Soluții pentru problema *Puzzle Bila*

Lotul Național de Informatică, Craiova 2025

Subtask 1 ($1 \leq N, M \leq 40$)

Vom folosi programare dinamică pentru a rezolva problema. Fie $dp_{i,j}$ numărul minim de shift-uri pe care trebuie să le facem pentru a ajunge din $(0, 0)$ în (i, j). Să considerăm că avem deja calculată dinamica pentru linia i din matrice și vrem acum să trecem la linia $i + 1$. Vom încerca un drum de la celula (i, l) la celula $(i + 1, r)$, unde $l \leq r$. Pentru a face asta, vrem să dăm shift liniei $i + 1$, astfel încât segmentul de pe linia $i + 1$ de la poziția l până la poziția r să conțină doar zerouri. Fie această valoare $cost(i+1, l, r)$.

În cazul acestui subtask, putem doar să fixăm fiecare secvență de lungime $r - l + 1$ plină de zerouri de pe linia $i + 1$ și să calculăm distanța până la intervalul $[l, r]$.

Formula de recurență este, deci

$$dp_{i+1,r} = \min(dp_{i,l} + cost(i + 1, l, r)),$$

iar complexitatea timp este $O(NM^3)$.

Subtask 2 ($1 \leq N, M \leq 500$)

Pentru al doilea subtask, putem să reducem complexitatea calculării funcției *cost*. Este suficient să precalculăm pentru fiecare linie o structură care ne spune, pentru fiecare poziție j, unde găsim capătul celei mai apropiate secvențe de lungime *len*. Putem precalcula această structură în $O(NM^2)$. Prin urmare, putem să calculăm $cost(i + 1, l, r)$ în $O(1)$, iar complexitatea finală devine $O(NM^2)$.

Subtask 3 ($1 \leq N, M \leq 50000$)

Vom încerca să optimizăm soluția de mai sus. Mai exact, când calculăm $dp_{i+1,r}$, vrem să găsim mai rapid l-ul de pe linia i care relaxează cel mai mult dinamica.

O primă observație este că, dacă ne uităm la secvențele compacte de zerouri de pe o linie, vom avea cel mult $O(\sqrt{M})$ lungimi distincte ale acestora.

Așadar, vom încerca să calculăm, pentru fiecare triplet de forma $(i+1, r, len)$, care este l-ul optim de pe linia i dacă știm că vrem să calculăm $dp_{i+1,r}$ aducând un bloc de lungime exact len peste poziția r. Bineînțeles, ne vom uita doar la cel mai apropiat bloc de lungime len din dreapta poziției r și la cel mai apropiat bloc de lungime len din stânga poziției r.

Să considerăm că aducem blocul din stânga și că acesta se termină la poziția k. Atunci, vom putea relaxa dinamica după următoarea formulă:

$$dp_{i+1,r} = k - r + \min(dp_{i,j}),$$

unde $r - len + 1 \leq j \leq r$.

Să considerăm acum că aducem blocul din dreapta și că acesta se termină la poziția k. Atunci, putem să relaxăm dinamica după următoarea formulă:

$$dp_{i+1,r} = \min\left(\min(dp_{i,j}), \min(dp_{i,j'} + k - r + 1 - j')\right),$$

unde $k - len + 1 \leq j \leq r$, iar $r - len + 1 \leq j' \leq \min(r, k - len)$.

Toate cele de mai sus se pot calcula în $O(1)$, după o precalculare în $O(M \log M)$ pentru fiecare linie, folosind *sparse tables*.

Prin urmare, complexitatea finală devine $O(NM\sqrt{M} + NM \log M)$.