Write a program to find a given item in a given 2-dimensional array in which each row is sorted and each column is sorted. The execution time must be linear in the sum of the dimensions.

Let the array be \( A \), let its dimensions be \( n \) by \( m \), and let the item we seek be \( x \). The problem, except for time, is \( P \), where

\[
P \quad \text{if } x: A(0..n)(0..m) \text{ then } x = A_{ij} \text{ else } i' = -1 \lor j' = m \quad \text{fi}
\]

The idea is to start at the lower left corner of the array, and by comparing that item with \( x \) we can cross off an entire row or column, and then repeat. We'll need integer variables \( i \) and \( j \) to keep track of the row and column. Define

\[
Q = \begin{cases} \text{if } x: A(0..i+1)(0..m) \text{ then } x = A_{ij} \text{ else } i' = -1 \lor j' = m \quad \text{fi} \end{cases}
\]

which specifies the search in the clear part of the picture.

Then

\[
P \quad \Leftrightarrow \quad i := n-1. \quad j := 0. \quad Q
\]

\[
Q \quad \Leftrightarrow \quad \text{if } i = -1 \lor j = m \text{ then } \text{ok}
\]

\[
\text{else if } A_{ij} > x \text{ then } i := i - 1. \quad Q
\]

\[
\text{else if } A_{ij} < x \text{ then } j := j + 1. \quad Q
\]

\[
\text{else } \text{ok} \quad \text{fi} \quad \text{fi} \quad \text{fi}
\]

Here is the proof. First the refinement of \( P \).

\[
i := n-1. \quad j := 0. \quad Q \quad \Rightarrow \quad \text{if } x: A(0..n)(0..m) \text{ then } x = A_{ij} \text{ else } i' = -1 \lor j' = m \quad \text{fi}
\]

\[
\text{antecedent is } \top
\]

Now the refinement of \( Q \). We use case analysis.

\[
Q \quad \Leftrightarrow \quad (i = -1 \lor j = m) \land \text{ok}
\]

\[
\Rightarrow \quad \text{if } x: A(0..i+1)(0..m) \text{ then } x = A_{ij} \text{ else } i' = -1 \lor j' = m \quad \text{fi}
\]

\[
\text{distribution, antidist}
\]

\[
\Rightarrow \quad \text{if } x: A(0..i+1)(0..m) \text{ then } x = A_{ij} \text{ else } i' = -1 \lor j' = m \quad \text{fi}
\]

\[
\text{expand } Q \quad \text{; substitution law twice}
\]

\[
\Rightarrow \quad P
\]

Each \textbf{if} condition is \( \bot \) because the bunch is \texttt{null}, and the \textbf{else}-part is \( \top \).
The timing proof is much easier. \( P \) becomes \( t' \leq t+n+m \) and \( Q \) becomes 
\[-1 \leq i \leq n \land 0 \leq j \leq m \Rightarrow t' \leq t+i+1+m-j\]