

## Анализ на задача raspberry

*Тагове: движение на североизток, динамично програмиране, оптимизация на вътрешния цикъл, 2D суфиксен максимум*

Задачата е модификация на стандартната задача за движение на североизток, като тук движението е в посока югоизток. Формално от една клетка може да скочим директно на всяка клетка с не по-малък номер на ред и не по-малък номер на колона или да напуснем таблицата. Трябва да намерим максималната сума на път от всяка клетка, разгледана като стартова, движейки се по тези правила. Причината да се търси сумата на отговорите за всяка клетка, не е за да се намира по изкуствен начин отговорът, а за да се отпечата само едно число и по-добре да се различават решенията, защото времето за изход става пренебрежително.

### Решение на втора подзадача - 7 точки

Тази подзадача е дадена с цел да може да се намалят нулите по задачата и да има подзадача с много лесно решение. Съществено е ограничението, че има само 1 ред. Тогава движението в този ред е просто от някоя клетка до някоя вдясно от нея. За всяка клетка търсим максималната сума на път, започващ от нея и завършващ в последната клетка. Очевидно е достатъчно да обходим реда от дясно наляво и да поддържаме най-голямата сума на път, завършващ в последната клетка. Съответно като сме на някоя клетка, добавяме нейната стойност към най-добрия път в момента, за да получим отговора за тази клетка. След това гледаме дали текущият път няма да е по-добър от досегашния най-добър (което става винаги, ако текущата клетка е с положителна стойност) и евентуално актуализираме стойността на най-добрия път до момента. Очакваната сложност на решение е  $O(M)$ .

### Решение на трета подзадача - 11 точки

Предвид малките ограничения, е ясно, че тази задача е предвидена за някакво решение с пълно изчерпване. Лесно можем да направим рекурсия, която да започва от всяка клетка и да разглежда всички възможни пътища от нея, като намира най-добрия път, който завършва в последната клетка. Интересен е въпросът каква е сложността на такова решение. Трябва да преброим всички възможни пътища, които могат да се получат по описаните правила. Ние ще посочим само груба горна оценка -  $2^{NM}$ , защото може да си мислим, че избираме произволни клетки от таблицата и пътят получаваме като ги наредим в правилен ред на преминаване (но има доста възможности, които не задават валиден път). Затова сложността на решението ще посочим като  $O(2^{NM})$ .

### Решение на четвърта подзадача - 32 точки

Това е първата съществена подзадача, стъпка към пълното решение. Решението на задачата е с динамично програмиране, както и на стандартната задача, от която произлиза тази. Много пъти, когато се пита за минимален/максимален път, може да се помисли за решение с динамично. В други случаи решението е графово, но тук построяването на граф не е приложимо, защото ще имаме много на брой ребра. Искаме да намерим  $dp[x][y]$ , което е отговорът на задачата за стартова клетка  $(x, y)$ . Тогава очевидно имаме следната рекурентна зависимост:

$$\begin{aligned} dp[x][y] &= \max_{x' \geq x, y' \geq y} (a[x][y] + dp[x'][y']) = \\ &= a[x][y] + \max_{x' \geq x, y' \geq y} dp[x'][y'] \end{aligned}$$

Ще допълним, че в индексите на максимума трябва да се уточни, че  $(x', y') \neq (x, y)$ , но е махнато за по-добра четимост. Така ако реализираме директно намирането на това динамично, получаваме решението на тази подзадача със сложност  $O(N^2M^2)$ .

### Решение на пета подзадача - 19 точки

Тази подзадача е още една независима. Имаме допълнително ограничение, че всички клетки са с неотрицателна стойност. Нека забележим следното лесно наблюдение - задачата се свежда до класическото движение на североизток. Това е така, защото нямаме изгода да от една клетка да скачаме в някоя далечна друга (която не е съседна по страна), защото просто можем да се движим по съседни клетки и пак да стигнем другата, но ще натрупаме по-голяма или равна сума по пътя, понеже всички стойности са неотрицателни. Допълнително, получаваме че оптималното решение от дадена клетка винаги ще минава и през последната - аналогично нямаме изгода да напуснем таблицата по-рано. От всичко казано, стигаме до извода, че можем да използваме класическото динамично за движение на североизток или:  $dp[x][y] = a[x][y] + \max(dp[x+1][y], dp[x][y+1])$ , за  $x < N, y < M$  в общия случай. Тогава решението е линейно по таблицата и постигнатата сложност е  $O(NM)$ .

### Решение на шеста подзадача - 79 точки

Вече ограниченията са максимални и единствено имаме допълнително ограничение, че оптималният път включва последната клетка. Както в много задачи с динамично, може да се направи някакъв тип оптимизация на вътрешния цикъл - този, който намира стойността на даден стейт. Често като има сумиране може да се намери нова формула на динамичното с малко изразяване, в случая това е неприложимо. Но можем да направим помощна матрица, в която да записваме максималното динамично от дадена клетка до долния десен ъгъл (югоизточния ъгъл):

$$max[x][y] = \max_{x' \geq x, y' \geq y} dp[x'][y']$$

Тогава, ако лесно можем да поддържаме тази матрица, намирането на динамичното става с константна сложност:  $dp[x][y] = a[x][y] + \max(max[x+1][y], max[x][y+1])$ . Разбира се, поддържането на матрицата с максимуми също е просто и реално тя поддържа 2D суфиксен максимум в таблицата на динамичните -  $max[x][y] = \max(dp[x][y], max[x+1][y], max[x][y+1])$ . Така постигаме решение с крайната сложност  $O(NM)$ .

### Решение на седма подзадача - 100 точки

Последно остана да видим какво става, ако оптималните пътища не е нужно да завършват в последната клетка. Един интересен факт е, че това реално се определя от това дали стойността на последната клетка е отрицателна или не. Можем да направим следната модификация на предното решение, с което да позволим да прекратяваме пътя и в произволна клетка. Когато смятаме динамичното  $dp[x][y]$ , можем да разгледаме случай, в който пътят приключва в клетка  $(x, y)$  и след това напуска таблицата. Единственият случай, когато това би било оптимално от гледна точка на минаване през клетка  $(x, y)$  е, ако  $\max(max[x+1][y], max[x][y+1]) < 0$ . Тогава просто слагаме  $dp[x][y] = a[x][y]$ . Така разрешаваме пътят да спре в произволна клетка, което означава, че и максимумите, които пазим в допълнителната матрица са за пътища, които също могат да приключват в произволна клетка. Крайното решение е със сложност  $O(NM)$ .

Автор: Илиян Йорданов