



Romvesner

Satellitten vår har nettop oppdaget en sivilisasjon på en fjern planet. Vi har allerede skaffet et lav-oppløsningsbilde av et kvadratisk område av planeten. Fotografiet viser mange tegn på intelligent liv. Ekspertene våre har identifisert n interessante områder på fotografiet. Områdene er nummerert fra 0 til $n - 1$. Vi ønsker nå å ta høyoppløste bilder som inneholder alle disse n områdene.

Internt har satellitten delt opp områdene i lav-oppløsningsbildet i et m ganger m rutenett av kvadratiske celler. Både radene og kolonnene er nummerert i rekkefølge fra 0 til $m - 1$ (fra henholdsvis øvert og til venstre). Vi bruker (s, t) til å betegne cellen i rad s og kolonne t . Område nummer i er fullstendig inneholdt i cellen (r_i, c_i) . Hver celle kan ha et vilkårlig antall interessante områder.

Satellitten vår er i en stabil bane som passerer direkte over *hoveddiagonalen* i rutenettet. Hoveddiagonalen er linjesegmentet som går fra hjørnet øverst til venstre til hjørnet nederst til høyre på rutenettet. Satellitten kan ta høyoppløste bilder av et hvilket som helst område som tilfredsstiller følgende krav:

- Området er kvadratisk,
- De to motsatte hjørnene av kvadratet ligger begge på hoveddiagonalen på rutenettet,
- Hver celle på rutenettet er enten fullstendig inneholdt eller fullstendig utenfor det fotograferte området.

Satellitten kan ta opptil k høyoppløste bilder.

Når satellitten er ferdig med å ta bilder så vil den overføre de høyoppløste bildene av alle de fotograferte cellene til hjembasen (uavhengig av om den cellen inneholder interessante områder eller ikke). Dataene for hver fotograferte celle vil bare bli overført *én gang*, selv om cellen har blitt fotografert flere ganger.

Vi må dermed velge opptil k kvadratiske områder til å bli fotografert, slik at:

- hver celle med ett eller flere interessante områder blir fotografert minst én gang, og
- antall celler som blir fotografert minst én gang er minimert.

Opgaven din er å finne det minste antallet fotograferte celler.

Implementationsdetaljer

Du skal implementere følgende funksjon:

- `int64 take_photos(int n, int m, int k, int[] r, int[] c)`
 - n : antall interessante områder,
 - m : antall rader (og kolonner) i rutenettet,

- k : det høyeste antall fotografier som satellitten kan ta,
- r og c : to arrayer av lengde n som beskriver koordinatene til cellene som har interessante områder. For hver $0 \leq i \leq n - 1$, så er det i -ende interessante området i celle $(r[i], c[i])$,
- funksjonen skal returnere det minste totale antall celler som blir fotografert minst én gang (gitt at fotografiet må dekke alle interessante områder).

Bruk de vedlagte templatfilene for detaljer om implementasjonen i ditt språk.

Eksempler

Eksempel 1

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

Her har vi et 7×7 rutenett med 5 interessante områder. Disse områdene befinner seg i fire forskjellige celler: $(0, 3)$, $(4, 4)$, $(4, 5)$ og $(4, 6)$. Du kan ta opptil 2 høyoppløste bilder.

En måte å fange alle fem områdene på er å ta to bilder: ett bilde av 6×6 kvadratet som inneholder cellene $(0, 0)$ og $(5, 5)$, og ett bilde av 3×3 kvadratet som inneholder cellene $(4, 4)$ og $(6, 6)$. Dersom vi tar disse to bildene så vil satellitten overføre bilder av 41 celler. Dette tallet er ikke optimalt.

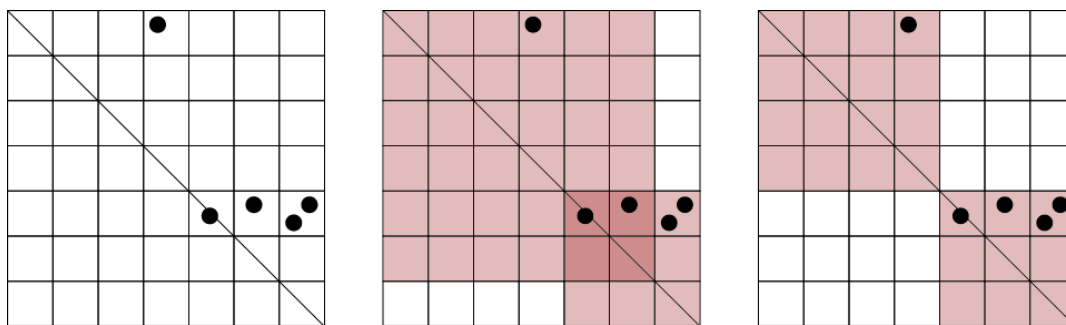
En måte å fange alle de fem områdene på er å ta to bilder: et kvadratisk bilde som er 6×6 celler som inneholder cellene $(0, 0)$ og $(5, 5)$, og et kvadratisk bilde som er 3×3 som inneholder cellene $(4, 4)$ og $(6, 6)$. Hvis satellitten tar disse to bildene, så vil den sende data om 41 celler. Dette antallet er ikke optimalt

Den optimale løsningen er å bruke ett foto til å dekke 4×4 kvadratet som inneholder cellene $(0, 0)$ og $(3, 3)$, og et annet foto til å dekke 3×3 kvadratet som dekker cellene $(4, 4)$ og $(6, 6)$. Dette resulterer i bare 25 fotograferte celler, som er den optimale løsningen. `take_photos` skal dermed returnere 25.

Merk at det holder å fotografere cellen $(4, 6)$ én gang, selv om den inneholder to interessante områder.

Begge måtene å ta bildene på er vist under. Figuren til høyre viser den optimale løsningen.

Dette eksempelet er vist i figurene under. Den venstre figuren viser rutenettet som tilsvarer eksempelet. Figuren i midten viser den suboptimale løsningen hvor 41 celler ble fotografert. Figuren til høyre viser den optimale løsningen.

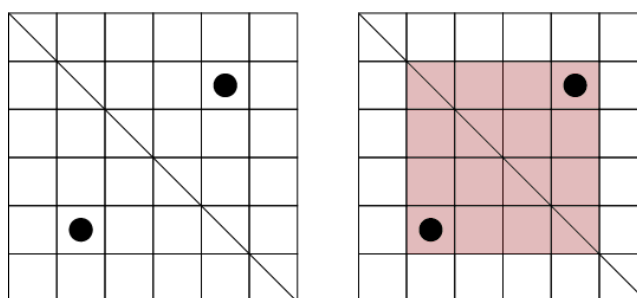


Eksempel 2

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

Her har vi 2 interessante områder som er plassert symmetrisk: i cellene (1, 4) og (4, 1). Alle gyldige bilder som inneholder ett av områdene vil nødvendigvis også inneholder det andre. Derfor er det tilstrekkelig med å ta ett bilde.

Figuren under viser dette eksempelet og den optimale løsningen. I denne løsningen tar satellitten ett enkelt bilde av 16 celler.



Subtasks

For alle subtasks, $1 \leq k \leq n$.

1. (4 poeng) $1 \leq n \leq 50$, $1 \leq m \leq 100$, $k = n$,
2. (12 poeng) $1 \leq n \leq 500$, $1 \leq m \leq 1000$, for alle i slik at $0 \leq i \leq n - 1$ så er $r_i = c_i$,
3. (9 poeng) $1 \leq n \leq 500$, $1 \leq m \leq 1000$,
4. (16 poeng) $1 \leq n \leq 4000$, $1 \leq m \leq 1\,000\,000$,
5. (19 poeng) $1 \leq n \leq 50\,000$, $1 \leq k \leq 100$, $1 \leq m \leq 1\,000\,000$,
6. (40 poeng) $1 \leq n \leq 100\,000$, $1 \leq m \leq 1\,000\,000$.

Sample grader

Sample graderen leser input i følgende format:

- linje 1: heltall n , m og k ,
- linje $2 + i$ ($0 \leq i \leq n - 1$): heltall r_i og c_i .