

## Problema origami - descriere soluție

Autori: Lucian Bicsi, Eugenie-Daniel Posdărăscu

### Soluție 1

O primă soluție constă în programarea dinamică:  $OK[i1][j1][i2][j2] = \text{true}$  dacă se poate forma submatricea cu colțurile în  $(i1, j1)$  și  $(i2, j2)$ . Popularea dinamicii se face de la mare la mic și se folosește de verificarea progresivă a egalității unor submatrice. Soluții de acest tip au complexități între  $O(M^6)$  și  $O(M^4)$  și nu ar trebui să obțină mai mult de 30 de puncte.

### Soluție 2

O altă soluție este să pornim de la matricea inițială și să o împăturim în toate modurile valide posibile (sau, optimizat, doar în primul mod pe cele patru direcții) și să introducem în coadă noile submatrice. Complexitatea acestei soluții depinde de numărul de soluții:  $O(X * M)$  și ar trebui să obțină circa 40 de puncte.

### Soluție 3

Observația cheie este că **răspunsul este independent pe linii și pe coloane**. Mai exact, operațiile pe linie și cele pe coloană se pot efectua (interclasa) în orice ordine. Putem astfel reduce la problema pe cazul 1D, în care valorile noilor vectori sunt “comprimări” ale liniilor, respectiv coloanelor matricei. La sfârșit, se vor înmulți cele două rezultate.

Adaptarea soluțiilor 1 și 2 pe aceste șiruri pot conduce la punctaje de până la 70.

Pentru rezolvarea completă, este necesară o altă observație foarte frumoasă:

Să considerăm  $OK(i, j) = \text{true}$  dacă și numai dacă se poate forma subsecvența  $[i \dots j]$ . Observația este:  **$OK(i, j) = \text{true}$  dacă și numai dacă  $OK(1, j) = \text{true}$  și  $OK(i, n) = \text{true}$ . Mai mult, operațiile pe stânga și cele pe dreapta sunt independente.**

Demonstrația se bazează pe un raționament inductiv și pe câteva proprietăți ale palindroamelor și nu o vom prezenta în cadrul acestui document.

Rămân de calculat  $OK(1, i) = \text{Pref}[i]$  și  $OK(i, n) = \text{Suff}[i]$ . Pentru a computa  $\text{Pref}[]$ , ne folosim de cazul de bază  $\text{Pref}[n] = \text{true}$  și programarea dinamică de la dreapta la stânga și avem că  $\text{Pref}[i] = \text{true}$  dacă și numai dacă  $\text{Pref}[i'] = \text{true}$  pentru măcar un  $i'$  în intervalul  $[i \dots i + \text{MaxPal}[i]]$ , unde  $\text{MaxPal}[i]$  este lungimea celui mai lung palindrom centrat între pozițiile  $i$  și  $i + 1$  și se poate calcula în  $O(n \log(n))$

cu hashuri + căutare binară (97 de puncte) sau în timp liniar cu algoritmul Manacher (100 de puncte).

Computarea șirului `Suff[]` se face în mod similar.

La final, numărarea soluțiilor se face printr-un algoritm ușor tip “baleiere”.

**Complexitate finală:**  $O(N * M + N + M) = O(N * M)$