

Практическая работа № 1

Алгоритмы поиска

Цель работы: Изучить способы применения алгоритмов поиска.

Оборудование: Windows 10, Visual Studio, Microsoft Word.

1. КРАТКАЯ ТЕОРИЯ

Когда поиск некоторого элемента необходимо осуществить в упорядоченной по возрастанию или убыванию последовательности, тогда применим алгоритм двоичного (бинарного) поиска. Метод использует стратегию «разделяй и властвуй», а именно: заданная последовательность делится на две равные части и поиск осуществляется в одной из этих частей, которая потом также делится надвое, и так до тех пор, пока обнаружится наличие искомого элемента или его отсутствие.

Использовать эту операцию, уменьшая каждый раз зону поиска вдвое, позволяют лишь исходя из того факта, что элементы последовательности заранее упорядочены. Найдя средний элемент (сделать это, зная число элементов массива, не составит труда), и сравнив его значение с искомым, можно уверенно сказать, где относительно среднего элемента находится искомый элемент.

Далее, будем полагать, что элементы массива располагаются в порядке возрастания, поскольку нет существенной разницы, как именно они упорядочены: по возрастанию или убыванию значение. Также обозначим границы зоны поиска через left и right – крайний левый и крайний правый элементы соответственно.

| | | | | | | | | | | | | | | |
|----------|------|----------------|---------|------|--|--|--|--|--|----------------------------|------|--------|--|--|
| | | | | | АиСД.09.03.02.110000.0000ПР | | | | | | | | | |
| Изм | Лист | № докум. | Подпись | Дата | | | | | | | | | | |
| Разраб. | | Куличенко Е.В. | | | Практическая работа №1 Алгоритмы поиска .. | | | | | Лит | Лист | Листов | | |
| Провер. | | Береза А.Н. | | | | | | | | | | 1 | | |
| | | | | | | | | | | ИСОuП(ф) ДГТУ ИСТ-Тb21 | | | | |
| Н.контр. | | | | | | | | | | | | | | |
| Утв. | | | | | | | | | | | | | | |

ИСХОДНЫЙ КОД

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

// функция с алгоритмом двоичного поиска
int Search_Binary(int arr[], int left, int right, int key)
{
    int midd = 0;
    while (1)
    {
        midd = (left + right) / 2;

        if (key < arr[midd])           // если искомое меньше значения в ячейке
            right = midd - 1;         // смещаем правую границу поиска
        else if (key > arr[midd])      // если искомое больше значения в ячейке
            left = midd + 1;          // смещаем левую границу поиска
        else                          // иначе (значения равны)
            return midd;              // функция возвращает индекс ячейки

        if (left > right)              // если границы сомкнулись
            return -1;
    }
}

int main()
{
    setlocale(LC_ALL, "rus");

    const int SIZE = 12;
    int array[SIZE] = {};
    int key = 0;
    int index = 0; // индекс ячейки с искомым значением

    for (int i = 0; i < SIZE; i++) // заполняем и показываем массив
    {
        array[i] = i + 1;
        cout << array[i] << " | ";
    }

    cout << "\n\nВведите любое число: ";
    cin >> key;

    index = Search_Binary(array, 0, SIZE, key);

    if (index >= 0)
        cout << "Указанное число находится в ячейке с индексом: " << index << "\n\n";
    else
        cout << "В массиве нет такого числа!\n\n";

    return 0;
}
```

| | | | | | | |
|-----|------|----------|---------|------|-----------------------------|------|
| | | | | | АиСД.09.03.02.110000.0000ПР | Лист |
| | | | | | | 2 |
| Изм | Лист | № докум. | Подпись | Дата | | |

```
C:\> Консоль отладки Microsoft Visual Studio

1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |

Введите любое число: 1
Указанное число находится в ячейке с индексом: 0

C:\Users\Lisa Cool\source\repos\ConsoleApplication3\Debug\ConsoleApplication3.exe (проц
м 0.
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис"
томатически закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно...
```