

Maps (Решение)

Задачата Maps беше по-лесна от обикновено за група С. За решаването ѝ състезателите трябваше да направят няколко прости наблюдения:

1. Тъй като не сме ограничени в далечината на "преходите" всеки ден, можем да стигнем от всяка клетка, до всяка друга, стига новата да е по-висока.
2. От предното наблюдение следва, че правоъгълната матрица не допринася с нищо – еквивалентно е да държим всички числа в едномерен масив.
3. Можем да сортираме едномерния масив, като така валидните клетки ще са винаги надясно от текущата.
4. За постигане на максимален отговор, ако можем да се разцепим, ще го направим.
5. За постигане на максимален отговор, ще ходим винаги в най-близките по-високи клетка (най-близките различни от текущото число надясно).

Реално трябва да приложим много лесно (едномерно) динамично: <http://www.informatika.bg/lectures/dynamic-programming-part1>.

Всъщност, от горните наблюдения следва, че ако всички височини са различни, то отговорите биха били числата на Фибоначи (тъй като винаги взимаме отговорите от динамичната таблица от съседните две клетки. Наистина, ако $N * M = 1$, отговорът би бил 1; ако $N * M = 2$, отговорът би бил 1; ако $N * M = 3$, отговорът би бил 2; ако $N * M = 4$ отговорът би бил 3; ако $M * M = 5$, отговорът би бил 5 и т.н.). Реално тогава не би имало и нужда изобщо да четем височините, а само N и M . Оригинално това беше и идеята ми за задачата – да е задача-бъзик.

Тъй като височините могат да се повтарят, обаче, решението е една идея по-сложно – вместо съседните две клетки отдясно, трябва да намерим най-близките две отдясно, които са с числа стриктно по-големи от текущото. Това не е нужно да правим всеки път; вместо това, можем да започнем от най-големите числа и да вървим към по-малките, пазейки къде са най-близките надясно, които са стриктно по-големи. Движейки се наляво, трябва да ъпдейтваме тези индекси, но това може лесно да стане с амортизирано $O(1)$ ($O(N * M)$ за цялата задача).

Забележете, че числата бяха относително малки, съответно вместо да ползваме $O(N * M * \log(N * M))$ сортиране, можеше да направим counting sort за (приблизително) $O(N * M)$, като така цялото решение би било с тази сложност. Това, обаче, беше ненужно при дадените ограничения.

Важно в задачата е да забележим, че се иска максималният отговор, изведен по модул, а не максималният модул! Лесно е да се обърка решението, като сравняваме числа по модул – което не винаги би дало максималния отговор. Например, 42 по модул 13 е 3, докато 17 по модул 13 е 4. Така бихме казали, че $42 < 17$, което, очевидно, не е вярно.

Автор: Александър Георгиев