АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА ПАЛИНДРОМИ

Означаваме буквите от входния низ с s[0], s[1], …, s[n-1].

Запълваме таблицата p[i][j] за i <= j със стойности true или false в зависимост от това, дали отрезът от низа s[i..j] е палиндром или не е. Запълването става като първо положим p[i][i]=true за всяко i=0,1,…,n-1, което означава, че всеки еднобуквен низ е палиндром. След това запълваме стойностите p[i][i+1] за i=0, 2, …, n-2 с true само когато s[i]=s[i+1].

След това запълваме последователно стойностите p[i][j] с разлика на индексите j-i равна на 2,3, .., и т.н. като имаме предвид, че s[i..j] е палиндром тогава и само тогава, когато s[i]=s[j] и s[i+1…j-1] е палиндром.

Така за всеки подниз може бързо да отговорим на въпроса дали е палиндром или не е.

Следващата стъпка е прилагане на метода на динамичното оптимиране. Нека c[j] означава минималният брой разделители, с които се решава задачата за подниза s[0..j]. Очевидно c[0]=0 и, ако s[0..j] е палиндром, то отново c[j]=0. За да пресметнем c[i] за i=j+1, разглеждаме всичко поднизове от вида s[k..i], където k=1, 2, …, i, и за тези k, за които s[k..i] е палиндром, намираме минимума на стойностите c[k]+1 и полагаме c[i] да е равно на този минимум. Така всъщност минималният брой разделители за подниза s[0..i] получаваме като вземем минималния брой разделители, който е най-добър при премахнат накакъв палиндром от позиция i назад донякъде към началото на s.

Отговорът на задачата е стойността c[n-1]. Сложността и паметта за пресмятанията са O(n*n).

Автор: Емил Келеведжиев