Лабораторная работа № 7
Генерация разбиений множества.
Треугольники Стирлинга (4 часа)

Краткий теоретический материал

Генерация всех разбиений множества

Разбиения множества представляют собой способы рассматривать множество как объединение непустых непересекающихся подмножеств, именуемых *блоками*. Например, пять различных разбиений множества $\{1, 2, 3\}$ можно компактно записать в виде

$$123,\ 12|3,\ 13|2,\ 1|23,\ 1|2|3 \tag{1}$$

с использованием вертикальной черты для отделения одного блока от другого. В этом списке элементы каждого блока могут быть записаны в любом порядке, как и сами блоки, так что “13|2”, “31|2”, “2|13”, и “2|31” – это одно и то же разбиение. Можно стандартизировать представление путем соглашения, например, о перечислении элементов каждого блока в возрастающем порядке, а сами блоки располагать в порядке возрастания их наименьших элементов. При этих соглашениях разбиениями множества $\{1, 2, 3, 4\}$ являются

$$\begin{gathered}1234,\ 123|4,\ 124|3,\ 12|34,\ 12|3|4,\ 134|2,\ 13|24,\ 13|2|4,\\ 14|23,\ 1|234,\ 1|23|4,\ 14|2|3,\ 1|24|3,\ 1|2|34,\ 1|2|3|4,\end{gathered} \tag{2}$$

получаемые путем добавления 4 всеми возможными способами к блокам (1).

Заданное на множестве отношение эквивалентности задаёт разбиение этого множества на *классы эквивалентности*. Верно и обратное: каждое разбиение множества определяет отношение эквивалентности; в частности, если П является разбиением множества $\{1, 2, \dots, n\}$, можно записать $j \sim k$, если j и k принадлежат одному и тому же блоку П.

Один из наиболее удобных способов представления разбиения множества в компьютере состоит в кодировании его *ограниченно растущей строкой*, т. е. строкой $a_1 a_2 \dots a_n$ в которой

$$a_1 = 0 \text{ и } a_{j+1} \le 1 + max(a_1, \dots, a_j) \text{ для } 1 \le j < n \tag{4}$$

Идея заключается в том, чтобы $a_j = a_k$ тогда и только тогда, когда $j \sim k$, причем выбирать наименьшее доступное число для a_j, где $j \sim k$ – наименьшее число в своем блоке. Например, ограниченно растущие строки для 15 разбиений (2) представляют собой соответственно

$$0000, 0001, 0010, 0011, 0012, 0100, 0101, 0102, \qquad (5)$$

$$0110, 0111, 0112, 0120, 0121, 0122, 0123$$

Такое соглашение наводит на мысль о следующей схеме генерации разбиений.

В этом алгоритме функция $[x]$ действует следующим образом:

$$[x] = \begin{cases} 1, \text{если } x \text{ истинно}, \\ 0, \text{если } x \text{ ложно}. \end{cases}$$

Алгоритм (Ограниченно растущие строки в лексикографическом порядке). Для данного $n \geq 2$ этот алгоритм генерирует все разбиения $\{1, 2, \dots, n\}$ путем посещения всех строк $a_1 a_2 \dots a_n$, удовлетворяющих условию ограниченного роста (4). Поддерживается вспомогательный массив $b_1 b_2 \dots b_n$, где $b_{j+1} = 1 + max(a_1, \dots, a_j)$; значение b_n из соображений эффективности реально содержится в отдельной переменной m.

Шаг 1. [Инициализация.] Установить $a_1 \dots a_n \leftarrow 0 \dots 0$, $b_1 \dots b_{n-1} \leftarrow 1 \dots 1$, $m \leftarrow 1$.

Шаг 2. [Посещение.] Посетить ограниченно растущую строку $a_1 \dots a_n$, которая представляет собой разбиение на $m + [a_n = m]$ блоков. Затем, если $a_n = m$, перейти к шагу 4.

Шаг 3. [Увеличение a_n.] Установить $a_n \leftarrow a_n + 1$ и вернуться к шагу 2.

Шаг 4. [Поиск j.] Установить $j \leftarrow n - 1$; затем, пока $a_j = b_j$ устанавливать $j \leftarrow j - 1$.

Шаг 5. [Увеличение a_j.] Завершить работу алгоритма, если $j = 1$. В противном случае установить $a_j \leftarrow a_j + 1$.

Шаг 6. [Обнуление $a_{j+1} \dots a_n$.] Установить $m \leftarrow b_j + [a_j = b_j]$ и $j \leftarrow j + 1$. Затем, пока $j < n$, устанавливать $a_j \leftarrow 0$, $b_j \leftarrow m$ и $j \leftarrow j + 1$. Наконец, установить $a_n \leftarrow 0$ и вернуться к шагу 2.

Задание

1. Разработать программу для генерации всех разбиений множества $\{1, 2, \ldots, n\}$ при заданном n.

2. Используя результат выполнения предыдущего задания, разработать программу, которая строит треугольника Стирлинга второго рода. Количество выводимых строк треугольника задаёт пользователь.

Убедиться, что полученный числовой треугольник совпадает с треугольником, приведённым на лекционных занятиях.

Важное замечание: для нахождения чисел Стирлинга второго рода необходимо выполнять генерацию всех разбиений множества; пользоваться рекуррентной формулой запрещено.

3. Построить треугольник Стирлинга второго рода, используя рекуррентное соотношение (см. лекции). Сравнить результат с результатом предыдущего задания.

4. Разработать программу, которая строит треугольник Стирлинга первого рода. Для генерации всех перестановок необходимо использовать результаты предыдущих лабораторных. Алгоритм нахождения количества циклов в перестановке разработать самостоятельно. Количество выводимых строк треугольника задаёт пользователь.

Убедиться, что полученный числовой треугольник совпадает с треугольником, приведённым на лекционных занятиях.

Важное замечание: для нахождения чисел Стирлинга первого рода необходимо использовать генерацию всех перестановок множества и подсчёт числа циклов в перестановках; пользоваться рекуррентной формулой запрещено.

5. Построить треугольник Стирлинга первого рода, используя рекуррентное соотношение (см. лекции). Сравнить результат с результатом предыдущего задания.

Требования к отчету

Отчет по лабораторной работе должен включать:

1. Титульный лист; задание; исходный код.
2. Примеры работы программы (скриншоты).
3. Выводы.

Литература

Кнут Д. Э. Искусство программирования, том 4, А. Комбинаторные алгоритмы, часть 1 / Москва: Вильямс, 2013. – Т. 4. – 960 с.