивирующая задача

Задача:

- У Алисы есть секретная строка из $2n$ букв.
- Известно, что в ней n букв «a» и n букв «b».
- Боб может спросить: «какая буква на позиции i?» и получить ответ.
- Цель Боба — получить ответ «a», задав $\leqslant 100$ вопросов.

Как решать?

- Выберем очередную позицию *случайно*: пусть это равномерно распределённое целое число от 1 до $2n$.

Мотивирующая задача

Задача:

- У Алисы есть секретная строка из $2n$ букв.
- Известно, что в ней n букв «a» и n букв «b».
- Боб может спросить: «какая буква на позиции i?» и получить ответ.
- Цель Боба — получить ответ «a», задав $\leqslant 100$ вопросов.

Как решать?

- Выберем очередную позицию *случайно*: пусть это равномерно распределённое целое число от 1 до $2n$.
- Вероятность неудачи равна $1/2$.

Мотивирующая задача

Задача:

- У Алисы есть секретная строка из $2n$ букв.
- Известно, что в ней n букв «a» и n букв «b».
- Боб может спросить: «какая буква на позиции i?» и получить ответ.
- Цель Боба — получить ответ «a», задав $\leqslant 100$ вопросов.

Как решать?

- Выберем очередную позицию *случайно*: пусть это равномерно распределённое целое число от 1 до $2n$.
- Вероятность неудачи равна $1/2$.
- Вероятность неудачи 100 раз подряд равна $1/2^{100}$.

Нужные свойства

Что нам может быть нужно от *случайных* чисел?

- Непредсказуемость: по предыдущим не узнать следующее.
- Достоверность: математическое ожидание, дисперсия...
- Повторяемость: тот же результат при повторе.
- Скорость: сотни миллионов в секунду?

Нужные свойства

Что нам может быть нужно от *случайных* чисел?

- Непредсказуемость: да.
- Достоверность: всё хорошо.
- Повторяемость: нет.
- Скорость: мала.

Как получать *случайные* числа:

- «Настоящие» случайные числа: аппаратные датчики.
- Например, младшие цифры датчика температуры на процессоре.

Нужные свойства

Что нам может быть нужно от *случайных* чисел?

- Непредсказуемость: нет.
- Достоверность: только самые базовые свойства.
- Повторяемость: да.
- Скорость: да.

Как получать *случайные* числа:

- Псевдослучайные числа: сгенерированные алгоритмом.
- Например, линейный конгруэнтный генератор:
- $s \leftarrow (s \cdot a + c) \bmod m$

Нужные свойства

Что нам может быть нужно от *случайных* чисел?

- Непредсказуемость: да.
- Достоверность: скорее да.
- Повторяемость: да.
- Скорость: меньше.

Как получать *случайные* числа:

- Псевдослучайные числа: сгенерированные алгоритмом.
- Например, какой-нибудь криптографически стойкий генератор.

Псевдослучайные числа

Как устроен генератор псевдослучайных чисел:

- Есть состояние s.
- Есть команда «инициализировать состояние» с параметром *seed*.
- Есть команда «выдать следующее случайное число»:
  - состояние меняется известной функцией, $s \leftarrow f(s)$;
  - выдаётся значение другой известной функции от нового состояния, $answer \leftarrow g(s)$.

Содержание

Устройство

Линейный конгруэнтный генератор:

- $s \leftarrow (s \cdot a + c) \bmod m$
- s — состояние
- a, c, m — константы

Устройство

Линейный конгруэнтный генератор:

- $s \leftarrow (s \cdot a + c) \bmod m$
- s — состояние
- a, c, m — константы
- `init (`$seed$`)`: $s \leftarrow seed$
- `next ()`: $s \leftarrow (s \cdot a + c) \bmod m$
- `random (`k`)`:

```
next ();
return s mod k;
```

Устройство

Линейный конгруэнтный генератор:

- $s \leftarrow (s \cdot a + c) \bmod m$
- s — состояние
- a, c, m — константы
- `init (`$seed$`)`: $s \leftarrow seed$
- `next ()`: $s \leftarrow (s \cdot a + c) \bmod m$
- `random (`k`)`:

```
    next ();
    return (s · k) div m;
```

Устройство

Линейный конгруэнтный генератор:

- $s \leftarrow (s \cdot a + c) \bmod m$
- s — состояние
- a, c, m — константы
- `init (`$seed$`)`: $s \leftarrow seed$
- `next ()`: $s \leftarrow (s \cdot a + c) \bmod m$
- `random (`k`)`:

  $m' = m - m \bmod k$`;`
  `do next (); while (`$s \geqslant m'$`);`
  `return` $(s \cdot k) \operatorname{div} m'$`;`
- Выборка с отклонением медленнее, но обеспечивает равномерность распределения.

Период

Какой период у генератора?

- $s \leftarrow (s \cdot a + c) \bmod m$
- Понятно, что состояний не больше m.
- И, если мы попали в одно и то же состояние, — дальше будет одна и та же последовательность.
- Когда состояний в периоде ровно m?

Период

Какой период у генератора?

- $s \leftarrow (s \cdot a + c) \bmod m$
- Понятно, что состояний не больше m.
- И, если мы попали в одно и то же состояние, — дальше будет одна и та же последовательность.
- Когда состояний в периоде ровно m?
  - m и c взаимно просты,
  - $a - 1$ делится на все простые делители m,
  - если m делится на 4, то и $a - 1$ делится на 4.

Код

```cpp
#include <iostream>

using namespace std;

const unsigned a = 1664525, c = 1013904223;
unsigned s;
void init (int seed) {s = seed;}
void next () {s = s * a + c;}
int random (int k) {

    next ();
    return s % k;
}

int main () {
    unsigned seed;
    int k;
    cin >> seed >> k;
    init (seed);
    for (int step = 0; step < 10; step++)
        cout << random (k) << endl;
    return 0;
}
```

Ввод:

```
123456
1000000000
```

Вывод:

```
351072415
870155634
704390697
406627700
759071299
62768838
939533677
945261032
903911975
50228058
```

Код

```cpp
#include <iostream>

using namespace std;

const unsigned a = 1664525, c = 1013904223;
unsigned s;
void init (int seed) {s = seed;}
void next () {s = s * a + c;}
int random (int k) {

    next ();
    return (s * 1LL * k) >> 32;
}

int main () {
    unsigned seed;
    int k;
    cin >> seed >> k;
    init (seed);
    for (int step = 0; step < 10; step++)
        cout << random (k) << endl;
    return 0;
}
```

Ввод:

```
123456
1000000000
```

Вывод:

```
81740416
202598896
164003739
560336676
642396346
14614508
451582874
220085734
443289050
943017205
```

Код

```cpp
#include <iostream>

using namespace std;

const unsigned a = 1664525, c = 1013904223;
unsigned s;
void init (int seed) {s = seed;}
void next () {s = s * a + c;}
int random (int k) {
    long long z = (1LL << 32) / k * k;
    do next (); while (s >= z);
    return (s * 1LL * k) / z;
}

int main () {
    unsigned seed;
    int k;
    cin >> seed >> k;
    init (seed);
    for (int step = 0; step < 10; step++)
        cout << random (k) << endl;
    return 0;
}
```

Ввод:

```
123456
1000000000
```

Вывод:

```
87768103
217538908
176097674
601656925
689767824
15692209
484883419
236315258
475977993
975811047
```

Код

```
 1  #include <iostream>
 2  #include <cstdlib>
 3  using namespace std;
 4
 5
 6
 7
 8
 9  int random (int k) {
10
11
12      return rand () % k;
13  }
14
15  int main () {
16      unsigned seed;
17      int k;
18      cin >> seed >> k;
19      srand (seed);
20      for (int step = 0; step < 10; step++)
21          cout << random (k) << endl;
22      return 0;
23  }
```

Ввод:

```
123456
1000000000
```

Вывод:

```
9977
22818
10150
16017
7706
20368
21548
8141
828
19946
```

Код

```cpp
#include <iostream>
#include <random>
using namespace std;









int main () {
    unsigned seed;
    int k;
    cin >> seed >> k;
    mt19937 rng (seed); // MT19937 is not an LCG!
    uniform_int_distribution <int> random (0, k - 1);
    for (int step = 0; step < 10; step++)
        cout << random (rng) << endl;
    return 0;
}
```

Ввод:

```
123456
1000000000
```

Вывод:

```
136332816
552883898
964354828
279683982
757868586
963400379
836262026
404531926
361015349
675133609
```

Обратный ход

Задача:

- $s \leftarrow (s \cdot a + c) \bmod m$
- Мы знаем a, c и m.
- Пусть мы знаем $s_1 = (s_0 \cdot a + c) \bmod m$.
- Как найти s_0?

Обратный ход

Задача:

- $s \leftarrow (s \cdot a + c) \bmod m$
- Мы знаем a, c и m.
- Пусть мы знаем $s_1 = (s_0 \cdot a + c) \bmod m$.
- Как найти s_0?

Решим сравнение по модулю:

- $s_1 - c \equiv s_0 \cdot a \pmod{m}$
- $s_0 = (s_1 - c) \cdot a^{-1} \pmod{m}$

Восстановление состояния

Задача:

- $s \leftarrow (s \cdot a + c) \bmod m$
- Мы знаем a, c и m.
- Пусть мы знаем $x = \texttt{random}(k)$.
- Каким могло быть s_0?

Восстановление состояния

Задача:

- $s \leftarrow (s \cdot a + c) \bmod m$
- Мы знаем a, c и m.
- Пусть мы знаем $x = \texttt{random}(k)$.
- Каким могло быть s_0?

Переберём, каким могло быть s_1:

- $x = s_1 \bmod k$.
- Значит, $s_1 = x$, или $s_1 = x + k$, или $s_1 = x + 2k$, или...
- Например, если $x = 12\,345$ и $k = 1\,000\,000$, то $s_1 = 12\,345$ или $1\,012\,345$ или $2\,012\,345$ или...

Восстановление состояния

Задача:

- $s \leftarrow (s \cdot a + c) \bmod m$
- Мы знаем a, c и m.
- Пусть мы знаем $x = \texttt{random}(k)$.
- Каким могло быть s_0?

Переберём, каким могло быть s_1:

- $x = s_1 \bmod k$.
- Значит, $s_1 = x$, или $s_1 = x + k$, или $s_1 = x + 2k$, или...
- Например, если $x = 12\,345$ и $k = 1\,000\,000$, то $s_1 = 12\,345$ или $1\,012\,345$ или $2\,012\,345$ или...
- В общем случае $s_1 = x + t \cdot k$, где t — неотрицательное целое число, и $0 \leqslant s_1 < m$.
- Осталось перебрать все такие s_1 — их примерно m/k — и для каждого найти s_0.

Значения через несколько шагов

Задача:

- $s \leftarrow (s \cdot a + c) \bmod m$
- Мы знаем a, c и m.
- Как по s_0 найти s_k для большого k?

Значения через несколько шагов

Задача:

- $s \leftarrow (s \cdot a + c) \bmod m$
- Мы знаем a, c и m.
- Как по s_0 найти s_k для большого k?

Распишем:

- $s_1 \equiv (s_0 \cdot a + c) \pmod{m}$
- $s_2 \equiv (s_1 \cdot a + c) \pmod{m}$
- $s_2 \equiv ((s_0 \cdot a + c) \cdot a + c) \pmod{m}$
- $s_2 \equiv s_0 \cdot a^2 + c \cdot (a + 1) \pmod{m}$

Значения через несколько шагов

Задача:

- $s \leftarrow (s \cdot a + c) \bmod m$
- Мы знаем a, c и m.
- Как по s_0 найти s_k для большого k?

Распишем:

- $s_1 \equiv (s_0 \cdot a + c) \pmod{m}$
- $s_2 \equiv (s_1 \cdot a + c) \pmod{m}$
- $s_2 \equiv ((s_0 \cdot a + c) \cdot a + c) \pmod{m}$
- $s_2 \equiv s_0 \cdot a^2 + c \cdot (a + 1) \pmod{m}$
- $s_3 \equiv s_0 \cdot a^3 + c \cdot (a^2 + a + 1) \pmod{m}$

Значения через несколько шагов

Задача:

- $s \leftarrow (s \cdot a + c) \bmod m$
- Мы знаем a, c и m.
- Как по s_0 найти s_k для большого k?

Распишем:

- $s_1 \equiv (s_0 \cdot a + c) \pmod{m}$
- $s_2 \equiv (s_1 \cdot a + c) \pmod{m}$
- $s_2 \equiv ((s_0 \cdot a + c) \cdot a + c) \pmod{m}$
- $s_2 \equiv s_0 \cdot a^2 + c \cdot (a + 1) \pmod{m}$
- $s_3 \equiv s_0 \cdot a^3 + c \cdot (a^2 + a + 1) \pmod{m}$
- $s_k \equiv s_0 \cdot a^k + c \cdot (a^{k-1} + \ldots + a + 1) \pmod{m}$

Значения через несколько шагов

Задача:

- $s \leftarrow (s \cdot a + c) \bmod m$
- Мы знаем a, c и m.
- Как по s_0 найти s_k для большого k?

Распишем:

- $s_1 \equiv (s_0 \cdot a + c) \pmod{m}$
- $s_2 \equiv (s_1 \cdot a + c) \pmod{m}$
- $s_2 \equiv ((s_0 \cdot a + c) \cdot a + c) \pmod{m}$
- $s_2 \equiv s_0 \cdot a^2 + c \cdot (a + 1) \pmod{m}$
- $s_3 \equiv s_0 \cdot a^3 + c \cdot (a^2 + a + 1) \pmod{m}$
- $s_k \equiv s_0 \cdot a^k + c \cdot (a^{k-1} + \ldots + a + 1) \pmod{m}$
- a^k и сумма геометрической прогрессии считаются за $O(\log k)$ техникой, аналогичной быстрому возведению в степень.

Значения через несколько шагов

Задача:

- $s \leftarrow (s \cdot a + c) \bmod m$
- Мы знаем a, c и m.
- Как по s_0 найти s_k для большого k?

Через матрицы над $\mathbb{Z}_m$:

- $s_1 \equiv (s_0 \cdot a + c) \pmod{m}$
- $\begin{pmatrix} s_k \\ 1 \end{pmatrix} = \begin{pmatrix} a & c \\ 0 & 1 \end{pmatrix} \begin{pmatrix} s_{k-1} \\ 1 \end{pmatrix}$

Значения через несколько шагов

Задача:

- $s \leftarrow (s \cdot a + c) \bmod m$
- Мы знаем a, c и m.
- Как по s_0 найти s_k для большого k?

Через матрицы над $\mathbb{Z}_m$:

- $s_1 \equiv (s_0 \cdot a + c) \pmod{m}$
- $\begin{pmatrix} s_k \\ 1 \end{pmatrix} = \begin{pmatrix} a & c \\ 0 & 1 \end{pmatrix} \begin{pmatrix} s_{k-1} \\ 1 \end{pmatrix}$
- $\begin{pmatrix} s_k \\ 1 \end{pmatrix} = \begin{pmatrix} a & c \\ 0 & 1 \end{pmatrix}^k \begin{pmatrix} s_0 \\ 1 \end{pmatrix}$
- Осталось за $O(\log k)$ возвести матрицу в степень.

Значения через несколько шагов

Задача:

- $s \leftarrow (s \cdot a + c) \bmod m$
- Мы знаем a, c и m.
- Как по s_0 найти s_k для большого k?

Через матрицы над $\mathbb{Z}_m$:

- $s_1 \equiv (s_0 \cdot a + c) \pmod{m}$
- $\begin{pmatrix} s_k \\ 1 \end{pmatrix} = \begin{pmatrix} a & c \\ 0 & 1 \end{pmatrix} \begin{pmatrix} s_{k-1} \\ 1 \end{pmatrix}$
- $\begin{pmatrix} s_k \\ 1 \end{pmatrix} = \begin{pmatrix} a & c \\ 0 & 1 \end{pmatrix}^k \begin{pmatrix} s_0 \\ 1 \end{pmatrix}$
- Осталось за $O(\log k)$ возвести матрицу в степень.
- Можно хранить не матрицу, а только линейную функцию — множитель и слагаемое. В матрице они занимают верхнюю строку.

Значения через несколько шагов

Задача:

- $s \leftarrow (s \cdot a + c) \bmod m$
- Мы знаем a, c и m.
- Как по s_0 найти s_k для большого k?

Ещё быстрее:

- Запишем линейную функцию, прыгающую на r шагов, как пару $f_r = (a_r, c_r)$:
- $s_r \equiv (s_0 \cdot a_r + c_r) \pmod m$

Значения через несколько шагов

Задача:

- $s \leftarrow (s \cdot a + c) \bmod m$
- Мы знаем a, c и m.
- Как по s_0 найти s_k для большого k?

Ещё быстрее:

- Запишем линейную функцию, прыгающую на r шагов, как пару $f_r = (a_r, c_r)$.
- Научимся делать композицию линейных функций:
- Пусть $f(x) = a_f x + c_f$ и $g(x) = a_g x + c_g$.
- $f(g(x)) = a_f(a_g x + c_g) + c_f$, раскроем скобки:
- $f(g(x)) = a_f a_g x + a_f c_g + c_f$

Значения через несколько шагов

Задача:

- $s \leftarrow (s \cdot a + c) \bmod m$
- Мы знаем a, c и m.
- Как по s_0 найти s_k для большого k?

Ещё быстрее:

- Запишем линейную функцию, прыгающую на r шагов, как пару $f_r = (a_r, c_r)$.
- $f(g(x)) = a_f a_g x + a_f c_g + c_f$
- Далее пусть $t = \lceil \sqrt{m} \rceil$.
- Вычислим $f_1, f_2, f_3, \ldots, f_t$ и $f_t, f_{2t}, f_{3t}, \ldots, f_{t \cdot t}$.
- Теперь $f_k(x) = f_{k \bmod t}(f_{(k \operatorname{div} t) \cdot t}(x))$.
- Время: $O(\sqrt{m})$ на предподсчёт, а затем $O(1)$ на вычисление.

Содержание

Хеширование строк

Идея:

- В строке $S = S_1 S_2 \dots S_n$ важен порядок символов.
- Хеш строки: $h(S) = (h(S_1 \dots S_{n-1}) \cdot p + S_n) \bmod q$.
- Подробнее разбирались на прошлом занятии.

Хеширование строк

Идея:

- В строке $S = S_1 S_2 \dots S_n$ важен порядок символов.
- Хеш строки: $h(S) = (h(S_1 \dots S_{n-1}) \cdot p + S_n) \bmod q$.
- Подробнее разбирались на прошлом занятии.
- То же самое работает для последовательностей: элементы — не обязательно символы.

Хеширование множеств

Идея:

- Во множестве порядок элементов не важен.
- Заранее сопоставим каждому возможному элементу случайное целое число — например, равномерно распределённое от 0 до $2^{63} - 1$.
- Хеш множества — это xor всех чисел, соответствующих его элементам.
- Заметим, что xor-сумма нескольких (для начала двух) независимых случайных чисел с таким распределением — тоже случайное число с таким распределением.

Хеширование множеств

Идея:

- Во множестве порядок элементов не важен.
- Заранее сопоставим каждому возможному элементу случайное целое число — например, равномерно распределённое от 0 до $2^{63} - 1$.
- Хеш множества — это xor всех чисел, соответствующих его элементам.
- Заметим, что xor-сумма нескольких (для начала двух) независимых случайных чисел с таким распределением — тоже случайное число с таким распределением.
- Какая операция с двумя множествами получается легко и удобно?

Хеширование множеств

Идея:

- Во множестве порядок элементов не важен.
- Заранее сопоставим каждому возможному элементу случайное целое число — например, равномерно распределённое от 0 до $2^{63} - 1$.
- Хеш множества — это xor всех чисел, соответствующих его элементам.
- Заметим, что xor-сумма нескольких (для начала двух) независимых случайных чисел с таким распределением — тоже случайное число с таким распределением.
- Какая операция с двумя множествами получается легко и удобно?
- Симметрическая разность: $h(A \triangle B) = h(A)$ xor $h(B)$.

Вопросы?

Вопросы?