1.3 Array Search

Andre Monteiro

Term 3, 2025

(a) Assume a 3 x 3 matrix of form:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

wherein valid entries for k are highlighted in green, invalid in red, and our queried target in black.

- **Case 1:** $A[1][n] > k$.

$$\begin{bmatrix} \color{green}{a_{11}} & \color{green}{a_{12}} & a_{13} \\ \color{green}{a_{21}} & \color{green}{a_{22}} & \color{red}{a_{23}} \\ \color{green}{a_{31}} & \color{green}{a_{32}} & \color{red}{a_{33}} \end{bmatrix}$$

  k can appear in the first row before the last element, in the first column, or in the submatrix below the first row and left of the last column.

  All these positions contain elements strictly less than $A[1][n]$, so they may include k without violating the sorted property.

- **Case 2:** $A[1][n] < k$

$$\begin{bmatrix} \color{red}{a_{11}} & \color{red}{a_{12}} & a_{13} \\ \color{green}{a_{21}} & \color{green}{a_{22}} & \color{green}{a_{23}} \\ \color{green}{a_{31}} & \color{green}{a_{32}} & \color{green}{a_{33}} \end{bmatrix}$$

  k cannot appear in the first row or at $A[1][n]$, but may appear in the submatrix below the first row and left of the last column, or in the last column below the first row.

  Since these positions are strictly greater than elements above or to their left but may still equal k, they are potential candidates.

- **Case 3:** $A[1][n] = k$

$$\begin{bmatrix} \color{red}{a_{11}} & \color{red}{a_{12}} & \color{green}{a_{13}} \\ \color{red}{a_{21}} & \color{red}{a_{22}} & \color{red}{a_{23}} \\ \color{red}{a_{31}} & \color{red}{a_{32}} & \color{red}{a_{33}} \end{bmatrix}$$

  k is found at the top-right corner; the search terminates immediately.

  The element matches k, so no further search is needed.

(b) Let $A \in \mathbb{R}^{n \times n}$ be strictly increasing in rows and columns, and let k be the target integer. Start at the top-right element: $i = 1$, $j = n$. Then repeat:

$$\text{while } i \leq n \text{ and } j \geq 1 : \begin{cases} A[i][j] = k & \Rightarrow \text{return True} \\ A[i][j] > k & \Rightarrow j \leftarrow j - 1 \\ A[i][j] < k & \Rightarrow i \leftarrow i + 1 \end{cases}$$

If the loop ends without finding k, return False.

Complexity Analysis

- **Time Complexity:** $O(n)$, since each step either moves down one row or left one column, and there are at most n rows and n columns.
- **Space Complexity:** $O(1)$, as only two index variables i and j are used.