



Obcy (Aliens)

Na jednej z odległych planet nasz satelita wykrył ślady cywilizacji. Na Ziemię dotarło już pierwsze, niskiej rozdzielczości zdjęcie kwadratowego obszaru tej planety. Zdjęcie wskazuje na wiele miejsc, gdzie mogą znajdować się ślady życia. Naszym ekspertom udało się zidentyfikować na zdjęciu n interesujących punktów. Punkty te są ponumerowane od 0 do $n - 1$. Kolejnym krokiem będzie wykonanie wysokiej rozdzielczości zdjęć zawierających wszystkie te n punktów.

Na obszarze znajdującym się na zdjęciu (tym o niskiej rozdzielczości) satelita naniósł siatkę o wymiarach m na m złożoną z jednostkowych pól. Tak wiersze, jak i kolumny siatki są ponumerowane kolejno od 0 do $m - 1$ (odpowiednio od góry i od lewej). Przez (s, t) oznaczamy pole znajdujące się na przecięciu wiersza s i kolumny t . Interesujący punkt numer i znajduje się na polu (r_i, c_i) . Każde pole może zawierać dowolnie wiele interesujących punktów.

Nasz satelita znajduje się na orbicie przebiegającej bezpośrednio nad *główną* przekątną siatki, przy czym główna przekątna to odcinek łączący lewy górny i prawy dolny róg siatki. Satelita może wykonać wysokiej rozdzielczości zdjęcie dowolnego obszaru, który spełnia następujące warunki:

- ma kształt kwadratu,
- dwa przeciwległe wierzchołki tego kwadratu znajdują się na głównej przekątnej siatki,
- każde pole siatki znajduje się albo całkowicie wewnątrz, albo całkowicie na zewnątrz fotografowanego obszaru.

Satelita może wykonać co najwyżej k zdjęć w wysokiej rozdzielczości.

Gdy tylko satelita wykona wszystkie zdjęcia, wyśle na Ziemię wysokiej rozdzielczości zdjęcie każdego z fotografowanych pól (niezależnie od tego, czy pole to zawiera jakiegokolwiek interesujące punkty). Dane z każdego z tych pól zostaną przesłane dokładnie *raz*, nawet jeżeli pole to zostało sfotografowane wielokrotnie.

Tak więc musimy wybrać co najwyżej k kwadratowych obszarów, które zostaną sfotografowane, tak aby:

- każde z pól zawierających interesujące punkty znalazło się na co najmniej jednym ze zdjęć oraz
- liczba pól, które znajdują się na co najmniej jednym zdjęciu, była jak najmniejsza.

Twoim zadaniem jest wyznaczenie najmniejszej możliwej łącznej liczby pól na zdjęciach.

Szczegóły implementacji

Powinieneś zaimplementować następującą funkcję (metodę):

- `int64 take_photos(int n, int m, int k, int[] r, int[] c)`
 - `n`: liczba interesujących punktów,
 - `m`: liczba wierszy (a zarazem liczba kolumn) siatki,
 - `k`: maksymalna liczba zdjęć, jakie może wykonać satelita,
 - `r` i `c`: dwie tablice rozmiaru `n` opisujące współrzędne pól siatki zawierających interesujące punkty. Dla każdego $0 \leq i \leq n - 1$, `i`-ty z interesujących punktów znajduje się na polu `(r[i], c[i])`.
 - Funkcja powinna zwrócić najmniejszą możliwą łączną liczbę pól, które znajdują się na co najmniej jednym zdjęciu (przy założeniu, że zdjęcia pokrywają wszystkie z interesujących punktów).

Szczegóły implementacji w Twoim języku programowania znajdują się w dostarczonych plikach z szablonami.

Przykłady

Przykład 1

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

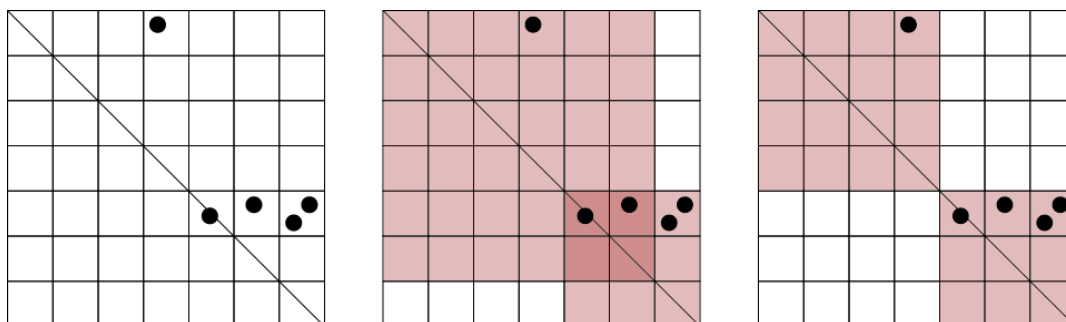
W tym przykładzie mamy siatkę o wymiarach 7×7 zawierającą 5 interesujących punktów. Interesujące punkty znajdują się na czterech różnych polach: $(0, 3)$, $(4, 4)$, $(4, 5)$ i $(4, 6)$. Satelita może wykonać co najwyżej 2 zdjęcia w wysokiej rozdzielczości.

Jednym ze sposobów uchwycenia wszystkich interesujących punktów jest wykonanie dwóch zdjęć: zdjęcia pokrywającego obszar 6×6 zawierający pola $(0, 0)$ i $(5, 5)$ i zdjęcia pokrywającego obszar 3×3 zawierający pola $(4, 4)$ i $(6, 6)$. Jeśli satelita wykona te dwa zdjęcia, wyśle na Ziemię dane z 41 pól. Nie jest to optymalny wynik.

Optymalnym rozwiązaniem jest wykonanie jednego zdjęcia pokrywającego obszar 4×4 zawierający pola $(0, 0)$ i $(3, 3)$ i drugiego zdjęcia pokrywającego obszar 3×3 zawierający pola $(4, 4)$ i $(6, 6)$. W ten sposób na zdjęciach znajdzie się łącznie tylko 25 pól, co jest optymalnym wynikiem, więc funkcja `take_photos` powinna zwrócić 25.

Zauważ, że pole $(4, 6)$ wystarczy sfotografować raz, mimo iż zawiera ono dwa interesujące punkty.

Przykład ten przedstawiono na poniższych rysunkach. Rysunek po lewej pokazuje siatkę z zaznaczonymi interesującymi punktami. Na środkowym rysunku zaznaczono nieoptymalne rozwiązanie, w którym na zdjęciach znajduje się 41 pól. Rysunek po prawej przedstawia optymalne rozwiązanie.

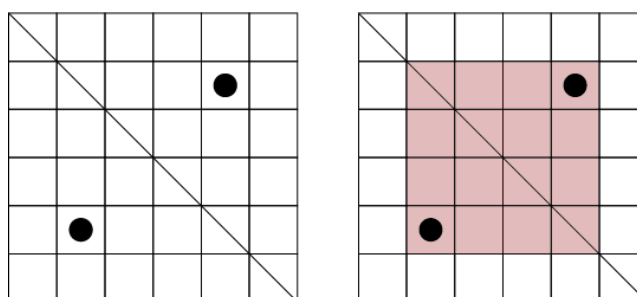


Przykład 2

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

W tym przypadku mamy 2 interesujące punkty położone symetrycznie, na polach $(1, 4)$ i $(4, 1)$. Każde poprawne zdjęcie zawierające jeden z nich zawiera także drugi z nich. Wystarczy zatem wykonać jedno zdjęcie.

Rysunki poniżej przedstawiają tenże przykład i jego optymalne rozwiązanie. W tym rozwiązaniu satelita wykonuje jedno zdjęcie pokrywające 16 pól.



Podzadania

We wszystkich podzadaniach zachodzi $1 \leq k \leq n$.

1. (4 punkty) $1 \leq n \leq 50$, $1 \leq m \leq 100$, $k = n$,
2. (12 punktów) $1 \leq n \leq 500$, $1 \leq m \leq 1000$, dla każdego i takiego że $0 \leq i \leq n - 1$, $r_i = c_i$,
3. (9 punktów) $1 \leq n \leq 500$, $1 \leq m \leq 1000$,
4. (16 punktów) $1 \leq n \leq 4000$, $1 \leq m \leq 1\,000\,000$,
5. (19 punktów) $1 \leq n \leq 50\,000$, $1 \leq k \leq 100$, $1 \leq m \leq 1\,000\,000$,
6. (40 punktów) $1 \leq n \leq 100\,000$, $1 \leq m \leq 1\,000\,000$.

Przykładowy program sprawdzający

Przykładowy program sprawdzający wczytuje dane w następującej formie:

- wiersz 1: liczby całkowite n , m oraz k ,
- wiersz $2 + i$ ($0 \leq i \leq n - 1$): liczby całkowite r_i oraz c_i .