Анализ на задача grid

Първото нещо което идва на ум, е да си създадем матрица NxN, в която да попълним начините за достигане на всяка клетка. За поредната клетка, те са равни на сумата от начините за достигане на горната клетка и начините за достигане на лявата клетка. Ако някоя от съответните клетки е извън масива, тя е недостижима, следователно добавяме 0. Начините за достигане на първата клетка е винаги 1, стига тя да не е премахната. При срещане на клетка с координати – координатите на премахната точка, записваме 0 в съответната клетка. Така получаваме резултата в последната клетка на последния ред.

Пример за попълнена матрица без премахнати точки:

1 1 1

1 2 3

1 3 6

Пример за попълнена матрица с 1 премахната точка:

1 1 1

1 2 0

1 3 3

Нека разгледаме ограниченията. При N = 6000, матрицата ще се състои от 36`000`000 елемента. Ако вземем модула 10000000019 в предвид, бързо установяваме, че се налага използване на тип long long int. Сега нашата матрица ще се състои от 36`000`000 \* 8 байта, което надхвърля лимита за памет, указан в условието. Така идеята за матрицата в този вид отпада.

Вместо да заделяме памет за N\*N, може да заделим само един масив a[N]. Попълваме масива с 0-ли, като само първият елемент е 1. За да получим последния ред от NxN матрицата, ще направим N итерации, като за всяка итерация, клеткa a[j] = a[j] + a[j-1], за j = 1, 2, …, N-1. Не трябва да забравяме да използваме модула, както и да проверяваме, дали сме достигнали премахната точка и ако да – попълваме 0 на съответната клетка. В случай, че точката с координати (1, 1) или (N, N) е премахната, крайният резултат е 0. Така отговорът ще се получи в последната клетка след последната итерация.

Решението има сложност O(N\*N).

Автор: Делян Добрев