

АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА ДАТЧИЦИ

Пълното изчерпване няма да върви в 50-60% от тестовете.

Решение на частните случаи с намаляваща и нарастваща редица ще хване 30%.

Решението за пълен брой точки е линейно, като не е необходимо дори използване на масив.

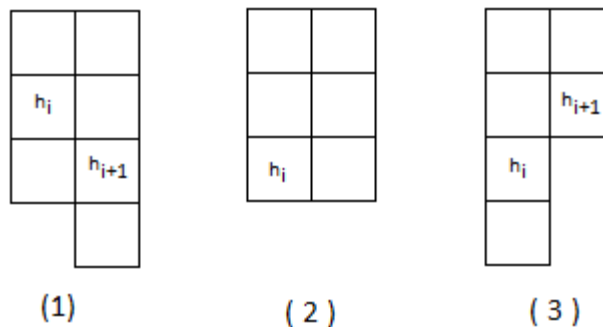
Най-близкото до ума е решение, при което се увеличава броя на датчиците. Тогава обаче възникват повече частни случаи за всеки от посочените по-долу варианти (1) – (3).

Най-краткото за писане е следното решение, при което приемаме в началото, че броят на датчиците е равен на N .

Освен брояч за датчиците, въвеждаме променлива h , определяща номера на реда на последния датчик, което означава, че всички редове от 1 до h са покрити. Променливата a е броя на квадратите в текущата колона.

Сравнявайки a и h , има три варианта:

- (1) $a > h$, в този случай се вдига с 1 само височината h /покрива се следващия ред $h+1$ /.
- (2) $a = h$ – дължината на текущата колона е h , следователно е покрита и намаляваме брояча на датчиците.
- (3) $a < h$ е аналогичен на случай (2) относно брояча, но тук „тънкия” момент е да се смени новата стойност на h да стане равна на a .



На картинката са илюстрирани трите случая, като h_i е стойността на h в колона i , а h_{i+1} е новата стойност на h в колона $i+1$.

Решението е фрагмента от програмата в цикъл от 1 до N .

```
cin >> a;
if (a>h) h++;
else
if (a==h) br--;
else
if (a<h) {
    br--;
    h = a;
}
```

Автори: Бисерка Йовчева, Павел Петров