

УДК 004

DOI 10.21661/r-497643

*Н.К. Петрова, Т.М. Алимов, М.Б. Алабдуллах,
А.Х.Х. Альхузайи, Р.З. Каттави*

ДИНАМИЧЕСКИЕ МАССИВЫ НА C++ И ЗАДАЧА ПОИСКА МИНИМАКСОВ

Аннотация: в представленной статье рассматривается классическая задача поиска седловой точки в двумерном массиве. Подобного рода задачи весьма востребованы в геодезии, при построении топографических или гравиметрических карт местности, в энергетике – при планировании нагрузки на электрические или тепловые сети. Код программ предлагаемого проекта написан в рамках структурного подхода на языке C++ в среде разработки Visual Studio 2017.

Ключевые слова: седловая точка, VS Studio, C++, динамические массивы.

*N.K. Petrova, T.M. Alimov, M.B. Alabdullah,
A.H.H. Alhuzaji, R.Z. Kattavi*

DYNAMIC ARRAYS IN C ++ AND MINIMAX SEARCH PROBLEM

Abstract: the article presents the classical saddle-point problem in a two-dimensional array. Such tasks are highly demanded in geodesy, in the construction of topographic or gravimetric maps of the terrain, in the energy sector – in planning the load on electric or thermal networks. The program code of the proposed project is written within the framework of the structured approach in the C ++ language in the Visual Studio 2017 programming environment.

Keywords: saddle point, VS Studio, C ++, dynamic arrays.

Введение. Возможности операций с динамическими массивами на C++ позволяет без особых затруднений решать классические задачи на их основе для закрепления навыков выполнения низкоуровневых операций при работе с отдельными элементами массива, а также передаче массивов в другие функции через ссылки и указатели, строя проект в рамках технологии структурного программирования. Дадим определение

ключевому понятию «седловая точка», рассматриваемого в нашей статье. Седловая точка – элемент матрицы, являющуюся наименьшим в своей строке и одновременно наибольшим в своем столбце или, наоборот, является наибольшим в своей строке и наименьшим в своем столбце. Составим программный проект, который для любого двумерного массива размера $(N \times M)$, где N – количество строк, M – количество столбцов, находит индексы седловых точек и их количество. Программу будем реализовывать в среде программирования Visual Studio 2017 на языке C++.

Контрольные примеры для разработки алгоритма и обеспечения полноты программы. Для реализации проекта необходимо начать разработку с составления контрольных примеров. В качестве контрольного примера для тестирования рассмотрим двумерный массив размером $(n = 3; m = 4)$, заполним его числами в пределах $[0; 100]$ (рис. 1).

	0	1	2	3
0	45	84	25	17
1	60	72	33	1
2	80	97	47	29

Рис. 1. Исходный массив

Далее найдем максимальные элементы в каждой строке (рис. 2):

	0	1	2	3
0	45	84	25	17
1	60	72	33	1
2	80	97	47	29

Рис. 2. Максимальные элементы строк

Далее найдем минимальные элементы в каждой строке (рис. 3):

	0	1	2	3
0	45	84	25	17
1	60	72	33	1
2	80	97	47	29

Рис. 3. Минимальные элементы строк

Далее найдем максимальные элементы в каждом столбце (рис. 4):

	0	1	2	3
0	45	84	25	17
1	60	72	33	1
2	80	97	47	29

Рис. 4. Максимальные элементы столбцов

Далее найдем минимальные элементы в каждом столбце (рис. 5):

	0	1	2	3
0	45	84	25	17
1	60	72	33	1
2	80	97	47	29

Рис. 5. Минимальные элементы столбцов

Далее сравниваем максимальные элементы строк и минимальные элементы столбцов, и наоборот, минимальные элементы строк и максимальные элементы столбцов, если они совпадают, то записываем индексы этой точки и увеличиваем количество седловых точек на 1.

Проанализировав результаты, видим, что есть 2 седловые точки с индексами (1; 1) и (2; 3) (рис. 6):

	0	1	2	3
0	45	84	25	17
1	60	72	33	1
2	80	97	47	29

Рис. 6. Седловые точки

Анализ контрольных примеров показывает, что решение данной задачи требует использование следующих алгоритмических методов:

- введение счётчика. Переменная k для подсчета количества седловых точек;
- использование операторов ввода/вывода. Операторы *cin* и *cout*;
- использование циклических алгоритмов: *for* и *while*;
- работа с индексами двумерного массива. N – количество строк, M – количество столбцов массива a ;

- введение временных переменных. *Min* – минимальный элемент строки; *Max* – максимальный элемент строки; *MinCol* – минимальный элемент столбца; *MaxCol* – максимальный элемент столбца;
- нахождение максимума;
- нахождение минимума;
- передача массива в функции. Массив передаётся в функции через указатели на указатели. Функция *MakeArr* – создает массив, *print* – выводит массив, *solution* – вычисляет седловые точки массива и выводит их индексы на экран.

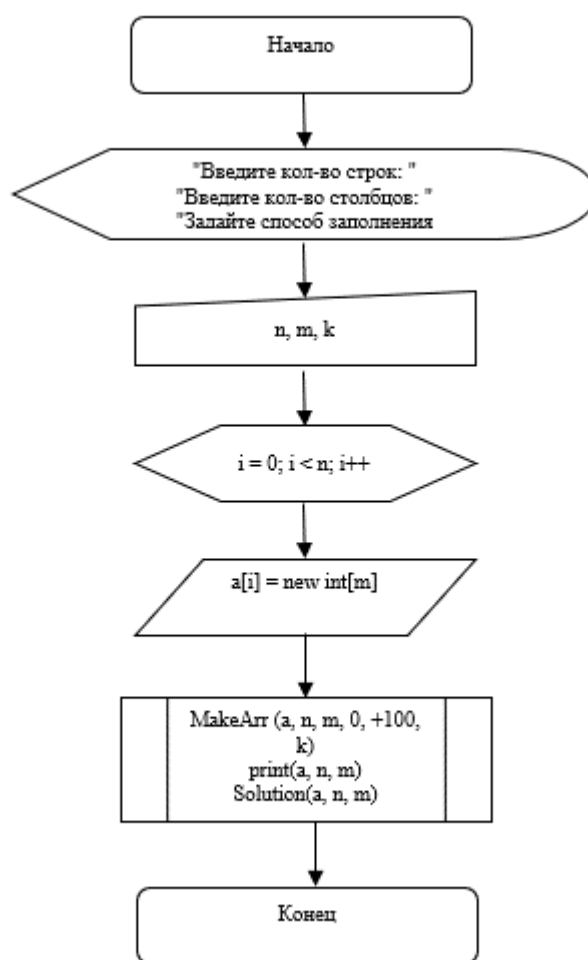


Рис. 7. Главная программа

В качестве примера приведем текст функции *MakeArr*:

```

void MakeArr(int **tf, int nf, int mf, int r_min, int r_max, int way)
{
    srand((unsigned int)time(NULL));

```

```

for (int i = 0; i < nf; i++)
for (int j = 0; j < mf; j++)
{
if (way == 1)
tf[i][j] = rand () % (r_max - r_min) + r_min;
else
{
cout << «Элемент (" << i << "; " << j << "): ";
cin >> tf[i][j];
}
}
return;
}

```

Int tf – исходный массив; *int nf* – количество строк; *int mf* – количество столбцов; *int r_min* – левая граница генерируемых чисел; *int r_max* – правая граница генерируемых чисел; *int way* – флаг, введенный для выбора способа инициализации массива. Если он равен 1, то инициализация идёт с помощью генератора случайных чисел в заданном диапазоне значений от *r_min* до *r_max*, в противном случае инициализация осуществляется вводом значений с консоли.

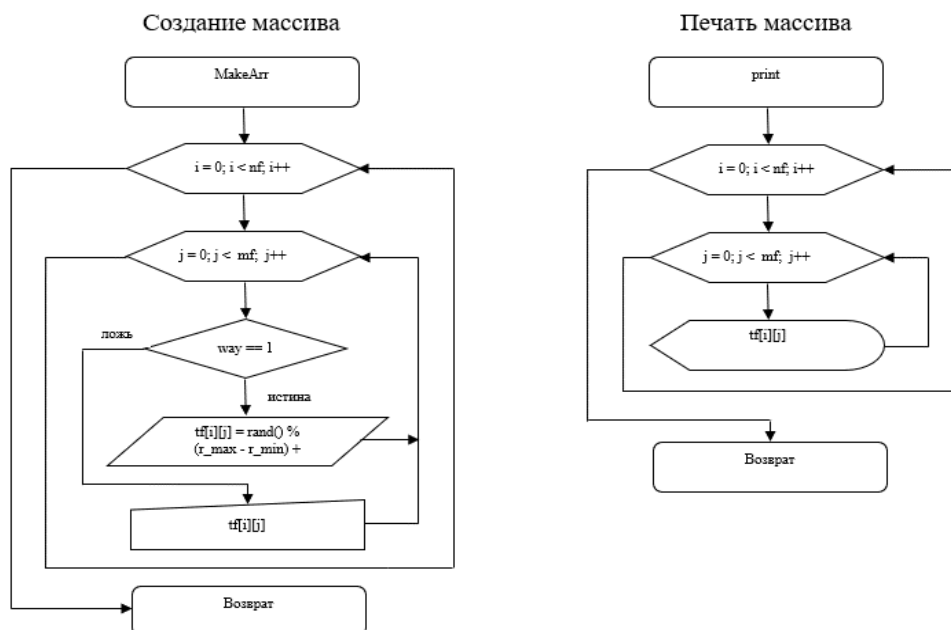


Рис. 8. Создание и печать массива

Заключение. По итогам проведенной работы нами разработан алгоритм, использующий приемы структурного программирования, для нахождения седловых точек в двумерном массиве, визуализированный словесно-блочной схемой, разработан контрольный пример для проверки корректности программного кода.

Программу протестировали 3 раза:

На 2 способа инициализации двумерного массива (1–2).

На неправильно введенный размер двумерного массива (3).

Результаты отладки программы в среде разработки Visual Studio 2017 представлены ниже:

1) способ инициализации вручную с консоли $k = 0$;

```

Введите кол-во строк: 3
Введите кол-во столбцов: 4
Задайте способ заполнения матрицы:
Введите 0, если хотите заполнить матрицу вручную, 1 - случайными числами: 0
Элемент (0; 0): 45
Элемент (0; 1): 84
Элемент (0; 2): 25
Элемент (0; 3): 17
Элемент (1; 0): 60
Элемент (1; 1): 72
Элемент (1; 2): 33
Элемент (1; 3): 1
Элемент (2; 0): 80
Элемент (2; 1): 97
Элемент (2; 2): 47
Элемент (2; 3): 29

  45  84  25  17
  60  72  33   1
  80  97  47  29

  1  1
  2  3

Количество седловых точек: 2
Для продолжения нажмите любую клавишу . . .
    
```

Рис. 7. Способ инициализации вручную с консоли

2) способ инициализации случайными числами $k = 1$;

```

Введите кол-во строк: 4
Введите кол-во столбцов: 5
Задайте способ заполнения матрицы:
Введите 0, если хотите заполнить матрицу вручную, 1 - случайными числами: 1

  74  61  52   2  53
  12  24  25   3  14
  88  27  94  47  88
  58  69  78  25  98

  1  2

Количество седловых точек: 1
Для продолжения нажмите любую клавишу . . .
    
```

Рис. 8. Способ инициализации случайными числами

3) неправильно введенный размер двумерного массива: $n = 0, m < 0$.

```

Введите кол-во строк: 0
Количество строк не может быть отрицательным или равным нулю. Введите кол-во строк: 5
Введите кол-во столбцов: -4
Количество столбцов не может быть отрицательным или равным нулю. Введите кол-во строк: 7
Задайте способ заполнения матрицы:
Введите 0, если хотите заполнить матрицу вручную, 1 - случайными числами: 1

66  38  76  21  0  24  7
20  58  28  17  1  10  15
29  36  49  42  60  63  25
20  98  57  55  80  14  29
29  61  27  66  94  3  51

Количество седловых точек: 0
Для продолжения нажмите любую клавишу . . .

```

Рис. 9. Проверка на неправильно введенные данные

Все седловые точки совпадают с представленным контрольным примером.

Результат: 1) – 2 седловые точки с индексами (1; 1) и (2; 3) (рис. 7).

Результат: 2) – 1 седловая точка с индексом (1; 2) (рис. 8).

Результат: 3) – проверка на неправильно введенные данные, седловые точки отсутствуют (рис. 9).

Список литературы

1. Петрова Н.К. Программирование на C++: конспект лекций [Электронный ресурс]. – Режим доступа: <https://lms.kgeu.ru/course/view.php?id=2594>.
2. Довбуш Г. Visual C++ на примерах / Г. Довбуш, А. Хомоненко. – М.: БХВ-Петербург, 2012. – 528 с.
3. Балена Ф. Современная практика программирования на Microsoft Visual Basic и Visual C# / Ф. Балена, Д. Димауро. – М.: Русская Редакция, 2015. – 640 с.
4. Лафоре Р. Объектно-ориентированное программирование в C++ / Р. Лафоре. – 4-е изд. – Питер, 2018. – 928 с.
5. Сидорина Т. Самоучитель Microsoft Visual Studio C++ и MFC / Т. Сидорина. – М.: БХВ-Петербург, 2014. – 848 с.

References

1. Petrova, N. K. Programmirovaniye na S++: konspekt lektsii.
2. Dovbush, G., & Khomonenko, A. (2012). Visual C++ na primerakh. M.: BKhV-Peterburg.
3. Balena, F., & Dimauro, D. (2015). Sovremennaya praktika programmirovaniya na Microsoft Visual Basic i Visual C#. M.: Russkaya Redaktsiya.
4. Lafore, R. (2018). Obektno-orientirovannoye programmirovaniye v S++, 928. Piter.
5. Sidorina, T. (2014). Samouchitel' Microsoft Visual Studio C++ i MFC. M.: BKhV-Peterburg.

Петрова Наталья Константиновна – канд. физ.-мат. наук, доцент кафедры информатики и информационно-управляющих систем, ФГБОУ ВО «Казанский государственный энергетический университет», Россия, Казань.

Petrova Natalya Konstantinovna – candidate of physical and mathematical sciences, associate professor of the informatics and information management systems department, FSBEI of HE «Kazan State Power Engineering University», Russia, Kazan.

Алимов Тимур Мансурович – студент ФГБОУ ВО «Казанский государственный энергетический университет», Россия, Казань.

Alimov Timur Mansurovich – a student of FSBEI of HE «Kazan State Power Engineering University», Russia, Kazan.

Алабдуллах Мохамад Белал – студент ФГБОУ ВО «Казанский государственный энергетический университет», Россия, Казань.

Alabdullah Mohamad Belal – a student of FSBEI of HE «Kazan State Power Engineering University», Russia, Kazan.

Альхузайи Ахмед Хушам Хусам – студент ФГБОУ ВО «Казанский государственный энергетический университет», Россия, Казань.

Alhuzai Ahmed Husham Hussam – a student of FSBEI of HE «Kazan State Power Engineering University», Russia, Kazan.

Каттави Раджих Зиад – студент ФГБОУ ВО «Казанский государственный энергетический университет», Россия, Казань.

Kattavi Radzhih Ziad – a student of FSBEI of HE «Kazan State Power Engineering University», Russia, Kazan.
