

# КОНТРОЛНО СЪСТЕЗАНИЕ НА РАЗШИРЕНИЯ НАЦИОНАЛЕН ОТБОР

Група С 7–8 клас

## Анализ на задача Поле

**Първа подзадача.** Тук разполагаме само с една лампа и е достатъчно да проверим всички  $N^2$  възможности. За да е достатъчно бързо трябва да използваме префиксни суми в таблица.  
Сложност –  $O(N \times N)$ .

**Втора подзадача.** С това решение започваме постепенно да развиваме главната идея за цялата задача, а именно динамично програмиране. Фактът, че съществува решение, в което всяка област е осветена от най-много една лампа улеснява стейта на динамичното.

Нека  $dp[n][k]$  е максималната ефективност, която можем да постигнем, ако разглеждаме първите  $n$  реда и колони и имаме  $k$  лампти. Тогава имаме две възможности за най-долната клетка от главния диагонал: не поставяме лампа или поставяме лампа с яркост  $s$ .

$$dp[n][k] = \max(dp[n-1][k], \max(dp[n-s-1][k-1] + \text{sum}(n-s, n)) \mid (0 \leq s \leq n))$$

В частност  $dp[0][k] = 0, dp[n][0] = 0$ . Тук  $\text{sum}(n-s, n)$  е сумата на клетките с реда и колона в интервала  $[n-s, s]$ . Забележете, че с тази формула приемаме, че няма застъпване между осветените региони.

Сложност –  $O(N^2 \times K)$ .

**Трета и четвърта подзадача.** Тези подзадачи са предвидени за всякакви верни решения с полиномиална сложност. Един възможен стейт за динамичното е  $dp[n][k][last]$  – максималната ефективност, ако разглеждаме първите  $n$  реда и колони, имаме  $k$  лампти и всички клетки с ред и колона поне  $last$  са осветени (с някаква лампа, която вече сме поставили). Така можем да пресмятаме сумата само на клетките, които покриваме за първи път с нова лампа:

$$dp[n][k][last] = \max(dp[n-1][k][last], \max(dp[n-s-1][k-1][n] + \text{sum}(n-s, n) - \text{sum}(last, n)) \mid (n - last < s \leq n))$$

В частност отговорът е  $dp[N][K][N+1]$ . Забележете, че в едно оптимално решение няма да има лампа, чийто обсег на осветяване да попада изцяло под обсега на друга лампа, затова приемаме, че всяка нова лампа допринася с някакъв ненулев брой области:  $n - last < s$ .

Сложност –  $O(N^3 \times K)$ .

**Пета подзадача.** Смисъла на тази подзадача е частична идея за пълното решение също да изкара точки, макар и не много тъй като решението от миналата подзадача лесно може да се

# КОНТРОЛНО СЪСТЕЗАНИЕ НА РАЗШИРЕНИЯ

## НАЦИОНАЛЕН ОТБОР

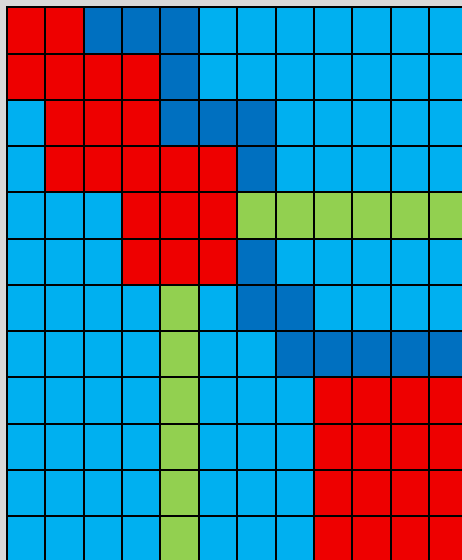
### Група С 7-8 клас

модифицира без параметър за  $k$ . Ще разкажем решение с квадратна сложност, което ще ни помогне и за последната подзадача.

Нека изчислим минималната сума на всички области, които НЕ осветяваме с никоя лампа и я извадим от сумата на всички области. За да я намерим отново ще използваме динамично, но с доста по-различен стейт. Можем да си представим как махайки всички осветени области оставаме с две огледални части спрямо главния диагонал, които могат да се припокриват единствено по него (илюстрацията вдясно).

Червените клетки са осветени, а сините не. Двете сини половини са симетрични спрямо диагонала и са едно значно определени от по-тъмно синята пътечка, която се допира до границите на всички осветени области.

Нека  $dp[row][col]$  е минималната сума на сините области, ако сме фиксирали тъмно синята пътечка до клетка  $(x, y)$  започвайки от  $(n + 1, n + 1)$  – за да е по-лесно и в имплементацията си представяме, че тя огражда и отвън винаги от  $(n + 1, n + 1)$  до  $(0, 0)$ . Искаме винаги да се движим над или по главния диагонал, затова приемаме, че  $row \leq col$ . Тогава имаме:



$$dp[row][col] = \max(dp[row - 1][col] + S, dp[row][col - 1])$$

В частност  $dp[0][col] = 0$ . Тук  $S$  е сумата на клетките, които се включват в синята зона след всяка стъпка по пътеката. Можем да смятаме, че всеки път щом се придвижим нагоре добавяме съответния участък от реда и огледния участък от другата страна (двете зелени линии). Когато се движим наляво не добавяме нищо.

Сложност –  $O(N^2)$ .

**Шеста подзадача.** Да се опитаме да разширим идеята от предната подзадача за ограничен брой лампи.

$$dp[row][col][k] = \max(dp[row - 1][col][k] + S, dp[row][col - 1][k - 1])$$

Отново ползваме същата идея с пътеката, но следим и колко лампи ползваме. Броя лампи съответства на броя пъти, в които правим завой от спускане по редове на спускане по колони с изключение на случаите, в които това е точно по главния диагонал (т.е.  $row = col - 1$  и разглеждаме  $dp[row][col - 1]$ ). Как обаче отчитаме, че правим завой? Нищо не ни пречи да

# КОНТРОЛНО СЪСТЕЗАНИЕ НА РАЗШИРЕНИЯ

## НАЦИОНАЛЕН ОТБОР

### Група С 7-8 клас

добавим още един параметър към динамичното:  $[0/1]$ , който показва в коя посока е направено последното ни движение. Нека 0 значи че сме дошли от  $row + 1, col$ , а 1 значи от  $row, col + 1$ . Тогава пълната формула придобива вида:

$$dp[row][col][k][dir] = \max(dp[row - 1][col][k][0] + S, dp[row][col - 1][k - G][1])$$

Където  $G = 1$ , ако  $row \neq col - 1$  (изключение) и  $dir = 0$ . С това имаме и пълната идея за решение. Единствения останал детайл, е че паметта на тримерното динамично е твърде голяма. Можем да забележим, че спрямо кой да е от 3те големи параметъра се интересваме само от същата стойност и тази с 1 по-малка, съответно можем да приложим оптимизация по памет за някой от тях. В авторовото решение е избран  $row$ .

Сложност –  $O(N^2 \times K)$ .

Автор: Александър Гатев