



Vesoljčki

Naš satelit je ravnokar odkril življenje na zelo oddaljenem planetu. Dobili smo prve slike kvadratnega območja planeta. Trenutno so nizke ločljivosti, vendar vse kaže, da smo odkrili zelo razvite in pametne vesoljčke. Naši strokovnjaki so našli n točk, ki se jim zdijo zanimive. Točke so označene s števili od 0 do $n - 1$. Sedaj pa moramo narediti še slike visoke ločljivosti tako, da bo na njih vseh n točk.

Satelit je že razdelil območje (ki je na sliki nizke ločljivosti) na polje $m \times m$ enotskih celic. Vrstice in stolpci so zaporedno označeni s števili od 0 do $m - 1$, kjer se celica $(0, 0)$ nahaja v zgornjem levem oglišču. Za vsako celico uporabimo zapis (s, t) , kjer s predstavlja vrstico in t stolpec. i -ta zanimiva točka se nahaja znotraj enotske celice označene z (r_i, c_i) . Vsaka celica lahko vsebuje poljubno število zanimivih točk.

Naš satelit leži v zelo stabilni orbiti in preleta celotno *glavno* diagonalo polja. Glavna diagonala je tista, ki prečka vse celice (i, i) za vsak $0 \leq i \leq m - 1$ oziroma povezuje zgornjo levo celico s spodnjo desno. Satelit lahko zajema slike visoke ločljivosti le, če je zadoščeno sledečim pogojem:

- oblika območja je kvadrat,
- dva nasprotna si oglišča kvadrata ležita na glavni diagonali mreže,
- ena celica mreže mora biti v celoti znotraj ali v celoti zunaj slikanega območja.

Satelit je zmožen narediti največ k slik visoke ločljivosti.

Ko satelit konča s slikanjem, pošlje slike visoke ločljivosti vseh slikanih celic domov. To je neodvisno od tega, ali celica vsebuje zanimive točke ali ne. Vsaka slikana celica bo poslana domov samo *enkrat*, tudi če je bila slikana večkrat.

Torej, izbrati moramo največ k kvadratnih območij, ki jih bomo slikali, in da bo veljalo:

- vsaka celica, ki vsebuje zanimivo točko, je slikana vsaj enkrat,
- število celic, ki so slikane vsaj enkrat, je čim manjše.

Tvoja naloga je, da poiščeš to število.

Opombe k implementaciji

Implementirati moraš eno funkcijo (metodo):

- `int64 take_photos(int n, int m, int k, int[] r, int[] c)`
 - n : število zanimivih točk,
 - m : število stolpcev in vrstic v polju,
 - k : omejitev števila slik, ki jih lahko satelit zajame,
 - r in c : dve polji dolžine n , ki opisujeta koordinate celic, kjer ležijo zanimive

točke. Za $0 \leq i \leq n - 1$, i -ta zanimiva točka leži v celici z lokacijo $(r[i], c[i])$,

- o funkcija naj vrne najmanjše možno število celic, ki so slikane vsaj enkrat (ob predpostavki, da so slikane vse zanimive točke)

Uporabi predložne datoteke za več informacij o implementaciji v izbranem programskem jeziku.

Primeri

1. primer

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

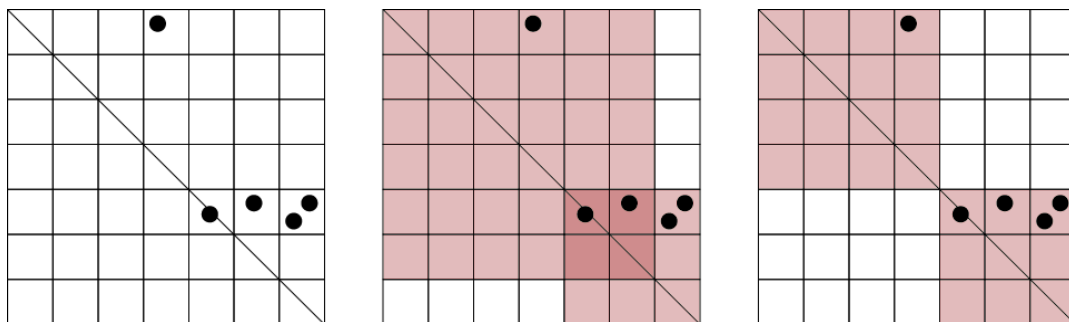
V tem primeru imamo 7×7 veliko polje s 5 zanimivimi točkami. Te točke so locirane v štirih celicah: $(0, 3)$, $(4, 4)$, $(4, 5)$ in $(4, 6)$. Narediš lahko največ 2 sliki visoke ločljivosti.

Eden izmed načinov, da zajameš vseh pet točk zanimanja z dvema slikama, je: ena slika s celicama $(0, 0)$ in $(5, 5)$ v nasprotnih si ogliščih ter drugo s celicama $(4, 4)$ in $(6, 6)$ v nasprotnih si ogliščih. Če storimo tako, bo satelit poslal slike 41 celic, vendar rešitev ni najboljša.

Najboljša rešitev naredi eno sliko kvadrata velikosti 4×4 , ki vsebuje celici $(0, 0)$ in $(3, 3)$, ter drugo sliko kvadrata velikosti 3×3 , ki vsebuje celici $(4, 4)$ in $(6, 6)$. Tako pošlje slike zgolj 25 celic, kar je najboljša rešitev, in `take_photos` vrne 25.

Pomni, da je dovolj, če celico $(4, 6)$ slikamo le enkrat, kljub temu da vsebuje dve zanimivi točki.

Ta primer je narisano na spodnji sliki. Leva slika predstavlja oris problema. Srednja slika predstavlja prvo rešitev, ki zajame 41 celic. Na desni sliki je prikazana najboljša rešitev.

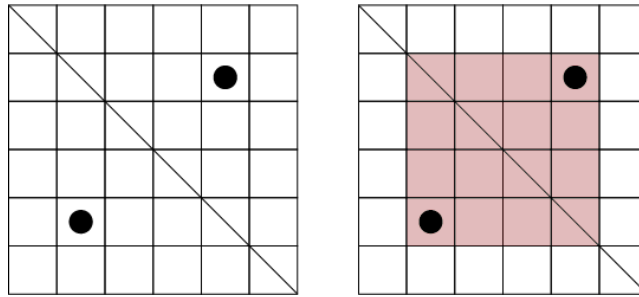


2. primer

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

V tem primeru imamo 2 dve zanimivi točki, ki ležita nasproti ena drugi: v celicah $(1, 4)$ in $(4, 1)$. Katera koli slika, ki vsebuje eno izmed točk, vsebuje tudi drugo. Zato je dovolj, da naredimo le eno sliko.

Spodnji sliki prikazujeta primer in najboljšo rešitev, pri kateri satelit z eno sliko zajame 16 celic.



Podnaloge

Za vse podnaloge velja: $1 \leq k \leq n$.

1. (4 točke) $1 \leq n \leq 50$, $1 \leq m \leq 100$, $k = n$,
2. (12 točk) $1 \leq n \leq 500$, $1 \leq m \leq 1000$, za vsak i ($0 \leq i \leq n - 1$):
 $r_i = c_i$,
3. (9 točk) $1 \leq n \leq 500$, $1 \leq m \leq 1000$,
4. (16 točk) $1 \leq n \leq 2000$, $1 \leq m \leq 1\,000\,000$,
5. (12 točk) $1 \leq n \leq 20\,000$, $1 \leq k \leq 100$, $1 \leq m \leq 1\,000\,000$,
6. (7 točk) $1 \leq n \leq 20\,000$, $1 \leq k \leq 1000$, $1 \leq m \leq 1\,000\,000$,
7. (40 točk) $1 \leq n \leq 200\,000$, $1 \leq m \leq 1\,000\,000$.

Vzorčni ocenjevalnik

Vzorčni ocenjevalnik bere vhod sledeče oblike:

- 1. vrstica: cela števila n , m in k ,
- $(2 + i)$ -ta vrstica ($0 \leq i \leq n - 1$): celi števili r_i i c_i .