

**МИНОБРНАУКИ РОССИИ**  
**ФГБОУ ВО «СГУ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**

**АЛГОРИТМ ТРАНСПОНИРОВАНИЯ МАТРИЦ**

**ЛАБОРАТОРНАЯ РАБОТА**

студента 3 курса 331 группы  
специальности 100501 — Компьютерная безопасность  
факультета КНиИТ  
Окунькова Сергея Викторовича

Проверил  
доцент

\_\_\_\_\_

А. Н. Гамова

## **СОДЕРЖАНИЕ**

1	Описание алгоритма .....	3
2	Эффективность алгоритма.....	4
3	Реализация .....	5
4	Тестирование программы .....	7
	СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	8

## 1 Описание алгоритма

Транспонирование — в линейной алгебре это операция над матрицами в результате которой матрица поворачивается относительно своей главной диагонали. При этом столбцы исходной матрицы становятся строками результирующей. Очевидно, что если исходная матрица  $A$  имела размер  $mn$ , то транспонированная матрица  $A^T$  будет размером  $nm$ . Матрица-строка в результате транспонирования преобразуется в матрицу-столбец и наоборот. Несложно увидеть, что  $A^T[i][j] = A[j][i]$ .

Идея моей реализации заключается в создании промежуточной матрицы  $b$  размерности  $nm$ ,  $b[i][j]$ , элемент которой присваивается  $a[j][i]$  элементу исходной матрицы. После чего присваиваем промежуточную переменную, в которой хранится промежуточная матрица, переменной основной матрицы, или, если в задаче необходимо сохранить основную матрицу, то просто возвращаем промежуточную матрицу.

## 2 Эффективность алгоритма

Сложность работы алгоритма определяется как  $O(nm)$ , так как мы проходим по всем элементам заданной матрицы и присваиваем их новой матрице. Однако данный алгоритм так же выделяет дополнительную память, которая определяется как  $O(nm)$ , что является главным недостатком данного алгоритма.

### 3 Реализация

```
#include <iostream>
#include <vector>

using namespace std;

vector<vector<int>> T (vector<vector<int>> &a) {
    vector<vector<int>> b (a[0].size(), vector<int> (a.size()));

    for(int i = 0; i < b.size(); ++i) {
        for(int j = 0; j < b[i].size(); ++j) {
            b[i][j] = a[j][i];
        }
    }

    return b;
}

int main(void) {
    setlocale(LC_ALL, "rus");
    int col, row;
    cout << "Введите размерности матрицы: ";
    cin >> col >> row;
    cout << "Введите исходную матрицу:\n";
    vector<vector<int>> a (col, vector<int> (row));

    for(int i = 0; i < col; ++i) {
        for(int j = 0; j < row; ++j) {
            cin >> a[i][j];
        }
    }

    vector<vector<int>> b = T(a);

    cout << "Транспонированная матрица:\n";
    for(int i = 0; i < b.size(); ++i) {
        for(int j = 0; j < b[i].size(); ++j) {
            cout << b[i][j] << " ";
        }
        cout << endl;
    }
}
```

```
    return 0;  
}
```

#### 4 Тестирование программы

```
sokunkov@C11310 complexity_calculation % ./a.out
Введите размерности матрицы: 5 3
Введите исходную матрицу:
1 2 3
-1 -2 -3
0 0 0
7 8 9
1 1 1
Транспонированная матрица:
1 -1 0 7 1
2 -2 0 8 1
3 -3 0 9 1
sokunkov@C11310 complexity_calculation %
```

Рисунок 1 – Тест1

## **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

- 1 Скиена Стивен "Алгоритмы. Руководство по разработке 2018 год. Яз. рус.
- 2 Нииколаус Вирт "Алгоритмы и структуры данных 2008 год. Яз. рус.