

Занятие 12 — Статические и динамические массивы. Рекурсивные функции. Строки. Дополнительные материалы

Программирование, численные методы и информатика

Александр Валерьевич Позднеев

Кафедра автоматизации научных исследований
Факультет вычислительной математики и кибернетики
Московский государственный университет имени М. В. Ломоносова
<http://ani.cmc.msu.ru/geol>

Осенний семестр 2011/2012



Содержание занятия

- 1 Статические массивы
 - Одномерные массивы
 - Многомерные массивы
- 2 Массивы как аргументы функций
- 3 Динамические массивы и указатели
 - Общие понятия
 - Выделение памяти
- 4 Рекурсивные функции
 - Сортировка слиянием
- 5 Строки
 - Символьный тип `char`
 - Строки в стиле языка C
 - Строки C++ `string`
- 6 Упражнения
 - Одномерные массивы
 - Многомерные массивы
 - Массивы как аргументы функций
 - Рекурсивные функции
 - Строки
- 7 Дополнительные материалы
 - Контейнер `vector`
 - Форматирование вывода
 - Шаблоны

Одномерные массивы

Одномерные массивы — пример

Заполнить массив десятью случайными числами из сегмента $[-0,5; 0,5]$ и вывести его на печать.

```
#include <iostream>
#include <cstdlib>
using namespace std;

int main() {
    const int N = 10;
    double a[N];

    for (int i = 0; i < N; ++i)
        a[i] = rand() / (double)RAND_MAX - 0.5;

    for (int i = 0; i < N; ++i)
        cout << i << "\t" << a[i] << endl;

    return 0;
}
```

Многомерные массивы

Многомерные массивы — пример

Заполнить массив размера 2×3 случайными числами из сегмента $[-0,5; 0,5]$ и вывести его на печать.

```
int main() {
    const int N = 2, M = 3;
    double a[N][M];

    for (int i = 0; i < N; ++i)
        for (int j = 0; j < M; ++j)
            a[i][j] = rand() / (double)RAND_MAX - 0.5;

    for (int i = 0; i < N; ++i) {
        for (int j = 0; j < M; ++j)
            cout << a[i][j] << "\t";
        cout << endl;
    }

    return 0;
}
```

Массивы как аргументы функций

Массивы как аргументы функций

```
void Init_Array(double a[], int N) {  
    for (int i = 0; i < N; ++i)  
        a[i] = rand() / (double)RAND_MAX - 0.5;  
}
```

```
void Print_Array(double a[], int N) {  
    for (int i = 0; i < N; ++i)  
        cout << i << "\t" << a[i] << endl;  
}
```

```
int main() {  
    const int N = 10;  
    double a[N];  
  
    Init_Array(a, N);  
    Print_Array(a, N);  
}
```


Динамические массивы и указатели

Динамические массивы и указатели — общие понятия

Динамическая память

- ▶ Размер массива не известен на момент компиляции
- ▶ Он задается на этапе выполнения
- ▶ *Динамическое распределение памяти*

Указатели

- ▶ Переменная типа *указатель* хранит адрес, по которому расположен массив или какая-то другая переменная
- ▶ имя_типа *идентификатор
- ▶ `int *a;`
- ▶ `double *x, *y;`
- ▶ Сразу после объявления значение указателя не определено

Динамические массивы и указатели — выделение памяти

Для создания массива в динамической памяти используется оператор `new` с указанием размера создаваемого массива:

```
double *a;  
a = new double[N];
```

Для уничтожения массива (освобождения занимаемой им памяти) служит оператор `delete []`:

```
delete [] a;
```

Динамические массивы и указатели — пример

```
int main() {
    int N;

    cout << "Enter N: ";
    cin >> N;

    double *a;
    a = new double[N];

    for (int i = 0; i < N; ++i)
        a[i] = rand() / (double)RAND_MAX - 0.5;

    for (int i = 0; i < N; ++i)
        cout << i << "\t" << a[i] << endl;

    delete [] a;
```

Рекурсивные функции

Рекурсивные функции

РЕКУРСИЯ (мат.) — см. рекурсия

- ▶ Функция внутри себя может обращаться к другой функции
- ▶ Но функция также может вызывать и саму себя

Рассмотрим определение факториала:

$$n! = \underbrace{1 \times 2 \times \dots \times (n-1)}_{(n-1)!} \times n = (n-1)! \times n$$

Таким образом,

$$n! = \begin{cases} 1, & n = 1 \\ (n-1)! \times n, & n > 1 \end{cases}$$

Например,

$$\begin{aligned} 3! &= (3-1)! \times 3 = 2! \times 3 = [(2-1)! \times 2] \times 3 = \\ &= [1! \times 2] \times 3 = [(1) \times 2] \times 3 \end{aligned}$$

Рекурсивные функции — пример

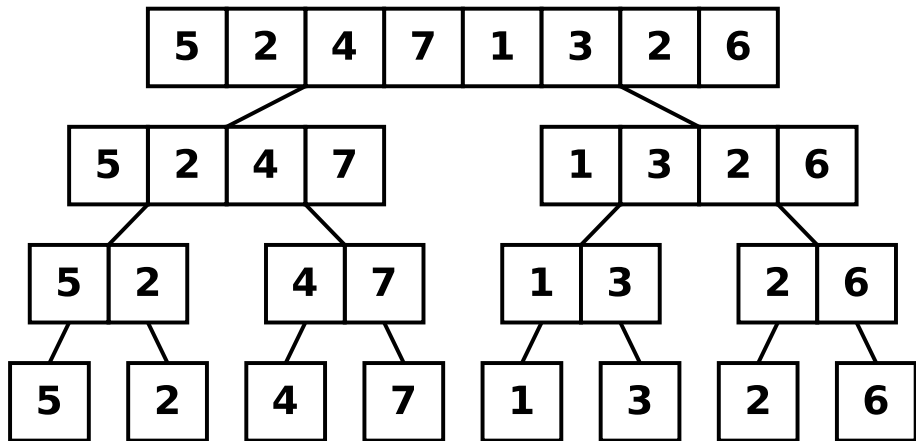
```
#include <iostream>
using namespace std;

int Factorial(int N)
{
    if (N == 1)
        return 1;
    else
        return N * Factorial(N-1);
}

int main() {
    const int n = 5;
    cout << n << "! = " << Factorial(n) << endl;
    return 0;
}
```

Сортировка слиянием

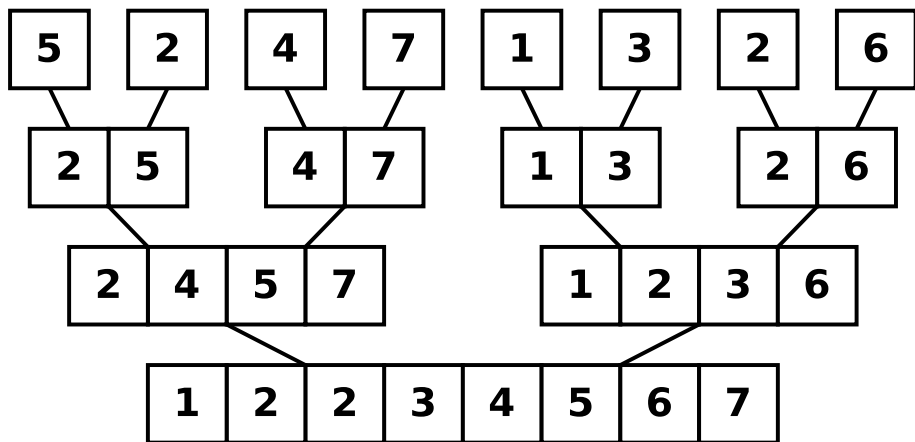
Разделение



Кормен Т., Лейзерсон Ч., Ривест Р. - Алгоритмы. Построение и анализ

Сортировка слиянием

Слияние



Кормен Т., Лейзерсон Ч., Ривест Р. - Алгоритмы. Построение и анализ

Сортировка слиянием — Слияние

```
void Merge(int A[], int p, int q, int r) {
    int nl = q - p + 1;
    int nr = r - (q + 1) + 1;

    int *L = new int[nl + 1];
    int *R = new int[nr + 1];

    for (int il = 0; il < nl; ++il)
        L[il] = A[p + il];
    for (int ir = 0; ir < nr; ++ir)
        R[ir] = A[q + 1 + ir];
    L[nl] = INT_MAX;
    R[nr] = INT_MAX;

    int il = 0, ir = 0;

    for (int k = p; k <= r; ++k)
        if (L[il] <= R[ir]) {
            A[k] = L[il];
            ++il;
        }
        else {
            A[k] = R[ir];
            ++ir;
        }
}
```

Сортировка слиянием — Разделение

```
void Merge_Sort(int A[], int p, int r)
{
    if (p < r)
    {
        int q = (p + r) / 2;
        Merge_Sort(A, p, q);
        Merge_Sort(A, q+1, r);
        Merge(A, p, q, r);
    }
}

int main()
{
    int A[] = { 5, 2, 4, 7, 1, 3, 2, 6 };
    int n = sizeof(A) / sizeof(A[0]);
    Print(A, n);

    Merge_Sort(A, 0, n-1);
    Print(A, n);

    return 0;
}
```

Строки

Символьный тип char

```
#include <iostream>
using namespace std;

int main() {
    char c = 'A';
    cout << c << endl;

    while (cin >> c) {
        cout << c;
    }

    cout << endl;

    return 0;
}
```

Строки в стиле языка C

```
#include <iostream>
using namespace std;

int main() {
    char str[] = "Hello, world";

    cout << str << endl;

    for (int i = 0; str[i] != '\0'; ++i)
        cout << str[i];

    cout << endl;

    return 0;
}
```

Строки C++ string

```
#include <iostream>
#include <string>
using namespace std;

int main() {
    string s1 = "Hello";
    string s2 = "World";
    string s = s1 + ", " + s2 + "!";

    cout << s << endl;

    for (int i = 0; i < s.length(); ++i)
        cout << s[i];
    cout << endl;

    return 0;
}
```

Ввод строк C++ string

```
#include <iostream>
#include <string>
using namespace std;

int main() {
    string s1, s2;

    cin >> s1; // "Lomonosov Moscow State University"
    cout << s1 << endl; // "Lomonosov"

    getline(cin, s2);
    cout << s2 << endl; // " Moscow State University"

    return 0;
}
```


Упражнения

Одномерные массивы — упражнения

1. Переставьте соседние элементы массива (0-й элемент поменять с 1-м, 2-й с 3-м и т. д.).
2. Определите, есть ли в данном массиве два соседних элемента с одинаковыми знаками и укажите их индексы,

Двумерные массивы — упражнения

1. Найти сумму элементов каждой строки матрицы размера 4×3 и записать эти значения в одномерный массив длины 4. Найти максимальную из этих сумм.

Массивы как аргументы функций — упражнения

1. Напишите функцию

```
int Search (int A[], int n, int x)
```

, которая находит в массиве `int A[n]` элемент, значение которого равно `x`. Функция возвращает индекс найденного элемента или `-1`, если такого элемента в массиве нет.

2. Напишите функцию

```
int CountMax (int A[], int n)
```

, которая подсчитывает, сколько раз в массиве встречается значение, являющееся максимальным. Функция должна выполнять однократный просмотр массива.

Рекурсивные функции — упражнения

1. Напишите рекурсивную функцию возведения в степень, воспользовавшись свойством: $a^n = a \cdot a^{n-1}$
2. Последовательность Фибоначчи определена следующим образом: $\varphi_0 = 1, \varphi_1 = 1, \varphi_n = \varphi_{n-1} + \varphi_{n-2}$ при $n > 1$. Начало ряда Фибоначчи выглядит следующим образом: 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, Напишите функцию `int Phi(int n)`, которая по данному натуральному n возвращает φ_n .
3. Напишите рекурсивную функцию возведения в степень, которая работала бы и для отрицательных значений n : $a^{-n} = 1/a^n$
4. Напишите функцию быстрого возведения в степень, которая пользовалась бы следующими свойствами: $a^n = (a^{n/2})^2$ при четном n , $a^n = a \cdot a^{n-1}$ при нечетном n
5. Для биномиальных коэффициентов (числа сочетаний из n по k) хорошо известна рекуррентная формула: $C_n^k = C_{n-1}^{k-1} + C_{n-1}^k$. Вычислите значение C_n^k , пользуясь этой формулой и учитывая, что $C_n^0 = C_n^n = 1$.

Строки — упражнения

1. Даны две строки. Определите, совпадают ли они, сравнив их посимвольно. Напишите для этого функцию `bool Compare(string S1, string S2)`
2. Напишите программу, которая по данному числу k от 1 до 120 печатает фразу «Мне k лет», меняя значение k на введенное число, а вместо слова «лет» печатая при необходимости слово «год» в правильном склонении
3. Дана строка, содержащая пробелы. Найдите, сколько в нем слов (слово — это последовательность непробельных символов, слова разделены одним пробелом, первый и последний символ строки — не пробел).
4. Дана строка, содержащая пробелы. Найдите в ней самое длинное слово, выведите на экран это слово и его длину.
5. Даны две строки. Определите, является ли первая строка подстрокой второй строки.

Дополнительные материалы

Дополнительные материалы — контейнер vector

```
#include <iostream>
#include <vector>
#include <algorithm>
#include <iterator> // ostream_iterator
using namespace std;

int main() {
    int N = 10;
    vector<double> a(N);

    for (int i = 0; i < N; ++i)
        a.at(i) = rand() / (double)RAND_MAX;

    sort(a.begin(), a.end());

    for (int i = 0; i < N; ++i)
        cout << i << "\t" << a[i] << endl;

    copy(a.begin(), a.end(), ostream_iterator<double>(cout, "\n"));

    return 0;
}
```


Дополнительные материалы — форматирование вывода

```
#include <iostream>
#include <cmath>
using namespace std;

int main() {
    const int N = 10;
    const double pi = 4.0 * atan(1.0);

    cout.width(5);
    cout << N << endl; // " 10"

    cout.width(10);
    cout.precision(3);
    cout << pi << endl; // " 3.14"

    cout.setf(ios::scientific | ios::showpos);
    cout << pi << endl; // "+3.142e+000"
```

Дополнительные материалы — шаблоны

```
#include <iostream>
using namespace std;

template <class T>
T Abs(T x) {
    if (x >= 0) return x;
    else return -x;
}

int main() {
    cout << "Abs(10):\t" << Abs(10) << endl;
    cout << "Abs(-10):\t" << Abs(-10) << endl;

    cout << "Abs(3.14):\t" << Abs(3.14) << endl;
    cout << "Abs(-3.14):\t" << Abs(-3.14) << endl;

    return 0;
}
```