PIRU&ETE: descrierea solutiei

- Cerinta:

Dandu-se **N** si **K**, in cate moduri din cele $\binom{2N}{K}$ pot plasa obstacole astfel incat dupa o reprezentatie completa de **T** pasi dansatorul sa se fi reintors in punctul **C=0** ? (mod $\mathbf{10^9+7}$)

- *Solutie brut* O($N!$)

Generez toate cele $\binom{2N}{K}$ variante posibile si simulez miscarile.

- *Solutie dinamica* O($N^4 * K$)

Tin dinamica pe urmatoarele stari:

DP[time(modulo 2)][pozitie_curenta][directie][explorat_st][explorat_dr][nr_obstacole]

Prin **explorat_dr** se intelege pentru cate pozitii din dreapta am certitudinea disparitiei sau a lipsei obstacolelor. Analog pentru **explorat_st**.

Fie ***R = 2*M - explorat_st - explorat_dr*** ce reprezinta **numarul de obstacole incerte** unei solutii obtinute din dinamica pentru un interval explorat de forma **[explorat_st, explorat_dr]**. Se observa ca pozitiile astfel ramase pot contine sau nu obstacole fara ca acestea sa afecteze dansatorul. Asadar este suficient si necesar sa consideram ca solutii finale toate cele $\mathbf{2^R}$ configuratii posibile. Acest lucru se poate realiza usor incrementand vectorul solutie incepand cu pozitia **K** cu vectorul **combinarilor** de **R**.

- *Solutie dinamica* O($N^2 * K * T$)

Solutia este aceeasi cu precedenta, dar se foloseste de observatia ca nu este neaparat necesar sa retinem **pozitie_curenta** ci putem privi miscarile ca niste salturi cu diferite costuri in urma carora dansatorul se va intoarce mereu in origine.

- *Solutie dinamica* O($T^2 * \sqrt{T} * K$) pentru **T <= 2 * N + 2**

Ne gandim sa reformulam problema in urmatorul mod pentru fiecare din cele doua directii: In cate moduri pot alege **K** obstacole in asa fel incat suma distantelor lor sa fie **S**, iar cel mai departat pe care il voi atinge sa fie la distanta cel mult **X**. Restrictia asupra lui **X** ne intereseaza pentru a trata cazurile privind **numarul de obstacole incerte** descrise mai sus.

Ne gandim sa precalculam urmatoarea dinamica: **P [X][S][K]** = in cate moduri se pot alege **K** numere distincte a caror suma sa fie **S**, si niciunul dintre ele sa nu depaseasca **X**. Recurenta este data de observatia ca solutiile pentru **X** provin mereu dintr-o shiftare cu **"1"** la dreapta a solutiilor pentru **X-1**, singura incertitudine ramasa fiind aceea a adaugarii sau nu a unui nou **"1"** la inceputul sirului solutie.

P[X][S][K] = P[X-1][S-K][K-1] + P[X-1][S-K][K]

Pentru a combina cele doua directii vom considera fixate doua capete ce descriu ultimele obstacole atinse pe cele doua directii pentru ca apoi sa putem incrementa vectorul solutie printr-un sir de combinari exact ca in solutiile precedente. De asemenea suntem nevoiti sa iteram atat distribuita sumei **N+1** pe directii, cat si numarul de obstacole atinse (toate <= **K**) spre a putea lua in considerare toti incrementii posibili pentru **K**. Ne vom folosi si de observatia ca numarul de obstacole atinse ce pot genera solutii este mereu cel mult $\sqrt{T}$, de unde si complexitatea finala a solutiei. O ultima observatie care reduce foarte mult numarul de operatii efectuate este aceea ca suma pentru o directie este cel mereu cel putin $\frac{K*(K+1)}{2}$ pentru un **K** fixat.

- *Solutie dinamica* O($T^2 * \sqrt{T} * N$)

Folosind dinamica **P[X][S][K]** de la solutia anterioara se poate obtine o adoua dinamica **D[V][S][K]** = in cate moduri se pot alege **V** numere ale caror suma este **S**. **K** din ele fiind distincte si mai mici ca **N + 1**, iar celelale **(V – K)** egale cu **N + 1**.

K trebuie neaparat sa fie mic (ca si la pasul anterior) si iar V poate fi cel mult egal cu **T**, deci calculul acestei dinamici se face in complexitate O($\sqrt{T} * T^2 * N$).

Dupa pentru a combina cele doua directii este nevoie sa fixam in partea stanga ce suma se obtine (**S1**) cu cate elemente (**V1**) si cate obstacole se construiesc (**K1**) si pentru partea dreapta va fi suma **T – S**, fie **V1**, fie **V1 + 1** elemente si **K – K1** obstacole. Complexitatea acestui pas este O($T^2 * K$).