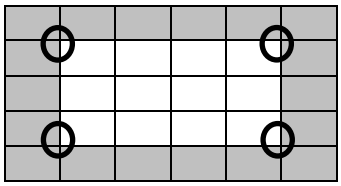
АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА ФИГУРИ

Поради ограничението в паметта е ясно, че цялата таблица не може да бъде съхранявана в паметта. Поради това, ще възстановяваме таблицата ред по ред, четейки последователно данните за ивиците.

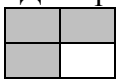
Първо да разгледаме първата част от задачата – преброяване на изрязаните клетки, т.е. нулите в таблицата. Да си представим по-прост вариант на задачата, в който има само хоризонтални ивици. Прочитаме данните за първата ивица (в програмната реализация в променливи *newi, newj* и *d*). След това ходим по индексите на клетките от таблицата докато не стигнем до индекси *(i,j),* които съвпадат с индексите на началната клетка на ивицата. До този момент всички клетки са били изрязани (в таблицата е имало нули) и трябва да бъдат преброени. Щом сме стигнали до началната клетка на първата ивица, то започва последователност от единици. В една променлива (в програмната реализация *maxj*) държим индекса на най-десния елемент от текущия ред, който е равен на 1. Когато сме срещнали първата ивица, имаме *maxj=newj+d-1*. Четем данните за следващата ивица и продължаваме да се движим по реда. Ако индекс *j* (номерът на елемента от текущия ред) е станал по-голям от *maxj*, то се намираме в изрязана клетка (0 в таблицата) и увеличаваме брояча на изрязаните клетки. Ако се прехвърлим на нов ред, правим *maxj*=0 и продължаваме по същия начин. Когато стигнем до началото на следващата прочетена ивица, то обновяваме *maxj*, като го правим равно на по-голямата от старата му стойност и *newj+d-1*, където *newj* е координатата на началото на новата хоризонтална ивица.

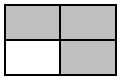
Сега да добавим и вертикалните ивици. За да отчитаме и тях, ще използваме масив *x* с размерност *N*, като в *x[j]* ще държим най-големия индекс *i*, до който достига вертикална ивица (от прочетените), която е разположена в стълб с номер *j*; *x[j]* се обновява, когато движейки се по клетките на таблицата достигнем до началото на следващата прочетена вертикална ивица, което е в стълб *j*. Обновяването е аналогично на това на *maxj* – по-голямата от старата стойност на *x[j]* и *newi+d-1*. Имайки тази информация, то, когато се намираме в клетка с индекси *(i,j),* към проверката дали не се покрива от хоризонтална ивица, добавяме аналогична проверка и дали не се покрива от вертикална ивица.

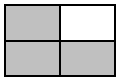
По-труден изглежда моментът с преброяване на изрязаните фигури. Тъй като нямаме в паметта цялата таблица, то и тях трябва да ги възстановяваме ред по ред. Можем да се опитаме да ги строим „по редово”, но това е трудно и бавно. Има много по лесен начин, който се основава на следното наблюдение. Нека най-напред си представим, че е изрязан само един правоъгълник.

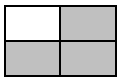


Да наречем външни ъгли на правоъгълника, шаблоните от вида







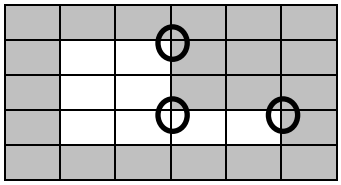


Тъй като фигурите на се допират една до друга, вкл. по диагонал, то това са четирите вида външни ъгли.

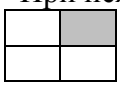
Ако Пешо беше изрязвал само правоъгълници, то за да ги преброим щеше да е достатъчно да преброим външните ъгли и намерия брой да разделим на 4.

Какво се променя, когато Пешо изрязва не само правоъгълници?

Да разгледаме следната фигура:



При нея се появи ъгъл от друг тип – да го наречем вътрешен. Неговият шаблон е:



Броят на външните ъгли се увеличи с 1 – един изчезна и се появиха 2 нови.

Лесно се съобразява (след като веднъж се сетите, че може да се търси нещо подобно), че, както и да реже Пешо (при условията на задачата), разликата между броя на външните и броя на вътрешните ъгли на фигурата е постоянен и е равен на 4. Тогава броят на фигурите е равен на разликата между всички преброени външни ъгли и всички преброени вътрешни ъгли, разделена на 4. За да броим вътрешните и външните ъгли, трябва да пазим само състоянието на двата последни реда от таблицата.

Трябва да се има предвид следната тънкост – външни ъгли възникват и по краищата на листа, т.е. трябва да си представяме, че целият лист е заобиколен от единици.