

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Чувашский государственный университет им. И.Н.Ульянова».
Кафедра вычислительной техники.
Предмет: Объектно-ориентированное программирование

Лабораторная работа №4. Исследование методов поиска

Вариант 4.

Выполнил: Васильев Егор Юрьевич
студент группы ИВТ-41-20
Проверил: Павлов.Л.А

Цель работы: Ознакомление с методами быстрого поиска; получение навыков программирования задач быстрого поиска.

1. Последовательный поиск в упорядоченной таблице.

```
Ссылка: 1
public static int SequentialSearch<Type>(this Array array, Type key)
{
    if (!array.IsSameTypeAsArrayElement(key))
    {
        throw new ArgumentException("Wrong types");
    }

    var index = 0;

    foreach (var item in array)
    {
        if (Compare(item, key) == 0)
        {
            return index;
        }

        index++;
    }

    return -1;
}
```

2. Обычный бинарный поиск.

```
Ссылка: 1
public static int BinarySearch<Type>(this Array array, Type key)
{
    if (!array.IsSameTypeAsArrayElement(key))
    {
        throw new ArgumentException("Wrong types");
    }

    int middleIndex,
        leftIndex = 0,
        rightIndex = array.Length - 1;

    while (leftIndex <= rightIndex)
    {
        middleIndex = (leftIndex + rightIndex) / 2;

        if (Compare(array.GetValue(middleIndex), key) == 0)
        {
            return middleIndex;
        }

        if (Compare(array.GetValue(middleIndex), key) > 0)
        {
            rightIndex = middleIndex - 1;
        }
        else
        {
            leftIndex = middleIndex + 1;
        }
    }

    return -1;
}
```

6. Поиск Фибоначчи.

Ссылка: 1

```
public static int FibonacciSearch<Type>(this Array array, Type key)
{
    if (!array.IsSameTypeAsArrayElement(key))
    {
        throw new ArgumentException("Wrong types");
    }

    int fibNum1 = 1, fibNum2 = 0;
    var fibSum = fibNum1 + fibNum2;

    while (fibSum < array.Length)
    {
        fibNum2 = fibNum1;
        fibNum1 = fibSum;
        fibSum = fibNum1 + fibNum2;
    }

    var offset = -1;

    while (fibSum > 1)
    {
        var index = Math.Min(offset + fibNum2, array.Length - 1);

        if (Compare(array.GetValue(index), key) < 0)
        {
            fibSum = fibNum1;
            fibNum1 = fibNum2;
            fibNum2 = fibSum - fibNum1;
            offset = index;
        }

        else if (Compare(array.GetValue(index), key) > 0)
        {
            fibSum = fibNum2;
            fibNum1 = fibNum1 - fibNum2;
            fibNum2 = fibSum - fibNum1;
        }

        else
        {
            return index;
        }
    }

    return -1;
}
```

7. Интерполяционный поиск. (c++)

```
long int interpolSearch(long int* array, long int size, long int key) {
    long int mid, left = 0, right = size - 1;
    long int count = 0;

    while (array[left] <= key && array[right] >= key) {
        count++;
        mid = left + ((key - array[left]) * (right - left)) / (array[right] - array[left]);
        if (array[mid] < key) {
            left = mid + 1;
        }
        else if (array[mid] > key) {
            right = mid - 1;
        }
        else return count;
    }
    return count;
}
```

9. Поиск в красно-черных деревьях. (c++)

```
int find(int const data) {
    RedBlackTree* father = root;
    int count = 0;
    while (father) {
        count++;
        if (data < father->data) {
            father = father->left;
        }
        else if (data > father->data) {
            father = father->right;
        }
        else {
            break;
        }
    }

    return count;
}
```

11. Поиск с хешированием.

```
Ссылка: 1
public string Search(string key)
{
    var hash = GetHash(key);

    if (!_items.ContainsKey(hash))
    {
        return null;
    }
    var hashTableItem = _items[hash];

    if (hashTableItem != null)
    {
        var item = hashTableItem.SingleOrDefault(i => i.Key == key);

        if (item != null)
        {
            return item.Value.ToString();
        }
    }

    return null;
}
```

Вывод: ознакомился с методами быстрого поиска, получил навыки программирования задач быстрого поиска.