

Алгоритмы поиска. Оценка сложности.

Алгоритмы и структуры данных

Мулюгин Николай

10.09.2022

Мотивация

- Зачем нам что то искать?

Мотивация

- Зачем нам что то искать?
- Зачем нам что то быстро искать?

Мотивация

- Зачем нам что то искать?
- Зачем нам что то быстро искать?
- Где нам искать?

Массив

0	1	2	3	4	5	6	7	8	9
5	7	2	3	20	11	17	3	4	8

Массив

0	1	2	3	4	5	6	7	8	9
5	7	2	3	20	11	17	3	4	8

Массив - структура данных, хранящая набор значений в памяти непосредственно друг за другом

Массив

0	1	2	3	4	5	6	7	8	9
5	7	2	3	20	11	17	3	4	8

Массив - структура данных, хранящая набор значений в памяти непосредственно друг за другом

Индекс массива - номер элемента в массиве в памяти непосред

Массив

0	1	2	3	4	5	6	7	8	9
5	7	2	3	20	11	17	3	4	8

Массив - структура данных, хранящая набор значений в памяти непосредственно друг за другом

Индекс массива - номер элемента в массиве в памяти непосред

Вопрос: сложно ли получить элемент по индексу?

Массив

0	1	2	3	4	5	6	7	8	9
5	7	2	3	20	11	17	3	4	8

Массив - структура данных, хранящая набор значений в памяти непосредственно друг за другом

Индекс массива - номер элемента в массиве в памяти непосред

Вопрос: сложно ли получить элемент по индексу?

Ответ: $O(1)$

Массив

0	1	2	3	4	5	6	7	8	9
5	7	2	3	20	11	17	3	4	8

Массивы в C++

```
1 //create array with 4 int type elements
2 int[4] arr = {5,7,2,3};
3 //read value at 3
4 int three = arr[3]; //three = 3
5 //write to 3
6 arr[3] = 4; // [5 7 2 4]
```

Линейный поиск

```
1 int LinearSearch1( int* arr, int n, int x)
2 {
3     int answer = -1;
4     for( int i = 0; i < n; i++ )
5         if(arr[i] == x)
6             answer = i;
7     return answer;
8 }
```

$f(n) =$

Линейный поиск

```
1 int LinearSearch1( int* arr, int n, int x)
2 {
3     int answer = -1;
4     for( int i = 0; i < n; i++ )
5         if(arr[i] == x)
6             answer = i;
7     return answer;
8 }
```

$$f(n) = (C_{=} + C_{<} + C_{+} + C_{==} + C_{=}) \cdot n = C \cdot n$$

Линейный поиск

```
1 int LinearSearch1( int* arr, int n, int x)
2 {
3     int answer = -1;
4     for( int i = 0; i < n; i++ )
5         if(arr[i] == x)
6             answer = i;
7     return answer;
8 }
```

$$f(n) = (C_{=} + C_{<} + C_{+} + C_{==} + C_{=}) \cdot n = C \cdot n$$

$$f(n) = \Theta(n)$$

Линейный поиск

```
1 int LinearSearch2( int* arr, int n, int x)
2 {
3     for( int i = 0; i < n; i++ )
4         if(arr[i] == x)
5             return i;
6     return -1;
7 }
```

Бинарный поиск

```
1 //sorted array 2 5 6 15 21 23 25
2 int BinarySearch(int* arr, int n, int key) {
3     int low = 0;
4     int high = n - 1;
5
6     while (low <= high) {
7         int mid = (low + high) / 2;
8         int midVal = arr[mid];
9
10        if (midVal < key)
11            low = mid + 1;
12        else if (midVal > key)
13            high = mid - 1;
14        else
15            return mid; // key found
16    }
17    return -(low + 1); // key not found.
18 }
```

Сложность бинарного поиска $f(n) = ?$

Сложность бинарного поиска $f(n) = ?$

```
1 int[7] arr = { 2,5,6,15,21,23,25 };  
2 int* arr_ptr = &arr[0];  
3 int n = 7;  
4 int ans_i = BinarySearch( arr_ptr, n, 2 );
```

0	1	2	3	4	5	6
2	5	6	15	21	23	25

[

Сложность бинарного поиска $f(n) = ?$

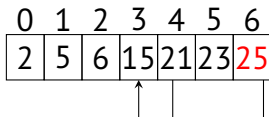
```
1 int[7] arr = { 2,5,6,15,21,23,25 };  
2 int* arr_ptr = &arr[0];  
3 int n = 7;  
4 int ans_i = BinarySearch( arr_ptr, n, 2 );
```

0	1	2	3	4	5	6
2	5	6	15	21	23	25
[↑]		

Сложность бинарного поиска $f(n) = ?$

```
1 int[7] arr = { 2,5,6,15,21,23,25 };  
2 int* arr_ptr = &arr[0];  
3 int n = 7;  
4 int ans_i = BinarySearch( arr_ptr, n, 2 );
```

0	1	2	3	4	5	6
2	5	6	15	21	23	25



Сложность бинарного поиска $f(n) = ?$

```
1 int[7] arr = { 2,5,6,15,21,23,25 };  
2 int* arr_ptr = &arr[0];  
3 int n = 7;  
4 int ans_i = BinarySearch( arr_ptr, n, 2 );
```


0	1	2	3	4	5	6
2	5	6	15	21	23	25

[↑]

Сложность бинарного поиска $f(n) = ?$

```
1 int[7] arr = { 2,5,6,15,21,23,25 };  
2 int* arr_ptr = &arr[0];  
3 int n = 7;  
4 int ans_i = BinarySearch( arr_ptr, n, 2 );
```

0	1	2	3	4	5	6
2	5	6	15	21	23	25



Сложность бинарного поиска

Сложность бинарного поиска $f(n) = ?$

Сложность бинарного поиска

Сложность бинарного поиска $f(n) = ?$

0 1 2 3 4 5 6 $n \left(\frac{n}{2^1} \right)$

Сложность бинарного поиска

Сложность бинарного поиска $f(n) = ?$

0 1 2 3 4 5 6 $n \left(\frac{n}{2^1} \right)$

0 1 2 3 4 5 6 $\frac{n}{2} \left(\frac{n}{2^1} \right) \dots$

Сложность бинарного поиска

Сложность бинарного поиска $f(n) = ?$

0 1 2 3 4 5 6 $n \left(\frac{n}{2^1} \right)$

0 1 2 3 4 5 6 $\frac{n}{2} \left(\frac{n}{2^1} \right) \dots$

0 1 2 3 4 5 6 ... $\frac{n}{4} \left(\frac{n}{2^2} \right) \dots$

0 $\frac{n}{2^y}$

Сложность бинарного поиска

Сложность бинарного поиска $f(n) = ?$

0 1 2 3 4 5 6 $n \left(\frac{n}{2^1} \right)$

0 1 2 3 4 5 6 $\frac{n}{2} \left(\frac{n}{2^1} \right) \dots$

0 1 2 3 4 5 6 ... $\frac{n}{4} \left(\frac{n}{2^2} \right) \dots$

0 $\frac{n}{2^y}$

$$\frac{n}{2^y} = 1;$$

Сложность бинарного поиска

Сложность бинарного поиска $f(n) = ?$

0 1 2 3 4 5 6 $n \left(\frac{n}{2^1} \right)$

0 1 2 3 4 5 6 $\frac{n}{2} \left(\frac{n}{2^1} \right) \dots$

0 1 2 3 4 5 6 ... $\frac{n}{4} \left(\frac{n}{2^2} \right) \dots$

0 $\frac{n}{2^y}$

$\frac{n}{2^y} = 1; n = 2^y;$

Сложность бинарного поиска

Сложность бинарного поиска $f(n) = ?$

0 1 2 3 4 5 6 $n \left(\frac{n}{2^1} \right)$

0 1 2 3 4 5 6 $\frac{n}{2} \left(\frac{n}{2^1} \right) \dots$

0 1 2 3 4 5 6 ... $\frac{n}{4} \left(\frac{n}{2^2} \right) \dots$

0 $\frac{n}{2^y}$

$\frac{n}{2^y} = 1; n = 2^y; y = ?;$

Сложность бинарного поиска

Сложность бинарного поиска $f(n) = ?$

0 1 2 3 4 5 6 $n \left(\frac{n}{2^1} \right)$

0 1 2 3 4 5 6 $\frac{n}{2} \left(\frac{n}{2^1} \right) \dots$

0 1 2 3 4 5 6 ... $\frac{n}{4} \left(\frac{n}{2^2} \right) \dots$

0 $\frac{n}{2^y}$

$\frac{n}{2^y} = 1; n = 2^y; y = ?; \boxed{y = \log_2 n}$

Сложность бинарного поиска

Сложность бинарного поиска $f(n) = ?$

0 1 2 3 4 5 6 $n \left(\frac{n}{2^1} \right)$

0 1 2 3 4 5 6 $\frac{n}{2} \left(\frac{n}{2^1} \right) \dots$

0 1 2 3 4 5 6 ... $\frac{n}{4} \left(\frac{n}{2^2} \right) \dots$

0 $\frac{n}{2^y}$

$\frac{n}{2^y} = 1; n = 2^y; y = ?; \boxed{y = \log_2 n}$

$\boxed{f(n) = \Theta(\log_2 n)}$

Домашнее задание

- Написать программу реализующую алгоритм Эратосфена.
- Оценить сложность - написать в программе комментариями мысли и результат.
- Сделать pull request в папку lecture_01/homework/ с файлом решения.