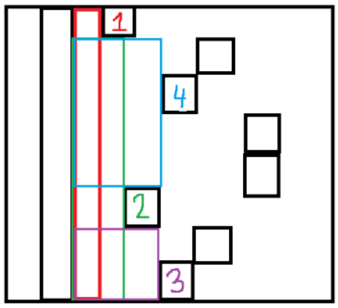
АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА БРОЙ ПРАВОЪГЪЛНИЦИ

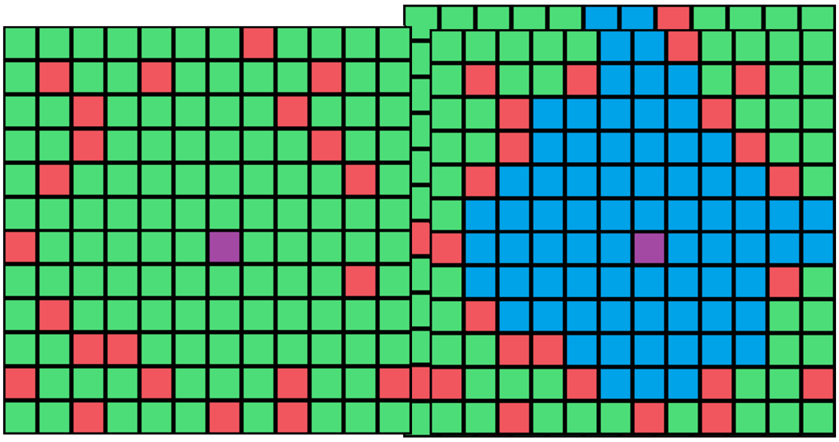
Задачата на практика се състои от две части. Първоначално намиране на броя правоъгълници и после изпълняване на заявките. За първоначалното намиране ще използваме следния алгоритъм със сложност (n*m*logn):

Първо правим по една опашка с индексите на единиците на всеки ред. После обхождаме стълбовете последователно отляво надясно и премахваме индексите на единици които са вляво от текущия разглеждан стълб. Така във върха на опашките получаваме най-лявата единица във всеки ред, която е вдясно или върху разглеждания стълб. Сортираме тези единици по това колко наляво са (може да се направи линейно с counting sort) и започваме да ги обхождаме (отляво надясно). При обхождането на дадена единица (на ред i) я вкарваме в set (по точно вкарваме реда на който е), така имаме бърз достъп до най-долната обходена единица над текущата и най-горната под текущата. Ще броим всички правоъглъници които се съдържат в следните и няма да броим тези които вече са броени. С други думи обхождайки точка 4 (от картинката). Намираме по-дясната от точки 1 и 2 (най-близките долна и горна, които са обходени). Знаем че за да не сме броили вече правоъгълника, то десният му край трябва да е по-голям или равен на *max*(колоната на клетката отгоре (клетка 1), колоната на клетката отдолу (клетка 2)). Освен това, горният и долният край трябва да са в интервала [реда на клетката отогоре (келтка 1) + 1, реда на клетката отдолу (клетка 2) - 1]. Можем да намерим броя на тези правоъгълници константно с проста формула, зависеща от координатите на трите клетки. В случая за клетка 4 те са 10 (един вариант за десния край на правоъгълника и 5*4/2 = 10 за горен и долен т.е. 1 * 10 = 10).



В крайна сметка, сложността за обхождане на дадена колона е n*log*(n), където *log*(n) идва от операциите с типа set.

Сега относно заявките. Първо ще отбележим, че независимо дали променяме нула на единица или обратното, трябва да преброим правоъгълниците, съдържащи тази клетка, които не съдържат други единици (различни от тази клетка) и съответно да ги прибавим или извадим от досегашния брой правоъгълници. Една такава заявка може да се изпълни за линейно време O(n+m). За целта ще имаме матрица, в която ще помним за всяка клетка от таблицата колко последователни нули има наляво, както и една такава матрица за надясно (след всяка заявка ще променяме само реда, на който е направена).



Нека имаме заявка за лилавата клетка. Нека зелените клетки са нули, а червените единици. Ще пазим информация за синята фигура. Тя е максималната, която се стеснява нагоре и надолу от лилавата клетка, не съдържа червени клетки и съдържа клетка от колоната на лилавата. Важното свойство в случая е, че всеки правоъгълник, съдържащ само лилави и зелени клетки в първата фигура, съдържа само лилави и сини клетки във втората, т. е., можем да игнорираме всички останали зелени клетки. Ще пазим фигурата по следния начин: за всяка клетка в колоната на лилавата ще пазим колко наляво от нея има сини и колко надясно от нея има сини. Сега ще обходим последователно отгоре надолу редовете с поне една синя клетка над лилавата (и самия ред с лилавата). На дадена стъпка ще броим всички правоъгълници, чиято горна страна е редът, който обхождаме вмомента. Ще разделим правоъгълниците на три групи, които ще броим поотделно. Да речем, че обхождаме сивия ред. Първата група е от правоъгълниците с долен край под двете черни линии (те тръгват от краищата на сивия ред и стигат до последните сини клетки в съответните колони). За тях няма значение докъде се простира сивият ред и можем да използваме информацията от предишния обходен ред, като само евентуално добавим още някои редове. В случая има един такъв ред и при него имаме два варианта за ляв край и два за десен – общо 4. Втората група е под края на едната черна линия, но не под другиата. Това означава, че от едната страна се простира до краищата на долните редпве (в случая отляво и реда е точно 1), а отдясно се простира до десния край на сивата линия. В случая ще поддържаме сбора от левите краища, който ще умножим по броя варианти за десен край (той е общ за всички такива редове : десният край на сивия ред). В случая те са 3*3 = 9 на брой. Остава третата група, която не е под нито една от черните линии, т. е., и левия, и десния край зависят от сивия ред. Тук има значение броят на тези редове и за всеки от тях умножаваме по броя варианти за ляв и за десен край (определени от сивия ред). В случая има 3 варианта за долен край 4 за ляв и 3 за десен т.е. 36. При разглеждане на следващ ред последователно повдигаме долните краища на черните линии с колкото е нужно и съответно увеличаваме правоъгълниците от първата група или сумата за втората група. За повече подробности вижте source кода – първата група се смята от променливата cnt, втората – с променливите lcnt и rcnt (в зависимост от това, коя черна линия е по-ниско, другата приема стойност нула) и третата група се смята директно. В gcnt се пази броят на всички правоъгълници за заявката.

Автор: Даниел Атанасов