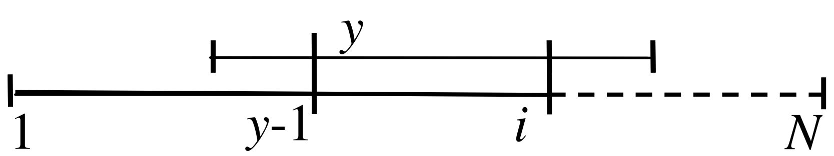
АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА
ПАЛИНДРОМИ

В една редица с 10000 елемента може да има твърде много палиндроми (например, ако всички знаци в редицата са еднакви). Затова нека първо намерим само тези палиндроми, които са *симетрично максимални*, т.е. знакът отляво на палиндрома и знакът отдясно са различни. При търсенето на симетрично максималните палиндроми трябва да имаме предвид, че палиндромите са два вида – с нечетна дължина или с четна дължина. В първия случай заставаме върху всеки от знаците (той е тривиален палиндром) и се опитваме да го разширим наляво и надясно до нечетен симетрично максимален. Във втория случай заставаме между два знака и отново се опитваме да разширим, този път празното множество, до четен симетрично максимален. За всеки намерен симетрично максимален палиндром записваме в масива `pal[2*MAXN][2]` началото и края му. Забележете, че за всеки от N-те знака ще имаме точно един нечетен симетрично максимален, а четните симетрично максимални не може да са повече от N – 1. Тъй като за намирането на един симетрично максимален палиндром ще са необходими между 1 и $N/2$ сравнения, то сложността в най-лошия случай на тази част на алгоритъма ще бъде $O(N^2)$. В сила е следното очевидно

Твърдение 1. *Всеки палиндром на редицата може да се получи от точно един симетрично максимален с премахване на един и същ брой (включително 0) елементи отляво и от дясно.*

Вторият етап на алгоритъма е реализиран по схемата Динамично програмиране. За целта построяваме две линейни таблици – в авторското решение те са разположени в масива `C[MAXN][2]`. В `C[i][0]` ще записваме минималния брой палиндроми, на които можем да разбием частта от подредицата, започваща в началото и завършваща в `i`-тия знак. В `C[i][1]` ще записваме началото на последния палиндром на минималното разбиване за `i`. В началото запълваме таблицата с тривиалното разбиване на N палиндрома, всеки от които е с дължина 1, т.е. `C[i][0]=C[i][1]=i`.



Твърдение 2. *Нека в* `C[p][0]`, `p`=1, 2,…,`i-1` *са намерени минималните разбивания на палиндроми на подредиците започващи в началото и завършващи в позиция* `p`. *Тогава* `C[i][0]`= min `C[y-1][0]+1`, *където минимумът е взет по всички палиндроми започващи в* `y` *и завършващи в* `i`.

В основната стъпка на ДП разглеждаме последователно позициите `i, i=2,3,…,N`. Съгласно *Твърдение 1*, всеки палиндром на редицата се получава от някой симетрично максимален. Затова, за да пресметнем `C[i][0]` за поредната стойност на `i`, за всеки симетрично максимален палиндром определяме симетрично разположен спрямо средата му подпалиндром, който завършва в позиция `i` (ако има такъв). Нека така намерения подпалиндром е с начало `y` (вж. Фигурата). Преизчисляваме `C[i][0]` съгласно *Твърдение 2*: `C[i][0]`= min{ `C[i][0]`, `C[y-1][0]+1` } и ако минимумът е `C[y-1][0]+1` – променяме `C[i][1]=y`.

Броят на изпълнените на втория етап операции е практически същия, като този на първия етап, затова и тук сложността е $O(N^2)$. За да завършим решението, остава да възстановим разбиването от запомнените в `C[i][1]` стойности, като за местата на срязване, T_1, T_2,…, T_P, P = `C[N][0]-1` , имаме T_P = `C[N][1]-1`, T_{P-1} = `C[`T_P`][1]-1` и т.н. Последният етап е със сложност $O(N)$ и затова сложността на решението е $O(N^2)$.

Автор: Красимир Манев