АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА НИНДЖА

Задачата може да се сведе до задача за намиране на най-кратък път в лабиринт със специални правила кога може и кога не може да се преминава през дадена клетка като тези правила зависят от момента на преминаване през клетките в него.

Важно нещо, което трябва да се съобрази, за да се оптимизира решението на задачата е, че героят в задачата (Лъчо) няма нужда да седи в една стая повече от 1 секунда, за да чака промяна на състоянието на осветлението. Ако Лъчо изчака 2 секунди, осветлението ще си е в първоначалното положение и той ще е загубил 2 секунди в безсмислено чакане, което само ще го забави, а все пак в задачата се търси минимално време.

След като се съобрази това, лесно се вижда, че задачата може да бъде решена с динамично оптимиране, тръгвайки от клетка (0,0) и движейки се само надолу и надясно до клетка (N-1,N-1).

Без да се налага да чака смяна на положението на осветлението в стаите, Лъчо може да влезе в текущата клетка от:

- Горната клетка в същата колона или лявата клетка от същия ред, за които секундите, в които е бил Лъчо, са четно число и осветлението на текущата стая в началото е било загасено. Тогава в момента на влизането в текущата стая лампите също ще са загасени.
- Горната клетка в същата колона или лявата клетка от същия ред, за която секундите, в които е бил Лъчо, са нечетно число и осветлението на текущата стая в началото е било светнато. Тогава в момента на влизането в текущата стая осветлението ще стане загасено.

Ако предните стаи не отговарят на тези условия, то Лъчо може просто да изчака една допълнителна секунда, за да се сменят положенията на лампите така че да може да отиде в текущата стая. Тогава обаче това ще му коства още 1 допълнителна секунда.

Спазвайки тези неща, за всяка клетка се търси минимумът от секунди, за които може да се влезе в нея от горната и от лявата клетка, съседни на текущата. Накрая отговорът се намира в броя секунди намерени за клетка (N-1,N-1).

По-надолу следва примерно решение, базирано на техниката динамично оптимиране, използвайки едномерна таблица за запазване на решенията. В това решение първо се попълва първата клетка, след това се попълва първият ред и после за всеки ред първата колона последвано от остатъка от реда. Попълването на едномерната таблица с резултатите от динамичното оптимиране става, спазвайки описаните по-горе правила за движение между стаите.

```
#include <stdio.h>
#define MAXN 1501
#define INIT_STATE_ON 1
#define INIT_STATE_OFF 0
using namespace std;
int dp[MAXN];
int N, initState;
int fromLeft, fromUp;
int main()
{
    scanf("%d", &N);
    scanf("%d", &initState);

    // first row, first column = first second
    dp[0] = 1;
```

```
    // first row
    for (int col = 1; col < N; col++)
    {
        scanf("%d", &initState);

        if (initState == INIT_STATE_ON && (dp[col-1] % 2) == 1)
            dp[col] = dp[col-1] + 1;
        else if (initState == INIT_STATE_OFF && (dp[col-1] % 2) == 0)
            dp[col] = dp[col-1] + 1;
        else dp[col] = dp[col-1] + 2;
    }

    for(int row = 1; row < N; row++)
    {
        // first column
        scanf("%d", &initState);
        if (initState == INIT_STATE_ON && (dp[0] % 2) == 1)
            dp[0] = dp[0] + 1;
        else if (initState == INIT_STATE_OFF && (dp[0] % 2) == 0)
            dp[0] = dp[0] + 1;
        else dp[0] = dp[0] + 2;

        // next columns
        for(int col = 1; col < N; col++)
        {
            scanf("%d", &initState);

            if (initState == INIT_STATE_ON && (dp[col-1] % 2) == 1)
                fromLeft = dp[col-1] + 1;
            else if (initState == INIT_STATE_OFF && (dp[col-1] % 2) == 0)
                fromLeft = dp[col-1] + 1;
            else fromLeft = dp[col-1] + 2;

            if (initState == INIT_STATE_ON && (dp[col] % 2) == 1)
                fromUp = dp[col] + 1;
            else if (initState == INIT_STATE_OFF && (dp[col] % 2) == 0)
                fromUp = dp[col] + 1;
            else fromUp = dp[col] + 2;

            dp[col] =  fromUp < fromLeft ? fromUp : fromLeft;
        }
    }

    printf("%d\n", dp[N-1]);

    return 0;
}
```

Забележка: За примерния вход от условието на задачата едно от възможните положения на Лъчо в стаите (разписано по секунди) може да бъде: (0,0), (0,1), (0,1), (1,1), (2,1), (2,2), (2,2), (2,3), (2,3), (3,3).

Автор: Николай Костов