

cautare (A, h, n)	C1
i = n - 1;	C2
j = 0;	C3
if (A[i][j] < h)	C4
directie = 1;	C5
else	C6
directie = -1;	C7
while (1)	C8
if (i < 0)	C9
return 0;	C10
if (directie == 1)	C11
if (A[i][j] <= h)	C12
if (A[i][j] == h)	C13
return 1;	C14
if (j != n - 1)	C15
j++;	C16
else	C17
return 0;	C18
else	C19
j--;	C20
directie = directie*(-1);	C21
if (directie == -1)	C22
if (A[i][j] == h)	C23
return 1;	C24
i--;	C25
directie = directie*(-1);	C26

Se observă că operațiile 12 și 22 sunt operații critice. Costul lor este $O(1)$.

În cel mai defavorabil caz, i va lua toate valorile de la $n-1$ la 1 (adică se va parcurge o linie de jos în sus), iar j va lua valori de la 1 la $n-1$. Astfel, antetul funcției while se va repeat de $a*n+b*n$ ori, a și b determinându-se în funcție de i și j .

Complexitatea va fi $T(n) = O(a*n+b*n) = O(n)$.