

## Ziar – Descrierea soluției

### Cerința 1

Putem genera vectorul de frecvență a literelor pentru fiecare cuvânt. Două cuvinte fac parte din același grup dacă au același vector de frecvență. Se parcurge lista de cuvinte. Pentru fiecare cuvânt se caută toate cuvintele cu care acesta este într-un grup și se marchează poziția acestor cuvinte, pentru a nu mai fi parcurse. Se elimină astfel un grup din lista de cuvinte. Procedul trebuie repetat până când toate cuvintele vor fi marcate.

Această soluție are complexitatea  $O(N^2)$  și obține 2·9 de puncte din cele 5·9 alocate pentru această cerință. O soluție mai eficientă care obține punctaj maxim ar fi următoarea:

Se reordonează literele fiecărui cuvânt astfel încât să fie în ordine crescătoare în interiorul cuvântului (ex. dacia →aacdi). Astfel, cuvintele din același grup devin identice. Se sortează lista de cuvinte modificate. Numărul grupurilor va fi numărul elementelor diferite ale listei sortate (care se poate determina într-o singură parcurgere a listei), sau echivalent, numărul de subsecvențe de lungime maximă formate de cuvinte identice. Complexitate:  $O(N \cdot \log(N))$ .

### Cerința 2

Un cuvânt poate fi transformat într-un palindrom dacă și numai dacă există cel mult o literă care apare de un număr impar de ori în acest cuvânt. Putem verifica această proprietate parcurgând lista de cuvinte și analizând vectorul de frecvență a literelor pentru fiecare cuvânt. Complexitate:  $O(N)$ .

## Caractere – Descrierea soluției

Prima dată verificăm dacă șirul citit satisface deja condiția cerută.

Dacă nu, o soluție poate fi găsită pentru orice caz (în care  $n \geq 4$ ) folosind următorul algoritm:

1. Parcurgem caracterele din șir de la stânga la dreapta.
2. Fie  $p$  următoarea (inițial, prima depistată) poziție pe care nu se află caracterul '<'.  
 3. Dacă  $p = n$  algoritmul se termină, șirul satisface condiția cerută.
4. Altfel, dacă  $p < n$ , vom muta grupul '\*\*' astfel încât prima steluță să fie pe poziția  $p$ .
  - a. Dacă grupul '\*\*' se află deja pe pozițiile  $(p, p+1)$ , nu avem nimic de făcut.
  - b. Dacă primul caracter al grupului '\*\*' se află la o distanță de cel puțin 2 față de  $p$ , avem de făcut o singură mutare:  $p$ . (Codificăm mutările conform formatului dat la descrierea datelor de ieșire, adică prin „mutarea  $p$ ” înțelegem mutarea în care grupul '\*\*' se schimbă cu cele două caractere care se află pe pozițiile  $(p, p+1)$ .)
  - c. Altfel, grupul '\*\*' se află pe pozițiile  $(p+1, p+2)$ . Efectuăm prima dată mutarea  $(2 \cdot n - 1)$ , adică ducem grupul de stelute la sfârșitul șirului (această mutare este validă, deoarece avem  $p < n$  și  $n \geq 4$ , de unde rezultă că  $p+2 < 2 \cdot n - 1$ , adică cea de-a doua stelută sigur nu se află pe poziția  $2 \cdot n - 1$ ). Putem continua acum cu mutarea  $p$ , deoarece acum distanța dintre poziția  $p$  și poziția primei stelute  $(2 \cdot n - 1)$  este mai mare decât 2.
5. Căutăm primul caracter '<' aflat după poziția  $p$ . Fie  $u$  poziția acestuia.
6. Dacă  $u < 2 \cdot n$ , putem efectua mutarea  $u$ , și astfel poziția  $p$  devine ocupată de un caracter '<'. Continuăm de la pasul 2.
7. Altfel avem  $u = 2 \cdot n$ , deci am ajuns la ultimul caracter '<' care trebuie mutat înaintea caracterelor '>'. În acest caz  $p = n - 1$  și este validă următoarea secvență de mutări:  $2 \cdot n - 1, p+2, p-1$ . După aceasta, algoritmul se termină.

Pentru  $n = 4$  și  $n = 5$  ilustrăm efectele mutărilor de la pasul 7 în următoarele tabele:

După mutarea:	Șirul:
	<<*>>><
$2 \cdot n - 1$	<<><>>**
$p+2$	<<<<*>>
$p-1$	<<*<<>>

După mutarea:	Șirul:
	<<<*>>><
$2 \cdot n - 1$	<<<><>>>**
$p+2$	<<<<<*>>>
$p-1$	<<*<<>>>