Jakub Pachocki — Treść zadania, Opracowanie
Tomasz Idziaszek — Program

Dostępna pamięć: 64 MB. *OI, etap III, dzień pierwszy, 28.03.2012*

Wyrównywanie terenu

Bajtazar postanowił wybudować dom. Jako lokalizację dla niego wybrał pewną bardzo wąską dolinę. Bajtazar musi najpierw wyrównać grunt pod budowę domu. Ma on do dyspozycji dwie koparki: pierwsza z nich może zwiększyć lub zmniejszyć poziom gruntu w dowolnym spójnym fragmencie doliny o dokładnie a metrów; druga z nich może natomiast zwiększyć lub zmniejszyć poziom gruntu w dowolnym spójnym fragmencie doliny o dokładnie b metrów. Zauważ, że zarówno przed wykonaniem każdej takiej operacji jak i po jej wykonaniu grunt w rozważanym kawałku doliny nie musi być równy.

Mając daną mapę terenu, wyznacz minimalną liczbę operacji, jakie trzeba wykonać, aby wyrównać teren w całej dolinie (tj. aby grunt w całej dolinie miał poziom równy 0). W trakcie wykonywania ciągu operacji poziom gruntu w każdym fragmencie doliny może być dowolnie duży lub dowolnie mały (w szczególności ujemny).

Wejście

W pierwszym wierszu standardowego wejścia znajdują się trzy liczby całkowite n, a, b ($1 \leqslant n \leqslant 100\,000$, $1 \leqslant a, b \leqslant 10^9$), pooddzielane pojedynczymi odstępami. Liczba n oznacza długość doliny, podaną w metrach. Drugi wiersz zawiera n liczb całkowitych $h_1, h_2, \ldots, h_n$ nieprzekraczających co do wartości bezwzględnej 10^9, pooddzielanych pojedynczymi odstępami. Liczby te reprezentują poziom gruntu (wyrażony w metrach) na kolejnych kawałkach ziemi długości jednego metra.

W testach wartych 30% punktów zachodzą dodatkowe warunki $n, a, b \leqslant 200$ oraz $-200 \leqslant h_1, \ldots, h_n \leqslant 200$.

W testach wartych 60% punktów zachodzą warunki $n, a, b \leqslant 2\,000$ oraz $-2\,000 \leqslant h_1, \ldots, h_n \leqslant 2\,000$.

W testach wartych 90% punktów zachodzi warunek $a, b \leqslant 10^6$.

Wyjście

W pierwszym i jedynym wierszu standardowego wyjścia Twój program powinien wypisać jedną liczbę całkowitą — minimalną liczbę operacji potrzebnych do wyrównania gruntu lub liczbę -1, jeśli wyrównanie gruntu w dolinie za pomocą podanych koparek nie jest w ogóle możliwe.

Przykład

Dla danych wejściowych:

```
5 2 3
1 2 1 1 -1
```

poprawnym wynikiem jest:

```
5
```

Wyjaśnienie do przykładu: *Jedno z możliwych rozwiązań dla przykładowego wejścia to:*

- *zwiększ poziom gruntu na pierwszych dwóch metrach doliny o 2 metry,*
- *zmniejsz poziom gruntu na pierwszych dwóch metrach doliny o 3 metry,*
- *zwiększ poziom gruntu na czterech końcowych metrach doliny o 2 metry,*
- *zwiększ poziom gruntu na ostatnim metrze doliny o 2 metry,*
- *zmniejsz poziom gruntu na czterech końcowych metrach doliny o 3 metry.*

Rozwiązanie

Weźmy ciąg z przykładu:

$$1,\ 2,\ 1,\ 1,\ -1$$

Możemy dołożyć po jego obu stronach nieskończone ciągi zer bez zmiany wyniku:

$$\ldots,\ 0,\ 1,\ 2,\ 1,\ 1,\ -1,\ 0,\ \ldots$$

Rozważmy najpierw przypadek z jedną koparką, zmieniającą poziom ziemi o 1 metr. Widać, że wówczas musi ona być użyta na kolejnych metrach odpowiednio -1, -2, -1, -1, 1 razy. Można udowodnić, że minimalna liczba operacji potrzebnych, aby zrealizować taki schemat (tj. wykonać X_i operacji na i-tym metrze), wynosi

$$(\ldots + |X_{-1} - X_0| + |X_1 - X_2| + \ldots + |X_n - X_{n+1}| + \ldots)/2,$$

a zatem w przykładzie

$$(|0-(-1)| + |-1-(-2)| + |-2-(-1)| + |-1-(-1)| + |(-1)-1| + |1-0|)/2 = 3.$$

Rozważmy teraz przypadek z dwiema koparkami. Przede wszystkim, wszystkie poziomy gruntu muszą być podzielne przez $\mathrm{nwd}(a, b)$ (w przeciwnym przypadku odpowiedź to -1). Dalej założymy, że $\mathrm{nwd}(a, b) = 1$. Jeżeli wyznaczymy dla każdego metra, ile razy będzie na nim użyta koparka $+a$, a ile razy koparka $+b$ — oznaczmy odpowiednie ciągi przez A_i oraz B_i — to minimalny koszt realizacji takiego schematu będzie równy, jak w poprzednim przypadku, połowie sumy $|A_i - A_{i+1}| + |B_i - B_{i+1}|$.

Na początku zauważmy, że możemy wyznaczyć pewne (niekoniecznie optymalne) rozwiązanie (ciągi A_i oraz B_i), korzystając np. z rozszerzonego algorytmu Euklidesa. Następnie można zauważyć, że ponieważ $\mathrm{nwd}(a, b) = 1$, wszystkie pary ciągów rozwiązujące problem powstają z otrzymanego ciągu przez zamienianie (być może wielokrotne) pewnych par (A_i, B_i) na $(A_i + b, B_i - a)$ lub $(A_i - b, B_i + a)$. Możemy uznać taką zamianę za naszą operację bazową i dążyć do zminimalizowania sumy wartości bezwzględnych różnic kolejnych elementów ciągów.

Przyjmijmy $X_i = A_i - A_{i-1}$, $Y_i = B_i - B_{i-1}$. Oczywiście pary (X, Y) oraz (A, B) wyznaczają się wzajemnie. Jeżeli będziemy pracować ze schematem w postaci (X, Y),

nasz wynik będzie równy po prostu połowie sumy $|X_i|+|Y_i|$, a nasza operacja bazowa będzie polegała na zamianie

$$(X_i, Y_i, X_{i+1}, Y_{i+1})$$

na

$$(X_i + b, Y_i - a, X_{i+1} - b, Y_{i+1} + a)$$

lub przeciwnie. Wykonując tę operację wielokrotnie, można zamienić każdą czwórkę postaci (X_i, Y_i, X_j, Y_j) na $(X_i + b, Y_i - a, X_j - b, Y_j + a)$. W istocie więc dowolna liczba operacji bazowych na (X, Y) sprowadza się do m-krotnej zamiany par (X_i, Y_i) na $(X_i + b, Y_i - a)$ oraz m-krotnej zamiany par (X_j, Y_j) na $(X_j - b, Y_j + a)$, dla pewnego m.

Niech $F_i(k)$ będzie równe $|X_i+kb|+|Y_i-ka|$, dla dowolnego k całkowitego. Funkcja F_i jest wypukła, co więcej, zbiór liczb całkowitych można podzielić na co najwyżej trzy przedziały, na których jest to funkcja liniowa. Naszym celem (wciąż równoważnym z początkowym zadaniem!) będzie teraz znalezienie ciągu liczb całkowitych D_i o sumie równej 0, dla którego suma postaci $\sum_i F_i(D_i)$ jest minimalna.

Każde F_i przyjmuje minimum na pewnym spójnym przedziale. Nietrudno wykazać, że istnieje rozwiązanie optymalne, w którym nie ma D_i, D_j, takich że D_i jest mniejsze od wszystkich liczb w optymalnym przedziale F_i, a D_j jest większe od wszystkich liczb w optymalnym przedziale F_j. Daje to dwa przypadki do rozpatrzenia. Załóżmy zatem bez straty ogólności, że D_i nigdy nie będzie mniejsze od wszystkich liczb w optymalnym przedziale F_i. Przyjmijmy M_i jako najmniejszą liczbę, w której F_i przyjmuje minimum. Będzie zatem zachodzić $D_i \geqslant M_i$. Funkcje F_i okrojone do przedziałów $[D_i, +\infty)$ są monotoniczne i wypukłe oraz można je podzielić na co najwyżej trzy funkcje liniowe. Przyjmijmy za wyjściowe wartości $D_i = M_i$, chcemy je zwiększyć o łącznie $-\sum_i M_i$ (jeśli $-\sum_i M_i$ jest ujemna, rozwiązanie nie istnieje w tym przypadku). Możemy posortować funkcje liniowe odpowiadające F_i po współczynnikach i zachłannie przydzielić przesunięcia o łącznie $-\sum_i M_i$ najmniej nieopłacalnym przesunięciom. Rozwiązaniem zadania będzie połowa z minimum po obu przypadkach z sumy $\sum_i F_i(D_i)$.

Złożoność czasowa rozwiązania to $O(n \cdot (\log n + \log (a + b)))$ w związku z wykorzystaniem sortowania oraz algorytmu Euklidesa. Implementacje rozwiązania można znaleźć w plikach `wyr.cpp` i `wyr1.pas`.

Testy

Rozwiązania zawodników były sprawdzane na 10 zestawach testowych. Większość testów została wygenerowana w sposób losowy. Opis pozostałych typów testów:

- „brak rozwiązania”: odpowiedzią jest -1;
- „dużo operacji”: wynik jest duży (relatywnie do rozmiaru danych);
- „długie przedziały”: operacje na czwórkach (X_i, Y_i, X_j, Y_j) wymagają czwórek o dużych różnicach $j - i$;
- „ciąg bardzo zmienny”: znaczące oscylacje w ciągu (h_i).

Nazwa	n	a	b	max h_i	Opis
wyr1a.in	100	85	2	99	losowy
wyr1b.in	97	90	75	90	brak rozwiązania
wyr1c.in	17	54	81	81	losowy
wyr2a.in	200	19	118	199	losowy
wyr2b.in	193	187	78	199	dużo operacji
wyr2c.in	200	17	17	187	losowy
wyr3a.in	200	117	42	198	losowy
wyr3b.in	160	3	2	200	długie przedziały
wyr4a.in	1 000	1 680	719	1 995	losowy
wyr4b.in	1 000	1 955	1 479	1 989	losowy
wyr5a.in	2 000	163	514	2 000	losowy
wyr5b.in	914	974	919	1 000	dużo operacji
wyr6a.in	2 000	709	542	1 999	losowy
wyr6b.in	1 600	3	2	2 000	długie przedziały
wyr7a.in	10 000	975 747	488 245	999 972	losowy
wyr7b.in	10 000	751 933	773 134	999 911	losowy
wyr7c.in	10 000	2	3	300 000	ciąg bardzo zmienny
wyr8a.in	50 000	448 000	495 000	999 993 000	losowy
wyr8b.in	50 000	6 293	541 632	999 965 078	losowy
wyr8c.in	50 000	3	4	1 000 000 000	ciąg bardzo zmienny
wyr9a.in	100 000	494 853	105 495	999 987 309	losowy
wyr9b.in	100 000	613 053	992 562	999 977 022	brak rozwiązania
wyr9c.in	100 000	4	5	1 000 000 000	ciąg bardzo zmienny
wyr10a.in	100 000	160 444 645	135 419 134	999 996 983	losowy
wyr10b.in	100 000	7	3	1 125 000	długie przedziały
wyr10c.in	100 000	166 216 387	279 312 073	999 011 893	losowy
wyr10d.in	100 000	1	2	1 000 000 000	ciąg bardzo zmienny