Maxpal-descrierea solutiei

Descrierea soluţiei:

Varianta 1. Rezolvare cu backtracking,

Se vor cauta toate palindroamele de lungime maxima din sir, avand grija sa actualizam numarul de solutii: daca se gaseste o solutie mai buna (de lungime mai mare), numerotarea va incepe de la 1, iar daca e solutie la fel de buna (de aceeasi lungime), atunci se incrementeaza numarul de solutii.

Backtrackingul se poate imbunatati daca stiva se umple alternativ cu pozitii din ambele capete ale sirului cu conditia a[st[2k-1]]=a[st[2k]], prelucrand sirul din ambele capete catre interior.

O astfel de solutie valoreaza 20% din teste.

Complexitate exponenţială.

Varianta 2. Rezolvare cu programare dinamică

Avem şirul x_1, x_2, ..., x_n. Vom folosi două matrici de dimensiuni n x n, şi anume *lmax* (pentru lungimea maximă) şi *nrsol* (pentru numărul de soluţii).

Lungimea maximă a unui subşir palindrom de la o poziţie *st* până la o poziţie *dr* se calculează cu ajutorul următoarei recurenţe

$$lmax(st,dr)=\begin{cases} 2+lmax(st+1, dr-1) & \text{dacă } st<dr, \text{ şi } x_{st}=x_{dr} \\ \max(lmax(st,dr-1),lmax(st+1,dr)) & \text{dacă } st<dr, \text{ şi } x_{st}<>x_{dr} \\ 1 & \text{dacă } st=dr \\ 0 & \text{dacă } st>dr \end{cases}$$

Cu această recursivitate se construieşte o matrice pătratică (triunghiulară), iniţializând diagonala principală cu 1, şi calculând elementele de la stânga la dreapta, pornind de la diagonala principală prima linie, până când se va calcula elementul din colţul din dreapta-sus a matricei *a*.

Elementul *a*[1,*n*]=*lmax*(1,n) va fi răspunsul pentru prima cerinţă.

Numărul soluţiilor distincte se va calcula cu ajutorul tabelului dinamic anterior astfel:

$$nrsol(st,dr) = \begin{cases} nrsol(st+1,dr-1) + nrsol(st+1,dr), & \text{daca } st<dr-1,\ x[st]=x[dr] \text{ si } lmax(st,dr)=lmax(st+1,dr) \\ nrsol(st+1,dr-1) + nrsol(st,dr-1), & \text{daca } st<dr-1,\ x[st]=x[dr] \text{ si } lmax(st,dr)=lmax(st,dr-1) \\ nrsol(st+1,dr-1), & \text{daca } st<dr-1,\ x[st]=x[dr] \text{ si } lmax(st+1,dr)<>lmax(st,dr) \\ & \quad \text{si } lmax(st,dr-1)<>lmax(st,dr) \\ nrsol(st+1,dr) & \text{daca } st<dr,\ x[st]<>x[dr], \text{ si } lmax(st+1,dr)>lmax(st,dr-1) \\ nrsol(st,dr-1) & \text{daca } st<dr,\ x[st]<>x[dr], \text{ si } lmax(st+1,dr)<lmax(st,dr-1) \\ nrsol(st+1,dr)+nrsol(st,dr-1) & \text{daca } st<dr,\ x[st]<>x[dr], \\ & \quad \text{si } lmax(st+1,dr)=lmax(st,dr-1)<>lmax(st+1,dr-1) \\ nrsol(st+1,dr)+nrsol(st,dr-1)-nrsol(st+1,dr-1) & \text{daca } st<dr,\ x[st]<>x[dr], \\ & \quad \text{si } lmax(st+1,dr)=lmax(st,dr-1)=lmax(st+1,dr-1) \\ 1 & \text{daca } st=dr \\ 1 & \text{daca } x[st]=x[dr]\ \ st=dr-1 \\ 0 & \text{daca } st>dr \end{cases}$$

Răspunsul pentru a doua cerinţă va fi *nrsol(1,n)*.

O astfel de abordare a problemei din cauza restricţiilor de memorie va obţine cel mult 60 de puncte.

Complexitate O(n^2).

Varianta 3. Optimizarea variantei 2.

Datorită faptului că cele două matrice *lmax* şi *nrsol* sunt triunghiulare, datele din ele se pot stoca într-o singură deasupra şi sub diagonala principală.

Astfel se obţine o soluţie de 70 de puncte.

Complexitate O(n^2).

Varianta 4. Optimizarea variantei 2.

Folosindu-ne de proprietatea că elementele matricelor *lmax* şi *nrsol* se pot calcula pe linii paralele cu diagonala principală folosindu-ne de elemente de pe două linii paralele anterioare, problema se poate rezolva cu 6 şiruri ajutătoare în loc de două matrice.

Această soluţie va obţine 100 puncte.

Complexitate O(n^2).