

Czarny market (A)

Limit pamięci: 256 MB

Limit czasu: 1.00 s

Znany w podziemiu netrunner o pseudonimie Bajtazar stworzył absolutnie przełomowy exploit (tzw. *B-day*), który pozwala na ominięcie ICE największych megakorporacji. Bajtazar postanowił zmonetyzować swoje dzieło i sprzedawać do niego dostęp w modelu subskrypcyjnym na anonimowym forum w Barknecie.

Na forum znajduje się N potencjalnych kupców. Każdy z nich ma swój określony budżet i jest w stanie zapłacić maksymalnie c_i kryptokredytów za dostęp.

Bajtazar musi ustalić **jedną sztywną cenę** P dla wszystkich. Jeśli ustalona cena będzie wyższa niż maksymalny budżet danego kupca ($P > c_i$), ten zrezygnuje z zakupu. W przeciwnym razie kupiec nabywa subskrypcję. Bajtazar chce ustalić taką cenę P , aby zsumowany zysk C ($C = P \times$ ile osób kupi subskrypcję) kupców był jak największy.

Pomóż mu obliczyć maksymalny możliwy zysk C oraz cenę P , która go daje. Jeśli istnieje kilka cen dających ten sam maksymalny zysk C , wybierz **najniższą** z nich.

Wejście

W pierwszym wierszu znajduje się jedna liczba całkowita N .

W drugim wierszu znajduje się N liczb całkowitych c_1, c_2, \dots, c_N oddzielonych pojedynczymi odstępami, oznaczających maksymalny budżet poszczególnych kupców.

Wyjście

Twój program powinien wypisać dwie liczby całkowite oddzielone spacją: Maksymalny zysk C , jaki może osiągnąć Bajtazar oraz optymalną cenę subskrypcji P dającą ten zysk. Jeśli istnieje kilka cen dających ten sam maksymalny zysk C , wybierz **najniższą** z nich.

Ograniczenia

$$1 \leq N \leq 10^6, 1 \leq c_i \leq 10^6$$

Podzadania

Podzadanie	Warunki	Punkty
1	$c_i \leq 1\,000$	20
2	$N \leq 1\,000$	20
3	brak dodatkowych ograniczeń	60

Przykład

Wejście

4
1 6 4 6

Wyjście

12 4

Wyjaśnienie

Przy cenie równej 4 produkt kupią trzy osoby (te z budżetem 4, 6 i 6), co daje maksymalny zysk 12 ($4 \cdot 3$). Ten sam zysk osiągniemy przy cenie 6 ($6 \cdot 2$), jednak zgodnie z poleceniem w przypadku remisu wybieramy najniższą kwotę, dlatego poprawny wynik to zysk 12 przy cenie 4.

Wejście

8
1 2 3 4 5 6 7 8

Wyjście

20 4

Wejście

2

1 4

Wyjście

4 4

Magiczny Dywan (B)

Limit pamięci: 256 MB

Limit czasu: 2.00 s

W królestwie Bajtii wznosi się pasmo majestatycznych gór. Składa się ono z N szczytów, ponumerowanych od 1 do N z zachodu na wschód. Szczyt o numerze i ma wysokość H_i metrów.

Młody czarodziej Bajtazar testuje swój nowo utkany artefakt – Magiczny Dywan. Artefakt ten pozwala mu przemieszczać się w powietrzu, zużywając magiczną energię (manę). W każdej sekundzie lotu Bajtazar wykonuje dwie akcje **jednocześnie**:

1. **Ruch w poziomie:** Bajtazar wybiera jedną z trzech opcji:

- Przemieszcza się na pozycję sąsiedniego wschodniego szczytu, zmieniając pozycję poziomą z i na $i + 1$.
- Przemieszcza się na pozycję sąsiedniego zachodniego szczytu, zmieniając pozycję poziomą z i na $i - 1$.
- Zatrzymuje się w miejscu (nie zmienia pozycji w poziomie).

2. **Zmiana wysokości:** Równoległe z ruchem poziomym, dywan zmienia swoją wysokość:

- **Wzniesienie się** o jeden metr (kosztuje 4 jednostki many).
- **Opadnięcie** o jeden metr (kosztuje 1 jednostkę many – dywan wspomaga się grawitacją).
- **Utrzymanie wysokości** (kosztuje 2 jednostki many, by przeciwstawić się ciężarowi).

Oczywiście, Bajtazar musi unikać zderzenia ze skałami – w każdym momencie, gdy znajduje się nad szczytem i , jego wysokość musi wynosić co najmniej H_i .

Wielki Mag Bajtii zlecił Bajtazarowi dostarczenie Q ważnych zwojów. Podczas j -tej misji Bajtazar startuje dokładnie ze szczytu góry S_j i musi wylądować idealnie na szczycie góry T_j . Ponieważ regeneracja many trwa bardzo długo, Bajtazar chce zużyć jej jak najmniej. Pomóż mu obliczyć minimalną ilość many wymaganą dla każdej z misji!

Wejście

W pierwszym wierszu wejścia znajduje się jedna liczba całkowita N , oznaczająca liczbę górskich szczytów. W drugim wierszu znajduje się N liczb całkowitych H_1, H_2, \dots, H_N oddzielonych pojedynczymi odstępami, oznaczających wysokości kolejnych szczytów w metrach. W trzecim wierszu wejścia znajduje się jedna liczba całkowita Q , oznaczająca liczbę zapytań. W kolejnych Q wierszach znajdują się opisy poszczególnych misji. W j -tym z tych wierszy podane są dwie liczby całkowite S_j oraz T_j , oznaczające odpowiednio numer szczytu startowego i docelowego.

Wyjście

Twój program powinien wypisać na standardowe wyjście dokładnie Q wierszy. W j -tym wierszu powinna znaleźć się jedna liczba całkowita – minimalna liczba jednostek many niezbędna do wykonania lotu ze szczytu S_j na szczyt T_j .

Ograniczenia

$$2 \leq N, Q \leq 200\,000, 1 \leq H_i \leq 10^9, 1 \leq S_j, T_j \leq N$$

Ocenianie

Podzadanie	Warunki	Punkty
1	$S_j + 1 = T_j$	5
2	$H_i = i$	6
3	$N, Q, H_i \leq 100$	18
4	$N, Q \leq 1\,000$	24
5	$S_j = 1$	20

Przykład

Wejście

4
9 1 8 2
2
1 3
4 2

Wyjście

3
31

Wyjaśnienie

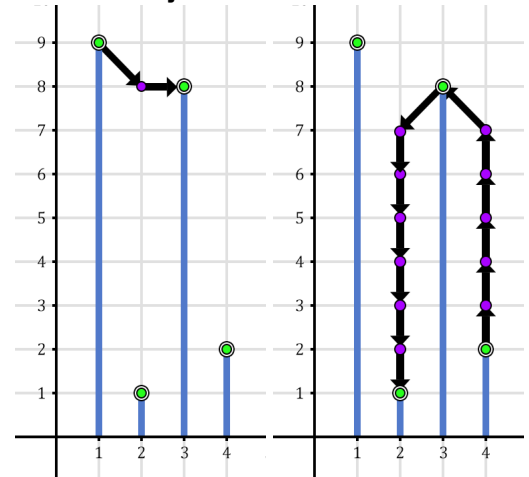
Lot z pierwszym zwojem: Start z góry 1 (wysokość 9). Przelot nad górę 2 ze spadkiem o jeden metr (koszt: 1). Przelot nad górę 3 z utrzymaniem wysokości (koszt: 2).

Łącznie: $1 + 2 = 3$ jednostki many.

Lot z drugim zwojem: Start z góry 4 (wysokość 2). Wzniesienie w miejscu o pięć metrów (koszt: $5 \cdot 4 = 20$). Przelot nad górę 3 ze wzniesieniem o jeden metr (koszt: 4). Przelot nad górę 2 ze spadkiem o jeden metr (koszt: 1). Opadnięcie w miejscu o sześć metrów (koszt: $6 \cdot 1 = 6$).

Łącznie: $20 + 4 + 1 + 6 = 31$ jednostek many.

Lot ze zwojem nr 1 i 2:



Wejście

9
1 2 3 2 1 2 3 2 1
4
1 9
4 6
2 6
5 2

Wyjście

18
4
9
9

Wejście

5
1 2 3 4 5
3
1 3
3 1
2 5

Wyjście

8
2
12

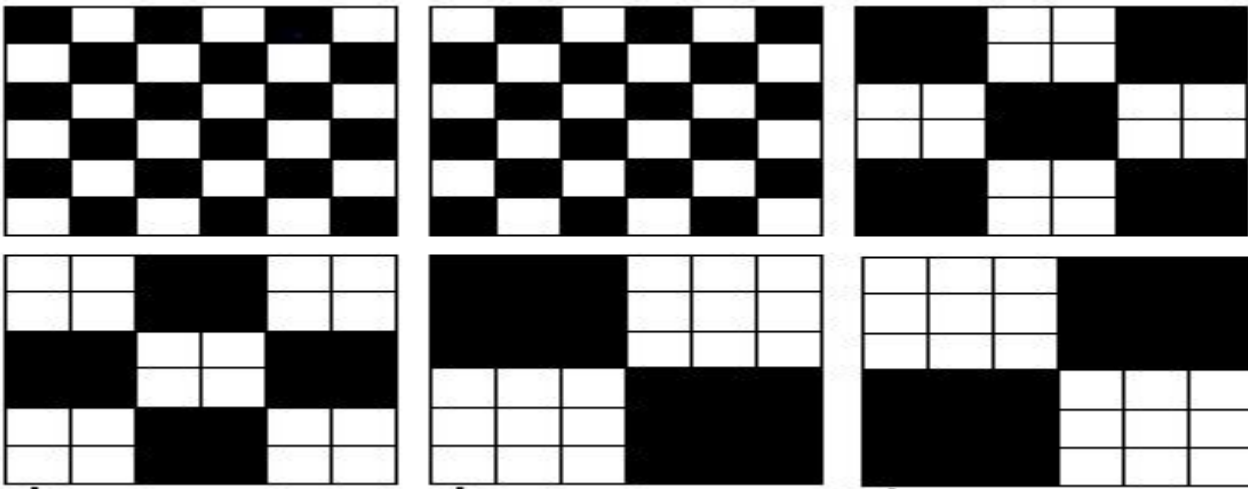
Plansza (c)

Limit pamięci: 256 MB

Limit czasu: 2.00 s

Bajtek ma planszę rozmiaru $N \times N$ podzieloną na N^2 pól. K prostokątnych obszarów na planszy jest pomalowanych na czarno, reszta planszy jest pomalowana na biało. Prostokątne obszary mają boki równoległe do boków planszy i przebiegające wzdłuż linii podziału planszy – każde pole planszy jest w obszarze albo w całości, albo wcale. Obszary nie przecinają się ze sobą. Wiersze i kolumny planszy są ponumerowane liczbami od 1 do N – wiersze od góry do dołu, a kolumny od lewej do prawej.

Planszę nazywamy **szachownicą**, jeżeli można ją podzielić na identyczne kwadraty o boku nie mniejszym od 1 i **ściśle** mniejszym od N , w taki sposób, że wszystkie pola wewnątrz każdego z tych kwadratów są tego samego koloru, a dwa sąsiednie kwadraty mają różne kolory. Dwa kwadraty nazywamy sąsiednimi, jeżeli mają wspólny bok. Poniżej pokazane są wszystkie możliwe szachownice dla $N = 6$:



Bajtek chce zamienić swoją planszę w szachownicę. W tym celu może on przemalowywać pola – jeśli pole było białe, po przemalowaniu staje się czarne i na odwrót. Pomóż Bajtkowi wyznaczyć najmniejszą liczbę przemalowań, jakich musi on dokonać, aby plansza stała się szachownicą.

Wejście

W pierwszym wierszu wejścia znajdują się dwie liczby całkowite N i K – rozmiar planszy oraz liczba czarnych prostokątnych obszarów.

W każdym z kolejnych K wierszy znajdują się cztery liczby całkowite x_1, y_1, x_2, y_2 oddzielone pojedynczymi odstępami – odpowiednio numer wiersza i kolumny lewego górnego pola oraz numer wiersza i kolumny prawego dolnego pola prostokątnego obszaru.

Możesz założyć, że żadne dwa obszary się nie przecinają.

Wyjście

Twój program powinien wypisać jedną liczbę całkowitą – najmniejszą liczbę przemalowań potrzebnych, aby plansza stała się szachownicą.

Ograniczenia

$$2 \leq N \leq 10^5.$$

$$0 \leq K \leq \min(N^2, 10^5).$$

$$1 \leq x_1 \leq x_2 \leq N, 1 \leq y_1 \leq y_2 \leq N.$$

Podzadania

Podzadanie	Warunki	Punkty
1	$N \leq 100, K = 0$	8
2	N jest liczbą pierwszą, pole każdego obszaru wynosi 1	8
3	$N \leq 100, K \leq 1000$, pole każdego obszaru wynosi 1	15
4	$N \leq 1000$, pole każdego obszaru wynosi 1	16
5	pole każdego obszaru wynosi 1	23
6	brak dodatkowych ograniczeń	30

Przykład

Wejście

2 0

Wyjście

2

Wyjaśnienie

Optymalne przemalowania dla wszystkich testów przykładowych znajdują się poniżej.

Wejście

6 8
3 3 3 3
1 2 1 2
3 4 3 4
5 5 5 5
4 3 4 3
4 4 4 4
2 1 2 1
3 6 3 6

Wyjście

14

Wejście

4 1
4 1 4 4

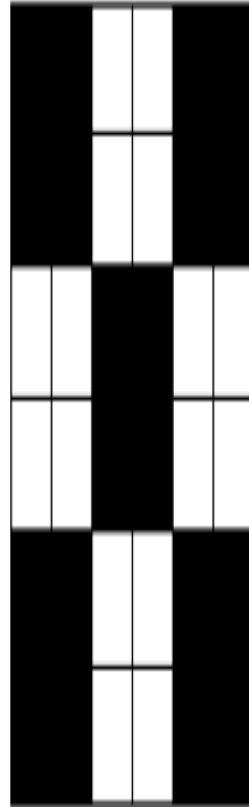
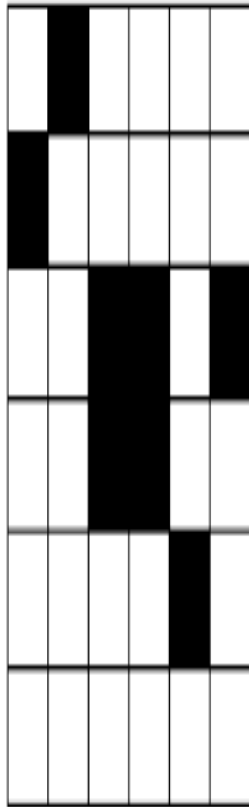
Wyjście

8

1)



2)



3)

