

Практическая работа №4

Тема «Алгоритмы поиска»

Цели работы:

1. Изучить алгоритмы поиска.

Оснащение:

Персональный компьютер под управлением ОС Windows 10.

Офисный пакет Microsoft Office 2016

Среда разработки Visual Studio 2019

1. Линейный поиск

Код программы

```
#include <iostream>
#include <ctime>
using namespace std;
int i, N;
//линейный поиск
int LineSearch(int A[], int key)
{
    for (i = 0; i < N; i++)
        if (A[i] == key) return i;
    return -1;
}
//главная функция
void main()
{
    setlocale(LC_ALL, "Rus");
    int key, A[1000];
    srand(time(NULL));
    cout << "Размер массива > "; cin >> N;
    cout << "Искомый элемент > "; cin >> key;
    cout << "Исходный массив: ";
    for (i = 0; i < N; i++)
    {
        A[i] = rand() % 10;
        cout << A[i] << " ";
    }
    if (LineSearch(A, key) == -1) cout << "\nЭлемент не найден";
    else cout << "\nНомер элемента: " << LineSearch(A, key) + 1;
    system("pause");
}
```

Результат

					<i>Au</i> СД.09.03.02.050000 ПР			
Изм.	Лист	№ докум.	Подпись	Дат				
Разраб.		Ермошина В.А			Практическая работа №4 «Алгоритмы поиска»	Лит.	Лист	Листов
Провер.		Берёза А. Н.					2	
Реценз						ИСОиП (филиал) ДГТУ в г.Шахты ИСТ-Тб11		
Н. Контр.								
Утверд.								

2. Бинарный поиск

Код программы

```
#include <stdio.h>
#include <malloc.h>
#include <conio.h>
#include <cstdlib>
#include <ctime>
#include <iostream>
int binarysearch(int a, int mass[], int n);
void InsertionSort(int n, int mass[]);
int main()
{
    setlocale(LC_ALL, "Rus");
    //ввод N
    int N, a;
    printf("Введите число элементов массива: ");
    scanf_s("%d", &N);
    //выделение памяти под массив
    int* mass;
    mass = (int*)malloc(N * sizeof(int));
    //ввод элементов массива
    srand(time(NULL));
    for (int i = 0; i < N; i++)
        mass[i] = rand();
    //сортировка вставками
    InsertionSort(N, mass);
    //вывод отсортированного массива на экран
    printf("Отсортированный массив:\n");
    for (int i = 0; i < N; i++)
        printf("%d ", mass[i]);
    printf("\n");
    //ввод искомого элемента
    printf("Введите искомый элемент: ");
    scanf_s("%d", &a);
    int k;
    //двоичный поиск
    k = binarysearch(a, mass, N);
    if (k != -1)
    {
        printf("Индекс элемента массива %d\n", k);
    }
    else
        printf("Искомый элемент не найден!\n");
    //освобождение памяти
    free(mass);
    _getch();
    return 0;
}
```

```

}
int binarysearch(int a, int mass[], int n)
{
    int low, high, middle;
    low = 0;
    high = n - 1;
    while (low <= high)
    {
        middle = (low + high) / 2;
        if (a < mass[middle])
            high = middle - 1;
        else if (a > mass[middle])
            low = middle + 1;
        else
            return middle;
    }
    return -1;
}
//сортировка вставками
void InsertionSort(int n, int mass[])
{
    int newElement, location;

    for (int i = 1; i < n; i++)
    {
        newElement = mass[i];
        location = i - 1;
        while (location >= 0 && mass[location] > newElement)
        {
            mass[location + 1] = mass[location];
            location = location - 1;
        }
        mass[location + 1] = newElement;
    }
}

```

Результат



3. Числа Фибоначчи

Код программы

```

#include <iostream>
int fibonacci(int number)
{
    if (number == 0)
        return 0; // базовый случай (условие завершения)
    if (number == 1)
        return 1; // базовый случай (условие завершения)

```

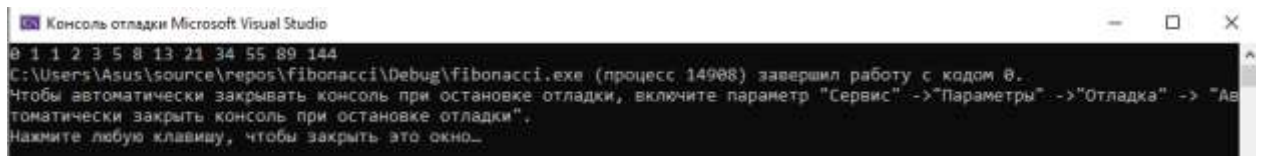
```

        return fibonacci(number - 1) + fibonacci(number - 2);
    }
    // Выводим первые 13 чисел Фибоначчи
    int main()
    {
        for (int count = 0; count < 13; ++count)
            std::cout << fibonacci(count) << " ";

        return 0;
    }

```

Результат



4. Интерполяционный поиск

Код программы

```

#include <iostream>
#include <cstdlib>
#include <ctime>
using namespace std;
const int N = 20;
int InterpolSearch(int A[], int key)
{
    int mid, left = 0, right = N - 1;
    while (A[left] <= key && A[right] >= key)
    {
        mid = left + ((key - A[left]) * (right - left)) / (A[right] - A[left]);
        if (A[mid] < key) left = mid + 1;
        else if (A[mid] > key) right = mid - 1;
        else return mid;
    }
    if (A[left] == key) return left;
    else return -1;
}
//главная функция
int main()
{
    setlocale(LC_ALL, "Rus");
    int i, key;
    srand(time(NULL));
    int A[N] = {};
    for (int i = 0; i < N; i++)
    {
        A[i] = rand();
    }
    cout << "Искомый элемент > "; cin >> key; //ввод ключа
    cout << "Исходный массив: ";
    for (i = 0; i < N; i++) cout << A[i] << " "; //вывод массива
    if (InterpolSearch(A, key) == -1) cout << "\nЭлемент не найден";
}

```

```

else cout << "\nНомер элемента: " << InterpolSearch(A, key) + 1;
system("pause>>void");
}

```

Результат



Вывод: В ходе выполнения лабораторной работы изучила алгоритмы поиска и написала программы, реализующие линейный поиск, бинарный поиск, вычисление чисел Фибоначчи и интерполяционный поиск.

					<i>АиСД.09.03.02.050000 ПР</i>	Лист
						6
Изм.	Лист	№ докум.	Подпись	Дата		