

Документация 1 бригады

Алгоритм 2: «Поиск элемента в одномерном массиве»

1.Описание входных и выходных данных

Данные	Наименование	Тип
Входные	Количество элементов(N)	Целое
	Выбор заполнения массива(C)	Целое
	Выбор поиска(C)	Целое
	Элементы массива(x[i])	Целое
	Массив(x[n])	Целое
Выходные	Номер элемента(i)	Целое
	Затраченное время(time)	Целое

2.Пример работы

1)Массив заполняется нами, выбор поиска – обычный (элемент присутствует в массиве)

```
введите количество элементов в массиве:
5
если вы хотите заполнить массив сами, введите 0, если хотите заполнить его автоматически введите 1:
0
1
2
3
4
5
введите число, которое хотите найти (от 0 до 100000)
3
выберете способ сортировка:
1)обычный поиск элемента в массиве напрямую
2)поиск элемента в массиве после сортировки
3)поиск элемента посредством бинарного поиска
1
искомый элемент найден под номером 2 считая с 0
затраченное на поиск время равно 0мс
если хотите повторить тест введите 1, в ином случае введите 0:
```

Результат: элемент найден, затрачено 0 мс

2)Массив заполняется нами, выбор поиска – обычный (элемент отсутствует в массиве)

```
введите количество элементов в массиве:
5
если вы хотите заполнить массив сами, введите 0, если хотите заполнить его автоматически введите 1:
0
1
2
3
4
5
введите число, которое хотите найти (от 0 до 100000)
6
выберете способ сортировки:
1)обычный поиск элемента в массиве напрямую
2)поиск элемента в массиве после сортировки
3)поиск элемента посредством бинарного поиска
1
искомого элемента в массиве нет
затраченное на поиск время равно 0мс
если хотите повторить тест введите 1, в ином случае введите 0:
```

Результат: выводится сообщение об отсутствии искомого элемента, затрачено 0 мс

3)Массив заполняется автоматически, выбор поиска – обычный

```
введите количество элементов в массиве:
5
если вы хотите заполнить массив сами, введите 0, если хотите заполнить его автоматически введите 1:
1
введите число, которое хотите найти (от 0 до 100000)
1111111111111111
выберете способ сортировки:
1)обычный поиск элемента в массиве напрямую
2)поиск элемента в массиве после сортировки
3)поиск элемента посредством бинарного поиска
1
искомого элемента в массиве нет
затраченное на поиск время равно 0мс
если хотите повторить тест введите 1, в ином случае введите 0:
```

Результат: выводится сообщение об отсутствии искомого элемента, затрачено 0 мс

4) Массив заполняется нами, выбор поиска - поиск элемента в массиве после сортировки (элемент присутствует)

```
введите количество элементов в массиве:
5
если вы хотите заполнить массив сами, введите 0, если хотите заполнить его автоматически введите 1:
0
1
2
3
4
5
введите число, которое хотите найти (от 0 до 100000)
4
выберете способ сортировка:
1)обычный поиск элемента в массиве напрямую
2)поиск элемента в массиве после сортировки
3)поиск элемента посредством бинарного поиска
2
искомый элемент найден
затраченное на поиск время равно 0мс
если хотите повторить тест введите 1, в ином случае введите 0:
```

Результат: выводится сообщение о наличии искомого элемента в массиве, затрачено 0мс

5)Массив заполняется нами, выбор поиска – поиск элемента в массиве после сортировки (элемент отсутствует)

```
введите количество элементов в массиве:
5
если вы хотите заполнить массив сами, введите 0, если хотите заполнить его автоматически введите 1:
0
1
2
3
4
5
введите число, которое хотите найти (от 0 до 100000)
6
выберите способ сортировки:
1)обычный поиск элемента в массиве напрямую
2)поиск элемента в массиве после сортировки
3)поиск элемента посредством бинарного поиска
2
искомого элемента в массиве нет
затраченное на поиск время равно 0мс
если хотите повторить тест введите 1, в ином случае введите 0:
```

Результат: выводится сообщение об отсутствии искомого элемента в массиве, затрачено 0мс

6)Массив заполняется автоматически, выбор поиска – поиск элемента в массиве после сортировки (элемент отсутствует)

```
введите количество элементов в массиве:
5
если вы хотите заполнить массив сами, введите 0, если хотите заполнить его автоматически введите 1:
1
введите число, которое хотите найти (от 0 до 100000)
6
выберите способ сортировки:
1)обычный поиск элемента в массиве напрямую
2)поиск элемента в массиве после сортировки
3)поиск элемента посредством бинарного поиска
2
искомого элемента в массиве нет
затраченное на поиск время равно 0мс
если хотите повторить тест введите 1, в ином случае введите 0:
```

7)Массив заполняется нами, выбор поиска – поиск посредством бинарного поиска (элемент присутствует в массиве)

```
введите количество элементов в массиве:
5
если вы хотите заполнить массив сами, введите 0, если хотите заполнить его автоматически введите 1:
0
1
2
3
4
5
введите число, которое хотите найти (от 0 до 100000)
4
выберите способ сортировки:
1)обычный поиск элемента в массиве напрямую
2)поиск элемента в массиве после сортировки
3)поиск элемента посредством бинарного поиска
3
искомый элемент найден
затраченное на поиск время равно 1мс
если хотите повторить тест введите 1, в ином случае введите 0:
```

8)Массив заполняется нами, выбор поиска - поиск посредством бинарного поиска (элемент отсутствует в массиве)

```
введите количество элементов в массиве:
5
если вы хотите заполнить массив сами, введите 0, если хотите заполнить его автоматически введите 1:
0
1
2
3
5
4
введите число, которое хотите найти (от 0 до 100000)
4
выберите способ сортировка:
1)обычный поиск элемента в массиве напрямую
2)поиск элемента в массиве после сортировки
3)поиск элемента посредством бинарного поиска
3
искомый элемент найден
затраченное на поиск время равно 0мс
если хотите повторить тест введите 1, в ином случае введите 0:
```

9)Массив заполняется автоматически, выбор поиска – поиск посредством бинарного поиска

```
введите количество элементов в массиве:
5
если вы хотите заполнить массив сами, введите 0, если хотите заполнить его автоматически введите 1:
1
введите число, которое хотите найти (от 0 до 100000)
4
выберите способ сортировка:
1)обычный поиск элемента в массиве напрямую
2)поиск элемента в массиве после сортировки
3)поиск элемента посредством бинарного поиска
3
искомого элемента в массиве нет
затраченное на поиск время равно 0мс
если хотите повторить тест введите 1, в ином случае введите 0:
```

3.Краткое описание логики работы

Для программы подается целое число(N), которое отвечает за количество элементов массива. Элементы массива могут быть сгенерированы пользователем, либо могут быть заполнены автоматически.

Пользователь вводит число, которое хочет найти в массиве, после чего предлагается 3 способа поиска:

- 1-обычный поиск элемента в массиве напрямую
- 2-поиск элемента в массиве после сортировки
- 3-поиск элемента посредством бинарного поиска

После выбора способа поиска, программа начинает поиск, и выдает сообщение о наличии/отсутствии искомого элемента в массиве и затраченное время на поиск.

4. Аналитический анализ сложности

1-обычный поиск элемента в массиве напрямую

```
case 1:
{
    unsigned long long start_time = clock();
    while (i != n - 1 && o != x[i]) LN
        i++;
    unsigned long long end_time = clock();
    unsigned long long search_time = end_time - start_time;
    if (o == x[i]) 1
        cout << "искомый элемент найден под номером " << i << " считая с 0\n";
    else
        cout << "искомого элемента в массиве нет\n";
    cout << "затраченное на поиск время равно " << search_time << "мс\n";
    break;
}
```

3N

$O(3N)$

2-поиск элемента в массиве после сортировки

```
case 2:
{
    unsigned long long start_time = clock();
    sort(x.begin(), x.end()); N
    while (i != n - 1 and o != x[i]) 2N
        i++;
    unsigned long long end_time = clock();
    unsigned long long search_time = end_time - start_time;
    if (o == x[i]) 1
        cout << "искомый элемент найден\n";
    else
        cout << "искомого элемента в массиве нет\n";
    cout << "затраченное на поиск время равно " << search_time << "мс\n";
    break;
}
```

4N

$O(4N)$

3-поиск элемента посредством бинарного поиска

```
case 3:
{
    unsigned long long start_time = clock();
    sort(x.begin(), x.end()); N
    auto it = lower_bound(x.begin(), x.end(), o); 2N
    unsigned long long end_time = clock();
    unsigned long long search_time = end_time - start_time;
    if (it != x.end()) N
        cout << "искомый элемент найден\n";
    else
        cout << "искомого элемента в массиве нет\n";
    cout << "затраченное на поиск время равно " << search_time << "мс\n";
    break;
}
```

4N

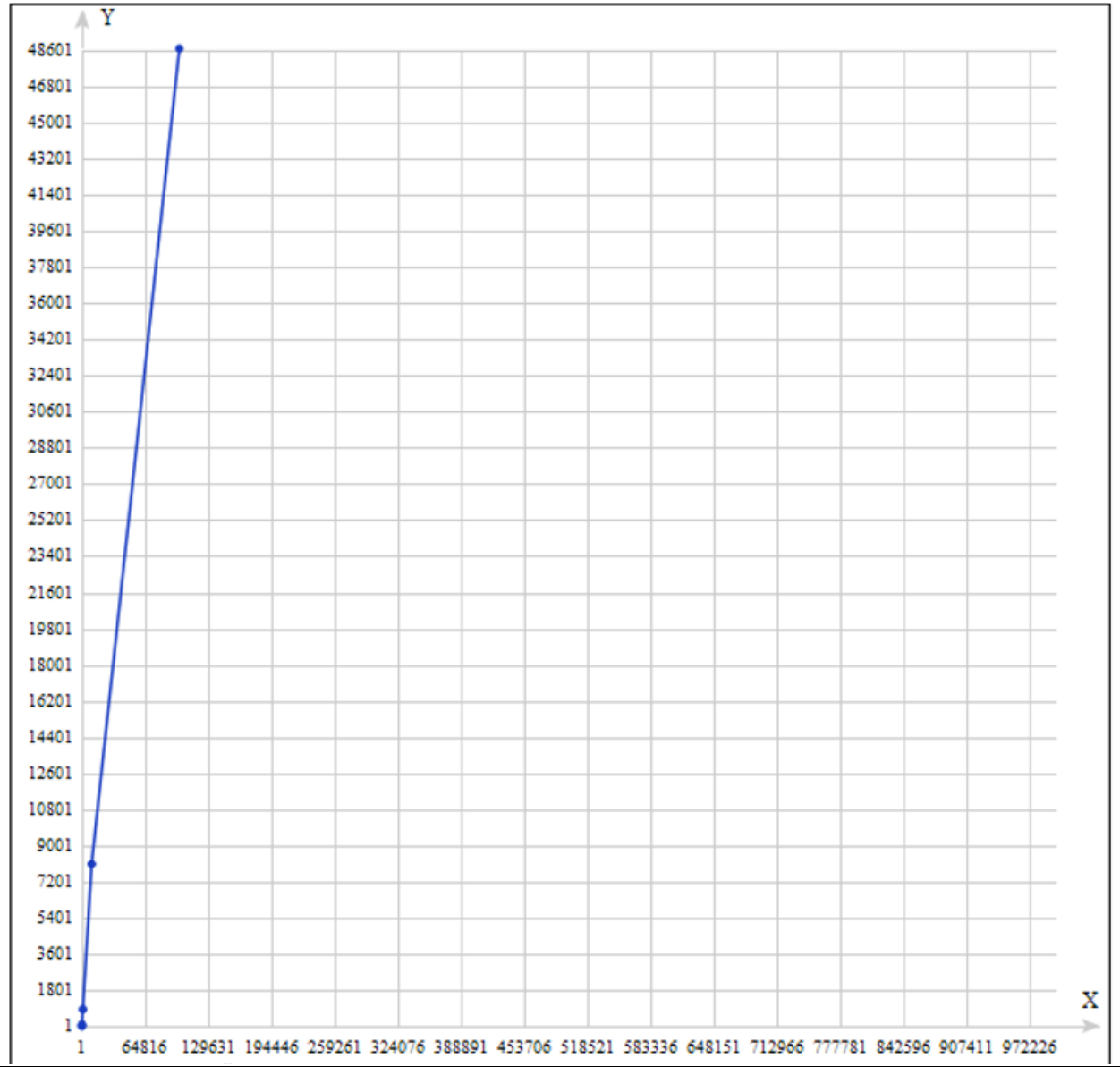
$O(4N)$

В ходе исследования, выяснилось, что наиболее сложным для вычислений является способ поиска после сортировки и способ посредством бинарного поиска

5.Экспериментальное исследование вычислительной сложности решения

У второго способа оказался наиболее наглядный график, потому ниже приведен именно график этого способа.

График 2 способа:



N	10 ³	10 ⁴	10 ⁵	10 ⁶	10 ⁷	10 ⁸
t, mc	1	7	85	836	8068	48684

1 способ:

N	10 ³	10 ⁴	10 ⁵	10 ⁶	10 ⁷	10 ⁸
t,mc	0	0	1	8	81	496

3 способ:

N	10 ³	10 ⁴	10 ⁵	10 ⁶	10 ⁷	10 ⁸
t,mc	0	6	83	820	8037	47183

Вывод :В ходе исследования алгоритма мы выяснили, что при работе с неотсортированным массивом способ обычного перебора эффективнее, но с отсортированным массивом наоборот лучше использовать бинарный поиск.