Комбинаторика: объект по номеру и номер по объекту

Иван Казменко

Летняя сессия по информатике, Лисий Нос

Понедельник, 9 июля 2012 года

Оглавление

Оглавление

Лексикографический порядок

Постановка задачи: нужно упорядочить составные объекты — например, строки или последовательности.

Лексикографический порядок

Постановка задачи: нужно упорядочить составные объекты — например, строки или последовательности.

1. Сначала зададим порядок на элементах:
   - для символов: $\text{‘a’} < \text{‘b’} < \ldots < \text{‘z’}$,
   - для чисел: $\ldots < -2 < -1 < 0 < 1 < 2 < 3 < \ldots$.
2. Затем будем сравнивать составные объекты поэлементно, просматривая их с начала:
   - если текущие элементы не равны, меньше тот составной объект, в котором текущий элемент меньше;
   - если один из составных объектов кончился, а другой ещё нет, меньше тот, который кончился;
   - если оба составных объекта кончились одновременно, они равны.

Лексикографический порядок

Постановка задачи: нужно упорядочить составные объекты — например, строки или последовательности.

1. Сначала зададим порядок на элементах:
   - для символов: ‘a’ $<$ ‘b’ $< \ldots <$ ‘z’,
   - для чисел: $\ldots < -2 < -1 < 0 < 1 < 2 < 3 < \ldots$.
2. Затем будем сравнивать составные объекты поэлементно, просматривая их с начала:
   - если текущие элементы не равны, меньше тот составной объект, в котором текущий элемент меньше;
   - если один из составных объектов кончился, а другой ещё нет, меньше тот, который кончился;
   - если оба составных объекта кончились одновременно, они равны.

Лексикографический порядок

Постановка задачи: нужно упорядочить составные объекты — например, строки или последовательности.

1. Сначала зададим порядок на элементах:
    - для символов: ‘a’ < ‘b’ < ... < ‘z’,
    - для чисел: $\ldots < -2 < -1 < 0 < 1 < 2 < 3 < \ldots$.
2. Затем будем сравнивать составные объекты поэлементно, просматривая их с начала:
    - если текущие элементы не равны, меньше тот составной объект, в котором текущий элемент меньше;
    - если один из составных объектов кончился, а другой ещё нет, меньше тот, который кончился;
    - если оба составных объекта кончились одновременно, они равны.

Примеры:

$$\mathtt{a} \sim \mathtt{b}$$

Лексикографический порядок

Постановка задачи: нужно упорядочить составные объекты — например, строки или последовательности.

1. Сначала зададим порядок на элементах:
   - для символов: ‘a’ $<$ ‘b’ $< \ldots <$ ‘z’,
   - для чисел: $\ldots < -2 < -1 < 0 < 1 < 2 < 3 < \ldots$.
2. Затем будем сравнивать составные объекты поэлементно, просматривая их с начала:
   - если текущие элементы не равны, меньше тот составной объект, в котором текущий элемент меньше;
   - если один из составных объектов кончился, а другой ещё нет, меньше тот, который кончился;
   - если оба составных объекта кончились одновременно, они равны.

Примеры:

$$\text{a} < \text{b}$$

Лексикографический порядок

Постановка задачи: нужно упорядочить составные объекты — например, строки или последовательности.

1. Сначала зададим порядок на элементах:
   - для символов: $\text{'a'} < \text{'b'} < \ldots < \text{'z'}$,
   - для чисел: $\ldots < -2 < -1 < 0 < 1 < 2 < 3 < \ldots$.
2. Затем будем сравнивать составные объекты поэлементно, просматривая их с начала:
   - если текущие элементы не равны, меньше тот составной объект, в котором текущий элемент меньше;
   - если один из составных объектов кончился, а другой ещё нет, меньше тот, который кончился;
   - если оба составных объекта кончились одновременно, они равны.

Примеры:

$$\mathtt{a} \sim \mathtt{ab}$$

Лексикографический порядок

Постановка задачи: нужно упорядочить составные объекты — например, строки или последовательности.

1. Сначала зададим порядок на элементах:
    - для символов: ‘a’ $<$ ‘b’ $< \ldots <$ ‘z’,
    - для чисел: $\ldots < -2 < -1 < 0 < 1 < 2 < 3 < \ldots$.
2. Затем будем сравнивать составные объекты поэлементно, просматривая их с начала:
    - если текущие элементы не равны, меньше тот составной объект, в котором текущий элемент меньше;
    - если один из составных объектов кончился, а другой ещё нет, меньше тот, который кончился;
    - если оба составных объекта кончились одновременно, они равны.

Примеры:

$$\mathrm{a} < \mathrm{ab}$$

Лексикографический порядок

Постановка задачи: нужно упорядочить составные объекты — например, строки или последовательности.

1. Сначала зададим порядок на элементах:
   - для символов: ‘a’ $<$ ‘b’ $< \ldots <$ ‘z’,
   - для чисел: $\ldots < -2 < -1 < 0 < 1 < 2 < 3 < \ldots$.
2. Затем будем сравнивать составные объекты поэлементно, просматривая их с начала:
   - если текущие элементы не равны, меньше тот составной объект, в котором текущий элемент меньше;
   - если один из составных объектов кончился, а другой ещё нет, меньше тот, который кончился;
   - если оба составных объекта кончились одновременно, они равны.

Примеры:

$$\mathtt{abc} \sim \mathtt{acab}$$

Лексикографический порядок

Постановка задачи: нужно упорядочить составные объекты — например, строки или последовательности.

1. Сначала зададим порядок на элементах:
   - для символов: ‘a’ $<$ ‘b’ $< \ldots <$ ‘z’,
   - для чисел: $\ldots < -2 < -1 < 0 < 1 < 2 < 3 < \ldots$.
2. Затем будем сравнивать составные объекты поэлементно, просматривая их с начала:
   - если текущие элементы не равны, меньше тот составной объект, в котором текущий элемент меньше;
   - если один из составных объектов кончился, а другой ещё нет, меньше тот, который кончился;
   - если оба составных объекта кончились одновременно, они равны.

Примеры:

abc < acab

Лексикографический порядок

Постановка задачи: нужно упорядочить составные объекты — например, строки или последовательности.

1. Сначала зададим порядок на элементах:
   - для символов: ‘a’ $<$ ‘b’ $< \ldots <$ ‘z’,
   - для чисел: $\ldots < -2 < -1 < 0 < 1 < 2 < 3 < \ldots$.
2. Затем будем сравнивать составные объекты поэлементно, просматривая их с начала:
   - если текущие элементы не равны, меньше тот составной объект, в котором текущий элемент меньше;
   - если один из составных объектов кончился, а другой ещё нет, меньше тот, который кончился;
   - если оба составных объекта кончились одновременно, они равны.

Примеры:

$$\mathrm{abc} \sim \mathrm{abaab}$$

Лексикографический порядок

Постановка задачи: нужно упорядочить составные объекты — например, строки или последовательности.

1. Сначала зададим порядок на элементах:
   - для символов: ‘a’ $<$ ‘b’ $< \ldots <$ ‘z’,
   - для чисел: $\ldots < -2 < -1 < 0 < 1 < 2 < 3 < \ldots$.
2. Затем будем сравнивать составные объекты поэлементно, просматривая их с начала:
   - если текущие элементы не равны, меньше тот составной объект, в котором текущий элемент меньше;
   - если один из составных объектов кончился, а другой ещё нет, меньше тот, который кончился;
   - если оба составных объекта кончились одновременно, они равны.

Примеры:

$$\mathtt{abc} > \mathtt{abaab}$$

Лексикографический порядок

Постановка задачи: нужно упорядочить составные объекты — например, строки или последовательности.

1. Сначала зададим порядок на элементах:
   - для символов: $\text{`a'} < \text{`b'} < \ldots < \text{`z'}$,
   - для чисел: $\ldots < -2 < -1 < 0 < 1 < 2 < 3 < \ldots$.
2. Затем будем сравнивать составные объекты поэлементно, просматривая их с начала:
   - если текущие элементы не равны, меньше тот составной объект, в котором текущий элемент меньше;
   - если один из составных объектов кончился, а другой ещё нет, меньше тот, который кончился;
   - если оба составных объекта кончились одновременно, они равны.

Примеры:

$$\mathtt{ab} \sim \mathtt{ab}$$

Лексикографический порядок

Постановка задачи: нужно упорядочить составные объекты — например, строки или последовательности.

1. Сначала зададим порядок на элементах:
   - для символов: ‘a’ $<$ ‘b’ $< \ldots <$ ‘z’,
   - для чисел: $\ldots < -2 < -1 < 0 < 1 < 2 < 3 < \ldots$.
2. Затем будем сравнивать составные объекты поэлементно, просматривая их с начала:
   - если текущие элементы не равны, меньше тот составной объект, в котором текущий элемент меньше;
   - если один из составных объектов кончился, а другой ещё нет, меньше тот, который кончился;
   - если оба составных объекта кончились одновременно, они равны.

Примеры:

$$\mathtt{ab} = \mathtt{ab}$$

Лексикографический порядок

Постановка задачи: нужно упорядочить составные объекты — например, строки или последовательности.

1. Сначала зададим порядок на элементах:
    - для символов: 'a' $<$ 'b' $< \ldots <$ 'z',
    - для чисел: $\ldots < -2 < -1 < 0 < 1 < 2 < 3 < \ldots$.
2. Затем будем сравнивать составные объекты поэлементно, просматривая их с начала:
    - если текущие элементы не равны, меньше тот составной объект, в котором текущий элемент меньше;
    - если один из составных объектов кончился, а другой ещё нет, меньше тот, который кончился;
    - если оба составных объекта кончились одновременно, они равны.

Примеры:

$$(1, 2, 3) \sim (1)$$

Лексикографический порядок

Постановка задачи: нужно упорядочить составные объекты — например, строки или последовательности.

1. Сначала зададим порядок на элементах:
   - для символов: $\text{‘a’} < \text{‘b’} < \ldots < \text{‘z’}$,
   - для чисел: $\ldots < -2 < -1 < 0 < 1 < 2 < 3 < \ldots$.
2. Затем будем сравнивать составные объекты поэлементно, просматривая их с начала:
   - если текущие элементы не равны, меньше тот составной объект, в котором текущий элемент меньше;
   - если один из составных объектов кончился, а другой ещё нет, меньше тот, который кончился;
   - если оба составных объекта кончились одновременно, они равны.

Примеры:

$$(1, 2, 3) > (1)$$

Лексикографический порядок

Постановка задачи: нужно упорядочить составные объекты — например, строки или последовательности.

1. Сначала зададим порядок на элементах:
   - для символов: ‘a’ $<$ ‘b’ $<$ … $<$ ‘z’,
   - для чисел: $\ldots < -2 < -1 < 0 < 1 < 2 < 3 < \ldots$.
2. Затем будем сравнивать составные объекты поэлементно, просматривая их с начала:
   - если текущие элементы не равны, меньше тот составной объект, в котором текущий элемент меньше;
   - если один из составных объектов кончился, а другой ещё нет, меньше тот, который кончился;
   - если оба составных объекта кончились одновременно, они равны.

Примеры:

$$(1,2) \sim (1,2)$$

Лексикографический порядок

Постановка задачи: нужно упорядочить составные объекты — например, строки или последовательности.

1. Сначала зададим порядок на элементах:
   - для символов: ‘a’ $<$ ‘b’ $< \ldots <$ ‘z’,
   - для чисел: $\ldots < -2 < -1 < 0 < 1 < 2 < 3 < \ldots$.
2. Затем будем сравнивать составные объекты поэлементно, просматривая их с начала:
   - если текущие элементы не равны, меньше тот составной объект, в котором текущий элемент меньше;
   - если один из составных объектов кончился, а другой ещё нет, меньше тот, который кончился;
   - если оба составных объекта кончились одновременно, они равны.

Примеры:

$$(1, 2) = (1, 2)$$

Лексикографический порядок

Постановка задачи: нужно упорядочить составные объекты — например, строки или последовательности.

1. Сначала зададим порядок на элементах:
   - для символов: $\text{‘a’} < \text{‘b’} < \ldots < \text{‘z’}$,
   - для чисел: $\ldots < -2 < -1 < 0 < 1 < 2 < 3 < \ldots$.
2. Затем будем сравнивать составные объекты поэлементно, просматривая их с начала:
   - если текущие элементы не равны, меньше тот составной объект, в котором текущий элемент меньше;
   - если один из составных объектов кончился, а другой ещё нет, меньше тот, который кончился;
   - если оба составных объекта кончились одновременно, они равны.

Примеры:

$$(1, 2) \sim (1, 2, 4)$$

Лексикографический порядок

Постановка задачи: нужно упорядочить составные объекты — например, строки или последовательности.

1. Сначала зададим порядок на элементах:
   - для символов: ‘a’ $<$ ‘b’ $< \ldots <$ ‘z’,
   - для чисел: $\ldots < -2 < -1 < 0 < 1 < 2 < 3 < \ldots$.
2. Затем будем сравнивать составные объекты поэлементно, просматривая их с начала:
   - если текущие элементы не равны, меньше тот составной объект, в котором текущий элемент меньше;
   - если один из составных объектов кончился, а другой ещё нет, меньше тот, который кончился;
   - если оба составных объекта кончились одновременно, они равны.

Примеры:

$$(1, 2) < (1, 2, 4)$$

Лексикографический порядок

Постановка задачи: нужно упорядочить составные объекты — например, строки или последовательности.

1. Сначала зададим порядок на элементах:
   - для символов: ‘a’ $<$ ‘b’ $< \ldots <$ ‘z’,
   - для чисел: $\ldots < -2 < -1 < 0 < 1 < 2 < 3 < \ldots$.
2. Затем будем сравнивать составные объекты поэлементно, просматривая их с начала:
   - если текущие элементы не равны, меньше тот составной объект, в котором текущий элемент меньше;
   - если один из составных объектов кончился, а другой ещё нет, меньше тот, который кончился;
   - если оба составных объекта кончились одновременно, они равны.

Примеры:

$$(1, 3, -1, 3) \sim (1, 3, 4, 6)$$

Лексикографический порядок

Постановка задачи: нужно упорядочить составные объекты — например, строки или последовательности.

1. Сначала зададим порядок на элементах:
   - для символов: ‘a’ < ‘b’ < ... < ‘z’,
   - для чисел: $\ldots < -2 < -1 < 0 < 1 < 2 < 3 < \ldots$.
2. Затем будем сравнивать составные объекты поэлементно, просматривая их с начала:
   - если текущие элементы не равны, меньше тот составной объект, в котором текущий элемент меньше;
   - если один из составных объектов кончился, а другой ещё нет, меньше тот, который кончился;
   - если оба составных объекта кончились одновременно, они равны.

Примеры:

$$(1, 3, -1, 3) < (1, 3, 4, 6)$$

Сравнение сложных объектов

- Эти правила можно применять рекурсивно: элементами последовательности могут быть не только числа или символы, но и строки, последовательности или другие сложные объекты.
- Если мы умеем сравнивать элементы последовательности, не составляет труда сравнить и сами последовательности лексикографически.

Сравнение сложных объектов

- Эти правила можно применять рекурсивно: элементами последовательности могут быть не только числа или символы, но и строки, последовательности или другие сложные объекты.
- Если мы умеем сравнивать элементы последовательности, не составляет труда сравнить и сами последовательности лексикографически.

Сравнение сложных объектов

- Эти правила можно применять рекурсивно: элементами последовательности могут быть не только числа или символы, но и строки, последовательности или другие сложные объекты.
- Если мы умеем сравнивать элементы последовательности, не составляет труда сравнить и сами последовательности лексикографически.

Примеры:

$$(«\mathrm{ab}», 2, 1.6, «\mathrm{aaa}») \sim («\mathrm{ab}», 2, 3.2, «\mathrm{aa}»)$$

Сравнение сложных объектов

- Эти правила можно применять рекурсивно: элементами последовательности могут быть не только числа или символы, но и строки, последовательности или другие сложные объекты.
- Если мы умеем сравнивать элементы последовательности, не составляет труда сравнить и сами последовательности лексикографически.

Примеры:

$$(\text{«ab»}, 2, 1.6, \text{«aaa»}) < (\text{«ab»}, 2, 3.2, \text{«aa»})$$

Сравнение сложных объектов

- Эти правила можно применять рекурсивно: элементами последовательности могут быть не только числа или символы, но и строки, последовательности или другие сложные объекты.
- Если мы умеем сравнивать элементы последовательности, не составляет труда сравнить и сами последовательности лексикографически.

Примеры:

$$((0,0),(0,1),(1,1),(1,0)) \sim ((0,0),(0,1),(1,0),(1,1))$$

Сравнение сложных объектов

- Эти правила можно применять рекурсивно: элементами последовательности могут быть не только числа или символы, но и строки, последовательности или другие сложные объекты.
- Если мы умеем сравнивать элементы последовательности, не составляет труда сравнить и сами последовательности лексикографически.

Примеры:

$$((0,0),(0,1),(1,1),(1,0)) \sim ((0,0),(0,1),(1,0),(1,1))$$

$$(0,0) = (0,0)$$

Сравнение сложных объектов

- Эти правила можно применять рекурсивно: элементами последовательности могут быть не только числа или символы, но и строки, последовательности или другие сложные объекты.
- Если мы умеем сравнивать элементы последовательности, не составляет труда сравнить и сами последовательности лексикографически.

Примеры:

$$((0,0),(0,1),(1,1),(1,0)) \sim ((0,0),(0,1),(1,0),(1,1))$$

$$(0,1) = (0,1)$$

Сравнение сложных объектов

- Эти правила можно применять рекурсивно: элементами последовательности могут быть не только числа или символы, но и строки, последовательности или другие сложные объекты.
- Если мы умеем сравнивать элементы последовательности, не составляет труда сравнить и сами последовательности лексикографически.

Примеры:

$$((0,0),(0,1),(1,1),(1,0)) > ((0,0),(0,1),(1,0),(1,1))$$

$$(1,1) > (1,0)$$

Другие типы сложных объектов

Стандартные формы представления и записи некоторых сложных объектов, удобные для упорядочивания:

- Упорядоченные пары, тройки, ... — по сути просто последовательности с фиксированным количеством элементов.
- Множества — сначала элементы упорядочиваются, а затем записываются в полученном порядке как последовательность.
- Обыкновенные графы из n вершин — представляются как множества рёбер, а рёбра представляются как множества из двух вершин.
- Матрицы одинакового размера — представляются как последовательность строк, а каждая строка — как последовательность элементов.
- Обыкновенные графы из n вершин — можно также представить как матрицы смежности.

Другие типы сложных объектов

Стандартные формы представления и записи некоторых сложных объектов, удобные для упорядочивания:

- Упорядоченные пары, тройки, ... — по сути просто последовательности с фиксированным количеством элементов.
- Множества — сначала элементы упорядочиваются, а затем записываются в полученном порядке как последовательность.
- Обыкновенные графы из n вершин — представляются как множества рёбер, а рёбра представляются как множества из двух вершин.
- Матрицы одинакового размера — представляются как последовательность строк, а каждая строка — как последовательность элементов.
- Обыкновенные графы из n вершин — можно также представить как матрицы смежности.

Другие типы сложных объектов

Стандартные формы представления и записи некоторых сложных объектов, удобные для упорядочивания:

- Упорядоченные пары, тройки, ... — по сути просто последовательности с фиксированным количеством элементов.
- Множества — сначала элементы упорядочиваются, а затем записываются в полученном порядке как последовательность.
- Обыкновенные графы из n вершин — представляются как множества рёбер, а рёбра представляются как множества из двух вершин.
- Матрицы одинакового размера — представляются как последовательность строк, а каждая строка — как последовательность элементов.
- Обыкновенные графы из n вершин — можно также представить как матрицы смежности.

Другие типы сложных объектов

Стандартные формы представления и записи некоторых сложных объектов, удобные для упорядочивания:

- Упорядоченные пары, тройки, ... — по сути просто последовательности с фиксированным количеством элементов.
- Множества — сначала элементы упорядочиваются, а затем записываются в полученном порядке как последовательность.
- Обыкновенные графы из n вершин — представляются как множества рёбер, а рёбра представляются как множества из двух вершин.
- Матрицы одинакового размера — представляются как последовательность строк, а каждая строка — как последовательность элементов.
- Обыкновенные графы из n вершин — можно также представить как матрицы смежности.

Другие типы сложных объектов

Стандартные формы представления и записи некоторых сложных объектов, удобные для упорядочивания:

- Упорядоченные пары, тройки, … — по сути просто последовательности с фиксированным количеством элементов.
- Множества — сначала элементы упорядочиваются, а затем записываются в полученном порядке как последовательность.
- Обыкновенные графы из n вершин — представляются как множества рёбер, а рёбра представляются как множества из двух вершин.
- Матрицы одинакового размера — представляются как последовательность строк, а каждая строка — как последовательность элементов.
- Обыкновенные графы из n вершин — можно также представить как матрицы смежности.

Другие типы сложных объектов

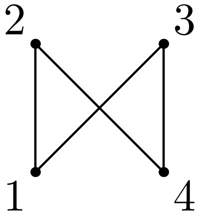
Стандартные формы представления и записи некоторых сложных объектов, удобные для упорядочивания:

- Упорядоченные пары, тройки, ... — по сути просто последовательности с фиксированным количеством элементов.
- Множества — сначала элементы упорядочиваются, а затем записываются в полученном порядке как последовательность.
- Обыкновенные графы из n вершин — представляются как множества рёбер, а рёбра представляются как множества из двух вершин.
- Матрицы одинакового размера — представляются как последовательность строк, а каждая строка — как последовательность элементов.
- Обыкновенные графы из n вершин — можно также представить как матрицы смежности.

Другие типы сложных объектов

Примеры:

- Множества: $\{1, 5, 3\} \longrightarrow (1, 3, 5)$
- Обыкновенные графы из n вершин:



$$\longrightarrow ((1,2),(1,3),(2,4),(3,4))$$

$$\longrightarrow \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

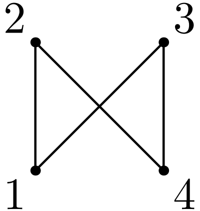
- Матрицы одинакового размера:

$$\begin{pmatrix} \cos\varphi & \sin\varphi \\ -\sin\varphi & \cos\varphi \end{pmatrix} \longrightarrow ((\cos\varphi, \sin\varphi), (-\sin\varphi, \cos\varphi))$$

Другие типы сложных объектов

Примеры:

- Множества: $\{1,5,3\} \longrightarrow (1,3,5)$
- Обыкновенные графы из n вершин:



$\longrightarrow \quad ((1,2),(1,3),(2,4),(3,4))$

$$\longrightarrow \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

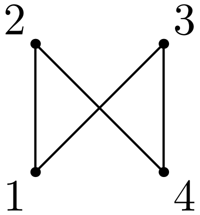
- Матрицы одинакового размера:

$$\begin{pmatrix} \cos\varphi & \sin\varphi \\ -\sin\varphi & \cos\varphi \end{pmatrix} \longrightarrow ((\cos\varphi, \sin\varphi), (-\sin\varphi, \cos\varphi))$$

Другие типы сложных объектов

Примеры:

- Множества: $\{1, 5, 3\} \longrightarrow (1, 3, 5)$
- Обыкновенные графы из n вершин:



$$\longrightarrow ((1,2),(1,3),(2,4),(3,4))$$

$$\longrightarrow \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

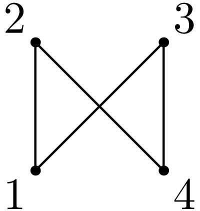
- Матрицы одинакового размера:

$$\begin{pmatrix} \cos\varphi & \sin\varphi \\ -\sin\varphi & \cos\varphi \end{pmatrix} \longrightarrow ((\cos\varphi, \sin\varphi), (-\sin\varphi, \cos\varphi))$$

Другие типы сложных объектов

Примеры:

- Множества: $\{1, 5, 3\} \longrightarrow (1, 3, 5)$
- Обыкновенные графы из n вершин:



$$\longrightarrow ((1,2),(1,3),(2,4),(3,4))$$

$$\longrightarrow \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

- Матрицы одинакового размера:

$$\begin{pmatrix} \cos\varphi & \sin\varphi \\ -\sin\varphi & \cos\varphi \end{pmatrix} \longrightarrow ((\cos\varphi, \sin\varphi), (-\sin\varphi, \cos\varphi))$$

Преимущества лексикографического порядка

- Сравнение можно производить до первого различия; всё, что после первого различия — даже если объекты бесконечные — значения не имеет.
- Объекты с одинаковыми префиксами находятся рядом.
- При упорядочивании можно свести задачу к подзадачам меньшего размера, строя при этом сложные объекты по частям из простых.
- Например, если строки складываются в хеш-таблицу и упорядочиваются в ней по хеш-коду — этих преимуществ нет.
- Обычно, чтобы удобно пронумеровать объекты, можно придумать какую-то их запись, по которой затем упорядочить их лексикографически.

Преимущества лексикографического порядка

- Сравнение можно производить до первого различия; всё, что после первого различия — даже если объекты бесконечные — значения не имеет.
- Объекты с одинаковыми префиксами находятся рядом.
- При упорядочивании можно свести задачу к подзадачам меньшего размера, строя при этом сложные объекты по частям из простых.
- Например, если строки складываются в хеш-таблицу и упорядочиваются в ней по хеш-коду — этих преимуществ нет.
- Обычно, чтобы удобно пронумеровать объекты, можно придумать какую-то их запись, по которой затем упорядочить их лексикографически.

Преимущества лексикографического порядка

- Сравнение можно производить до первого различия; всё, что после первого различия — даже если объекты бесконечные — значения не имеет.
- Объекты с одинаковыми префиксами находятся рядом.
- При упорядочивании можно свести задачу к подзадачам меньшего размера, строя при этом сложные объекты по частям из простых.
- Например, если строки складываются в хеш-таблицу и упорядочиваются в ней по хеш-коду — этих преимуществ нет.
- Обычно, чтобы удобно пронумеровать объекты, можно придумать какую-то их запись, по которой затем упорядочить их лексикографически.

Преимущества лексикографического порядка

- Сравнение можно производить до первого различия; всё, что после первого различия — даже если объекты бесконечные — значения не имеет.
- Объекты с одинаковыми префиксами находятся рядом.
- При упорядочивании можно свести задачу к подзадачам меньшего размера, строя при этом сложные объекты по частям из простых.
- Например, если строки складываются в хеш-таблицу и упорядочиваются в ней по хеш-коду — этих преимуществ нет.
- Обычно, чтобы удобно пронумеровать объекты, можно придумать какую-то их запись, по которой затем упорядочить их лексикографически.

Преимущества лексикографического порядка

- Сравнение можно производить до первого различия; всё, что после первого различия — даже если объекты бесконечные — значения не имеет.
- Объекты с одинаковыми префиксами находятся рядом.
- При упорядочивании можно свести задачу к подзадачам меньшего размера, строя при этом сложные объекты по частям из простых.
- Например, если строки складываются в хеш-таблицу и упорядочиваются в ней по хеш-коду — этих преимуществ нет.
- Обычно, чтобы удобно пронумеровать объекты, можно придумать какую-то их запись, по которой затем упорядочить их лексикографически.

Оглавление

Строки Фибоначчи: формулировки

- Определение: строка Фибоначчи из n символов — это строка из нулей и единиц такая, что в ней нет двух единиц подряд.
- Постановка задачи: по заданным числам n и k требуется найти k-ю в лексикографическом порядке строку Фибоначчи из n символов.
- Обратная задача: по заданному числу n и строке Фибоначчи из n символов требуется найти лексикографический номер этой строки среди всех строк Фибоначчи из n символов.
- Мотивация: часто удобно работать со сложными объектами в целом, оперируя лишь их номерами. Поэтому нужно уметь получать объект по номеру и номер по объекту.

Строки Фибоначчи: формулировки

- Определение: строка Фибоначчи из n символов — это строка из нулей и единиц такая, что в ней нет двух единиц подряд.
- Постановка задачи: по заданным числам n и k требуется найти k-ю в лексикографическом порядке строку Фибоначчи из n символов.
- Обратная задача: по заданному числу n и строке Фибоначчи из n символов требуется найти лексикографический номер этой строки среди всех строк Фибоначчи из n символов.
- Мотивация: часто удобно работать со сложными объектами в целом, оперируя лишь их номерами. Поэтому нужно уметь получать объект по номеру и номер по объекту.

Строки Фибоначчи: формулировки

- Определение: строка Фибоначчи из n символов — это строка из нулей и единиц такая, что в ней нет двух единиц подряд.
- Постановка задачи: по заданным числам n и k требуется найти k-ю в лексикографическом порядке строку Фибоначчи из n символов.
- Обратная задача: по заданному числу n и строке Фибоначчи из n символов требуется найти лексикографический номер этой строки среди всех строк Фибоначчи из n символов.
- Мотивация: часто удобно работать со сложными объектами в целом, оперируя лишь их номерами. Поэтому нужно уметь получать объект по номеру и номер по объекту.

Строки Фибоначчи: формулировки

- Определение: строка Фибоначчи из n символов — это строка из нулей и единиц такая, что в ней нет двух единиц подряд.
- Постановка задачи: по заданным числам n и k требуется найти k-ю в лексикографическом порядке строку Фибоначчи из n символов.
- Обратная задача: по заданному числу n и строке Фибоначчи из n символов требуется найти лексикографический номер этой строки среди всех строк Фибоначчи из n символов.
- Мотивация: часто удобно работать со сложными объектами в целом, оперируя лишь их номерами. Поэтому нужно уметь получать объект по номеру и номер по объекту.

Строки Фибоначчи: тривиальное решение

Тривиальное решение:

1. Выпишем все 2^n строк из нулей и единиц длины n.
2. Вычеркнем все строки, содержащие две единицы подряд.
3. Отсортируем полученную последовательность строк лексикографически.
4. Для решения прямой задачи будем просто выводить строку с заданным номером.
5. Для решения обратной задачи будем искать заданную строку двоичным поиском.

Конечно, хорошо бы использовать меньше времени на подготовку и меньше памяти.

Строки Фибоначчи: тривиальное решение

Тривиальное решение:

1. Выпишем все 2^n строк из нулей и единиц длины n.
2. Вычеркнем все строки, содержащие две единицы подряд.
3. Отсортируем полученную последовательность строк лексикографически.
4. Для решения прямой задачи будем просто выводить строку с заданным номером.
5. Для решения обратной задачи будем искать заданную строку двоичным поиском.

Конечно, хорошо бы использовать меньше времени на подготовку и меньше памяти.

Строки Фибоначчи: тривиальное решение

Тривиальное решение:

1. Выпишем все 2^n строк из нулей и единиц длины n.
2. Вычеркнем все строки, содержащие две единицы подряд.
3. Отсортируем полученную последовательность строк лексикографически.
4. Для решения прямой задачи будем просто выводить строку с заданным номером.
5. Для решения обратной задачи будем искать заданную строку двоичным поиском.

Конечно, хорошо бы использовать меньше времени на подготовку и меньше памяти.

Строки Фибоначчи: тривиальное решение

Тривиальное решение:

1. Выпишем все 2^n строк из нулей и единиц длины n.
2. Вычеркнем все строки, содержащие две единицы подряд.
3. Отсортируем полученную последовательность строк лексикографически.
4. Для решения прямой задачи будем просто выводить строку с заданным номером.
5. Для решения обратной задачи будем искать заданную строку двоичным поиском.

Конечно, хорошо бы использовать меньше времени на подготовку и меньше памяти.

Строки Фибоначчи: тривиальное решение

Тривиальное решение:

1. Выпишем все 2^n строк из нулей и единиц длины n.
2. Вычеркнем все строки, содержащие две единицы подряд.
3. Отсортируем полученную последовательность строк лексикографически.
4. Для решения прямой задачи будем просто выводить строку с заданным номером.
5. Для решения обратной задачи будем искать заданную строку двоичным поиском.

Конечно, хорошо бы использовать меньше времени на подготовку и меньше памяти.

Строки Фибоначчи: тривиальное решение

Тривиальное решение:

1. Выпишем все 2^n строк из нулей и единиц длины n.
2. Вычеркнем все строки, содержащие две единицы подряд.
3. Отсортируем полученную последовательность строк лексикографически.
4. Для решения прямой задачи будем просто выводить строку с заданным номером.
5. Для решения обратной задачи будем искать заданную строку двоичным поиском.

Конечно, хорошо бы использовать меньше времени на подготовку и меньше памяти.

Строки Фибоначчи: на примере

Рассмотрим строки Фибоначчи длины $n = 4$ (нумерация с нуля):

0	0000
1	0001
2	0010
3	0100
4	0101
5	1000
6	1001
7	1010

Строки Фибоначчи: на примере

Рассмотрим строки Фибоначчи длины $n = 4$ (нумерация с нуля):

0	$0\|000$
1	$0\|001$
2	$0\|010$
3	$0\|100$
4	$0\|101$
5	$10\|00$
6	$10\|01$
7	$10\|10$

Посмотрим на первый символ.

- Если это 0, дальше может быть любая строка Фибоначчи длины $n - 1$.
- Если же это 1, за ним обязательно следует 0, а дальше может быть любая строка Фибоначчи длины $n - 2$.

Строки Фибоначчи: на примере

Рассмотрим строки Фибоначчи длины $n = 4$ (нумерация с нуля):

0	0|000
1	0|001
2	0|010
3	0|100
4	0|101
5	10|00
6	10|01
7	10|10

Посмотрим на первый символ.

- Если это 0, дальше может быть любая строка Фибоначчи длины $n - 1$.
- Если же это 1, за ним обязательно следует 0, а дальше может быть любая строка Фибоначчи длины $n - 2$.

Строки Фибоначчи: на примере

Рассмотрим строки Фибоначчи длины $n = 4$ (нумерация с нуля):

0	0|000
1	0|001
2	0|010
3	0|100
4	0|101
5	10|00
6	10|01
7	10|10

Посмотрим на первый символ.

- Если это 0, дальше может быть любая строка Фибоначчи длины $n-1$.
- Если же это 1, за ним обязательно следует 0, а дальше может быть любая строка Фибоначчи длины $n-2$.

Строки Фибоначчи: на примере

Рассмотрим строки Фибоначчи длины $n = 4$ (нумерация с нуля):

0	0|000	$(n = 4, k = 0)$	$\longrightarrow$	«0»	$+$	$(n = 3, k = 0)$
1	0|001	$(n = 4, k = 1)$	$\longrightarrow$	«0»	$+$	$(n = 3, k = 1)$
2	0|010	$(n = 4, k = 2)$	$\longrightarrow$	«0»	$+$	$(n = 3, k = 2)$
3	0|100	$(n = 4, k = 3)$	$\longrightarrow$	«0»	$+$	$(n = 3, k = 3)$
4	0|101	$(n = 4, k = 4)$	$\longrightarrow$	«0»	$+$	$(n = 3, k = 4)$
5	10|00	$(n = 4, k = 5)$	$\longrightarrow$	«10»	$+$	$(n = 2, k = 0)$
6	10|01	$(n = 4, k = 6)$	$\longrightarrow$	«10»	$+$	$(n = 2, k = 1)$
7	10|10	$(n = 4, k = 7)$	$\longrightarrow$	«10»	$+$	$(n = 2, k = 2)$

Посмотрим на первый символ.

- Если это 0, дальше может быть любая строка Фибоначчи длины $n - 1$.
- Если же это 1, за ним обязательно следует 0, а дальше может быть любая строка Фибоначчи длины $n - 2$.

Строки Фибоначчи: на примере

Рассмотрим строки Фибоначчи длины $n = 4$ (нумерация с нуля):

0	0|000	$(n = 4, k = 0)$	$\longrightarrow$	«0»	+	$(n = 3, k = 0)$
1	0|001	$(n = 4, k = 1)$	$\longrightarrow$	«0»	+	$(n = 3, k = 1)$
2	0|010	$(n = 4, k = 2)$	$\longrightarrow$	«0»	+	$(n = 3, k = 2)$
3	0|100	$(n = 4, k = 3)$	$\longrightarrow$	«0»	+	$(n = 3, k = 3)$
4	0|101	$(n = 4, k = 4)$	$\longrightarrow$	«0»	+	$(n = 3, k = 4)$
5	10|00	$(n = 4, k = 5)$	$\longrightarrow$	«10»	+	$(n = 2, k = 0)$
6	10|01	$(n = 4, k = 6)$	$\longrightarrow$	«10»	+	$(n = 2, k = 1)$
7	10|10	$(n = 4, k = 7)$	$\longrightarrow$	«10»	+	$(n = 2, k = 2)$

Заметим, что сначала в лексикографическом порядке идут все строки Фибоначчи, начинающиеся на 0, а затем — все, начинающиеся на 1.

При этом в обеих половинах оставшиеся части строк также отсортированы лексикографически.

Таким образом, мы свели задачу к подзадаче меньшего размера ($n - 1$ или $n - 2$).

Строки Фибоначчи: на примере

Рассмотрим строки Фибоначчи длины $n = 4$ (нумерация с нуля):

0	0|000	$(n = 4, k = 0)$	$\longrightarrow$	«0»	$+$	$(n = 3, k = 0)$
1	0|001	$(n = 4, k = 1)$	$\longrightarrow$	«0»	$+$	$(n = 3, k = 1)$
2	0|010	$(n = 4, k = 2)$	$\longrightarrow$	«0»	$+$	$(n = 3, k = 2)$
3	0|100	$(n = 4, k = 3)$	$\longrightarrow$	«0»	$+$	$(n = 3, k = 3)$
4	0|101	$(n = 4, k = 4)$	$\longrightarrow$	«0»	$+$	$(n = 3, k = 4)$
5	10|00	$(n = 4, k = 5)$	$\longrightarrow$	«10»	$+$	$(n = 2, k = 0)$
6	10|01	$(n = 4, k = 6)$	$\longrightarrow$	«10»	$+$	$(n = 2, k = 1)$
7	10|10	$(n = 4, k = 7)$	$\longrightarrow$	«10»	$+$	$(n = 2, k = 2)$

Каким будет k в этой подзадаче?

- Если строка начинается на 0, то номер k не поменялся.
- Если же строка начинается на 1, то номер k уменьшился на количество строк, начинающихся на 0.

Строки Фибоначчи: на примере

Рассмотрим строки Фибоначчи длины $n = 4$ (нумерация с нуля):

0	0|000	$(n = 4, k = 0)$	$\longrightarrow$	«0»	$+$	$(n = 3, k = 0)$
1	0|001	$(n = 4, k = 1)$	$\longrightarrow$	«0»	$+$	$(n = 3, k = 1)$
2	0|010	$(n = 4, k = 2)$	$\longrightarrow$	«0»	$+$	$(n = 3, k = 2)$
3	0|100	$(n = 4, k = 3)$	$\longrightarrow$	«0»	$+$	$(n = 3, k = 3)$
4	0|101	$(n = 4, k = 4)$	$\longrightarrow$	«0»	$+$	$(n = 3, k = 4)$
5	10|00	$(n = 4, k = 5)$	$\longrightarrow$	«10»	$+$	$(n = 2, k = 0)$
6	10|01	$(n = 4, k = 6)$	$\longrightarrow$	«10»	$+$	$(n = 2, k = 1)$
7	10|10	$(n = 4, k = 7)$	$\longrightarrow$	«10»	$+$	$(n = 2, k = 2)$

Каким будет k в этой подзадаче?

- Если строка начинается на 0, то номер k не поменялся.
- Если же строка начинается на 1, то номер k уменьшился на количество строк, начинающихся на 0.

Строки Фибоначчи: решение

- Сначала посчитаем количество строк Фибоначчи длины 0, 1, 2, $\ldots$, n. Это просто число Фибоначчи: $f_0 = 1$, $f_1 = 2$, $f_n = f_{n-1} + f_{n-2}$. Среди строк длины n есть f_{n-1}, начинающихся на 0, и f_{n-2}, начинающихся на 1.
- Объект по номеру: $(n, k) \to s$. Если $k < f_{n-1}$, запишем в строку 0 и решим подзадачу $(n-1, k)$. Если же $k \geqslant f_{n-1}$, запишем в строку 10 и решим подзадачу $(n-2, k - f_{n-1})$.
- Номер по объекту: $(n, s) \to k$. Если строка начинается на 0, решим подзадачу для $n-1$ и строки s без первого символа. Если же строка начинается на 1, прибавим к номеру f_{n-1} и решим подзадачу для $n-2$ и строки s без первых двух символов.
- Случаи $n = 0$ и $n = 1$ в обеих задачах являются базой и рассматриваются отдельно.

Строки Фибоначчи: решение

- Сначала посчитаем количество строк Фибоначчи длины 0, 1, 2, …, n. Это просто число Фибоначчи: $f_0 = 1$, $f_1 = 2$, $f_n = f_{n-1} + f_{n-2}$. Среди строк длины n есть f_{n-1}, начинающихся на 0, и f_{n-2}, начинающихся на 1.
- Объект по номеру: $(n, k) \to s$. Если $k < f_{n-1}$, запишем в строку 0 и решим подзадачу $(n-1, k)$. Если же $k \geqslant f_{n-1}$, запишем в строку 10 и решим подзадачу $(n-2, k - f_{n-1})$.
- Номер по объекту: $(n, s) \to k$. Если строка начинается на 0, решим подзадачу для $n-1$ и строки s без первого символа. Если же строка начинается на 1, прибавим к номеру f_{n-1} и решим подзадачу для $n-2$ и строки s без первых двух символов.
- Случаи $n = 0$ и $n = 1$ в обеих задачах являются базой и рассматриваются отдельно.

Строки Фибоначчи: решение

- Сначала посчитаем количество строк Фибоначчи длины 0, 1, 2, ..., n. Это просто число Фибоначчи: $f_0 = 1$, $f_1 = 2$, $f_n = f_{n-1} + f_{n-2}$. Среди строк длины n есть f_{n-1}, начинающихся на 0, и f_{n-2}, начинающихся на 1.
- Объект по номеру: $(n, k) \to s$. Если $k < f_{n-1}$, запишем в строку 0 и решим подзадачу $(n-1, k)$. Если же $k \geqslant f_{n-1}$, запишем в строку 10 и решим подзадачу $(n-2, k - f_{n-1})$.
- Номер по объекту: $(n, s) \to k$. Если строка начинается на 0, решим подзадачу для $n-1$ и строки s без первого символа. Если же строка начинается на 1, прибавим к номеру f_{n-1} и решим подзадачу для $n-2$ и строки s без первых двух символов.
- Случаи $n = 0$ и $n = 1$ в обеих задачах являются базой и рассматриваются отдельно.

Строки Фибоначчи: решение

- Сначала посчитаем количество строк Фибоначчи длины 0, 1, 2, ..., n. Это просто число Фибоначчи: $f_0 = 1$, $f_1 = 2$, $f_n = f_{n-1} + f_{n-2}$. Среди строк длины n есть f_{n-1}, начинающихся на 0, и f_{n-2}, начинающихся на 1.
- Объект по номеру: $(n, k) \to s$. Если $k < f_{n-1}$, запишем в строку 0 и решим подзадачу $(n-1, k)$. Если же $k \geqslant f_{n-1}$, запишем в строку 10 и решим подзадачу $(n-2, k-f_{n-1})$.
- Номер по объекту: $(n, s) \to k$. Если строка начинается на 0, решим подзадачу для $n-1$ и строки s без первого символа. Если же строка начинается на 1, прибавим к номеру f_{n-1} и решим подзадачу для $n-2$ и строки s без первых двух символов.
- Случаи $n = 0$ и $n = 1$ в обеих задачах являются базой и рассматриваются отдельно.

Строки Фибоначчи: пример решения

Числа Фибоначчи:

n	0	1	2	3	4	5	6
f_n	1	2	3	5	8	13	21

Объект по номеру:

n	k	s
6	15	ε
4	2	10
3	2	100
2	2	1000
0	0	100010

Номер по объекту:

n	s	k
6	100010	0
4	0010	13
3	010	13
2	10	13
0	ε	15

Строки Фибоначчи: пример решения

Числа Фибоначчи:

n	0	1	2	3	4	5	6
f_n	1	2	3	5	8	13	21

Объект по номеру:

n	k	s
6	15	ε
4	2	10
3	2	100
2	2	1000
0	0	100010

Номер по объекту:

n	s	k
6	100010	0
4	0010	13
3	010	13
2	10	13
0	ε	15

Строки Фибоначчи: пример решения

Числа Фибоначчи:

n	0	1	2	3	4	5	6
f_n	1	2	3	5	8	13	21

Объект по номеру:

n	k	s
6	15	ε
4	2	10
3	2	100
2	2	1000
0	0	100010

Номер по объекту:

n	s	k
6	100010	0
4	0010	13
3	010	13
2	10	13
0	ε	15

Оглавление

Условные обозначения

$$\begin{array}{rcll} F_0 & = & (\varepsilon) & \\ F_1 & = & (0, 1) & \\ F_n & = & 0 & F_{n-1} \\ & & 10 & F_{n-2} \end{array}$$

- Каждый объект (строку Фибоначчи) запишем как строку.
- Все объекты с фиксированными параметрами (строки Фибоначчи фиксированной длины) выпишем в одну лексикографически отсортированную последовательность.
- Выражение <строка> <последовательность> означает новую последовательность, в которой к каждому элементу старой последовательности спереди приписана строка.
- Запись последовательностей друг под другом в правой части равенства означает конкатенацию последовательностей.

Условные обозначения

$$
\begin{array}{rcl}
F_0 & = & (\varepsilon) \\
F_1 & = & (0, 1) \\
F_n & = & 0 \quad F_{n-1} \\
 & & 10 \; F_{n-2}
\end{array}
$$

- Каждый объект (строку Фибоначчи) запишем как строку.
- Все объекты с фиксированными параметрами (строки Фибоначчи фиксированной длины) выпишем в одну лексикографически отсортированную последовательность.
- Выражение <строка> <последовательность> означает новую последовательность, в которой к каждому элементу старой последовательности спереди приписана строка.
- Запись последовательностей друг под другом в правой части равенства означает конкатенацию последовательностей.

Условные обозначения

$$
\begin{array}{rcl}
F_0 & = & (\varepsilon) \\
F_1 & = & (0, 1) \\
F_n & = & 0 \quad F_{n-1} \\
 & & 10 \; F_{n-2}
\end{array}
$$

- Каждый объект (строку Фибоначчи) запишем как строку.
- Все объекты с фиксированными параметрами (строки Фибоначчи фиксированной длины) выпишем в одну лексикографически отсортированную последовательность.
- Выражение <строка> <последовательность> означает новую последовательность, в которой к каждому элементу старой последовательности спереди приписана строка.
- Запись последовательностей друг под другом в правой части равенства означает конкатенацию последовательностей.

Условные обозначения

$$
\begin{array}{lcl}
F_0 & = & (\varepsilon) \\
F_1 & = & (0, 1) \\
F_n & = & 0 \quad F_{n-1} \\
 & & 10 \; F_{n-2}
\end{array}
$$

- Каждый объект (строку Фибоначчи) запишем как строку.
- Все объекты с фиксированными параметрами (строки Фибоначчи фиксированной длины) выпишем в одну лексикографически отсортированную последовательность.
- Выражение <строка> <последовательность> означает новую последовательность, в которой к каждому элементу старой последовательности спереди приписана строка.
- Запись последовательностей друг под другом в правой части равенства означает конкатенацию последовательностей.

Условные обозначения

$$
\begin{array}{lcl}
F_0 & = & (\varepsilon) \\
F_1 & = & (0, 1) \\
F_n & = & 0 \;\; F_{n-1} \\
 & & 10 \;\; F_{n-2}
\end{array}
$$

- Каждый объект (строку Фибоначчи) запишем как строку.
- Все объекты с фиксированными параметрами (строки Фибоначчи фиксированной длины) выпишем в одну лексикографически отсортированную последовательность.
- Выражение <строка> <последовательность> означает новую последовательность, в которой к каждому элементу старой последовательности спереди приписана строка.
- Запись последовательностей друг под другом в правой части равенства означает конкатенацию последовательностей.

Условные обозначения: пример

$$F_4 = \begin{array}{l} 0\ F_3 \\ 10\ F_2 \end{array} \qquad F_4 = \begin{array}{l} 0\ (000, 001, 010, 100, 101) \\ 10\ (00, 01, 10) \end{array}$$

$$F_4 = \begin{array}{l} 0\ (000, \\ \quad 001, \\ \quad 010, \\ \quad 100, \\ \quad 101) \\ 10\ (00, \\ \quad 01, \\ \quad 10) \end{array} \qquad F_4 = \begin{array}{l} (0000, \\ \ 0001, \\ \ 0010, \\ \ 0100, \\ \ 0101) \\ (1000, \\ \ 1001, \\ \ 1010) \end{array} \qquad F_4 = \begin{array}{l} (0000, \\ \ 0001, \\ \ 0010, \\ \ 0100, \\ \ 0101, \\ \ 1000, \\ \ 1001, \\ \ 1010) \end{array}$$

Условные обозначения: пример

$$F_4 = \begin{array}{l} 0\ F_3 \\ 10\ F_2 \end{array} \qquad F_4 = \begin{array}{l} 0\ (000, 001, 010, 100, 101) \\ 10\ (00, 01, 10) \end{array}$$

$$F_4 = \begin{array}{l} 0\ (000, 001, 010, 100, 101) \\ 10\ (00, 01, 10) \end{array} \qquad F_4 = \begin{array}{l} (0000, 0001, 0010, 0100, 0101) \\ (1000, 1001, 1010) \end{array} \qquad F_4 = (0000, 0001, 0010, 0100, 0101, 1000, 1001, 1010)$$

Условные обозначения: пример

$$F_4 = \begin{array}{ll} 0 & F_3 \\ 10 & F_2 \end{array} \qquad F_4 = \begin{array}{ll} 0 & (000, 001, 010, 100, 101) \\ 10 & (00, 01, 10) \end{array}$$

$$F_4 = \begin{array}{ll} 0 & (000, \\ & \ 001, \\ & \ 010, \\ & \ 100, \\ & \ 101) \\ 10 & (00, \\ & \ 01, \\ & \ 10) \end{array} \qquad F_4 = \begin{array}{l} (0000, \\ \ 0001, \\ \ 0010, \\ \ 0100, \\ \ 0101) \\ (1000, \\ \ 1001, \\ \ 1010) \end{array} \qquad F_4 = \begin{array}{l} (0000, \\ \ 0001, \\ \ 0010, \\ \ 0100, \\ \ 0101, \\ \ 1000, \\ \ 1001, \\ \ 1010) \end{array}$$

Условные обозначения: пример

$$F_4 = \begin{array}{ll} 0 & F_3 \\ 10 & F_2 \end{array} \qquad F_4 = \begin{array}{ll} 0 & (000, 001, 010, 100, 101) \\ 10 & (00, 01, 10) \end{array}$$

$$F_4 = \begin{array}{ll} 0 & (000, \\ & \ 001, \\ & \ 010, \\ & \ 100, \\ & \ 101) \\ 10 & (00, \\ & \ 01, \\ & \ 10) \end{array} \qquad F_4 = \begin{array}{l} (0000, \\ \ 0001, \\ \ 0010, \\ \ 0100, \\ \ 0101) \\ (1000, \\ \ 1001, \\ \ 1010) \end{array} \qquad F_4 = \begin{array}{l} (0000, \\ \ 0001, \\ \ 0010, \\ \ 0100, \\ \ 0101, \\ \ 1000, \\ \ 1001, \\ \ 1010) \end{array}$$

Условные обозначения: пример

$$F_4 = \begin{array}{ll} 0 & F_3 \\ 10 & F_2 \end{array} \qquad F_4 = \begin{array}{ll} 0 & (000, 001, 010, 100, 101) \\ 10 & (00, 01, 10) \end{array}$$

$$F_4 = \begin{array}{ll} 0 & (000, \\ & \ 001, \\ & \ 010, \\ & \ 100, \\ & \ 101) \\ 10 & (00, \\ & \ 01, \\ & \ 10) \end{array} \qquad F_4 = \begin{array}{l} (0000, \\ \ 0001, \\ \ 0010, \\ \ 0100, \\ \ 0101) \\ (1000, \\ \ 1001, \\ \ 1010) \end{array} \qquad F_4 = \begin{array}{l} (0000, \\ \ 0001, \\ \ 0010, \\ \ 0100, \\ \ 0101, \\ \ 1000, \\ \ 1001, \\ \ 1010) \end{array}$$

Условные обозначения: длина

Заметим, что из формулы для упорядоченных последовательностей легко получить формулу для их длин:

- заменим все последовательности на их длины,
- уберём приписываемые строки,
- конкатенацию последовательностей заменим на сложение их длин.

$$
\begin{array}{lll@{\qquad}lll@{\qquad}lll}
F_0 & = & (\varepsilon) & |F_0| & = & |(\varepsilon)| & f_0 & = & 1 \\
F_1 & = & (0,1) & |F_1| & = & |(0,1)| & f_1 & = & 2 \\
F_n & = & 0\ F_{n-1} & |F_n| & = & |F_{n-1}| & f_n & = & f_{n-1} \\
 & & 10\ F_{n-2} & & + & |F_{n-2}| & & + & f_{n-2}
\end{array}
$$

Условные обозначения: длина

Заметим, что из формулы для упорядоченных последовательностей легко получить формулу для их длин:

- заменим все последовательности на их длины,
- уберём приписываемые строки,
- конкатенацию последовательностей заменим на сложение их длин.

$$
\begin{array}{lll}
F_0 & = & (\varepsilon) \\
F_1 & = & (0, 1) \\
F_n & = & 0 \;\; F_{n-1} \\
 & & 10 \;\; F_{n-2}
\end{array}
\qquad
\begin{array}{lll}
|F_0| & = & |(\varepsilon)| \\
|F_1| & = & |(0, 1)| \\
|F_n| & = & |F_{n-1}| \\
 & + & |F_{n-2}|
\end{array}
\qquad
\begin{array}{lll}
f_0 & = & 1 \\
f_1 & = & 2 \\
f_n & = & f_{n-1} \\
 & + & f_{n-2}
\end{array}
$$

Условные обозначения: длина

Заметим, что из формулы для упорядоченных последовательностей легко получить формулу для их длин:

- заменим все последовательности на их длины,
- уберём приписываемые строки,
- конкатенацию последовательностей заменим на сложение их длин.

$$
\begin{array}{lllllllll}
F_0 & = & (\varepsilon) & |F_0| & = & |(\varepsilon)| & f_0 & = & 1 \\
F_1 & = & (0,1) & |F_1| & = & |(0,1)| & f_1 & = & 2 \\
F_n & = & 0 \; F_{n-1} & |F_n| & = & |F_{n-1}| & f_n & = & f_{n-1} \\
 & & 10 \; F_{n-2} & & + & |F_{n-2}| & & + & f_{n-2}
\end{array}
$$

Условные обозначения: длина

Заметим, что из формулы для упорядоченных последовательностей легко получить формулу для их длин:

- заменим все последовательности на их длины,
- уберём приписываемые строки,
- конкатенацию последовательностей заменим на сложение их длин.

$$
\begin{array}{lcl@{\qquad}lcl@{\qquad}lcl}
F_0 & = & (\varepsilon) & |F_0| & = & |(\varepsilon)| & f_0 & = & 1 \\
F_1 & = & (0,1) & |F_1| & = & |(0,1)| & f_1 & = & 2 \\
F_n & = & 0 \;\; F_{n-1} & |F_n| & = & |F_{n-1}| & f_n & = & f_{n-1} \\
 & & 10 \;\; F_{n-2} & & + & |F_{n-2}| & & + & f_{n-2}
\end{array}
$$

Оглавление

Подмножества 1: формулировка

- Рассмотрим все подмножества множества из n элементов.
- Пронумеруем элементы множества числами $0, 1, 2, \ldots, n-1$.
- Каждое подмножество будем записывать в виде строки из n символов — нулей и единиц. Ноль в позиции i означает, что i-й элемент отсутствует в подмножестве, единица — что присутствует.
- Объект по номеру: $(n, k) \to s$.
- Номер по объекту: $(n, s) \to k$.

Подмножества 1: формулировка

- Рассмотрим все подмножества множества из n элементов.
- Пронумеруем элементы множества числами $0, 1, 2, \ldots, n-1$.
- Каждое подмножество будем записывать в виде строки из n символов — нулей и единиц. Ноль в позиции i означает, что i-й элемент отсутствует в подмножестве, единица — что присутствует.
- Объект по номеру: $(n, k) \to s$.
- Номер по объекту: $(n, s) \to k$.

Подмножества 1: формулировка

- Рассмотрим все подмножества множества из n элементов.
- Пронумеруем элементы множества числами $0, 1, 2, \ldots, n-1$.
- Каждое подмножество будем записывать в виде строки из n символов — нулей и единиц. Ноль в позиции i означает, что i-й элемент отсутствует в подмножестве, единица — что присутствует.
- Объект по номеру: $(n, k) \to s$.
- Номер по объекту: $(n, s) \to k$.

Подмножества 1: формулировка

- Рассмотрим все подмножества множества из n элементов.
- Пронумеруем элементы множества числами 0, 1, 2, ..., $n - 1$.
- Каждое подмножество будем записывать в виде строки из n символов — нулей и единиц. Ноль в позиции i означает, что i-й элемент отсутствует в подмножестве, единица — что присутствует.
- Объект по номеру: $(n, k) \to s$.
- Номер по объекту: $(n, s) \to k$.

Подмножества 1: формулировка

- Рассмотрим все подмножества множества из n элементов.
- Пронумеруем элементы множества числами $0, 1, 2, \ldots, n-1$.
- Каждое подмножество будем записывать в виде строки из n символов — нулей и единиц. Ноль в позиции i означает, что i-й элемент отсутствует в подмножестве, единица — что присутствует.
- Объект по номеру: $(n, k) \to s$.
- Номер по объекту: $(n, s) \to k$.

Подмножества 1: решение

$$\begin{array}{lll} U_0 & = & (\varepsilon) \\ U_n & = & 0\ U_{n-1} \\ & & 1\ U_{n-1} \end{array} \qquad \begin{array}{lll} u_0 & = & 1 \\ u_n & = & u_{n-1} \\ & + & u_{n-1} \end{array} \qquad \begin{array}{lll} u_0 & = & 1 \\ u_n & = & 2 \cdot u_{n-1} \\ u_n & = & 2^n \end{array}$$

Получается, что код подмножества — это его лексикографический номер, записанный в двоичной системе счисления.

- Объект по номеру: $(n, k) \to s$. Если $k < u_{n-1}$, запишем в строку 0 и решим подзадачу $(n-1, k)$. Если же $k \geqslant u_{n-1}$, запишем в строку 1 и решим подзадачу $(n-1, k-u_{n-1})$.
- Номер по объекту: $(n, s) \to k$. Если строка начинается на 0, решим подзадачу для $n-1$ и строки s без первого символа. Если же строка начинается на 1, прибавим к номеру u_{n-1} и также решим подзадачу для $n-1$ и строки s без первого символа.

Подмножества 1: решение

$$
\begin{array}{lll lll lll}
U_0 & = & (\varepsilon) & u_0 & = & 1 & u_0 & = & 1 \\
U_n & = & 0\ U_{n-1} & u_n & = & u_{n-1} & u_n & = & 2 \cdot u_{n-1} \\
 & & 1\ U_{n-1} & & + & u_{n-1} & u_n & = & 2^n
\end{array}
$$

Получается, что код подмножества — это его лексикографический номер, записанный в двоичной системе счисления.

- Объект по номеру: $(n, k) \to s$. Если $k < u_{n-1}$, запишем в строку 0 и решим подзадачу $(n-1, k)$. Если же $k \geqslant u_{n-1}$, запишем в строку 1 и решим подзадачу $(n-1, k-u_{n-1})$.
- Номер по объекту: $(n, s) \to k$. Если строка начинается на 0, решим подзадачу для $n-1$ и строки s без первого символа. Если же строка начинается на 1, прибавим к номеру u_{n-1} и также решим подзадачу для $n-1$ и строки s без первого символа.

Подмножества 1: решение

$$
\begin{array}{lcl}
U_0 & = & (\varepsilon) \\
U_n & = & 0\ U_{n-1} \\
 & & 1\ U_{n-1}
\end{array}
\qquad
\begin{array}{lcl}
u_0 & = & 1 \\
u_n & = & u_{n-1} \\
 & + & u_{n-1}
\end{array}
\qquad
\begin{array}{lcl}
u_0 & = & 1 \\
u_n & = & 2 \cdot u_{n-1} \\
u_n & = & 2^n
\end{array}
$$

Получается, что код подмножества — это его лексикографический номер, записанный в двоичной системе счисления.

- Объект по номеру: $(n, k) \to s$. Если $k < u_{n-1}$, запишем в строку 0 и решим подзадачу $(n-1, k)$. Если же $k \geqslant u_{n-1}$, запишем в строку 1 и решим подзадачу $(n-1, k-u_{n-1})$.
- Номер по объекту: $(n, s) \to k$. Если строка начинается на 0, решим подзадачу для $n-1$ и строки s без первого символа. Если же строка начинается на 1, прибавим к номеру u_{n-1} и также решим подзадачу для $n-1$ и строки s без первого символа.

Подмножества 1: решение

$$
\begin{array}{lclclclcl}
U_0 & = & (\varepsilon) & u_0 & = & 1 & u_0 & = & 1 \\
U_n & = & 0\ U_{n-1} & u_n & = & u_{n-1} & u_n & = & 2 \cdot u_{n-1} \\
 & & 1\ U_{n-1} & & + & u_{n-1} & u_n & = & 2^n
\end{array}
$$

Получается, что код подмножества — это его лексикографический номер, записанный в двоичной системе счисления.

- Объект по номеру: $(n, k) \to s$. Если $k < u_{n-1}$, запишем в строку 0 и решим подзадачу $(n-1, k)$. Если же $k \geqslant u_{n-1}$, запишем в строку 1 и решим подзадачу $(n-1, k-u_{n-1})$.
- Номер по объекту: $(n, s) \to k$. Если строка начинается на 0, решим подзадачу для $n-1$ и строки s без первого символа. Если же строка начинается на 1, прибавим к номеру u_{n-1} и также решим подзадачу для $n-1$ и строки s без первого символа.

Подмножества 1: решение

$$
\begin{array}{lllllllll}
U_0 & = & (\varepsilon) & u_0 & = & 1 & u_0 & = & 1 \\
U_n & = & 0\ U_{n-1} & u_n & = & u_{n-1} & u_n & = & 2 \cdot u_{n-1} \\
 & & 1\ U_{n-1} & & + & u_{n-1} & u_n & = & 2^n
\end{array}
$$

Получается, что код подмножества — это его лексикографический номер, записанный в двоичной системе счисления.

- Объект по номеру: $(n, k) \to s$. Если $k < u_{n-1}$, запишем в строку 0 и решим подзадачу $(n-1, k)$. Если же $k \geqslant u_{n-1}$, запишем в строку 1 и решим подзадачу $(n-1, k-u_{n-1})$.
- Номер по объекту: $(n, s) \to k$. Если строка начинается на 0, решим подзадачу для $n-1$ и строки s без первого символа. Если же строка начинается на 1, прибавим к номеру u_{n-1} и также решим подзадачу для $n-1$ и строки s без первого символа.

Подмножества 1: пример

$$U_3 = \begin{array}{l} 0\ U_2 \\ 1\ U_2 \end{array} \qquad U_3 = \begin{array}{l} 0\ (00, 01, 10, 11) \\ 1\ (00, 01, 10, 11) \end{array}$$

$$U_3 = \begin{array}{l} 0\ (00, \\ \quad 01, \\ \quad 10, \\ \quad 11) \\ 1\ (00, \\ \quad 01, \\ \quad 10, \\ \quad 11) \end{array} \qquad U_3 = \begin{array}{l} (000, \\ 001, \\ 010, \\ 011) \\ (100, \\ 101, \\ 110, \\ 111) \end{array} \qquad U_3 = \begin{array}{l} (000, \\ 001, \\ 010, \\ 011, \\ 100, \\ 101, \\ 110, \\ 111) \end{array}$$

Подмножества 1: пример

$$U_3 = \begin{matrix} 0\ U_2 \\ 1\ U_2 \end{matrix} \qquad U_3 = \begin{matrix} 0\ (00, 01, 10, 11) \\ 1\ (00, 01, 10, 11) \end{matrix}$$

$$U_3 = \begin{matrix} 0\ (00, \\ 01, \\ 10, \\ 11) \\ 1\ (00, \\ 01, \\ 10, \\ 11) \end{matrix} \qquad U_3 = \begin{matrix} (000, \\ 001, \\ 010, \\ 011) \\ (100, \\ 101, \\ 110, \\ 111) \end{matrix} \qquad U_3 = \begin{matrix} (000, \\ 001, \\ 010, \\ 011, \\ 100, \\ 101, \\ 110, \\ 111) \end{matrix}$$

Подмножества 1: пример

$$U_3 = \begin{array}{l} 0\ U_2 \\ 1\ U_2 \end{array} \qquad U_3 = \begin{array}{l} 0\ (00, 01, 10, 11) \\ 1\ (00, 01, 10, 11) \end{array}$$

$$U_3 = \begin{array}{r} 0\ (00, \\ 01, \\ 10, \\ 11) \\ 1\ (00, \\ 01, \\ 10, \\ 11) \end{array} \qquad U_3 = \begin{array}{r} (000, \\ 001, \\ 010, \\ 011) \\ (100, \\ 101, \\ 110, \\ 111) \end{array} \qquad U_3 = \begin{array}{r} (000, \\ 001, \\ 010, \\ 011, \\ 100, \\ 101, \\ 110, \\ 111) \end{array}$$

Подмножества 1: пример

$$U_3 = \begin{array}{l} 0\ U_2 \\ 1\ U_2 \end{array} \qquad U_3 = \begin{array}{l} 0\ (00, 01, 10, 11) \\ 1\ (00, 01, 10, 11) \end{array}$$

$$U_3 = \begin{array}{ll} 0 & (00, \\ & 01, \\ & 10, \\ & 11) \\ 1 & (00, \\ & 01, \\ & 10, \\ & 11) \end{array} \qquad U_3 = \begin{array}{l} (000, \\ 001, \\ 010, \\ 011) \\ (100, \\ 101, \\ 110, \\ 111) \end{array} \qquad U_3 = \begin{array}{l} (000, \\ 001, \\ 010, \\ 011, \\ 100, \\ 101, \\ 110, \\ 111) \end{array}$$

Подмножества 1: пример

$$U_3 = \begin{array}{l} 0\ U_2 \\ 1\ U_2 \end{array} \qquad U_3 = \begin{array}{l} 0\ (00, 01, 10, 11) \\ 1\ (00, 01, 10, 11) \end{array}$$

$$U_3 = \begin{array}{r} 0\ (00, \\ 01, \\ 10, \\ 11) \\ 1\ (00, \\ 01, \\ 10, \\ 11) \end{array} \qquad U_3 = \begin{array}{r} (000, \\ 001, \\ 010, \\ 011) \\ (100, \\ 101, \\ 110, \\ 111) \end{array} \qquad U_3 = \begin{array}{r} (000, \\ 001, \\ 010, \\ 011, \\ 100, \\ 101, \\ 110, \\ 111) \end{array}$$

Оглавление

Подмножества 2: формулировка

- Рассмотрим все подмножества множества из n элементов.
- Пронумеруем элементы множества числами $1, 2, 3, \ldots, n$.
- Каждое подмножество будем записывать в виде возрастающей последовательности номеров его элементов.
- Объект по номеру: $(n, k) \to s$.
- Номер по объекту: $(n, s) \to k$.

Подмножества 2: формулировка

- Рассмотрим все подмножества множества из n элементов.
- Пронумеруем элементы множества числами 1, 2, 3, ..., n.
- Каждое подмножество будем записывать в виде возрастающей последовательности номеров его элементов.
- Объект по номеру: $(n, k) \to s$.
- Номер по объекту: $(n, s) \to k$.

Подмножества 2: формулировка

- Рассмотрим все подмножества множества из n элементов.
- Пронумеруем элементы множества числами 1, 2, 3, ..., n.
- Каждое подмножество будем записывать в виде возрастающей последовательности номеров его элементов.
- Объект по номеру: $(n, k) \to s$.
- Номер по объекту: $(n, s) \to k$.

Подмножества 2: формулировка

- Рассмотрим все подмножества множества из n элементов.
- Пронумеруем элементы множества числами 1, 2, 3, ..., n.
- Каждое подмножество будем записывать в виде возрастающей последовательности номеров его элементов.
- Объект по номеру: $(n, k) \to s$.
- Номер по объекту: $(n, s) \to k$.

Подмножества 2: формулировка

- Рассмотрим все подмножества множества из n элементов.
- Пронумеруем элементы множества числами 1, 2, 3, ..., n.
- Каждое подмножество будем записывать в виде возрастающей последовательности номеров его элементов.
- Объект по номеру: $(n, k) \to s$.
- Номер по объекту: $(n, s) \to k$.

Подмножества 2: формулы

$$\begin{array}{lcl}
V_0 & = & (()) \\
V_n & = & (()) \\
 & & 1\ V_{n-1}^{+1} \\
 & & 2\ V_{n-2}^{+2} \\
 & & \dots \\
 & & n\ V_{n-n}^{+n}
\end{array}
\qquad
\begin{array}{lcl}
v_0 & = & 1 \\
v_n & = & 1 \\
 & + & v_{n-1} \\
 & + & v_{n-2} \\
 & + & \dots \\
 & + & v_{n-n}
\end{array}
\qquad
\begin{array}{lcl}
v_n & = & 2^n \\
2^n & = & 1 \\
 & + & 2^{n-1} \\
 & + & 2^{n-2} \\
 & + & \dots \\
 & + & 2^0
\end{array}$$

- Здесь S^{+k} означает увеличение на k всех чисел во всех элементах последовательности S: мы решаем ту же задачу для номеров $k+1$, $k+2$, …, n.
- Последовательно перебираем случаи в лексикографическом порядке: либо запись подмножества пуста, либо она начинается с единицы, либо с двойки, …, либо с числа n (и им же и заканчивается).
- Как и следовало ожидать, количество подмножеств множества из n элементов от смены способа записи не изменилось.

Подмножества 2: формулы

$$
\begin{array}{rclcrclcrcl}
V_0 & = & (()) & & v_0 & = & 1 & & v_n & = & 2^n \\
V_n & = & (()) & & v_n & = & 1 & & 2^n & = & 1 \\
 & & 1\ V_{n-1}^{+1} & & & + & v_{n-1} & & & + & 2^{n-1} \\
 & & 2\ V_{n-2}^{+2} & & & + & v_{n-2} & & & + & 2^{n-2} \\
 & & \ldots & & & + & \ldots & & & + & \ldots \\
 & & n\ V_{n-n}^{+n} & & & + & v_{n-n} & & & + & 2^0
\end{array}
$$

- Здесь S^{+k} означает увеличение на k всех чисел во всех элементах последовательности S: мы решаем ту же задачу для номеров $k+1$, $k+2$, …, n.
- Последовательно перебираем случаи в лексикографическом порядке: либо запись подмножества пуста, либо она начинается с единицы, либо с двойки, …, либо с числа n (и им же и заканчивается).
- Как и следовало ожидать, количество подмножеств множества из n элементов от смены способа записи не изменилось.

Подмножества 2: формулы

$$\begin{array}{lcl lcl lcl}
V_0 & = & (()) & v_0 & = & 1 & v_n & = & 2^n \\
V_n & = & (()) & v_n & = & 1 & 2^n & = & 1 \\
 & & 1\ V_{n-1}^{+1} & & + & v_{n-1} & & + & 2^{n-1} \\
 & & 2\ V_{n-2}^{+2} & & + & v_{n-2} & & + & 2^{n-2} \\
 & & \dots & & + & \dots & & + & \dots \\
 & & n\ V_{n-n}^{+n} & & + & v_{n-n} & & + & 2^0
\end{array}$$

- Здесь S^{+k} означает увеличение на k всех чисел во всех элементах последовательности S: мы решаем ту же задачу для номеров $k+1$, $k+2$, $\dots$, n.
- Последовательно перебираем случаи в лексикографическом порядке: либо запись подмножества пуста, либо она начинается с единицы, либо с двойки, $\dots$, либо с числа n (и им же и заканчивается).
- Как и следовало ожидать, количество подмножеств множества из n элементов от смены способа записи не изменилось.

Подмножества 2: формулы

$$
\begin{array}{rcl}
V_0 & = & (()) \\
V_n & = & (()) \\
 & & 1\ V_{n-1}^{+1} \\
 & & 2\ V_{n-2}^{+2} \\
 & & \dots \\
 & & n\ V_{n-n}^{+n}
\end{array}
\qquad
\begin{array}{rcl}
v_0 & = & 1 \\
v_n & = & 1 \\
 & + & v_{n-1} \\
 & + & v_{n-2} \\
 & + & \dots \\
 & + & v_{n-n}
\end{array}
\qquad
\begin{array}{rcl}
v_n & = & 2^n \\
2^n & = & 1 \\
 & + & 2^{n-1} \\
 & + & 2^{n-2} \\
 & + & \dots \\
 & + & 2^0
\end{array}
$$

- Здесь S^{+k} означает увеличение на k всех чисел во всех элементах последовательности S: мы решаем ту же задачу для номеров $k+1$, $k+2$, $\dots$, n.
- Последовательно перебираем случаи в лексикографическом порядке: либо запись подмножества пуста, либо она начинается с единицы, либо с двойки, $\dots$, либо с числа n (и им же и заканчивается).
- Как и следовало ожидать, количество подмножеств множества из n элементов от смены способа записи не изменилось.

Подмножества 2: решение

- Объект по номеру: $(n, k) \to s$.
  - Будем просматривать строки формулы сверху вниз.
  - Если номер k строго меньше, чем размер x последовательности в правой части текущей строки, запишем в ответ префикс и перейдём к соответствующей подзадаче.
  - В противном случае вычтем из номера k размер x и перейдём к следующей строке формулы.
- Номер по объекту: $(n, s) \to k$.
  - Будем просматривать строки формулы сверху вниз.
  - Если строка начинается так же, как правая часть текущей строки, пропустим общий префикс и перейдём к соответствующей подзадаче.
  - В противном случае прибавим к номеру размер последовательности в правой части текущей строки и перейдём к следующей строке формулы.

Подмножества 2: решение

- Объект по номеру: $(n, k) \to s$.
  - Будем просматривать строки формулы сверху вниз.
  - Если номер k строго меньше, чем размер x последовательности в правой части текущей строки, запишем в ответ префикс и перейдём к соответствующей подзадаче.
  - В противном случае вычтем из номера k размер x и перейдём к следующей строке формулы.
- Номер по объекту: $(n, s) \to k$.
  - Будем просматривать строки формулы сверху вниз.
  - Если строка начинается так же, как правая часть текущей строки, пропустим общий префикс и перейдём к соответствующей подзадаче.
  - В противном случае прибавим к номеру размер последовательности в правой части текущей строки и перейдём к следующей строке формулы.

Подмножества 2: пример

$$
\begin{aligned}
V_3 &= (())\\
&\ 1\ {V_2}^{+1}\\
&\ 2\ {V_1}^{+2}\\
&\ 3\ {V_0}^{+3}
\end{aligned}
\qquad
\begin{aligned}
V_3 &= (())\\
&\ 1\ ((),(1),(1,2),(2))^{+1}\\
&\ 2\ ((),(1))^{+2}\\
&\ 3\ (())^{+3}
\end{aligned}
$$

$$
\begin{aligned}
V_3 = &\ (())\\
&1\ ((),\\
&\quad (2),\\
&\quad (2,3),\\
&\quad (3))\\
&2\ ((),\\
&\quad (3))\\
&3\ (())
\end{aligned}
\qquad
\begin{aligned}
V_3 = &\ (())\\
&((1),\\
&\ (1,2),\\
&\ (1,2,3),\\
&\ (1,3))\\
&((2),\\
&\ (2,3))\\
&((3))
\end{aligned}
\qquad
\begin{aligned}
V_4 = &\ ((),\\
&\ (1),\\
&\ (1,2),\\
&\ (1,2,3),\\
&\ (1,3),\\
&\ (2),\\
&\ (2,3),\\
&\ (3))
\end{aligned}
$$

Подмножества 2: пример

$$V_3 = \begin{array}{l} (()) \\ 1\ V_2{}^{+1} \\ 2\ V_1{}^{+2} \\ 3\ V_0{}^{+3} \end{array} \qquad V_3 = \begin{array}{l} (()) \\ 1\ ((),(1),(1,2),(2))^{+1} \\ 2\ ((),(1))^{+2} \\ 3\ (())^{+3} \end{array}$$

$$V_3 = \begin{array}{l} (()) \\ 1\ ((), \\ \quad (2), \\ \quad (2,3), \\ \quad (3)) \\ 2\ ((), \\ \quad (3)) \\ 3\ (()) \end{array} \qquad V_3 = \begin{array}{l} (()) \\ ((1), \\ \ (1,2), \\ \ (1,2,3), \\ \ (1,3)) \\ ((2), \\ \ (2,3)) \\ ((3)) \end{array} \qquad V_4 = \begin{array}{l} ((), \\ \ (1), \\ \ (1,2), \\ \ (1,2,3), \\ \ (1,3), \\ \ (2), \\ \ (2,3), \\ \ (3)) \end{array}$$

Подмножества 2: пример

$$V_3 = \begin{array}{l} (()) \\ 1\ {V_2}^{+1} \\ 2\ {V_1}^{+2} \\ 3\ {V_0}^{+3} \end{array} \qquad V_3 = \begin{array}{l} (()) \\ 1\ ((), (1), (1,2), (2))^{+1} \\ 2\ ((), (1))^{+2} \\ 3\ (())^{+3} \end{array}$$

$$V_3 = \begin{array}{l} (()) \\ 1\ ((), \\ \quad (2), \\ \quad (2,3), \\ \quad (3)) \\ 2\ ((), \\ \quad (3)) \\ 3\ (()) \end{array} \qquad V_3 = \begin{array}{l} (()) \\ ((1), \\ \ (1,2), \\ \ (1,2,3), \\ \ (1,3)) \\ ((2), \\ \ (2,3)) \\ ((3)) \end{array} \qquad V_4 = \begin{array}{l} ((), \\ \ (1), \\ \ (1,2), \\ \ (1,2,3), \\ \ (1,3), \\ \ (2), \\ \ (2,3), \\ \ (3)) \end{array}$$

Подмножества 2: пример

$$V_3 = (()) \quad 1\ {V_2}^{+1} \quad 2\ {V_1}^{+2} \quad 3\ {V_0}^{+3}$$

$$V_3 = (()) \quad 1\ ((), (1), (1,2), (2))^{+1} \quad 2\ ((), (1))^{+2} \quad 3\ (())^{+3}$$

$$V_3 = (()) \quad 1\ ((), (2), (2,3), (3)) \quad 2\ ((), (3)) \quad 3\ (())$$

$$V_3 = (()) \; ((1), (1,2), (1,2,3), (1,3)) \; ((2), (2,3)) \; ((3))$$

$$V_4 = ((), (1), (1,2), (1,2,3), (1,3), (2), (2,3), (3))$$

Подмножества 2: пример

$$\begin{aligned} V_3 &= (()) \\ &\quad 1\ {V_2}^{+1} \\ &\quad 2\ {V_1}^{+2} \\ &\quad 3\ {V_0}^{+3} \end{aligned} \qquad \begin{aligned} V_3 &= (()) \\ &\quad 1\ ((), (1), (1,2), (2))^{+1} \\ &\quad 2\ ((), (1))^{+2} \\ &\quad 3\ (())^{+3} \end{aligned}$$

$$\begin{aligned} V_3 = {} & (()) \\ & 1\ ((), \\ & \quad (2), \\ & \quad (2,3), \\ & \quad (3)) \\ & 2\ ((), \\ & \quad (3)) \\ & 3\ (()) \end{aligned} \qquad \begin{aligned} V_3 = {} & (()) \\ & ((1), \\ & \ (1,2), \\ & \ (1,2,3), \\ & \ (1,3)) \\ & ((2), \\ & \ (2,3)) \\ & ((3)) \end{aligned} \qquad \begin{aligned} V_4 = {} & ((), \\ & \ (1), \\ & \ (1,2), \\ & \ (1,2,3), \\ & \ (1,3), \\ & \ (2), \\ & \ (2,3), \\ & \ (3)) \end{aligned}$$

Подмножества 2: объект по номеру

$$
\begin{aligned}
n &= 6\\
k &= 43\\
s &= ()\\
\Delta &= 0
\end{aligned}
$$

$$
\begin{array}{rcll}
V_6 & = & (()) & 1\\
 & & 1\ V_5^{+1} & 32\\
 & & 2\ V_4^{+2} & 16\\
 & & 3\ V_3^{+3} & 8\\
 & & 4\ V_2^{+4} & 4\\
 & & 5\ V_1^{+5} & 2\\
 & & 6\ V_0^{+6} & 1
\end{array}
$$

Подмножества 2: объект по номеру

$$\begin{aligned} n &= 6 \\ k &= 42 \\ s &= () \\ \Delta &= 0 \end{aligned}$$

$$\begin{array}{lcll} V_6 & = & (()) & 1 \\ & & 1\ V_5^{+1} & \color{red}{32} \\ & & 2\ V_4^{+2} & 16 \\ & & 3\ V_3^{+3} & 8 \\ & & 4\ V_2^{+4} & 4 \\ & & 5\ V_1^{+5} & 2 \\ & & 6\ V_0^{+6} & 1 \end{array}$$

Подмножества 2: объект по номеру

$$n = 6$$
$$k = 10$$
$$s = ()$$
$$\Delta = 0$$

$$\begin{array}{lllll} V_6 & = & (()) & & 1 \\ & & 1 & V_5^{+1} & 32 \\ & & 2 & V_4^{+2} & 16 \\ & & 3 & V_3^{+3} & 8 \\ & & 4 & V_2^{+4} & 4 \\ & & 5 & V_1^{+5} & 2 \\ & & 6 & V_0^{+6} & 1 \end{array}$$

Подмножества 2: объект по номеру

$$\begin{array}{rcl} n & = & 6 \\ k & = & 10 \\ s & = & (2) \\ \Delta & = & 2 \end{array}$$

$$\begin{array}{rcll} V_4^{+2} & = & (()) & 1 \\ & & 3\ V_3^{+3} & 8 \\ & & 4\ V_2^{+4} & 4 \\ & & 5\ V_1^{+5} & 2 \\ & & 6\ V_0^{+6} & 1 \end{array}$$

Подмножества 2: объект по номеру

$$\begin{aligned} n &= 6 \\ k &= 9 \\ s &= (2) \\ \Delta &= 2 \end{aligned}$$

$$\begin{array}{lllll} V_4^{+2} & = & (()) & & 1 \\ & & 3 & V_3^{+3} & 8 \\ & & 4 & V_2^{+4} & 4 \\ & & 5 & V_1^{+5} & 2 \\ & & 6 & V_0^{+6} & 1 \end{array}$$

Подмножества 2: объект по номеру

$$
\begin{aligned}
n &= 6 \\
k &= 1 \\
s &= (2) \\
\Delta &= 2
\end{aligned}
$$

$$
\begin{array}{lcll}
V_4^{+2} & = & (()) & 1 \\
 & & 3\ V_3^{+3} & 8 \\
 & & 4\ V_2^{+4} & 4 \\
 & & 5\ V_1^{+5} & 2 \\
 & & 6\ V_0^{+6} & 1
\end{array}
$$

Подмножества 2: объект по номеру

$$\begin{aligned} n &= 6 \\ k &= 1 \\ s &= (2,4) \\ \Delta &= 4 \end{aligned}$$

$$\begin{array}{llll} V_2^{+4} & = & (()) & 1 \\ & & 5\ V_1^{+5} & 2 \\ & & 6\ V_0^{+6} & 1 \end{array}$$

Подмножества 2: объект по номеру

$$
\begin{aligned}
n &= 6 \\
k &= 0 \\
s &= (2, 4) \\
\Delta &= 4
\end{aligned}
$$

$$
\begin{array}{llll}
V_2^{+4} & = & (()) & 1 \\
 & & 5\ V_1^{+5} & 2 \\
 & & 6\ V_0^{+6} & 1
\end{array}
$$

Подмножества 2: объект по номеру

$$
\begin{aligned}
n &= 6 \\
k &= 0 \\
s &= (2,4,5) \\
\Delta &= 5
\end{aligned}
$$

$$
\begin{array}{rcll}
V_1^{+5} & = & (()) & 1 \\
 & & 6\ V_0^{+6} & 1
\end{array}
$$

Подмножества 2: номер по объекту

$$\begin{aligned} n &= 6 \\ s &= (2,4,5) \\ k &= 0 \\ \Delta &= 0 \end{aligned}$$

$$\begin{array}{rcll} V_6 & = & (()) & 1 \\ & & 1\ V_5^{+1} & 32 \\ & & 2\ V_4^{+2} & 16 \\ & & 3\ V_3^{+3} & 8 \\ & & 4\ V_2^{+4} & 4 \\ & & 5\ V_1^{+5} & 2 \\ & & 6\ V_0^{+6} & 1 \end{array}$$

Подмножества 2: номер по объекту

$$
\begin{aligned}
n &= 6 \\
s &= (2, 4, 5) \\
k &= 1 \\
\Delta &= 0
\end{aligned}
$$

$$
\begin{array}{lllr}
V_6 & = & (()) & 1 \\
 & & 1\ V_5^{+1} & \color{red}{32} \\
 & & 2\ V_4^{+2} & 16 \\
 & & 3\ V_3^{+3} & 8 \\
 & & 4\ V_2^{+4} & 4 \\
 & & 5\ V_1^{+5} & 2 \\
 & & 6\ V_0^{+6} & 1
\end{array}
$$

Подмножества 2: номер по объекту

$$\begin{aligned} n &= 6 \\ s &= (2, 4, 5) \\ k &= 33 \\ \Delta &= 0 \end{aligned}$$

$$\begin{array}{lllr} V_6 & = & (()) & 1 \\ & & 1\ V_5^{+1} & 32 \\ & & 2\ V_4^{+2} & 16 \\ & & 3\ V_3^{+3} & 8 \\ & & 4\ V_2^{+4} & 4 \\ & & 5\ V_1^{+5} & 2 \\ & & 6\ V_0^{+6} & 1 \end{array}$$

Подмножества 2: номер по объекту

$$\begin{aligned} n &= 6 \\ s &= (4, 5) \\ k &= 33 \\ \Delta &= 2 \end{aligned}$$

$$\begin{array}{lclr} V_4^{+2} & = & (()) & 1 \\ & & 3\ V_3^{+3} & 8 \\ & & 4\ V_2^{+4} & 4 \\ & & 5\ V_1^{+5} & 2 \\ & & 6\ V_0^{+6} & 1 \end{array}$$

Подмножества 2: номер по объекту

$$\begin{aligned} n &= 6 \\ s &= (4,5) \\ k &= 34 \\ \Delta &= 2 \end{aligned}$$

$$\begin{array}{lcll} V_4^{+2} & = & (()) & 1 \\ & & 3\ V_3^{+3} & \color{red}{8} \\ & & 4\ V_2^{+4} & 4 \\ & & 5\ V_1^{+5} & 2 \\ & & 6\ V_0^{+6} & 1 \end{array}$$

Подмножества 2: номер по объекту

$$\begin{aligned} n &= 6 \\ s &= (4, 5) \\ k &= 42 \\ \Delta &= 2 \end{aligned}$$

$$\begin{array}{lcll} V_4^{+2} & = & (()) & 1 \\ & & 3\ V_3^{+3} & 8 \\ & & 4\ V_2^{+4} & 4 \\ & & 5\ V_1^{+5} & 2 \\ & & 6\ V_0^{+6} & 1 \end{array}$$

Подмножества 2: номер по объекту

$$\begin{aligned} n &= 6 \\ s &= (5) \\ k &= 42 \\ \Delta &= 4 \end{aligned}$$

$$\begin{array}{llll} V_2^{+4} & = & (()) & 1 \\ & & 5\ V_1^{+5} & 2 \\ & & 6\ V_0^{+6} & 1 \end{array}$$

Подмножества 2: номер по объекту

$$\begin{aligned} n &= 6 \\ s &= (5) \\ k &= 43 \\ \Delta &= 4 \end{aligned}$$

$$\begin{array}{lcll} V_2^{+4} & = & (()) & 1 \\ & & 5\ V_1^{+5} & 2 \\ & & 6\ V_0^{+6} & 1 \end{array}$$

Подмножества 2: номер по объекту

$$
\begin{aligned}
n &= 6 \\
s &= () \\
k &= 43 \\
\Delta &= 5
\end{aligned}
$$

$$
\begin{array}{rclr}
V_1^{+5} & = & (()) & 1 \\
 & & 6 \;\; V_0^{+6} & 1
\end{array}
$$

Оглавление

Перестановки: формулировка

- Рассмотрим все перестановки множества из n элементов.
- Пронумеруем элементы множества числами $1, 2, 3, \ldots, n$.
- Каждую перестановку будем записывать в виде последовательности из n чисел: какой номер у элемента, стоящего на первом месте, какой — у элемента, стоящего на втором, ..., какой — у элемента, оказавшегося на последнем месте.
- Объект по номеру: $(n, k) \to s$.
- Номер по объекту: $(n, s) \to k$.

Перестановки: формулировка

- Рассмотрим все перестановки множества из n элементов.
- Пронумеруем элементы множества числами $1, 2, 3, \ldots, n$.
- Каждую перестановку будем записывать в виде последовательности из n чисел: какой номер у элемента, стоящего на первом месте, какой — у элемента, стоящего на втором, …, какой — у элемента, оказавшегося на последнем месте.
- Объект по номеру: $(n, k) \to s$.
- Номер по объекту: $(n, s) \to k$.

Перестановки: формулировка

- Рассмотрим все перестановки множества из n элементов.
- Пронумеруем элементы множества числами $1, 2, 3, \ldots, n$.
- Каждую перестановку будем записывать в виде последовательности из n чисел: какой номер у элемента, стоящего на первом месте, какой — у элемента, стоящего на втором, . . . , какой — у элемента, оказавшегося на последнем месте.
- Объект по номеру: $(n, k) \to s$.
- Номер по объекту: $(n, s) \to k$.

Перестановки: формулировка

- Рассмотрим все перестановки множества из n элементов.
- Пронумеруем элементы множества числами $1, 2, 3, \ldots, n$.
- Каждую перестановку будем записывать в виде последовательности из n чисел: какой номер у элемента, стоящего на первом месте, какой — у элемента, стоящего на втором, . . . , какой — у элемента, оказавшегося на последнем месте.
- Объект по номеру: $(n, k) \to s$.
- Номер по объекту: $(n, s) \to k$.

Перестановки: формулировка

- Рассмотрим все перестановки множества из n элементов.
- Пронумеруем элементы множества числами $1, 2, 3, \ldots, n$.
- Каждую перестановку будем записывать в виде последовательности из n чисел: какой номер у элемента, стоящего на первом месте, какой — у элемента, стоящего на втором, . . . , какой — у элемента, оказавшегося на последнем месте.
- Объект по номеру: $(n, k) \to s$.
- Номер по объекту: $(n, s) \to k$.

Перестановки: решение

$$
\begin{aligned}
P_0 &= (()) \\
P_n &= 1\ P_{n-1}^{(1)} \\
&\quad\ 2\ P_{n-1}^{(2)} \\
&\quad\ \dots \\
&\quad\ n\ P_{n-1}^{(n)}
\end{aligned}
$$

$$
\begin{aligned}
p_0 &= 1 \\
p_n &= n \cdot p_{n-1} \\
p_n &= n!
\end{aligned}
$$

Перестановки: решение

$$\begin{aligned} P_0 &= (()) \\ P_n &= 1\ P_{n-1}^{(1)} \\ &\ 2\ P_{n-1}^{(2)} \\ &\ \dots \\ &\ n\ P_{n-1}^{(n)} \end{aligned}$$

$$\begin{aligned} p_0 &= 1 \\ p_n &= n \cdot p_{n-1} \\ p_n &= n! \end{aligned}$$

Здесь $S^{(k)}$ означает «вставку» числа k во все элементы последовательности S: в каждой перестановке из S все элементы, не меньшие k, увеличиваются на единицу.

Перестановки: решение

$$\begin{aligned} P_0 &= (()) \\ P_n &= 1\ P_{n-1}^{(1)} \\ &\quad\ 2\ P_{n-1}^{(2)} \\ &\quad\ \dots \\ &\quad\ n\ P_{n-1}^{(n)} \end{aligned}$$

$$\begin{aligned} p_0 &= 1 \\ p_n &= n \cdot p_{n-1} \\ p_n &= n! \end{aligned}$$

Здесь $S^{(k)}$ означает «вставку» числа k во все элементы последовательности S: в каждой перестановке из S все элементы, не меньшие k, увеличиваются на единицу.

Пример:

$$P_2 = ((1,2), (2,1)) \qquad P_2^{(1)} = ((2,3), (3,2))$$

Перестановки: решение

$$
\begin{aligned}
P_0 &= (()) \\
P_n &= 1\ P_{n-1}^{(1)} \\
&\ 2\ P_{n-1}^{(2)} \\
&\ \dots \\
&\ n\ P_{n-1}^{(n)}
\end{aligned}
\qquad
\begin{aligned}
p_0 &= 1 \\
p_n &= n \cdot p_{n-1} \\
p_n &= n!
\end{aligned}
$$

Здесь $S^{(k)}$ означает «вставку» числа k во все элементы последовательности S: в каждой перестановке из S все элементы, не меньшие k, увеличиваются на единицу.

Пример:

$$
P_2 = ((1,2),\ (2,1)) \qquad P_2^{(2)} = ((1,3),\ (3,1))
$$

Перестановки: решение

$$
\begin{aligned}
P_0 &= (()) \\
P_n &= 1\ P_{n-1}^{(1)} \\
&\quad\ 2\ P_{n-1}^{(2)} \\
&\quad\ \ldots \\
&\quad\ n\ P_{n-1}^{(n)}
\end{aligned}
\qquad
\begin{aligned}
p_0 &= 1 \\
p_n &= n \cdot p_{n-1} \\
\\
p_n &= n!
\end{aligned}
$$

Здесь $S^{(k)}$ означает «вставку» числа k во все элементы последовательности S: в каждой перестановке из S все элементы, не меньшие k, увеличиваются на единицу.

Пример:

$$
P_2 = ((1,2), (2,1)) \qquad P_2^{(3)} = ((1,2), (2,1))
$$

Перестановки: решение

$$
\begin{aligned}
P_0 &= (()) \\
P_n &= 1\ P_{n-1}^{(1)} \\
&\ 2\ P_{n-1}^{(2)} \\
&\ \dots \\
&\ n\ P_{n-1}^{(n)}
\end{aligned}
\qquad
\begin{aligned}
p_0 &= 1 \\
p_n &= n \cdot p_{n-1} \\
\\
p_n &= n!
\end{aligned}
$$

- Здесь у нас n подзадач, каждая из которых добавляет $p_{n-1} = (n-1)!$ перестановок к нашей последовательности P_n.
- Объект по номеру: $(n, k) \to s$. Первый элемент перестановки равен $v = k \operatorname{div} (n-1)! + 1$. Переход осуществляется в подзадачу v (считая с единицы), а номер искомой перестановки в ней равен $k \bmod p_{n-1}$.
- Номер по объекту: $(n, s) \to k$. Пусть строка начинается на v. Тогда к номеру прибавляется $(v-1) \cdot (n-1)!$, а переход осуществляется к подзадаче v (считая с единицы).

Перестановки: решение

$$P_0 = (())$$

$$P_n = 1\ P_{n-1}^{(1)},\ 2\ P_{n-1}^{(2)},\ \dots,\ n\ P_{n-1}^{(n)}$$

$$p_0 = 1$$

$$p_n = n \cdot p_{n-1}$$

$$p_n = n!$$

- Здесь у нас n подзадач, каждая из которых добавляет $p_{n-1} = (n-1)!$ перестановок к нашей последовательности P_n.
- Объект по номеру: $(n, k) \to s$. Первый элемент перестановки равен $v = k \operatorname{div} (n-1)! + 1$. Переход осуществляется в подзадачу v (считая с единицы), а номер искомой перестановки в ней равен $k \bmod p_{n-1}$.
- Номер по объекту: $(n, s) \to k$. Пусть строка начинается на v. Тогда к номеру прибавляется $(v-1) \cdot (n-1)!$, а переход осуществляется к подзадаче v (считая с единицы).

Перестановки: решение

$$\begin{array}{rcl} P_0 & = & (()) \\ P_n & = & 1\ P^{(1)}_{n-1} \\ & & 2\ P^{(2)}_{n-1} \\ & & \dots \\ & & n\ P^{(n)}_{n-1} \end{array} \qquad \begin{array}{rcl} p_0 & = & 1 \\ p_n & = & n \cdot p_{n-1} \\ & & \\ p_n & = & n! \end{array}$$

- Здесь у нас n подзадач, каждая из которых добавляет $p_{n-1} = (n-1)!$ перестановок к нашей последовательности P_n.
- Объект по номеру: $(n, k) \to s$. Первый элемент перестановки равен $v = k \operatorname{div} (n-1)! + 1$. Переход осуществляется в подзадачу v (считая с единицы), а номер искомой перестановки в ней равен $k \bmod p_{n-1}$.
- Номер по объекту: $(n, s) \to k$. Пусть строка начинается на v. Тогда к номеру прибавляется $(v-1) \cdot (n-1)!$, а переход осуществляется к подзадаче v (считая с единицы).

Перестановки: пример

$$P_3 = \begin{array}{l} 1\ P_2^{(1)} \\ 2\ P_2^{(2)} \\ 3\ P_2^{(3)} \end{array}$$

$$P_3 = \begin{array}{l} 1\ ((2,3),(3,2)) \\ 2\ ((1,3),(3,1)) \\ 3\ ((1,2),(2,1)) \end{array}$$

$$P_3 = \begin{array}{l} 1\ ((2,3), \\ \quad (3,2)) \\ 2\ ((1,3), \\ \quad (3,1)) \\ 3\ ((1,2), \\ \quad (2,1)) \end{array}$$

$$P_3 = \begin{array}{l} ((1,2,3), \\ \ (1,3,2)) \\ ((2,1,3), \\ \ (2,3,1)) \\ ((3,1,2), \\ \ (3,2,1)) \end{array}$$

$$P_3 = \begin{array}{l} ((1,2,3), \\ \ (1,3,2), \\ \ (2,1,3), \\ \ (2,3,1), \\ \ (3,1,2), \\ \ (3,2,1)) \end{array}$$

Перестановки: пример

$$P_3 = \begin{array}{l} 1\ P_2^{(1)} \\ 2\ P_2^{(2)} \\ 3\ P_2^{(3)} \end{array} \qquad P_3 = \begin{array}{l} 1\ ((2,3),(3,2)) \\ 2\ ((1,3),(3,1)) \\ 3\ ((1,2),(2,1)) \end{array}$$

$$P_3 = \begin{array}{l} 1\ ((2,3), \\ \quad (3,2)) \\ 2\ ((1,3), \\ \quad (3,1)) \\ 3\ ((1,2), \\ \quad (2,1)) \end{array} \qquad P_3 = \begin{array}{l} ((1,2,3), \\ \ (1,3,2)) \\ ((2,1,3), \\ \ (2,3,1)) \\ ((3,1,2), \\ \ (3,2,1)) \end{array} \qquad P_3 = \begin{array}{l} ((1,2,3), \\ \ (1,3,2), \\ \ (2,1,3), \\ \ (2,3,1), \\ \ (3,1,2), \\ \ (3,2,1)) \end{array}$$

Перестановки: пример

$$P_3 = \begin{array}{l} 1\ P_2^{(1)} \\ 2\ P_2^{(2)} \\ 3\ P_2^{(3)} \end{array} \qquad P_3 = \begin{array}{l} 1\ ((2,3),(3,2)) \\ 2\ ((1,3),(3,1)) \\ 3\ ((1,2),(2,1)) \end{array}$$

$$P_3 = \begin{array}{l} 1\ ((2,3), \\ \quad (3,2)) \\ 2\ ((1,3), \\ \quad (3,1)) \\ 3\ ((1,2), \\ \quad (2,1)) \end{array} \qquad P_3 = \begin{array}{l} ((1,2,3), \\ \ (1,3,2)) \\ ((2,1,3), \\ \ (2,3,1)) \\ ((3,1,2), \\ \ (3,2,1)) \end{array} \qquad P_3 = \begin{array}{l} ((1,2,3), \\ \ (1,3,2), \\ \ (2,1,3), \\ \ (2,3,1), \\ \ (3,1,2), \\ \ (3,2,1)) \end{array}$$

Перестановки: пример

$$P_3 = \begin{array}{l} 1\ P_2^{(1)} \\ 2\ P_2^{(2)} \\ 3\ P_2^{(3)} \end{array} \qquad P_3 = \begin{array}{l} 1\ ((2,3),(3,2)) \\ 2\ ((1,3),(3,1)) \\ 3\ ((1,2),(2,1)) \end{array}$$

$$P_3 = \begin{array}{ll} 1 & ((2,3), \\ & \ (3,2)) \\ 2 & ((1,3), \\ & \ (3,1)) \\ 3 & ((1,2), \\ & \ (2,1)) \end{array} \qquad P_3 = \begin{array}{l} ((1,2,3), \\ \ (1,3,2)) \\ ((2,1,3), \\ \ (2,3,1)) \\ ((3,1,2), \\ \ (3,2,1)) \end{array} \qquad P_3 = \begin{array}{l} ((1,2,3), \\ \ (1,3,2), \\ \ (2,1,3), \\ \ (2,3,1), \\ \ (3,1,2), \\ \ (3,2,1)) \end{array}$$

Перестановки: пример

$$P_3 = \begin{array}{l} 1\ P_2^{(1)} \\ 2\ P_2^{(2)} \\ 3\ P_2^{(3)} \end{array} \qquad P_3 = \begin{array}{l} 1\ ((2,3),(3,2)) \\ 2\ ((1,3),(3,1)) \\ 3\ ((1,2),(2,1)) \end{array}$$

$$P_3 = \begin{array}{l} 1\ ((2,3), \\ \quad (3,2)) \\ 2\ ((1,3), \\ \quad (3,1)) \\ 3\ ((1,2), \\ \quad (2,1)) \end{array} \qquad P_3 = \begin{array}{l} ((1,2,3), \\ \ (1,3,2)) \\ ((2,1,3), \\ \ (2,3,1)) \\ ((3,1,2), \\ \ (3,2,1)) \end{array} \qquad P_3 = \begin{array}{l} ((1,2,3), \\ \ (1,3,2), \\ \ (2,1,3), \\ \ (2,3,1), \\ \ (3,1,2), \\ \ (3,2,1)) \end{array}$$

Всё.