Практическая работа №4 Тема «Алгоритмы поиска»

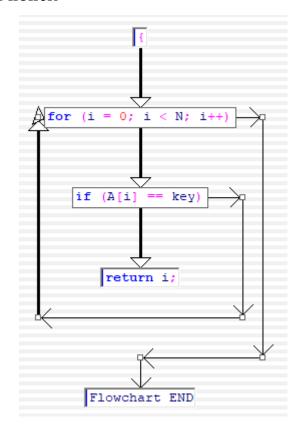
Цели работы:

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;
```

					АиСД.09.03.02.05	50000	ПР	
Изм.	Лист	№ докум.	Подпись	Дат				
Разра	аб.	Ермошина В.А			Практическая работа №4	Лит.	Лист	Листов
Провер.		Берёза А. Н.			«Алгоритмы поиска»		2	
Реценз					1	ИСОиП (филиал) ДГТУ в		ил) ДГТУ в
Н. Контр.						г.Шахты		
Утверд.						ИСТ-Ть11		o11

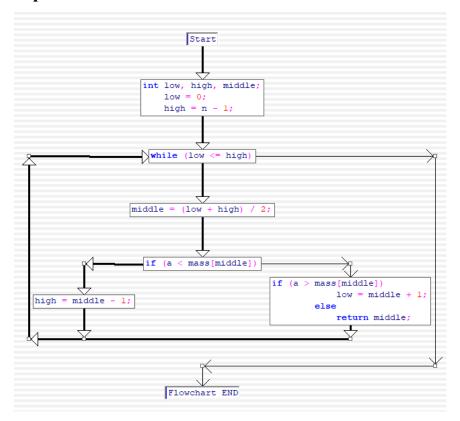
Результат

```
M C\Users\Asus\source\repos\LineSearch\Debug\LineSearch.exe — ☐ X

Размер массива > 50
Искомый элемент > 5
Искомый элемент > 5
Искофий массив: 0 5 2 3 6 4 1 1 2 9 8 8 1 3 2 9 1 0 3 8 2 0 3 7 4 3 6 1 4 8 7 3 1 7 2 9 0 3 4 1 9 5 1 0 2 9 2 2 1 4

Номер элемента: 2
```

2. Бинарный поиск



Код программы

#include <stdio.h>

					l
					l
Изм.	Лист	№ докум.	Подпись	Дата	

```
#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;
```

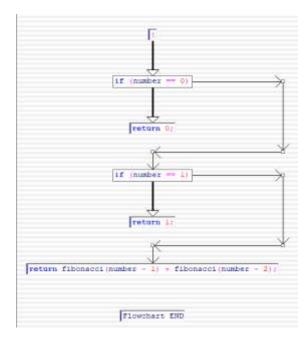
Изм.	Лист	№ докум.	Подпись	Дата
		J		<i>/</i> ,

```
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 \geq 0 \&\& mass[location] > newElement)
       mass[location + 1] = mass[location];
       location = location - 1;
     mass[location + 1] = newElement;
}
```

Результат

```
© C\(Users\Assus\source\repos\binary\Debug\binary\ese — □ ×
Введите число алементов массива: 25
Отсортированный массива: 25
1686 2946 3489 5153 6789 7468 12916 13847 13468 13551 13579 14782 17854 19818 21281 21751 22991 23826 24274 27191 27581 28647 38848 38758 32586
Введите искомый элемент: 2946
Индекс алемента массива 1
```

3. Числа Фибоначчи



11	77	Ma 2	77 - 3	77
ИЗМ.	Лист	№ докум.	Подпись	дата

```
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

```
тов Консоль отладки Microsoft Visual Studio — □ ×

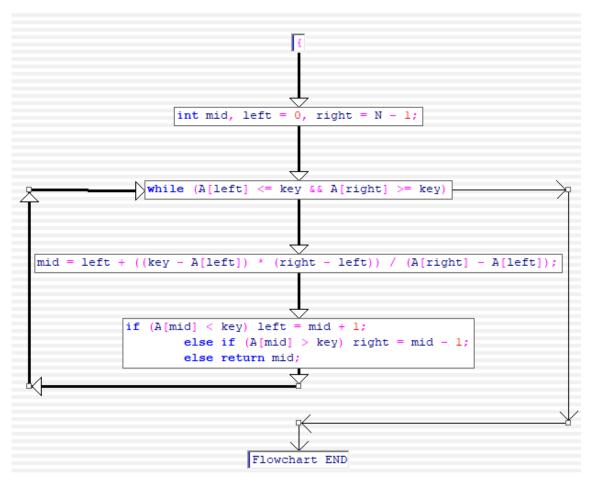
8 1 1 2 3 5 8 13 21 34 55 89 144

С:\Users\Asus\source\repos\fibonacci\Debug\fibonacci.exe (процесс 14908) завершил работу с кодом д.

Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" ->"Параметры" ->"Отладка" -> "Автоматически закрыть консоль при остановке отладки".

Нажмите любую клавишу, чтобы закрыть это окно…
```

4. Интерполяционный поиск



Изм.	Лист	№ докум.	Подпись	Дата
V13M.	Jiucm	л⊻ оокум.	1100пись	дит

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] \le 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

C/\Users\Asus\source\repos\poisk\Debug\poisk.exe П комый элемент > 0 ходный массию: 22553 21010 32532 31734 15242 21589 21715 15979 7516 22158 27991 21652 1024 20096 3111 17774 9685 30491 9798 128

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

Изм.	Лист	№ докум.	Подпись	Дата