

```

1 #include <iostream>
2 #include <vector>
3
4 void findKey(std::vector<std::vector<int>> A, int key) {
5     int n = A.size();
6
7     int row = 0;
8     int col = 0;
9     for (int i = 0; i < 2 * n; ++i) {
10         if (A[row][col] == key) {
11             std::cout << "Result: " << row << ", " << col << '\n';
12             return;
13         }
14         if (A[row][col] > key) {
15             ++row;
16         } else {
17             ++col;
18         }
19     }
20     std::cout << "There isn't " << key << " in the matrix" << '\n';
21 }

```

	затраты	итераций
C ₁	1	
C ₂	1	
C ₃	1	
C ₄	2n	
C ₅	2n-1	
C ₆	2n-1	
C ₇	2n-1	
C ₈	2n-1	
C ₉	2n-1	
C ₁₀	2n-1	
C ₁₁	1	

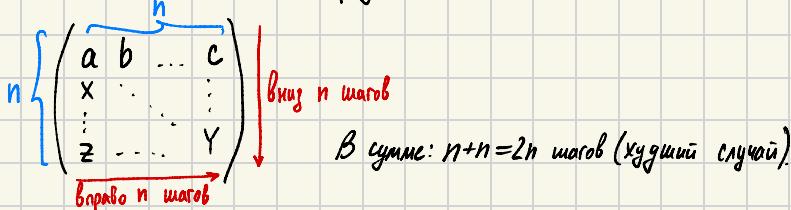
1. Начинаем поиск key слева вверху матрицы (элемент с индексом A[0][0]). Для индекса забогум переменные row и col где номера строки и столбца.

Удём в цикле. Если key == A[row][col], то выбогум ответ.

Если key < A[row][col], то спускаемся вниз (row+=1), т.к. справа будут только элементы > A[row][col], а следовательно и дальше key.

Аналогично, если key > A[row][col], то идём вправо (col+=1), т.к. внизу будут элементы < key.

До самого дальнего элемента (с индексом [n-1][n-1]) необходимо 2n шагов, так как матрица размера n×n.



Если после прохода по циклу key не найден, выбогум сообщение об этом.

$$\begin{aligned}
 2. \quad T(n) &= C_1 \cdot 1 + C_2 \cdot 1 + C_3 \cdot 2n + (2n-1) \cdot (C_4 + C_5 + C_6 + C_7 + C_8 + C_9 + C_{10}) + 1 \cdot C_{11} = \\
 &= (C_1 + C_2 - C_4 - C_5 - C_6 - C_7 - C_8 - C_9 - C_{10} - C_{11}) + \\
 &\quad + (C_3 + C_4 + C_5 + C_6 + C_7 + C_8 + C_9 + C_{10}) \cdot 2n = \\
 &= \left[\begin{array}{l} C_1 + C_2 - C_4 - C_5 - C_6 - C_7 - C_8 - C_9 - C_{10} - C_{11} = C_1 \\ 2(C_3 + C_4 + C_5 + C_6 + C_7 + C_8 + C_9 + C_{10}) = C_2 \end{array} \right] = C_1 + C_2 \cdot n \\
 &\quad C_1 \text{ и } C_2 - \text{ некоторые константы}
 \end{aligned}$$

$$T(n) = O(n)$$

Покажем, что $T(n) = O(n)$. Необходимо найти такие C и n , что

$$C_1 + C_2 n \leq Cn$$

Пусть $n_0 = 1$ и $\forall n \geq n_0$, тогда $C = \max(C_1, C_2) + 2024$.

Следовательно, $T(n) = O(n)$.