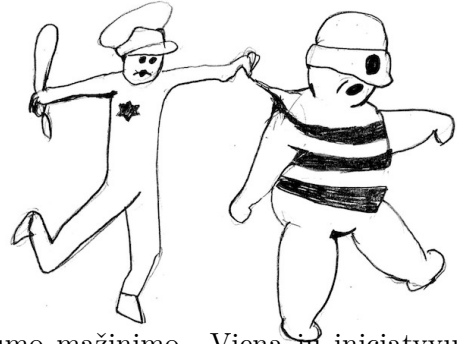


## Policininkas ir plėšikas

Bitangoje nusikalstamumo lygis muša visų laikų rekordus. Be kitų nusikaltimų, kasdien vyksta apiplėšimai. Kaskart įvykus nusikaltimui, siaurais skersgatviais, kurie jungia įvairius miesto užkampius, sprunkantį plėšiką tenka gaudyti vienišam policininkui. Deja, dažniausiai plėšikai pasprunka nuo persekiotojų, kadangi jie miestą pažįsta žymiai geriau už pareigūnus.



Bitangos miesto policijos departamentas (BMPD) organizuoja aukščiausio lygio susitikimą dėl nusikalstamumo mažinimo. Viena jų iniciatyvų yra persekioti plėšikus į pagalbą pasitelkiant kompiuterį. Tam BMPD sudarė tikslų miesto žemėlapi. Dabar jiems reikia programinės įrangos, kuri sukurtų persekiojimo strategijas.

Vieno pareigūno persekiojančio vieną plėšiką uždavinys modeliuojamas taip:

1. Policijos pareigūnas pasirenka vieną užkampį ir ima jame patruliuoti.
2. Tada plėšikas pasirenka užkampį apiplėšimui (jis žino, kur yra pareigūnas). Daroma prielaida, jog nuo šios akimirkos ir pareigūnas, ir plėšikas žino vienas kito buvimo vietą.
3. Policijos pareigūno ėjimo metu jis nukeliauja į gretimą užkampį (t. y. užkampį, kurį su dabartiniu užkampiu jungia skersgatvis) arba laukia (t. y. nepajuda niekur).
4. Plėšiko ėjimo metu jis nukeliauja į gretimą užkampį. Atkreipkite dėmesį, jog priešingai negu policininkas, plėšikas laukti negali. Instinktas jiems liepia nepaliaujamai bėgti.
5. Policijos pareigūnas ir plėšikas vienas po kito atlieka ėjimus (pirmas eina pareigūnas) kol kuri nors iš šių sąlygų tampa patenkinta:
  - (a) situacija pasikartoja (situacija yra apibrėžta abiejų dalyvių pozicijomis bei kieno eilė yra daryti ėjimą). Tai reiškia, jog plėšikas sugebės išvengti policijos kiek norima ilgai, taigi plėšikas pasprunka;
  - (b) policijos pareigūnas ir plėšikas susitinka tame pačiame užkampyje po bet kurio iš jų ėjimo. Šiuo atveju policijos pareigūnas pagauna plėšiką.

## Užduotis

Turite parašyti programą, kuri pagal duotą miesto žemėlapi nustatytų, ar įmanoma pagauti plėšiką, ir jeigu tai yra įmanoma, pagautų jį darydama ėjimus už policijos pareigūną.

Jūsų programa turi daryti prielaidą, jog plėšikas juda optimaliai.

## Realizacija

Turite realizuoti dvi funkcijas:

- **start**( $N$ ,  $A$ ), kuri priima šiuos argumentus:
  - $N$  – užkampių skaičius (užkampiai yra sunumeruoti nuo 0 iki  $N - 1$ );

- $A$  – dvimatis masyvas, apibūdinantis skersgatvius: kiekvienam  $0 \leq i, j \leq N - 1$ ,

$$A[i, j] \text{ is } \begin{cases} \text{false} & \text{jeigu } i \text{ ir } j \text{ nejungia skersgatvis} \\ \text{true} & \text{jeigu } i \text{ ir } j \text{ jungia skersgatvis} \end{cases}$$

Visi skersgatviai bus dvikrypčiai (t. y.  $A[i, j] = A[j, i]$  su kiekviena galima  $i$  ir  $j$  reikšme) ir nebus jokių skersgatvių, jungiančių užkampį su juo pačiu (t. y.  $A[i, i]$  bus **false** kiekvienam galimam  $i$ ). Be to, galite daryti prielaidą, jog judant skersgatviais visada bus įmanoma iš bet kurio užkampio pasiekti bet kurį kitą.

Jeigu duotame žemėlapyje įmanoma pagauti plėšiką, funkcija **start** turi gražinti numerį užkampio, kuriame policijos pareigūnas pasirenka patruliuoti. Priešingu atveju, ji turi gražinti  $-1$ .

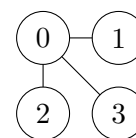
- **nextMove(R)**, kurios argumentas  $R$  yra numeris užkampio, kuriame dabar yra plėšikas, ir turi gražinti numerį užkampio, kuriame policijos pareigūnas bus po savo ėjimo.

Funkcija **start** bus iškviesta lygiai kartą, prieš bet kurį funkcijos **nextMove** iškvietimą. Jeigu **start** gražina  $-1$ , tada **nextMove** nebus iškviesta. Priešingu atveju, **nextMove** bus pakartotinai kviečiama tol, kol baigsis persekiojimas. Tiksliau, programa baigs darbą atsiradus kuriai nors iš šių sąlygų:

- **nextMove** gražina neleistiną ėjimą;
- situacija pasikartoja;
- plėšikas pagaunamas.

## Pavyzdys

Pažiūrėkime į dešinėje pavaizduotą pavyzdį. Šiuo atveju, bet kuris užkampis policijos pareigūnui yra gera pradinė pozicija. Jeigu jis pradės užkampyje Nr. 0, jis gali laukti savo pirmą ėjimą ir plėšikas pats pas jį atbėgs. Kitu atveju, jei jis pradės kuriame nors kitame užkampyje, jis gali laukti, kol plėšikas atsидurs užkampyje Nr. 0, ir tada nueiti į jį.



Funkcijų vykdymo eiga galėtų būti tokia:

Funkcijos kvietimas	Gražina
<b>start</b> (4, [[0, 1, 1, 1], [1, 0, 0, 0], [1, 0, 0, 0], [1, 0, 0, 0]])	3
<b>nextMove</b> (1)	3
<b>nextMove</b> (0)	0

Pastaba: **start** kvietime trumpumo vardan **false** žymima 0 ir **true** žymima 1.