

## Ateiviai

Palydovas aptiko ateivių civilizaciją tolimoje planetoje. Jis nufotografavo ir pateikė žemos raiškos kvadratinės teritorijos nuotrauką. Joje matomi protingų būtybių egzistavimo požymiai. Ekspertai nuotraukoje išvelgė  $n$  įdomių taškų. Taškai sunumeruoti nuo  $0$  iki  $n - 1$ .

Norime gauti aukštos raiškos nuotraukas, kuriose būtų visi šie  $n$  taškai.

Palydovinė žemos raiškos nuotrauka padalinta į  $m \times m$  vienetinių kvadratėlių tinklelį. Eilutės ir stulpeliai numeruojami nuo  $0$  iki  $m - 1$  (atitinkamai pradedant nuo viršaus ir iš kairės).  $(s, t)$  žymi kvadratėlį, esantį eilutės  $s$  ir stulpelio  $t$  sankirtoje. Įdomus taškas, kurio numeris  $i$ , yra kvadratėlyje  $(r_i, c_i)$ . Viename kvadratėlyje gali būti daug įdomių taškų.

Palydovas skrenda ta pačia orbita kuri yra tiesiai virš *pagrindinės* tinklelio įstrižainės. Pagrindinė įstrižainė yra atkarpa, jungianti viršutinį kairinį ir apatinį dešinį kvadratėlio kampus. Palydovas gali padaryti aukštos raiškos nuotrauką, jeigu fotografuojama teritorija tenkina šias sąlygas:

- teritorija yra kvadrato formos,
- du priešingi teritorijos kampai priklauso pagrindinei įstrižainei.
- kiekvienas tinklelio kvadratėlis arba visiškai patenka į nuotrauką, arba visiškai nepatenka.

Palydovas gali padaryti ne daugiau kaip  $k$  aukštos raiškos nuotraukų.

Baigęs fotografuoti, palydovas visų nufotografuotų kvadratėlių aukštos raiškos nuotraukas persiunčia į žemę nepriklausomai nuo to, ar kvadratėlyje yra, ar nėra įdomių taškų. Kiekvieno nufotografuoto kvadratėlio duomenys perduodami tik *vieną* kartą, net jei jis buvo nufotografuotas kelis kartus.

Turime ne daugiau nei  $k$  nufotografuotų kvadrato formos teritorijų ir žinome, kad:

- bent kartą nufotografuoti visi kvadratėliai, kuriuose yra nors vienas įdomus taškas, ir
- bent kartą nufotografuotų kvadratėlių skaičius yra minimalus.

Raskite bendrą mažiausią galimą nors kartą nufotografuotų kvadratėlių skaičių.

## Realizacija

Parašykite šią funkciją (metodą):

- `int64 take_photos(int n, int m, int k, int[] r, int[] c)`
  - $n$ : įdomių taškų skaičius,
  - $m$ : eilučių (ir stulpelių) skaičius tinklelyje,

- **k**: didžiausias galimas aukštos rezoliucijos nuotraukų, kurias gali padaryti palydovas, skaičius,
- **r** ir **c**: du  $n$  ilgio masyvai, nusakantys kvadratėlių, kuriuose yra įdomūs taškai, koordinatės. Kiekvienam  $0 \leq i \leq n-1$  kvadratėlyje ( $r[i]$ ,  $c[i]$ ) yra  $i$ -asis įdomus taškas,
- funkcija turi pateikti mažiausią galimą nors vieną kartą nufotografuotų kvadratėlių skaičių (kai žinoma, kad visi įdomūs taškai patenka į aukštos raiškos nuotraukas).

Pateiktuose failų šablonuose rasite realizacijai reikalingą informaciją. Pasirinkite šabloną, atitinkantį jūsų programavimo kalbą.

## Pavyzdžiai

### Pavyzdys Nr. 1

```
take_photos(5, 7, 2, [0, 4, 4, 4, 4], [3, 4, 6, 5, 6])
```

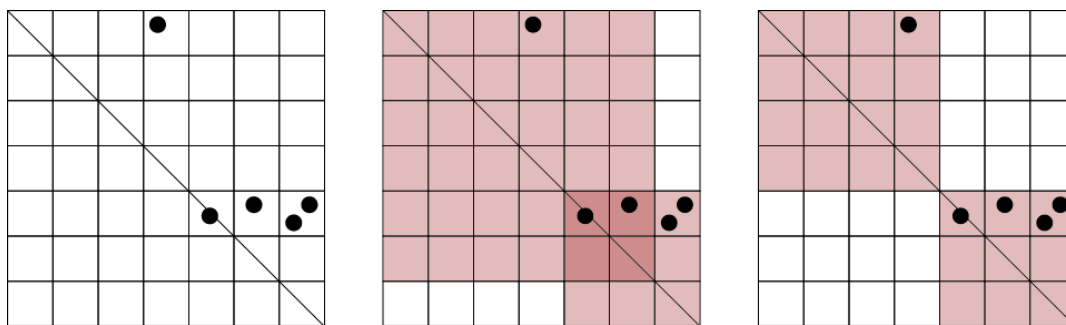
Šiame pavyzdyje turime  $7 \times 7$  tinklėlį, kuriame yra 5 įdomūs taškai. Šie taškai išsidėstę keturiuose kvadratėliuose:  $(0, 3)$ ,  $(4, 4)$ ,  $(4, 5)$  ir  $(4, 6)$ . Palydovas gali padaryti ne daugiau kaip 2 žemos raiškos nuotraukas.

Vienas iš būdų sutalpinti visus penkis įdomius taškus yra padaryti dvi tokias nuotraukas:  $6 \times 6$  dydžio nuotrauką, kuriai priklausytų kvadratėliai  $(0, 0)$  ir  $(5, 5)$ , bei  $3 \times 3$  dydžio nuotrauką, kuriai priklausytų kvadratėliai  $(4, 4)$  ir  $(6, 6)$ . Padaręs šias nuotraukas, palydovas perduos duomenis apie 41 kvadratėlį. Tai nėra optimalus sprendinys.

Optimaliu atveju reikėtų padaryti vieną  $4 \times 4$  dydžio nuotrauką, kurios kampiniai kvadratėliai yra  $(0, 0)$  ir  $(3, 3)$ , bei antrą  $3 \times 3$  dydžio nuotrauką, kurios kampiniai kvadratėliai  $(4, 4)$  ir  $(6, 6)$ . Šiuo atveju palydovui pakaktų perduoti 25 nufotografuotus kvadratėlius. Tai optimalus sprendinys ir funkcija `take_photos` turi grąžinti 25.

Atkreipiame dėmesį, kad kvadratėlį  $(4, 6)$  užtenka nufotografuoti vieną kartą, nors jame yra du įdomūs taškai.

Šis pavyzdys iliustruojamas žemiau. Kairiajame paveikslėlyje pavaizduotas tinklėlis. Viduriniame paveikslėlyje pateiktas sprendinys, kuris nėra optimalus. Jame nufotografuotas 41 kvadratėlis. Dešiniame paveikslėlyje pateiktas optimalus sprendinys.

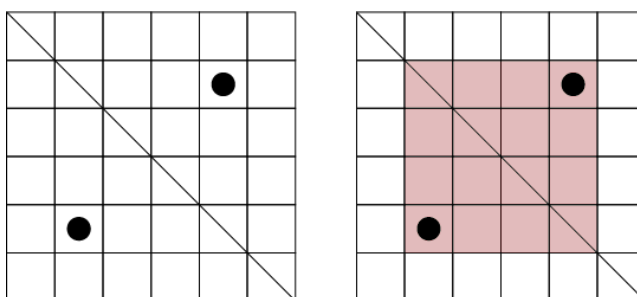


## Pavyzdys Nr. 2

`take_photos(2, 6, 2, [1, 4], [4, 1])`

Šiuo atveju 2 įdomūs taškai išdėstyti simetriškai:  $(1, 4)$  ir  $(4, 1)$  kvadratėliuose. Jei nuotraukoje bus vienas iš šių taškų, tai joje būtinai bus ir kitas. Todėl pakanka vienos nuotraukos.

Žemiau pateiktuose paveikslėliuose pateiktas tinklelis ir optimalus sprendinys. Šiame pavyzdyje palydovas padarys vieną nuotrauką iš 16 kvadratėlių.



## Dalinės užduotys

Visoms dalinėms užduotims galioja  $1 \leq k \leq n$ .

1. (4 taškai)  $1 \leq n \leq 50$ ,  $1 \leq m \leq 100$ ,  $k = n$ ,
2. (12 taškų)  $1 \leq n \leq 500$ ,  $1 \leq m \leq 1000$ , kiekvienam  $i$ , tokiam, kad  $0 \leq i \leq n - 1$ :  $r_i = c_i$ ,
3. (9 taškai)  $1 \leq n \leq 500$ ,  $1 \leq m \leq 1000$ ,
4. (16 taškų)  $1 \leq n \leq 4000$ ,  $1 \leq m \leq 1\,000\,000$ ,
5. (19 taškų)  $1 \leq n \leq 50\,000$ ,  $1 \leq k \leq 100$ ,  $1 \leq m \leq 1\,000\,000$ ,
6. (40 taškų)  $1 \leq n \leq 100\,000$ ,  $1 \leq m \leq 1\,000\,000$ .

## Pavyzdinė vertinimo programa

Pavyzdinė vertinimo programa skaito duomenis tokiu formatu:

- 1-oji eilutė: sveikieji skaičiai  $n$ ,  $m$  ir  $k$ ,
- $(2 + i)$ -oji eilutė, čia  $(0 \leq i \leq n - 1)$ : sveikieji skaičiai  $r_i$  ir  $c_i$ .