Let $A$ be the array, let its size be $m \times n$, and let $x$ be the item (these are constants). The solution will use variables $r, a, b, c : nat$. Variable $r$ will indicate a row; on that row there may be some occurrences of item $x$; $a$ will indicate the start of these occurrences, and $b$ will indicate the end of these occurrences. Variable $c$ will accumulate the count of the occurrences of $x$. As $r$ decreases from $m$ to $0$, variables $a$ and $b$ will trace the envelope of the occurrences of $x$, as in the following picture.

![Diagram](image)

The problem is $P$, defined as

$$P \equiv c' = \Sigma i : 0..m \cdot \Sigma j : 0..n \cdot \text{if } A_{ij} = x \text{ then } 1 \text{ else } 0$$

Define

$$Q = c = \left( \Sigma i : r..m \cdot \Sigma j : 0..n \cdot \text{if } A_{ij} = x \text{ then } 1 \text{ else } 0 \right) \land \left( \forall i : 0..r \cdot \forall j : 0..a \cdot A_{ij} < x \right) \land \left( \forall i : r..m \cdot \forall j : b..n \cdot x < A_{ij} \right)$$

$$\Rightarrow P$$

$$L = \left( \forall j : 0..a \cdot A_{rj} < x \right)$$

$$\Rightarrow \left( \forall j : 0..a' \cdot A_{rj} < x \right) \land \left( A_{r a'} \geq x \lor a' = n \right) \land r' = r \land b' = b \land c' = c$$

$$R = \left( \forall j : 0..b' \cdot A_{rj} \leq x \right)$$

$$\Rightarrow \left( \forall j : 0..b' \cdot A_{rj} \leq x \right) \land \left( A_{r b'} > x \lor b' = n \right) \land r' = r \land a = a' \land c' = c$$

Or, alternatively, define

$$Q = (a = 0 \lor A r (a - 1) < x) \land (b = 0 \lor A r (b - 1) \leq x)$$

$$\Rightarrow c' = c + \left( \Sigma i : 0..r \cdot \Sigma j : 0..n \cdot \text{if } A_{ij} = x \text{ then } 1 \text{ else } 0 \right)$$

$$L = (a' = 0 \lor A r (a' - 1) < x) \land (A r a' \geq x \lor a' = n) \land r' = r \land b' = b \land c' = c$$

$$R = (b' = 0 \lor A r (b' - 1) \leq x) \land (A r b' > x \lor b' = n) \land r' = r \land a = a' \land c' = c$$

Now the refinements:

$$P \iff r : = m. \ a : = 0. \ b : = 0. \ c : = 0. \ Q$$

$$Q \iff \text{if } r = 0 \text{ then ok else } r := r - 1. \ L. \ R. \ c := c + b - a. \ Q \text{ fi}$$

$$L \iff \text{if } a = n \text{ then ok else if } A r \geq x \text{ then ok else } a := a + 1. \ L \text{ fi \ fi}$$

$$R \iff \text{if } b = n \text{ then ok else if } A r > x \text{ then ok else } b := b + 1. \ R \text{ fi \ fi \ fi}$$

For the time, put $t := t + 1$ before each of the three recursive calls, replace specification $P$ with $t' \leq t + m + 2 \times n$, and replace all the other specifications $Q$, $L$, and $R$ with $a \leq n \land b \leq n \implies t' \leq t + r + m + 2 \times n - a - b$. 

* (sorted two-dimensional count) Write a program to count the number of occurrences of 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.*