

**Министерство образования и науки Российской Федерации**

**Федеральное агентство по образованию**

**Государственное образовательное учреждение высшего профессионального образования**

**«Московский государственный технический университет имени Н.Э. Баумана»**

**(МГТУ им. Н.Э. Баумана)**

**Факультет «Робототехника и комплексная автоматизация» (РК)**

**Кафедра «Системы автоматизированного проектирования» (РК6)**

****

**Отчет по лабораторной работе №2 по курсу**

**«Методы комбинаторных вычислений»**

**Студент: Петраков Станислав**

**Группа:** РК6-56Б

**Преподаватель:** Волосатова Т.М.

Проверил:

Дата:

2021 год

**10С вариант**

**Задание:**

Перечислить все перестановки из 5-ти десятичных цифр от 1 до 5, располагая их в порядке невозрастания суммы инверсий своих элементов.

**Описание алгоритма:**

Любая пара элементов, где имеет место нарушение их взаимного положения в перестановке, называется инверсией. Формально любая пара элементов *(pi, pj)* из перестановки *(p1,…,pi,…,pj,…,pn)* образует инверсию, если *pi>pj* и *i<j*.

Для компактной записи инверсий по элементам перестановки применяются вектор инверсий *(V1,…Vj,…Vn)* и таблица инверсий *(W1,…Wj,…Wn).* Эти обе формы записи идентифицируют числа инверсий для всех элементов перестановки *(p1,…pj,…pn)* и отличаются только порядком их перечисления. Формально, компоненты вектора и таблицы инверсий, определяются следующим образом:

Состав вектора и таблицы инверсий для цифровой перестановки *P* = (3,2,1,5,4) показан в следующем примере, где для наглядности индексированы позиции всех элементов:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *j* | 1 | 2 | 3 | 4 | 5 |
| *pj* | 3 | 2 | 1 | 5 | 4 |
| *Vj* | 0 | 1 | 2 | 0 | 1 |
| *Wj* | 2 | 1 | 0 | 1 | 0 |

Сумма значений всех компонентов вектора или таблицы инверсий определяет общее число инверсий *I* перестановки. В частности, для рассмотренного примера оно равно 4:

*I*(3,2,1,5,4) = 4.

При восстановлении из вектора инверсий*V*, элементы искомой перестановки *P* нужно последовательно вычислять справа налево в порядке уменьшения индекса, а их значения должны выбираться из списка *L* упорядоченных по убыванию целых чисел от *n* до 1. При этом значение очередного элемента перестановки *pi* принимается равным (*Vi+*1)*-*му по величине числу списка*L*. После этого выбранное число должно быть исключено из списка. Выполнение алгоритма иллюстрирует следующий пример восстановления перестановки *P* = (3,2,1,5,4) по вектору инверсий *V* = (0,1,2,0,1):

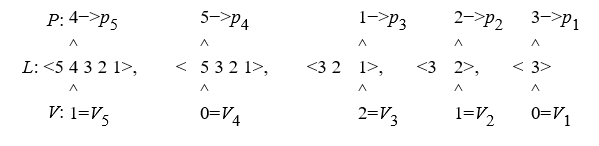


Рисунок 1. Схема восстановления последовательности по вектору инверсий

В этом примере символ *'^'* указывает элемент, который исключается из списка *L* на каждой итерации, для передачи его значения очередному элементу перестановки.

Поэтому векторы и таблицы инверсий можно рассматривать как целочисленные значения, представленные в системах счисления с возрастающими и убывающими факториалами. Их величины вычисляются, соответственно, по следующим формулам, где учтено равенство нулю значений *V1* и *Wn:*

При этом всегда возможно выполнить обратное преобразование, которое позволяет однозначно восстановить компоненты вектора или таблицы инверсий по любому заданному целому числу.

Алгоритм инверсий порождает последовательность перестановок, которая начинается с лексиграфически наименьшей перестановки с нулевым числом инверсий и заканчивается лексиграфически наибольшей перестановкой, где число инверсий максимально. При этом у каждой промежуточной перестановки число инверсий не меньше, чем у предыдущей перестановки. Таким образом, получаются последовательности перестановок, частично упорядоченные по суммам инверсий своих элементов.

**Исходный код:**

#include <iostream>

#include <vector>

#define LEN 5

int factorial(int x)

{

return (x < 2) ? 1 : x \* factorial(x - 1);

}

void print\_next\_transposition(int n)

{

std::vector<int> max\_transp; //5 4 3 2 1

int ans[LEN]; //result vector

int v[LEN]; //vector of inversions

for (int i = LEN; i >= 1; i--)

{

max\_transp.push\_back(i);

}

v[0] = 0;

for (int i = LEN - 1; i > 0; i--) //finding vector of inversions

{

v[i] = n / factorial(i);

n = n - factorial(i) \* v[i];

}

for (int i = LEN - 1; i >= 0; i--)

{

int j = v[i];

ans[i] = max\_transp[j];

max\_transp.erase(max\_transp.begin() + j);

}

for (int i = 0; i < LEN; i++)

{

std::cout << ans[i] << ' ';

}

std::cout << '\n';

}

int main()

{

int numb = 1;

for (int n = factorial(LEN) - 1; n >= 0; n--)

{

std::cout << numb << ": ";

print\_next\_transposition(n);

}

}

**Результат работы программы:**

5 4 3 2 1  
4 5 3 2 1  
5 3 4 2 1  
3 5 4 2 1  
4 3 5 2 1  
3 4 5 2 1  
5 4 2 3 1  
4 5 2 3 1  
5 2 4 3 1  
2 5 4 3 1  
4 2 5 3 1  
2 4 5 3 1  
5 3 2 4 1  
3 5 2 4 1  
5 2 3 4 1  
2 5 3 4 1  
3 2 5 4 1  
2 3 5 4 1  
4 3 2 5 1  
3 4 2 5 1  
4 2 3 5 1  
2 4 3 5 1  
3 2 4 5 1  
2 3 4 5 1  
5 4 3 1 2  
4 5 3 1 2  
5 3 4 1 2  
3 5 4 1 2  
4 3 5 1 2  
3 4 5 1 2  
5 4 1 3 2  
4 5 1 3 2  
5 1 4 3 2  
1 5 4 3 2  
4 1 5 3 2  
1 4 5 3 2  
5 3 1 4 2  
3 5 1 4 2  
5 1 3 4 2  
1 5 3 4 2  
3 1 5 4 2  
1 3 5 4 2  
4 3 1 5 2  
3 4 1 5 2  
4 1 3 5 2  
1 4 3 5 2  
3 1 4 5 2  
1 3 4 5 2  
5 4 2 1 3  
4 5 2 1 3  
5 2 4 1 3  
2 5 4 1 3  
4 2 5 1 3  
2 4 5 1 3  
5 4 1 2 3  
4 5 1 2 3  
5 1 4 2 3  
1 5 4 2 3  
4 1 5 2 3  
1 4 5 2 3  
5 2 1 4 3  
2 5 1 4 3  
5 1 2 4 3  
1 5 2 4 3  
2 1 5 4 3  
1 2 5 4 3  
4 2 1 5 3  
2 4 1 5 3  
4 1 2 5 3  
1 4 2 5 3  
2 1 4 5 3  
1 2 4 5 3  
5 3 2 1 4  
3 5 2 1 4  
5 2 3 1 4  
2 5 3 1 4  
3 2 5 1 4  
2 3 5 1 4  
5 3 1 2 4  
3 5 1 2 4  
5 1 3 2 4  
1 5 3 2 4  
3 1 5 2 4  
1 3 5 2 4  
5 2 1 3 4  
2 5 1 3 4  
5 1 2 3 4  
1 5 2 3 4  
2 1 5 3 4  
1 2 5 3 4  
3 2 1 5 4  
2 3 1 5 4  
3 1 2 5 4  
1 3 2 5 4  
2 1 3 5 4  
1 2 3 5 4  
4 3 2 1 5  
3 4 2 1 5  
4 2 3 1 5  
2 4 3 1 5  
3 2 4 1 5  
2 3 4 1 5  
4 3 1 2 5  
3 4 1 2 5  
4 1 3 2 5  
1 4 3 2 5  
3 1 4 2 5  
1 3 4 2 5  
4 2 1 3 5  
2 4 1 3 5  
4 1 2 3 5  
1 4 2 3 5  
2 1 4 3 5  
1 2 4 3 5  
3 2 1 4 5  
2 3 1 4 5  
3 1 2 4 5  
1 3 2 4 5  
2 1 3 4 5  
1 2 3 4 5

**Используемая литература:**

## Т.М. Волосатова, С.В. Родионов. Методы комбинаторных вычислений, учебное пособие. Издательство МГТУ им. Н.Э. Баумана. Москва. 2011