

## Citplanētieši

Mūsu satelīts tikko atklāja citplanētiešu civilizāciju tālajā planētā. Mēs jau esam ieguvuši planētas virsmas zemas izšķirtspējas fotogrāfiju. Fotogrāfijā redzamas vairākas saprātīgu būtņu darbības pēdas. Mūsu eksperti fotogrāfijā ir identificējuši  $n$  tos interesējošus *punktus*. Punkti ir numurēti no  $0$  līdz  $n - 1$ . Tagad mēs gribam uzņemt augstas izšķirtspējas fotogrāfijas, kurās būtu redzami visie šie  $n$  punkti.

Satelīts ir sadalījis zemās izšķirtspējas fotogrāfijas laukumu  $m$  reiz  $m$  vienības kvadrātšūnu režģī. Gan rindas, gan kolonnas ir pēc kārtas numurētas ar skaitļiem no  $0$  līdz  $m - 1$  (skaitot attiecīgi no augšas un no kreisās puses). Mēs izmantojam  $(s, t)$  lai apzīmētu šūnu  $s$ -tajā rindā un  $t$ -tajā kolonnā. Punkts ar numuru  $i$  atrodas  $(r_i, c_i)$  režģa šūnā. Katra šūna var saturēt patvaļīgu punktu skaitu.

Mūsu satelīts atrodas stabilā orbītā, kas iet precīzi pa režģa *galveno* diagonāli. Galvenā diagonāle ir taisns segments kas savieno režģa kreiso augšējo un labo apakšējo stūri. Satelīts var uzņemt augstas izšķirtspējas fotogrāfiju jebkuram laukumam kas atbilst šādiem nosacījumiem:

- laukuma forma ir kvadrāts,
- divi pretējie kvadrāta stūri abi ir uz režģa galvenās diagonāles,
- katra režģa šūna vai nu pilnībā pieder fotografētajam laukumam, vai arī ir pilnībā ārā no tā.

Satelīts var uzņemt ne vairāk kā  $k$  augstas izšķirtspējas fotogrāfiju.

Kad satelīts pabeidz uzņemt fotogrāfijas, tas bāzei pārraida katras nofotografētās šūnas augstas izšķirtspējas fotogrāfiju (neskatoties uz to vai šūna satur kādus ekspertus interesējošus punktus). Katrai nofotografētajai šūnai dati tiks pārraidīti tikai *vienreiz*, pat ja šūna tika nofotografēta vairākas reizes.

Līdz ar to, mums ir jāizvēlas ne vairāk kā  $k$  kvadrātveida laukumi, kurus nofotografēs tā, lai:

- katra šūna ar vismaz vienu interesējošo punktu tiktu nofotografēta vismaz vienreiz, un
- vismaz vienreiz nofotografēto šūnu skaits būtu mazākais iespējamais.

Jūsu uzdevums ir atrast mazāko iespējamo kopējo nofotografēto šūnu skaitu.

## Implementācijas detaļas

Jums ir jāimplementē viena funkcija (metode):

- `int64 take_photos(int n, int m, int k, int[] r, int[] c)`
  - $n$ : interesējošo punktu skaits,
  - $m$ : rindu (un arī kolonnu) skaits režģī,

- $k$ : maksimālais fotogrāfiju skaits, ko satelīts var uzņemt,
- $r$  un  $c$ : divi masīvi garumā  $n$ , kas apraksta interesējošo punktu režģa šūnu koordinātas. Katram  $0 \leq i \leq n - 1$ ,  $i$ -tais interesējošais punkts atrodas šūnā  $(r[i], c[i])$ ,
- funkcijai ir jāatgriež mazākais iespējamais vismaz vienreiz nofotografēto šūnu skaits (ņemot vērā, ka fotogrāfijās ir jābūt visiem interesējošajiem punktiem).

Implementācijas detaļām lūdzu izmantotiet piedāvātos šablona failus jūsu izmantotajā programmēšanas valodā.

## Piemēri

### 1. piemērs

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

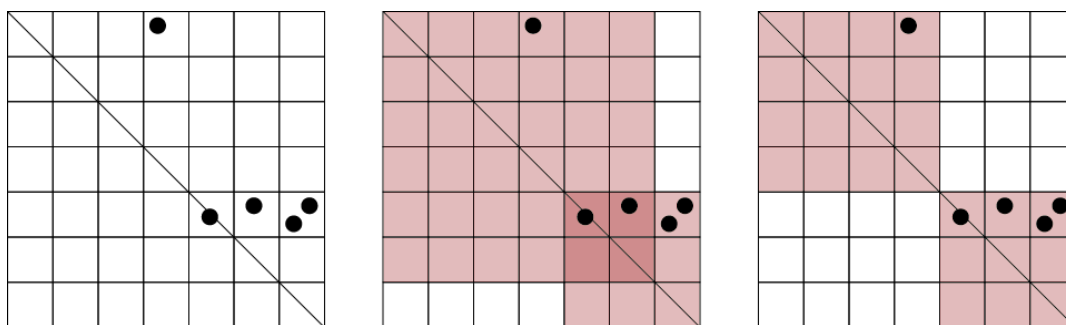
Šajā piemērā mums ir  $7 \times 7$  režģis ar 5 interesējošajiem punktiem. Interesējošie punkti ir četrās dažādās šūnās:  $(0, 3)$ ,  $(4, 4)$ ,  $(4, 5)$  un  $(4, 6)$ . Jūs varat uzņemt ne vairāk kā 2 augstas izšķirtspējas fotogrāfijas.

Viens veids kā nofotografēt visus piecus interesējošos punktus ir uzņemt divas fotogrāfijas:  $6 \times 6$  fotogrāfiju ar šūnām  $(0, 0)$  un  $(5, 5)$  un  $3 \times 3$  fotogrāfiju ar šūnām  $(4, 4)$  un  $(6, 6)$ . Ja satelīts uzņems šīs divas fotogrāfijas, tas pārraidīs informāciju par 41 šūnu. Šis skaits nav optimāls.

Optimāls risinājums izmanto  $4 \times 4$  fotogrāfiju ar šūnām  $(0, 0)$  un  $(3, 3)$  un  $3 \times 3$  fotogrāfiju ar šūnām  $(4, 4)$  un  $(6, 6)$ . Šajā gadījumā tiek pārraidīta informācija par 25 nofotografētam šūnam, kas ir optimāli, tāpēc `take_photos` ir jāatgriež 25.

Ņemiet vērā, ka ir pietiekami nofotografēt šūnu  $(4, 6)$  vienreiz, neskatoties uz to, ka tā satur divus interesējošos punktus.

Šis piemērs ir parādīts attēlos zemāk: pa kreisi ir parādīts režģis, vidū redzams neoptimālais (kurā tiek nofotografēta 41 šūna), bet pa labi - optimālais risinājums.



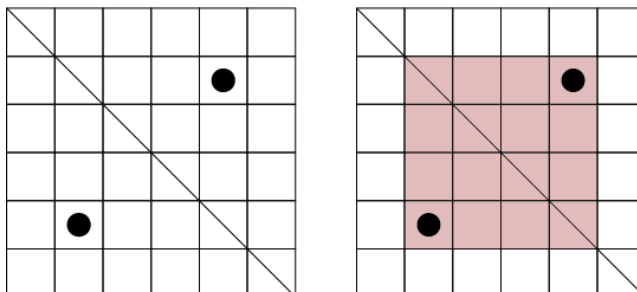
### 2. piemērs

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

Šajā piemērā ir 2 simetriski izvietoti interesējošie punkti, kas atrodas šūnās  $(1, 4)$

un  $(4, 1)$ . Jebkura derīga fotogrāfija, kas satur vienu no tiem, satur arī otru. Tātad ir pietiekams uzņemt vienu fotogrāfiju.

Attēlā zemāk parādīts šī piemēra režģis un optimālais atrisinājums. Šajā atrisinājumā satelīts uzņēma vienu fotogrāfiju ar **16** šūnām.



### Apakšuzdevumi

Visiem apakšuzdevumiem,  $1 \leq k \leq n$ .

1. (4 punkti)  $1 \leq n \leq 50$ ,  $1 \leq m \leq 100$ ,  $k = n$ ,
2. (12 punkti)  $1 \leq n \leq 500$ ,  $1 \leq m \leq 1000$ , visiem  $i$  ar  $0 \leq i \leq n - 1$ ,  
 $r_i = c_i$ ,
3. (9 punkti)  $1 \leq n \leq 500$ ,  $1 \leq m \leq 1000$ ,
4. (16 punkti)  $1 \leq n \leq 4000$ ,  $1 \leq m \leq 1\,000\,000$ ,
5. (19 punkti)  $1 \leq n \leq 50\,000$ ,  $1 \leq k \leq 100$ ,  $1 \leq m \leq 1\,000\,000$ ,
6. (40 punkti)  $1 \leq n \leq 100\,000$ ,  $1 \leq m \leq 1\,000\,000$ .

### Piemēru vērtētājs

Piemēru vērtētājs lasa ievaddatus sekojošā formātā:

- 1. rinda: vesēlie skaitļi  $n$ ,  $m$  un  $k$ ,
- $2 + i$ . rinda ( $0 \leq i \leq n - 1$ ): vesēlie skaitļi  $r_i$  un  $c_i$ .