

Алгоритмы и структуры данных-1

SET 1. Задача A5.

Осень 2024. Клычков М. Д.

```
1 std::pair<int, int> FindInSortedMatrix(std::vector<std::vector<int>>& matrix, int key) {
2     int first_row{0};
3     int last_row = matrix.size() - 1;
4     int first_col{0};
5     int last_col = matrix[0].size() - 1;
6
7     while (first_row <= last_row && first_col <= last_col) {
8         if (matrix[first_row][first_col] == key) {
9             return {first_row, first_col};
10        }
11        if (!(matrix[last_row][first_col] <= key &&
12              key <= matrix[last_row][last_col])) {
13            --last_row;
14        } else if (!(matrix[last_row][first_col] <= key &&
15                    key <= matrix[first_row][first_col])) {
16            ++first_col;
17        } else if (!(matrix[first_row][first_col] <= key &&
18                    key <= matrix[first_row][last_col])) {
19            ++first_row;
20        } else {
21            --last_col;
22        }
23    }
24    return {-1, -1};
25 }
```

Пункт 1. Приведенный алгоритм находит положение числа **key** в матрице **matrix** и возвращает индекс этого числа. Если в матрице не содержится **key**, то возвращается пара $(-1, -1)$.

Шаг циклического алгоритма — рассмотрение подматрицы (не обязательно квадратной), внутри которой (включая границы) может находиться искомое число.

Утверждается, что за один шаг можно вычеркнуть один крайний столбец или одну крайнюю строку рассматриваемой подматрицы, так как на нем/ней не может находиться искомое число.

Доказательство: От противного. Предположим, что число может лежать на любой из границ.

$$\begin{pmatrix} b & \dots & \dots & c \\ \vdots & \ddots & \ddots & \vdots \\ a & \dots & \dots & d \end{pmatrix}$$

Тогда верно:

$$a \leq key \leq b$$

$$b \leq key \leq c$$

$$d \leq key \leq c$$

$$a \leq key \leq d$$

Но тогда, например, $a \leq key \leq b \leq key \leq c$, а значения уникальны. \square

Алгоритм линеен, так как всего в худшем случае можно сделать $2n - 1$ вычеркиваний. Докажем это в пункте 2.

Пункт 2. Будем анализировать лишь худший случай. Он характеризуется так:

- $key \notin matrix$
- числа располагаются так, что будет выполнено максимальное количество вычеркиваний

Такая матрица существует, приведем пример:

$$\begin{pmatrix} 12 & 13 & 18 \\ 8 & 10 & 14 \\ 5 & 6 & 7 \end{pmatrix}, key = 11$$

Рассмотрим теперь тело цикла. Сложно определить количество вхождений в каждую из ветвей, чтобы посчитать сколько сравнений будет сделано внутри **if**-ов, поэтому усредним и будем считать, что 2 из 3 сложных условий будет проверено (всего возможных сценария зачеркиваний 4). В теле **if** всегда выполняется 2 операции: уменьшение на единицу и присваивание.

Теперь все готово к тому, чтобы составлять $T(n)$.

$$\begin{aligned} T(n) &= \underbrace{6}_{\text{var init}} + \underbrace{2n}_{\text{loop compares}} + \underbrace{(2n-1)}_{\text{loop iter}} \cdot \left(\underbrace{1}_{\text{not an ans}} + 2 \cdot \underbrace{3}_{\text{if head}} + \underbrace{2}_{\text{in/decrement}} \right) \\ &= 6 + 2n + 9(2n - 1) = 20n + 5 \end{aligned}$$

Покажем, что $T(n) = O(n)$:

$$T(n) = 20n + 5 \leq cn \Rightarrow c = 25, N_0 = 1$$