

Hashtag - Descrierea soluției

*stud. Okros Alexandru,
Universitatea Politehnica Timișoara*

Varianta 1 (35-40 puncte) - Complexitate $O(2^N)$ (Mircea Lupșe-Turpan)

Se utilizează un algoritm backtracking. Se generează un șir binar, în care valoarea 0 corespunde mutării pionului lui Yolo, iar valoarea 1 corespunde mutării pionului lui Swag. O soluție parțială este validă dacă diferența dintre numărul de mutări al lui Yolo (numărul de zerouri) și numărul de mutări al lui Swag (numărul de valori de 1) este mai mică decât $K-1$, unde $K-1$ reprezintă diferența inițială de căsuțe dintre Yolo și Swag. O soluție parțială este și soluție dacă numărul de mutări al lui Swag este $N-K$ (diferența dintre poziția N și poziția inițială a lui Swag), adică numărul de mutări necesare lui Swag pentru a ajunge pe poziția N .

Varianta 2 (35-40 puncte) - Complexitate $O(2^N)$ (Mircea Lupșe-Turpan)

Se poate optimiza algoritmul de mai sus prin renunțarea la generarea șirului binar și prin utilizarea a două variabile `YoloMoves` și `SwagMoves` care rețin în orice moment numărul de mutări al lui Yolo, respectiv numărul de mutări al lui Swag. În acest fel verificarea dacă o soluție parțială este validă, respectiv verificarea dacă o soluție parțială este soluție se face prin compararea acestor 2 variabile.

Varianta 3 (40 puncte) - Complexitate $O(2^N)$

Se bazează pe programare dinamică. Se scrie funcția $L(A, B)$ care calculează numărul de jocuri câștigate de Swag dacă Yolo începe pe poziția A și Swag pe poziția B .

Recurența este $L(A, B) = L(A+1, B) + L(A, B+1)$. Trebuie să calculăm $L(1, K)$, deoarece Yolo începe pe poziția 1, iar Swag pe poziția K .

Varianta 4 (100 puncte) - Complexitate $O(N^2)$

Este optimizarea soluției precedente folosind memoizare. Valorile deja calculate se rețin într-o matrice, astfel că se evită calcularea unor valori de două ori.

Complexitatea de memorie $O(N^2)$.

Varianta 5 (100 puncte) - Complexitate $O(N^2)$

Folosim programare dinamică. $DP[i][j]$ = numărul de jocuri câștigate de Swag dacă el e pe poziția j , iar jocul se termină pe poziția i . Recurența: $DP[i][j] = DP[i][j+1] + DP[i-1][j-1]$ (primul caz Swag se mută o căsuță la dreapta, cazul doi Yolo se mută o căsuță și astfel Swag și finish-ul se aproprie cu o căsuță, de unde “-1” la parametri).

Complexitatea de memorie $O(N^2)$.

Varianta 6 (100 puncte) - Complexitate $O(N^2)$

La soluția precedentă folosim doar ultimile două linii ale matricei și observăm că putem obține număr de operații $N \cdot (N-K)$ deoarece nu trebuie calculate toate valorile din matrice (rezultă din recurență).

Prin utilizarea unei matrice cu 2 linii și N coloane, complexitatea de memorie scade la $O(N)$.