**Анализ на решението на задача**

**ИГРА**

Преди Крис да хвърли първия камък за всяка колона изчисляваме и запомняме пътя, който би изминал камъка, ако Крис го хвърли в тази колона.

Когато Крис посочва в коя колона иска да хвърли камък, ние знаем точно къде ще застане камъка - последната клетка на пътя, която преди това сме изчислили за тази колона.

За да можем да направим същото за всеки следващ камък, трябва да опресняваме всеки път, по който последната клетка е тази, в която спира камъка. Можем да направим това, като симулираме ефекта на гравитацията от втората последна клетка в пътя (като пътя преди това не се променя).

При реализацията е удачно да се използва масив от тип структура и две функции, с помощта на които определяме къде ще падне съответния камък:

struct board {

 char row[30000];

 int r;

 void stoprock() {

 a[r-1][row[r-1]] = 'O';

 }

 void sr() {

 for( ;; ) {

 int s = row[r-1];

 if( r > 1 && a[r-1][s] != '.' ) { --r; continue; }

 if( r == R ) break;

 if( a[r][s] == 'X' ) break;

 if( a[r][s] == '.' ) {row[r++] = s;}

 else {

 if( s > 0 && a[r][s-1] == '.' && a[r-1][s-1] == '.' ) {

 row[r++] = s-1;}

 else if( s+1 < S && a[r][s+1] == '.' && a[r-1][s+1] == '.')

 { row[r++] = s+1;

 } else break;

 }

 }

 }

} P[30];

Общата времева сложност е O(C(R+N)).

Автор Пламенка Христова