

$[1, n]$ в $[1, n]$, при этом коэффициент μ_α определяется суммой

$$\mu_\alpha = \sum_{\epsilon_i \in \{0,1\}} (-1)^{\epsilon_1 + \dots + \epsilon_n} \epsilon_{\alpha(1)} \dots \epsilon_{\alpha(n)}.$$

Если α — перестановка, коэффициент μ_α равен $(-1)^n$, поскольку единственный ненулевой член суммы — это тот, для которого все ϵ_i , равны 1. В противном случае можно разложить μ_α следующим образом:

$$\sum_{\epsilon_i, i \neq k} (-1)^{\epsilon_1 + \dots + \epsilon_n} \sum_{\epsilon_k=0,1} (-1)^{\epsilon_k} \epsilon_{\alpha(1)} \dots \epsilon_{\alpha(n)},$$

где k — элемент из $[1, n]$, который не принадлежит образу α , и хорошо видно, что внутренний член (сумма по ϵ_k) равен нулю; это доказывает, что μ_α в этом случае равно нулю. Второе искомое выражение (суммы которого записываются на множествах) просто выводится из первого, исключением ненулевых ϵ_i , появляющихся в сумме.

b. *// here goes some text and code*

$$\sum_{E'} = \sum_E + (-1)^{|E'|} \prod_{1 \leq i \leq n} S_i(E'). \quad (7)$$

Вклад наименьшего элемента \emptyset в эту сумму — нулевой; значит, начнем с его последователя, который может быть $\{1\}$, $\{n\}$ или какое-нибудь одноэлементное множество в соответствии с порядком, заданным на $[1, n]$. Можно заметить, что $S_i(E') = S_i(E) \pm a_{ij}$, где $\{j\} = E \triangle E'$.

В этой записи, как в алгоритме 12, \pm должен пониматься как $+$, если $E \subset E'$ и $-$, если $E' \subset E$.

Чтобы оценить $\sum_{E'}$, исходя из \sum_E с использованием соотношения (7), нужно осуществить n сложений (для подсчета каждого $S_i(E')$), затем $n - 1$ перемножений: $S_1(E') \times \dots \times S_n(E')$, и, наконец, 1 сложение. Имеем $2^n - 2$ операций для осуществления (7), при этом первый член $\sum = -\prod_{1 \leq i \leq n} \alpha_{in}$ требует $n - 1$ перемножений; это доказывает сформулированный результат о сложности. Сложность $O(n2^n)$ значительна, но остается того же порядка, что и сложность, индуцированная определением ($n!(n - 1)$ перемножений и $n! - 1$ сложений). Формула Стирлинга позволяет сравнить эти два значения сложности: $n \cdot n! / (n \cdot 2^n) \approx (n/2e)^n \sqrt{2\pi n}$.

22. Перманент матрицы (продолжение)

а. Правая часть может рассматриваться как многочлен (от переменных a_{ij}), равный $\sum_{\alpha} \mu_{\alpha} a_{1\alpha(1)} \dots a_{n\alpha(n)}$, где сумма распространяется на все отображения $[1, n]$ в $[1, n]$. Коэффициент μ_{α} задан формулой

$$\mu_{\alpha} = \sum_{\omega} \mu_{\alpha}(\omega) \quad \text{с} \quad \mu_{\alpha}(\omega) = \omega_1 \dots \omega_{n-1} \omega_{\alpha(1)} \dots \omega_{\alpha(n)},$$

в которой полагаем $\omega_n = 1$. Если α — перестановка, то каждое $\mu_{\alpha}(\omega)$, присутствующее в сумме μ_{α} , равно 1 и, следовательно, $\mu_{\alpha} = 2^{n-1}$. Напротив, если α не является перестановкой, то сумма μ_{α} — нулевая. Действительно, образ α отличен от $[1, n]$, и различаем два случая:

- (i) $\exists k < n$, не принадлежащий образу α ,
- (ii) $\exists k < n$, дважды полученный из α .

В обоих случаях члены $\mu_{\alpha}(\omega)$, присутствующие в сумме, группируются попарно, один соответствуя $\omega_k = 1$, другой — $\omega_k = -1$, и взаимно уничтожаются (в случае (ii) $\mu_{\alpha}(\omega) = \omega_k$).

б. Формула пункта **а** может быть записана в следующем виде:

$$\frac{\text{per} A}{2} = \sum \omega_1 \dots \omega_{n-1} \prod_{1 \leq i \leq n} (a_{in} + \omega_1 a_{i1} + \dots + \omega_{n-1} a_{in-1}) / 2.$$

Как и в предыдущем упражнении, вычисление перманента получается генерированием перебора при линейной упорядоченности на $\{-1, 1\}^{n-1}$, в которой два последовательных элемента отличаются только одной компонентой. Если для $\omega \in \{-1, 1\}^{n-1}$ и $i \leq n$ положить

$$S_i(\omega) = (a_{in} + \omega_1 a_{i1} + \dots + \omega_{n-1} a_{in-1}) / 2,$$

то можно, благодаря перебору на $\{-1, 1\}^{n-1}$, вычислить последовательно $\sum_{\omega} = \sum_{\rho \leq \omega} \dots$, используя формулу

$$\sum_{\omega'} = \sum_{\omega} + \omega'_1 \dots \omega'_{n-1} \prod_{1 \leq i \leq n} S_i(\omega'),$$

где ω' — последователь для ω в $\{-1, 1\}^{n-1}$. (8)

Если j является индексом, по которому различаются два слова ω и ω' , то сумма $S_i(\omega')$ вычисляется, исходя из $S_i(\omega)$, через $S_i(\omega') = S_i(\omega) - a_{ij}$, если $\omega_j = 1$, и через $S_i(\omega') = S_i(\omega) + a_{ij}$, если $\omega_j = -1$.

// here goes some code

Алгоритм 13. Вычисление перманента в кольце, где 2 обратимо

С практической точки зрения для генерации адекватного перебора $\{-1, 1\}^{n-1}$, выбираем соответствие между $\{0, 1\}$ и $\{-1, 1\}$ вида $\epsilon \mapsto (-1)^\epsilon$ и классическую генерацию кода Грея на $\{0, 1\}^{n-1}$, что достигается с помощью алгоритма 13. Мультипликативная сложность получается, если заметить, что нужно вычислить $2^n - 1$ членов \sum_{ω} , каждый из которых требует $n - 1$ перемножений (см. формулу (8)), значит, всего $2^{n-1}(n - 1)$ произведений, к которым нужно добавить последнее умножение на 2. С точки зрения сложений первоначальный член $\sum_{(1, \dots, 1)}$ требует $n(n - 1)$ сложений и n делений на 2, тогда как общий член \sum_{ω} вычисляется, исходя из предыдущего, с помощью n сложений; наконец, нужно сложить все эти члены, что требует в целом $n(n - 1) + (2^{n-1} - 1)(n + 1)$ сложений и n делений на 2. Заметим относительно предыдущего упражнения, что сложность была приблизительно разделена на 2.

с. Рассмотрения полностью аналогичны предыдущему пункту, если только невозможно деление на 2; деление (точное) на 2^{n-1} будет иметь место уже в конце. Результатом является алгоритм 14.

// here goes some code

23. Массив инверсий подстановки

д. Массив инверсий перестановки α имеет вид $(0, 0, 0, 1, 4, 2, 1, 5, 7)$. Свойство $0 \leq \alpha_k < k$ легко получается из того, что имеется точно $k-1$ целых чисел, заключенных строго между 0 и k . Массив инверсий возрастающей перестановки интервала $[1, n]$ есть, очевидно, $(0, 0, \dots, 0)$, и таблица для единственной убывающей перестановки — $(0, 1, 2, \dots, n)$.

е. Можно использовать тот факт, что $a_{\alpha(j)}$ есть число индексов таких, что $i > j$ и $\alpha(i) < \alpha(j)$, что приводит к нижеследующему алгоритму:

// here goes some code

Сложность полученного способа, конечно, имеет порядок квадрата длины перестановки.

f. Пусть a — элемент из $[0, 1[\times [0, 2[\times \cdots \times [0, n[$. Построим перестановку α , для которой a является массивом инверсий, следующим способом:

- элемент n помещаем в массив, индексированный с помощью $[1, n]$, представляющий α , оставляя a_n **пустых ячеек** справа от n ; это означает в точности, что $\alpha^{-1}(n) = n - a_n + 1$;

- затем помещаем $n - 1$ в массив α , оставляя α_{n-1} **пустых ячеек** справа от $n - 1$;

- продолжаем, зная, что на k -м этапе этого процесса $k - 1$ величин уже размещены в массиве, следовательно, в массиве α остается $n - k$ свободных мест, и, с другой стороны, величина α_{n-k} строго меньше, чем $n - k$.

// here goes some code

В алгоритме 15 использована оптимизация: свободные места в массиве, представляющем перестановку α , отмечены числами 1, что позволяет не помещать это последнее значение в массив, представляющий α , в конце алгоритма.

24. Перебор перестановок транспозициями $(i, i + 1)$

a. Рассуждаем индукцией по n , при этом случаи $n = 1$ и $n = 2$ очевидны. С помощью перестановки σ интервала $[1, n]$ можно построить $n + 1$ перестановок $\sigma^1, \sigma^2, \dots, \sigma^{n+1}$ интервала $[1, n + 1]$, где перестановка σ^i получается включением в σ элемента $n + 1$ на i -ое место; например,