186 (sorted two-dimensional search) 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.

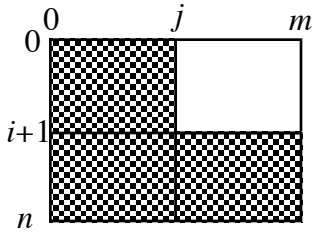
§ Let the array be A, let its dimensions be n by m, and let the item we seek be x. The problem, except for time, is P, where

$P \;=\; \textbf{if } x\colon A(0,..n)(0,..m) \textbf{ then } x = Ai'j' \textbf{ else } i'{=}{-}1 \vee j'{=}m \textbf{ fi}$

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

$Q \;=\; -1{\le}i{<}n \wedge 0{\le}j{\le}m \;\Rightarrow\; \textbf{if } x\colon A(0,..i{+}1)(j,..m) \textbf{ then } x = Ai'j' \textbf{ else } i'{=}{-}1 \vee j'{=}m \textbf{ fi}$

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



Then

$P \;\Leftarrow\; i{:=}\,n{-}1.\;\; j{:=}\,0.\;\; Q$

$Q \;\Leftarrow\; \textbf{if } i{=}{-}1 \vee j{=}m \textbf{ then } ok$
$\quad\textbf{else if } Aij > x \textbf{ then } i{:=}\,i{-}1.\;\; Q$
$\quad\quad\textbf{else if } Aij < x \textbf{ then } j{:=}\,j{+}1.\;\; Q$
$\quad\quad\quad\textbf{else } ok \textbf{ fi fi fi}$

Here is the proof. First the refinement of P.

$i{:=}\,n{-}1.\;\; j{:=}\,0.\;\; Q$ expand Q; substitution law twice

$=\quad -1{\le}n{-}1{<}n \wedge 0{\le}0{\le}m \;\Rightarrow\; \textbf{if } x\colon A(0,..n)(0,..m) \textbf{ then } x = Ai'j' \textbf{ else } i'{=}{-}1 \vee j'{=}m \textbf{ fi}$ antecedent is $\top$

$=\quad P$

Now the refinement of Q. We use case analysis.

$Q \;\Leftarrow\; (i{=}{-}1 \vee j{=}m) \wedge ok$ expand Q, portation

$=\quad (i{=}{-}1 \vee j{=}m) \wedge ok \wedge -1{\le}i{<}n \wedge 0{\le}j{\le}m$

$\Rightarrow\; \textbf{if } x\colon A(0,..i{+}1)(j,..m) \textbf{ then } x = Ai'j' \textbf{ else } i'{=}{-}1 \vee j'{=}m \textbf{ fi}$ distribution, antidist

$=\quad (\quad i{=}{-}1 \wedge ok \wedge -1{\le}i{<}n \wedge 0{\le}j{\le}m$
$\quad\Rightarrow \textbf{if } x\colon A(0,..i{+}1)(j,..m) \textbf{ then } x = Ai'j' \textbf{ else } i'{=}{-}1 \vee j'{=}m \textbf{ fi})$
$\wedge\;\; (\quad j{=}m \wedge ok \wedge -1{\le}i{<}n \wedge 0{\le}j{\le}m$
$\quad\Rightarrow \textbf{if } x\colon A(0,..i{+}1)(j,..m) \textbf{ then } x = Ai'j' \textbf{ else } i'{=}{-}1 \vee j'{=}m \textbf{ fi})$ expand ok

$=\quad (\quad i{=}{-}1 \wedge i'{=}i \wedge j'{=}j \wedge -1{\le}i{<}n \wedge 0{\le}j{\le}m$
$\quad\Rightarrow \textbf{if } x\colon A(0,..i{+}1)(j,..m) \textbf{ then } x = Ai'j' \textbf{ else } i'{=}{-}1 \vee j'{=}m \textbf{ fi})$
$\wedge\;\; (\quad j{=}m \wedge i'{=}i \wedge j'{=}j \wedge -1{\le}i{<}n \wedge 0{\le}j{\le}m$
$\quad\Rightarrow \textbf{if } x\colon A(0,..i{+}1)(j,..m) \textbf{ then } x = Ai'j' \textbf{ else } i'{=}{-}1 \vee j'{=}m \textbf{ fi})$ antecedent context

$=\quad (\quad i{=}{-}1 \wedge i'{=}i \wedge j'{=}j \wedge -1{\le}i{<}n \wedge 0{\le}j{\le}m$
$\quad\Rightarrow \textbf{if } x\colon A(0,..0)(j,..m) \textbf{ then } x = Ai'j' \textbf{ else } {-}1{=}{-}1 \vee j'{=}m \textbf{ fi})$
$\wedge\;\; (\quad j{=}m \wedge i'{=}i \wedge j'{=}j \wedge -1{\le}i{<}n \wedge 0{\le}j{\le}m$
$\quad\Rightarrow \textbf{if } x\colon A(0,..i{+}1)(m,..m) \textbf{ then } x = Ai'j' \textbf{ else } i'{=}{-}1 \vee m{=}m \textbf{ fi})$

Each **if** condition is $\bot$ because the bunch is *null*, and the **else**-part is $\top$.

$=\quad \top$

$Q \Leftarrow i{\neq}{-}1 \wedge j{\neq}m \wedge Aij > x \wedge (i{:=}\, i{-}1.\ Q)$ expand first Q, portation

$= \quad i{\neq}{-}1 \wedge j{\neq}m \wedge Aij > x \wedge (i{:=}\, i{-}1.\ Q) \wedge {-}1{\le}i{<}n \wedge 0{\le}j{\le}m$

$\Rightarrow$ **if** x: $A(0,..i{+}1)(j,..m)$ **then** $x = Ai'j'$ **else** $i'{=}{-}1 \vee j'{=}m$ **fi** simplify antecedent

$= \quad 0{\le}i{<}n \wedge 0{\le}j{<}m \wedge Aij > x \wedge (i{:=}\, i{-}1.\ Q)$

$\Rightarrow$ **if** x: $A(0,..i{+}1)(j,..m)$ **then** $x = Ai'j'$ **else** $i'{=}{-}1 \vee j'{=}m$ **fi** expand Q, substitution

$= \quad 0{\le}i{<}n \wedge 0{\le}j{<}m \wedge Aij > x$

$\wedge\ ({-}1{\le}i{-}1{<}n \wedge 0{\le}j{\le}m \Rightarrow$ **if** x: $A(0,..i)(j,..m)$ **then** $x = Ai'j'$ **else** $i'{=}{-}1 \vee j'{=}m$ **fi**)

$\Rightarrow$ **if** x: $A(0,..i{+}1)(j,..m)$ **then** $x = Ai'j'$ **else** $i'{=}{-}1 \vee j'{=}m$ **fi** discharge

$= \quad 0{\le}i{<}n \wedge 0{\le}j{<}m \wedge Aij > x$

$\wedge$ **if** x: $A(0,..i)(j,..m)$ **then** $x = Ai'j'$ **else** $i'{=}{-}1 \vee j'{=}m$ **fi**

$\Rightarrow$ **if** x: $A(0,..i{+}1)(j,..m)$ **then** $x = Ai'j'$ **else** $i'{=}{-}1 \vee j'{=}m$ **fi** If $Aij > x$ and row i is sorted, then $\neg\ x$: $Ai(j,..m)$ and so x: $A(0,..i)(j,..m) = x$: $A(0,..i{+}1)(j,..m)$

$= \quad \top$

$Q \Leftarrow i{\neq}{-}1 \wedge j{\neq}m \wedge Aij < x \wedge (j{:=}\, j{+}1.\ Q)$ just like the previous case

$= \quad \top$

$Q \Leftarrow i{\neq}{-}1 \wedge j{\neq}m \wedge Aij = x \wedge ok$ expand Q, portation

$= \quad i{\neq}{-}1 \wedge j{\neq}m \wedge Aij = x \wedge ok \wedge {-}1{\le}i{<}n \wedge 0{\le}j{\le}m$

$\Rightarrow$ **if** x: $A(0,..i{+}1)(j,..m)$ **then** $x = Ai'j'$ **else** $i'{=}{-}1 \vee j'{=}m$ **fi** simplify antecedent

$= \quad 0{\le}i{<}n \wedge 0{\le}j{<}m \wedge Aij = x \wedge ok$

$\Rightarrow$ **if** x: $A(0,..i{+}1)(j,..m)$ **then** $x = Ai'j'$ **else** $i'{=}{-}1 \vee j'{=}m$ **fi** expand ok

$= \quad 0{\le}i{<}n \wedge 0{\le}j{<}m \wedge Aij = x \wedge i'{=}i \wedge j'{=}j$

$\Rightarrow$ **if** x: $A(0,..i{+}1)(j,..m)$ **then** $x = Ai'j'$ **else** $i'{=}{-}1 \vee j'{=}m$ **fi** context $Aij = x$ makes **if** condition $\top$, and context $Aij = x \wedge i'{=}i \wedge j'{=}j$ makes **then** part $\top$.

$= \quad \top$

The timing proof is much easier. P becomes $t' \le t{+}n{+}m$ and Q becomes

$\quad {-}1{\le}i{<}n \wedge 0{\le}j{\le}m \Rightarrow t' \le t{+}i{+}1{+}m{-}j$