

## Analiza Inițială

Limita de timp astronomică de  $R = 10^{20}$  secunde sugerează imediat că o simulare directă (cum ar fi un algoritm Dijkstra pe un graf infinit sau foarte mare) este exclusă.

De asemenea, cerința de a furniza un răspuns cu o eroare relativă de  $10^{-6}$  ne indică faptul că problema nu cere un număr întreg exact, ci o aproximare foarte precisă a ariei explorabile.

## Observații Cheie

### 1. Periodicitatea Reliefului

Relieful se repetă periodic la fiecare  $n$  unități. Aceasta înseamnă că structura locală a costurilor este aceeași în fiecare "bloc" de  $n \times n$ . Pe măsură ce distanța față de origine crește, comportamentul drumului minim devine asimptotic liniar.

### 2. Convergența către o Formă Geometrică

Pentru valori extrem de mari ale timpului  $R$ , mulțimea punctelor  $(x, y)$  accesibile formează o figură geometrică convexă scalată. Într-un mediu discret (grid), numărul de puncte cu coordonate întregi din interiorul unei figuri foarte mari este aproximativ egal cu aria acelei figuri.

## Reducerea Matematică

Fie  $T(i, j)$  costul minim de a ne deplasa de la o celulă  $(x, y)$  la celula  $(x + i \cdot n, y + j \cdot n)$ . Dacă privim problema la scară macro, viteza de propagare în direcția  $(i, j)$  este invers proporțională cu acest cost.

Definim un set de puncte în planul 2D sub forma:

$$P_{i,j} = (i / (T(i, j)/n), j / (T(i, j)/n))$$

Aceste puncte reprezintă "distanța" maximă parcursă într-o unitate de timp în diverse direcții. Învelișul convex al acestor puncte definește un poligon  $P$  care descrie zona accesibilă într-o secundă.

- Calcularea costurilor locale:** Rulăm un algoritm Dijkstra dintr-un set de puncte de start (pentru a acoperi toate fazele posibile față de grila  $n \times n$ ). Limităm explorarea la o distanță Manhattan de  $2n$  pentru a captura costurile optime de tranziție între blocuri adiacente.
- Extragerea tranzițiilor:** Determinăm  $T(i, j)$  pentru toate perechile  $|i| + |j| \leq n$ . Acestea vor fi "vectorii de bază" ai mișcării noastre macroscopice.
- Construirea Poligonului:** Generăm punctele  $P_{i,j}$  și calculăm învelișul convex (Convex Hull) folosind un algoritm precum *Monotone Chain* sau *Graham Scan*.
- Calculul Ariei:** Calculăm aria poligonului convex.