|  |  |
| --- | --- |
|  | **Министерство образования и науки Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ Робототехники и комплексной автоматизации

КАФЕДРА Системы автоматизированного проектирования (РК-6)

**ОТЧЕТ О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ**

Студент Долженко Анастасия Тимофеевна

Группа РК6-52Б

Вариант 7

Студент **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Долженко А.Т.**

*подпись, дата фамилия, и.о.*

Преподаватель **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Волосатова Т.М.**

*подпись, дата фамилия, и.о.*

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Москва, 2024 г.*

**Оглавление**

[Задание 3](#_Toc178644053)

[Алгоритм работы программы 3](#_Toc178644054)

[Приложение 1. Код программы 4](#_Toc178644055)

[Приложение 2. Результаты работы программы 5](#_Toc178644056)

[Список литературы 5](#_Toc178644057)

# Задание

Вариант 7

Перечислить все траектории из левого нижнего в правый верхний угол прямоугольной сетки размером 4х5, которые проходят через минимальное число ее клеток. Для перечисления этих траекторий нужно применить алгоритм транспозиции бинарных сочетаний с правым сдвигом, кодируя проход через клетки по горизонтали и вертикали символами Н и V, соответственно.

# Алгоритм работы программы

Для начала определим, что для прямоугольной сетки размером 4 на 5 минимальная длина траектории составит 7 перемещений – 3 вертикально и 4 горизонтально.

Алгоритм работы программы основывается на алгоритме бинарной транспозиции с правым сдвигом. Траекторию представим последовательностью нулей и единиц, где 0 обозначает V, то есть сдвиг по вертикали, а 1 обозначает H, то есть сдвиг по горизонтали.

Далее применяем алгоритм бинарной транспозиции с правым сдвигом. Рассмотрим его работу на примере:

VVVHHHH – транспозиция -> VVHVHHH – транспозиция -> VVHHVHH – транспозиция -> VVHHHVH – транспозиция -> (VVHHHHV – транспозиция -> VHVHHHV – сдвиг ->VHVVHHH) ->… -> HHHHVVV

Конец перебора наступает, когда в строке траектории не осталось «VH» или «01», или все V или 0 перемещены вправо.

В реализации программы используется подход, при котором строка траектории строится по сохраненным индексам, соответствующим в данной траектории V (0). При транспозиции и сдвиге мы меняем лишь эти сохраненные индексы.

# Приложение 1. Код программы

#include <stdio.h>

#include <string.h>

#define N 7 // Общее количество шагов (4 H и 3 V)

#define K 3 // Количество шагов по вертикали V

// печать текущей траектории

void print\_path(int v\_position[], int n){

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

if (v\_position[i] == 1) printf("H");

else printf("V");

}

printf("\n");

}

// генерация следующей траектории

int next\_path(int v\_position[], int n, int k){

int cur\_nul\_pos = k - 1;

// проверяем, сколько нулей в конце - наша цель - переместить в конец все нули

while(cur\_nul\_pos >= 0 && v\_position[cur\_nul\_pos] == n - k + cur\_nul\_pos) {

cur\_nul\_pos--;

}

// если i < 0 - в конце 000, все пути перебрали,

// если i >= 0, то i показывает с каким из 0 в v\_position мы работаем

if (cur\_nul\_pos < 0) return 0;

// транспозиция - двигаем самый правый 0, который еще не в конце (меняем 01 на 10)

v\_position[cur\_nul\_pos]++;

// если мы работаем не с самым правым 0,

// то делаем сдвиг - смещаем все нули правее текущего на текущую позицию+1

for (int j = cur\_nul\_pos + 1; j < k; j++) v\_position[j] = v\_position[j - 1] + 1;

return 1;

}

int main(){

// позиции V в пути

int v\_position[K];

// определяем первый путь - vvvhhhh - позиции V: 0,1,2

for (int i = 0; i < K; i++) v\_position[i] = i;

// массив для хранения текущего пути (V = 0, H = 1, старт vvvhhhh)

int path[N];

do{

// заполняем (обновляем) путь: сначала заполняем все 1 (H), потом ставим 0 (V) на позиции из v\_position

for (int i = 0; i < N; i++) path[i] = 1;

for (int i = 0; i < K; i++) path[v\_position[i]] = 0;

// печать текущего пути

print\_path(path, N);

} while(next\_path(v\_position, N, K));

return 0;}

# Приложение 2. Результаты работы программы

Изображение выглядит как текст, Шрифт, снимок экрана, шаблон

Автоматически созданное описание Изображение выглядит как текст, Шрифт, снимок экрана, типография

Автоматически созданное описание Изображение выглядит как текст, Шрифт, типография, снимок экрана

Автоматически созданное описание

# Список литературы

1. Методы комбинаторных вычислений – Т.М. Волосатова, С.В. Родионов, 2011
2. http://bigor.bmstu.ru/?cnt/?doc=Comby/base.cou