

Ограбление банка

Автор задачи и разработчик: Антон Вдовин

Обозначим i -й символ j -й полученной строки за s_i^j . Тогда исходная строка равна $s^1 = \overline{s_0^1, \dots, s_{n-1}^1}$, а для каждой следующей $s_i^{j+1} = s_i^j \oplus s_j^{i+1}$. Сходу стоит заметить, что эта формула чем-то напоминает формулу для подсчета числа сочетаний: $C_{n+1}^k = C_n^k + C_{n-1}^k$.

Также далее будем пользоваться эквивалентностью операции исключающего ИЛИ сложению по модулю 2. Поэтому будем записывать $x \oplus x \oplus \dots \oplus x$, где x встречается y раз, как $y \cdot x$.

Распишем процесс полностью:

- Изначально строка равна s^1 .

- Строка s^2 получается как

$$s_i^2 = s_i^1 \oplus s_1^{i+1}$$

- На следующем шаге s^3 так же выражается через s^2 . Раскроем то, что получили выше, и получим, что

$$s_i^2 = s_1^i + 2 \cdot s_{i+1}^1 + s_{i+2}^1$$

- Если сложить полученные выражения для s_i^3 и s_{i+1}^3 , получим

$$s_i^4 = s_i^1 + 3 \cdot s_{i+1}^1 + 3 \cdot s_{i+2}^1 + s_{i+3}^1$$

- И так далее, на каждом шаге будем выписывать XOR соседних элементов предыдущей строки.

Несложно заметить, что коэффициенты, получающиеся при битах исходной строки — это значения из треугольника Паскаля. А как известно, значения в треугольнике Паскаля — это биномиальные коэффициенты C_n^k . Таким образом:

$$s_1^{n-1} = C_{n-1}^0 s_0^1 \oplus C_{n-1}^1 s_1^1 \oplus \dots \oplus C_{n-1}^{n-1} s_{n-1}^1.$$

Чтобы посчитать это значение, вспомним, что на самом деле мы считаем не сумму, а XOR, а значит нам важен только остаток полученной суммы по модулю 2. А найти четность биномиального коэффициента можно по *теореме Люка*, которая утверждает, что C_n^k нечетно тогда и только тогда, когда нет бита, равного 1 в k и 0 в n . Это можно проверить за $\mathcal{O}(1)$ следующим выражением: $k \wedge \neg n \neq 0$ (либо же просто проитерироваться по битам и сравнить каждый).

Таким образом, посчитаем четность каждого биномиального коэффициента C_{n-1}^i , умножим на соответствующий бит исходной строки, и сложим. Получим наш ответ за время $\mathcal{O}(n)$.