

**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 №1 по курсу**

**«Методы комбинаторных вычислений»**

**Студент: Сергеева Диана**

**Группа:** РК6-56Б

**Преподаватель:** Родионов С.В.

Проверил:

Дата:

2021 год

**13С вариант**

**Задание:**

Перечислить все траектории из левого нижнего в правый верхний угол прямоугольной сетки размером 4x5, которые проходят через минимальное число её клеток. Для перечисления этих траекторий нужно применить алгоритм транспозиции бинарных сочетаний с левым сдвигом, кодируя проход через клетки по горизонтали и вертикали символом H и V, соответственно.

**Описание алгоритма:**

В общем случае задача перечисления бинарных сочетаний, в которых n элементов по m элементов следует задавать в форме n-разрядного двоичного числа, где m единичных разрядов обозначают элементы сочетания, а остальные (n-m) разрядов имеют нулевые значения, сводится к систематическому перебору всех n-разрядных двоичных наборов с различным расположением m единичных и (n-m) нулевых разрядов. В наиболее простой форме такой перебор реализуют различные методы транспозиции смежных разрядов со сдвигом (транспозитивно-сдвиговые алгоритмы). Это итерационные алгоритмы, а их названия отражают характер операций, выполняемых на каждом шаге. Итерационные процедуры транспозитивно-сдвиговых алгоритмов формируют последовательности бинарных сочетаний, которые начинаются двоичным набором, где все единицы сосредоточены в младших разрядах (справа), и завершаются, когда все единицы будут находиться в старших разрядах (слева).

В алгоритме транспозиции с левым сдвигом на каждом шаге очередное бинарное сочетание получается из текущего заменой крайней левой пары разрядов 01 на 10 (транспозиция) и смещением группы лидирующих единичных разрядов, если таковые имеются, вплотную к паре 10, полученной после транспозиции (сдвиг). Если при этом в текущем бинарном сочетании нет единиц в старших разрядах, то сдвиг не производится, даже когда лидирующая единица получается после транспозиции на данном шаге. Сдвиг также не производится, когда в старших разрядах перед парой 10, полученной после транспозиции нет нулей.

В текущей задаче необходимо перечислить все траектории (комбинации путей) из нижнего левого угла сетки 4х5 в правый верхний, где траектории проходят через минимальное число клеток. По сути условие минимальности в этой задаче сводится к ограничению направления движения, т. е. смещение должно происходить только вправо и вверх, не выходя за границы сетки.

Посчитаем число путей из левого нижнего угла (0;0) в правый верхний (3;4):

**Исходный код:**

#include <stdio.h>

#define N 7

int compare\_arrays(int A[], int B[], int n) {

for( int i = 0; i < n; i++) {

if (A[i] != B[i])

return 0;

}

return 1;

}

void copy\_arrays(int A[], int B[], int n) {

for(int i = 0; i < n; i++) {

B[i] = A[i];

}

}

void print\_array(int A[], int n) {

for (int i = 0; i < n; i++ ) {

if(A[i] == 1) {

printf("%s", "H"); //Horizontal step

}

if(A[i] == 0) {

printf("%s", "V"); //Vertical step

}

}

putchar('\n');

}

void LST(int A[]) {

int B[N];

int i, j, k;

int n = 1; //Number of combinations

while(!compare\_arrays(A,B, N)) { //exit when A=B

copy\_arrays(A, B, N); //=A on previous transposition step

for (i = 0; i < N-1; i++) {

if (A[i] == 0 && A[i + 1] == 1) { //search 01

A[i] = 1;

A[i+1] = 0; //replace 01 by 10

print\_array(A, N);

n++;

if (i > 1) {

k = 1;

for (j = i - 2; j >= 0; j--) { //search group of ones preceding 10

if (A[j] == 1 && A[j+1] == 0) {

A[i - k] = 1; //shift group of ones to 10

A[j] = 0;

k++;

}//endif

}//endfor

}//endif

break; //escape from for when transposition step completed

}//endif

}//endfor

}//endwhile

printf("Number of combinations: %d\n", n);

}

int main() {

int A[N] = {0, 0, 0, 0, 1, 1, 1}; //4 vertical steps, 3 horizontal step

print\_array(A, N); //first combination

LST(A); //Left Shift Transposition

return 0;

}

**Результат работы программы:**

VVVVHHH  
VVVHVHH  
VVHVVHH  
VHVVVHH  
HVVVVHH  
HVVVHVH  
VVHVHVH  
VHVVHVH  
HVVVHVH  
HVVHVVH  
VHVHVVH  
HVVHVVH  
HVHVVVH  
HVHVVVH  
HHVVVVH  
HHVVVHV  
VVHVHHV  
VHVVHHV  
HVVVHHV  
HVVHVHV  
VHVHVHV  
HVVHVHV  
HVHVVHV  
HVHVVHV  
HHVVVHV  
HHVVHVV  
VHVHHVV  
HVVHHVV  
HVHVHVV  
HVHVHVV  
HHVVHVV  
HHVHVVV  
HVHHVVV  
HHVHVVV  
HHHVVVV  
Number of combinations: 35

**Используемая литература:**

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