# Kpal descriere soluție

*Prof. Dana Lica, Centrul Județean de Excelență Prahova*

Notăm lungimea cuvântului egală cu **L**.
Vom începe prin a calcula **CostMatch[i][j]** = costul minim pentru a schimba a **i-**a și a **j-**a litera a alfabetului astfelîncât acestea să devină egale. Pentru o pereche **(i,j)** avem:
**CostMatch[i][j]** = **min(Cost[i][k] + Cost[j][k])**, pentru **1**≤**k**≤**X**.
Cu alte cuvinte, iteram prin orice literăîn care pot fi tranformate **i**și **j**. Aceasta matrice poate fi calculatăîn **O(X^3).**

Având matricea **CostMatch**, putem încerca toate modalitățilede a împărți cuvântul inițial în șiruri de lungimi egale. Aceste lungimi sunt divizorii lui **L**. Pentru o lungime fixată, fiecare secvență obținută prin tăieturi trebuie schimbată individual într-un palidrom. Costul de a schimba o astfel de secvență într-un palidrom poate fi calculat comparând toate pozițiile simetrice față de mijloc și adunând **CostMatch**-ul corespunzator acestora. De exemplu, pentru cuvântul **aabbbc** și lungimea **3**, obținem secvențele **aab** și **bbc**. Costul lui **aab** este de fapt costul de a schimba atât primul **a** cât și **b**-ul în aceeasi literă. În acest caz, costul secvenței **aab** este **CostMatch['a']['b']**. Similar, costul secvenței **bbc** este **CostMatch['b']['c'].**

În continuare vom face următoarea observație. Presupunem că un cuvânt inițial poate fi împărțit în palindroame de lungime **x** având un cost total egal cu **cost(x)** și în palidroame de lungime **y** având cost total egal cu **cost(y)**. Dacă **x < y** și **cost(x)**≥**cost(y)** atunci va fi mereu preferat să împărțim cuvantul în palindroame de lungime y - Cu alte cuvinte, dacă o împărțire generează mai multe palindroame la un cost mai mare, atunci acea împărțire este neoptimă. Folosind această observație, vom itera prin toate modurile de a împărți cuvântul inițial în ordine crescătoare după lungimea palindroamelor obținute. Vom menține pe parcurs o stivă crescătoare $x_1$, $x_2$, … $x_k$ ($x_i$ reprezintă a **i**-a lungime) astfel încât $x_1 < x_2 < \ldots < x_k$și **cost**($x_1$) <**cost**($x_2$)< … <**cost**($x_k$). La fiecare pas al iterației **x**,dorim să adaugăm pe **x** în stivă. Ne uităm la vârful stivei, verificând dacă**cost**($x_k$) ≥**cost(x)**.Într-o astfel de situație, $x_k$-ul poate fi eliminată deoarece **x**-ul este o împărțire mult mai bună. Se repeta procesul până când stiva devine goala sau până ce vârful nu oferă o împărțire neoptimă. Dupa aceea, **x**-ul este adăugat ca vârf al stivei.

La final, vom avea stiva $x_1$, $x_2$, … $x_K$. Să presupunem că dorim să calculăm numărul minim de palindroame în care poate fi impărțit cuvântul inițial folosind schimbări de un cost total≤ **c**. Pentru aceasta, trebuie să gasim valoarea maximă din stivă$x_T$, astfel încât cost($x_T$) ≤**c**. În acest fel, pentru **c** considerat, numărul minim de palindroame este L/$x_T$. Ultima observație este faptul că orice **c** am alege între **cost**($x_T$) și **cost**($x_{T+1}$) **- 1** are răspunsul **L/**$x_T$.Tot ce ne rămâne acum de făcut este să împărțim intervalul **[0..Q]** folosind intervalele **[cost(**$x_T$**)..cost(**$x_{T+1}$**)-1]** și să calculăm valoarea dorită.

De exemplu, să presupum că avem **L** = 12, **Q** = 80 și stiva (1, 0) (2, 10) (3, 20) (4, 100) (12, 101) - unde primul element al perechii reprezintă lungimea palindroamelor tăiate, iar al doilea reprezintă costul. Dacă suntem interesați de un cost maxim **c** = 21, este evident că preferăm să împărțim în palindroame de lungime 3. Dacă suntem interesați de cost maxim **c** = 19,

atunci nu putem împărți mai bine de palidroame având lungimea 2. Acum, ultima observație ne spune că orice limita **c** am avea între 20 și 99 (spre exemplu), atunci cel mai bine este să împărțim în 4 palindroame de lungime 3.
Împărțim intervalul 0..80 în intervalele următoare - în funcție de costurile din stivă:

1. [0 .. 9] cu răspunsul pe orice element din interval:lungime=1
2. [10 ..19] cu răspunsul: lungime=2
3. [20..80] cu răspunsul: lungime=3

În total, răspunsul la exemplu este (9 - 0 + 1) * (12 / 1) + (19 - 10 + 1) * (12 / 2) + (80 - 20 + 1) * (12 / 3).