

АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА

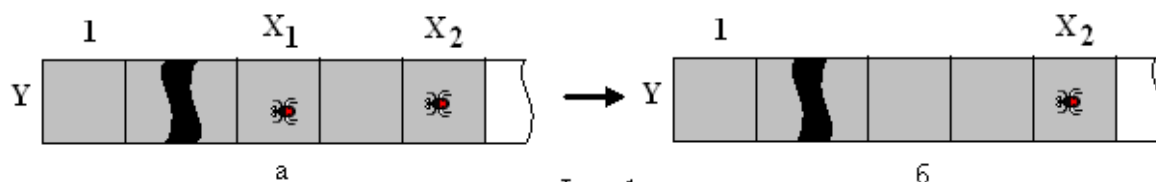
bio

Първият начин е чрез използване на двумерен масив в който се отбелязват квадратчетата през които минава роботчето. Неефективно решение поради големите ограничения.

Другите решения са свързани с използване на масиви, които съхраняват данни за роботите.

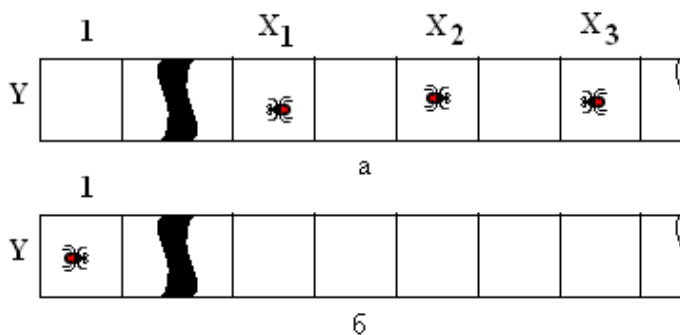
Нека в елементите на масива се пази за всеки елемент посоката, реда (стълба) и от кой до кой номер квадратче се движи робота. Явно единият от двата номера ще е 1,М или N.

Размерът на този масив ще се намали няколко пъти, ако се има пред вид следното: за всеки ред или стълб броят на роботчетата може да се сведе до **максимум два**.



Ще разгледаме случай за ред (за стълб е аналогично). От Фиг.1 се вижда, че двете роботчета може да се заменят с едно, и то ще е това, което се движи от т.(Y,X2) към т.(Y,1).

На Фиг.2 трите роботчета се заменят пак с едно, което ще е в т.(Y,1) и ще се движи надясно или ще е в т.(Y,N) и ще се движи наляво (т.е. ще покрие целия ред Y). Друга ситуация е двете роботчета да са в съседни квадрати, като левият се движи наляво, а десният – надясно. Тогава двата се заменят с един, който отново покрива целия ред.



Ясно е, че на всеки ред може да има най-много 1 робот, с изключение на ситуацията, в която ляв робот се движи наляво, десен надясно и между тях има повече от 1 квадрат.

След оптимизацията два робота преминават през един квадрат само тогава, когато единият се движи хоризонтално, а другия вертикално. Може да изчислим всички квадрати през които минават роботите и да извадим само застъпващите се. Това ще е решението.

Имайки предвид гореказаното, вторият начин ще е чрез използване на един едномерен масив.

Нека в масива има вече P на брой елемента. Четем данните за следващия робот („кандидат“ да влезе в масива) и трябва да направим съответните проверки с всички P елемента на масива дали „кандидатът“ ще влезе или не.

Има различни варианти - ако сме в случая от Фиг.1, трябва да запомним номера на полевия елемент и да го изтрием, или да заместим данните на този по-ляв елемент с нови. Последното изглежда добре, защото се избягва обхождане на масива заради изтриването на

елементите. Но ако сме в случая на Фиг.2 и елементите от т.(Y,X1) и т.(Y,X3) са вече в масива, а роботът от т.(Y,X2) е „кандидат“, лесно ще заместим един от съществуващите с нови данни. Но за другия? - Или трябва да го изтрием, или да го маркираме по някакъв начин, че повече няма да ни е нужен.

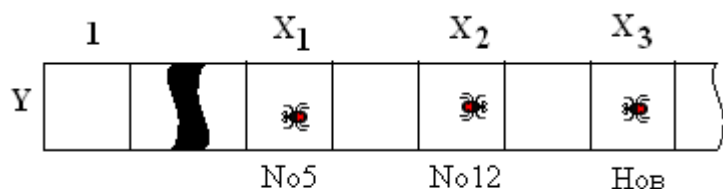
При използване на един масив трябва да се направи подходяща структура, защото в него ще са работи, движещи се както хоризонтално, така и вертикално. Като оптимизация може да ги разделим в два масива – единият за редовете, другия за стълбовете. Това разделяне на 2 масива оптимизира, но тестовете, които ще покрие ще са около 60%.

Проблем относно времето за изпълнение на програмата (1 сек.) се явява единствено търсенето на елемент, който е на същия ред или стълб, на който е „кандидата“. Може да се използва сортиране на съответния масив и с двоично търсене да се намери искания елемент, или с двоично търсене да се вмъкне нов и т.н., но това добавя $N \cdot \log N$ операции. Ако се използва подобно решение, очаква се да хване 70-80% от тестовете.

Операциите стават само N , ако се добави нова структура, в която пазим на всеки ред (стълб) номерата на роботчетата, които се движат по него. Тъй като не може да има повече от две роботчета на ред (стълб), структурата ще съдържа само 3 променливи B_r , n_1 и n_2 , които съответно са броя на роботите на реда, номера на единия и номера на другия. Необходими са ни 2 масива с такава структура, които са с размер по 15000. Те съхраняват за всеки номер на ред и всеки номер на стълб колко и кои работи са там. Да ги означим с RR и SS – масиви съответно за редовете и стълбовете

Например ако прочетем робот No10 с данни (7, 12, 1) и $RR[7]=(1,4,0)$, то е ясно, че в ред 7 има един робот, който е с номер 4. Ако данните на робот No 4 са (1,5) – т.е. се движи от позиция 1 до позиция 5 (ситуация на Фиг.1), то само сменяме данните на No 4 с(1,12). Т.е. премахваме кандидата с No10.

По-особен е случаят когато от 2 робота остава 1. За да избегнем триенето на елемент 12 (Фиг.3), може да му зададем лява и дясна граница 2 и 1 (ако се движи хоризонтално), което ще го „елиминира“ при събиране на замърсените квадратчета.



Фиг.3

Автор:Павел Петров