

Dokumentacija

Osnove

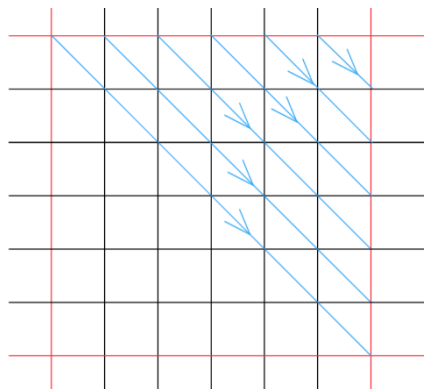
Ovaj projekat sadrži 3 funkcije:

1. `int broj_puteva(int, int);`
2. `int broj_puteva(int, int, vector<vector<bool>>&);`
3. `int broj_puteva(int, int, vector<int>&);`

Funkcija 1

Ova funkcija radi tako što u napravi matricu dp i u polje $dp[i][j]$ stavi broj načina da se dođe do polja $[i][j]$. Pošto se možemo kretati na polja $([i + 1][j + 1], [i][j + 1], [i - 1][j + 1])$ na polje $[i][j]$ možemo doći sa polja $[i + 1][j - 1], [i][j - 1], [i - 1][j - 1]$ pa je relacija sljedeća:

$$dp[i][j] = dp[i + 1][j - 1] + dp[i][j - 1] + dp[i - 1][j - 1].$$



U koliko neki od ovih prethodnika ne postoje stavimo da je 0. Matricu popunjavamo kao na slici ispod:

Kompleksnost je $O(m \cdot n)$ jer samo prolazimo kroz matricu konačan broj puta.

Funkcija 2

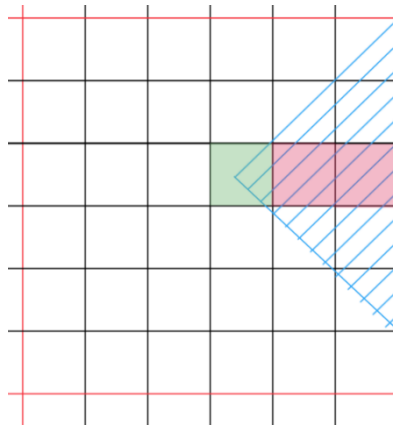
Radi isto kao i funkcija 1 samo na zidove se stavi 0 i nikad se ne računa $dp[i][j]$ ako je na tom mjestu zid.

Kompleksnost je očigledno ista kao i za funkciju 1 $O(m \cdot n)$

Funkcija 3

Ova funkcija radi slično kao funkcija 1 samo u k -toj iteraciji $dp[i][j]$ predstavlja na koliko načina se može doći do $[i][j]$ tako da se pređe preko prvih k redova zadanih u vektoru redoslijedom kojim su zadani. Neka je vektor<int> indeks = $\{2,5,1\}$. Funkcija 3 će da prvo da odredi na koliko načina se može doći do svakog $[i][j]$ pa će zatim odrediti na koliko načina se može doći

do $[i][j]$ tako da se pređe preko 2. reda, zatim da pređe preko 2. pa onda preko 5. i na kraju da pređe preko sva 3 reda zadanim redoslijedom. Broj puteva do $dp[i][j]$ sa uslovom da se posjeti neki od redova se računa tako što označimo prvi (onaj sa najmanjim brojem kolone, na slici to je



zeleni) element tog reda u koji možemo doći. Nacrtamo dijagonalu od tog elementa prema gore i dole kao na slici, te sve elemente koji se ne nalazi unutar tih dijagonala postavimo na 0 jer do njih ne možemo doći. (na slici su to elementi koji nisu osjenčeni plavom bojom). Ostali elementi u tom redu ne mijenjaju svoju vrijednost a sve ostale elemente računamo na naveden način u funkciji 1.

Vremenska kompleksnost ovoga je $O(I.size \cdot m \cdot n)$ gdje je I niz redova koji se mora posjetiti jer za svaki $i \in I$ vršimo funkciju 1. Prostorna kompleksnost svih funkcija je $O(m \cdot n)$ jer sve radimo na istoj matrici.