АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА

СТЕНИ

Подзадача 1: В тази подзадача ограниченията за координати на изходи и стени са малки, затова можем да представим клетките като върхове на граф. В този граф можем да пуснем стандартно търсене на минимален път.

Подзадача 2: Тук можем да направим наблюдението, че никога няма да направим стъпка надолу. Следователно отговорът за всеки изход е ( минималния брой движения по Х ) + ( Y координатата на изхода ). Сега трябва да сметнем минималния брой движения по Х за достигане на всяка една клетка. За целта сортираме стените по нарастващ Y и ги добавяме в този ред. При едно добавяне клетките, над които няма стена запазват стойността си ( реално пътят се увеличава с едно изкачване нагоре, но сега можем да го пренебрегнем и чак накрая да добавим У координатата ). А до клетките, над които има стена отговорът може да се получи по 2 начина- да минем през стената или да дойдем от някоя от клетките в краищата на стената. Така след всяко добавяне на стена имаме отговора за всяка клетка, която е на горния ред след стената . Това решение може да се реализира по наивен начин за О( \*NмахХ ).

Подзадача 3: Тук координатите са по-големи, но лесно се вижда, че ни интересуват само клетки, които са в краищата на стени. След като компресираме тези интересни точки лесно можем да възтановим отговорите дори за „не-интересни“ точки – за тях минималният път може да дойде само от лявата или дясната им „интересна“ точка.

За следващите подзадачи трябва да направим наблюдението, че най-краткият път до две клетки, които са на един и същи Y и се различават с 1 по Х, може да се различава с най-много 1. Това ни подсказва, че вместо да пазим целия масив със минимални стойности до всяка клетка от ред, можем да ги представим като множество от лъчи( ориентирани надясно например ), за които си пазим Х-координатата на началото, Y-координата( която ще представлява минималния път до клетката в началото на лъча) и ъгловият коефициент( който е 1 или -1 ). Сега, за да намерим минималния път до някоя клетка, намираме кой е лъчът, който започва най-близо отляво на точката и с помощта на ъгловия коефициент можем да сметнем отговора за нея.

Да разгледаме как се изменят лъчите, когато се добави нова стена. За клетките, над които не минава стената, стойностите не се променят. За останалите можем за момента да прибавим плътността на стената, което в представянето с лъчи представлява „отрязване“ на този участък и повдигането му с колкото е плътността на стената. Сега остава да се модифицират лъчите в левия и десния край на този участък. За клетките, които попадат там, отговоръ може да дойде от лявата ( дясната) клетка в края на участъка. Тоест можем да пуснем лъч от левия край с ъглов коеф. 1 и още един лъч от десния край с коеф. -1 . При това, ако лъчите, които са били в левия( десния ) край са по-високо от новият лъч, то старите биват премахнати.

За да се реализира това решение достатъчно бързо е удобно да се пази информацията за лъчите в структурата от данни treap. Така добавянето на лъчи отделянето на цял участък става за O(logN). Премахването на лъчи в краищата на участъка става отляво надясно ( отдясно наляво ) отново за O(log) за лъч.

Възможно е да се направи подобно решение с помощта на индексно дърво, но отново е необходима компресия на координатите, която забавя решението и повечето такива решения не взeмат точките за последната подзадача.

*Автор: Виктор Терзиев*