# Анализ на задача Игра

Задачата е разширение на game (АВ6, НОИ 3, 2014), като интересен факт е, че решението в този по-общ случай не се състои в надграждане с други алгоритми, а с още от същото.

**Първа подзадача.** Разглеждаме всички възможни изходи на играта рекурсивно и определяме кой от тях е с най-голям краен резултат. Сложност – **O( ( N / K )! × N ).**

**Втора подзадача.** На всеки ход трием по един елемент и слепваме получените две редици. Постановката е доста стандартна за един по-специален вид динамично, а именно интервалното динамично. Нека $dp[i][j]$ $(1 \le i \le j \le N)$ е оптималният резултат при игра, развиваща се върху числа в интервала $[i, j]$. Как можем да разбием задачата ни, на по-малки подзадачи от същия вид? Очевидно няма как да изтрием крайните два елемента. Нека $p$ $(i < p < j)$ е индекса на последния елемент, който ще премахнем от дадения интервал. Тогава ще трябва първо да сме изтрили по оптимален начин всички елементи в интервалите $[i + 1, p - 1]$ и $[p + 1, j - 1]$. Но това са подзадачите $dp[i][p]$ и $dp[p][j]$ съответно. Получаваме:

$$dp[i][j] = \max( dp[i][p] + dp[p][j] + a[p] ) - a[i] \times a[j] \quad p \in (i, j)$$

Където $a[i]$ е $i$-тият елемент от редицата. Сложност - **O( N³ ).**

**Трета подзадача.** Допълнителното ограничение задава, че след $(N - 2)/(K - 2)$ хода ще останат само двете крайни числа, което както ще видим по-късно, ни улеснява.

Запазваме идеята от втора подзадача. Тук обаче трием по два елемента от редицата. Едно грешно решение е да пробваме всички двойки $(p, p + 1)$ $i < p < j - 1$ за индекси на последните два елемента, които ще махнем от дадения интервал, защото не е задължително те да са на съседни позиции. Фиксирането пък на всички двойки на не задължително съседни позиции е твърде бавно, но все пак ни кара да се замислим над един въпрос.

Как се справяме с проблема, в който при игра върху някоя от получените три подзадачи $dp[i][p], dp[p][t], dp[t][j]$ $(i < p < t < j)$ накрая остават и някои елементи освен двата крайни, или по-формално казано $j - i - 1$ не се дели на $K - 2$? Истината е, че той е само илюзия. Ако се случи да остават още неизтрити елементи освен 2-та последни, които сме фиксирали, то това е в противоречие с факта, че сме приели, че те могат да са последни (не е трудно да се види, че не всички двойки $(p, t)$ могат да са индекси на последни елементи). Следователно изобщо не трябва да разглеждаме такъв случай! Това означава, че трябва да е спазено условието $K - 2 \,/\, p - i - 1$ за всеки индекс $p$ на последен елемент. И допълнителното ограничение ни помага с това, че по начало $K - 2 \,/\, j - i - 1$.

Сега как да избегнем разглеждането на всички двойки? Да погледнем подзадачите $dp[i-2][j]$ и $dp[i][j]$. И при двете за фиксирано $p$ ще правим едни и същи сметки за $t$. Или по друг начин казано многократно решаваме една съща подзадача от тип „фиксирай по оптимален начин втори последен елемент в интервал $[p, j]$“. Затова в един нов двумерен масив $calc[i][j]$ ще пазим отговорите на тези подзадачи.

$$calc[i][j] = \max(\ dp[i][t] + dp[t][j] + a[t]\ ) \quad t \in (i, j)$$
$$dp[i][j] = \max(\ dp[i][p] + calc[p][j] + a[p]\ ) - a[i] \times a[j] \quad p \in (i, j)$$

Сложност - **O( N$^3$ )**.

**Четвърта подзадача.** Защо да се ограничаваме само до $calc[i][j]$? Със същите разсъждения можем да стигнем до извода, че изчисляваме подзадачата „фиксирай по оптимален начин втори и трети последен елемент в интервал $[p, j]$“. Нека $calc[i][j][k]$ е подзадачата „фиксирай $k$ последни елемента в интервала $[i, j]$”. Лесно получаваме следната рекурента формула:

$$calc[i][j][0] = dp[i][j]$$
$$calc[i][j][k] = \max(\ dp[i][p] + calc[p][j][k-1] + a[p]\ ) \quad p \in (i, j)$$
$$dp[i][j] = calc[i][j][K-2] - a[i] \times a[j]$$

За всеки интервал изчисляваме отговора за всяко $k$, но възможните индекси на последен елемент са през $k-1$, затова крайната сложност е **O( N$^3$ )**.

**Пета подзадача.** Без допълнителното ограничение възниква очевидният проблем, че накрая ще останат повече от два елемента. Не е нещо, което още едно динамично не може да оправи. Идеята е да изчислим оптималния отговор при всички възможности за $N - 2\ mod(K-2)$ елемента, които ще останат накрая заедно с двата крайни. Нека $best[n][k]$ е оптималният резултат при оптимално избрани $k$ елемента, оставащи докрай в интервала $[1, n]$.

$$best[n][k] = \max(\ best[n-1-i*(K-2)][k-1] + dp[n-1-i*(K-2)][n]\ )$$
$$i \in [0, n/(K-2)]$$

Сложност - **O( N$^3$ )**.

**Шеста подзадача.** Можем да забързаме решението приблизително 2 пъти, като го приведем в итеративна вместо рекурсивна форма.

*Автор: Александър Гатев*