

Problema Drei

Fișier de intrare `drei.in`
Fișier de ieșire `drei.out`

Arheologii au descoperit, printre alte vestigii ale unei civilizații dispărute, o reprezentare neobișnuită a numerelor pe care au denumit-o DREI. În reprezentarea DREI apar semne care au fost echivalate de arheologi cu literele mici din alfabetul englez. O *reprezentare DREI* este un șir de litere distincte care fie constă dintr-o singură literă, fie respectă următoarele două condiții:

1. litera maximă apare la unul dintre capete;
2. șirul obținut prin eliminarea literei maxime este o reprezentare DREI.

Primele douăsprezece reprezentări, corespunzătoare numerelor 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 sunt $a, ab, b, ba, bac, bc, abc, ac, c, ca, cab, cb$.

Să notăm cu $\max(x)$ cea mai mare literă din reprezentarea DREI x .

Pentru a compara două reprezentări DREI notate cu x și respectiv y aplicăm, în ordine, următoarele reguli:

1. Dacă $\max(x) > \max(y)$, atunci $x > y$.
2. Reprezentarea formată dintr-o singură literă este mai mare decât orice altă reprezentare care are aceeași literă maximă în capătul drept și este mai mică decât orice altă reprezentare care are aceeași literă maximă în capătul stâng.
3. Dacă ambele reprezentări x și y au aceeași literă maximă poziționată la capătul drept, atunci se elimină litera maximă din ambele reprezentări și se compară reprezentările obținute (să le notăm x_{dr} și y_{dr}). Dacă $x_{dr} < y_{dr}$, atunci $x > y$, respectiv dacă $x_{dr} > y_{dr}$, atunci $x < y$.
4. Dacă ambele reprezentări x și y au aceeași literă maximă poziționată la capătul stâng, atunci se elimină litera maximă din ambele și se compară reprezentările obținute (să le notăm cu x_{st} și y_{st}). Dacă $x_{st} < y_{st}$, atunci $x < y$, respectiv dacă $x_{st} > y_{st}$, atunci $x > y$.

De exemplu:

- $edabc < edc$ (se aplică mai întâi regula 4, apoi se compară $dabc$ cu dc ; se aplică din nou regula 4 și se compară abc cu c , apoi se aplică regula 2)
- $ab < b < ba$, conform regulii 2
- $abe < caf$, conform regulii 1
- $bac < cba$, conform regulii 2
- $bac < abc$ (se aplică mai întâi regula 3, apoi se compară ba cu ab , aplicând regula 2)

Cerință

Scrieți un program care să determine răspunsurile pentru Q interogări de următoarele 4 tipuri:

Tip interogare	Răspuns
1 x	se afișează 1 dacă șirul x este o reprezentare DREI, respectiv 0 în caz contrar
2 $x y$	se compară reprezentările DREI x și y și se afișează -1 dacă $x < y$, 0 dacă $x = y$, respectiv 1 dacă $x > y$
3 N	se afișează cea de a N -a reprezentare DREI
4 x	se afișează numărul de ordine al reprezentării DREI x

Date de intrare

Fișierul de intrare `drei.in` conține pe prima linie numărul natural Q reprezentând numărul de interogări. Pe următoarele Q linii se află cele Q interogări, câte o interogare pe o linie, în formatul descris mai sus.

Date de ieșire

Fișierul de ieșire `drei.out` va conține Q linii. Pe linia i va fi scris răspunsul pentru cea de a i -a interogare din fișierul de intrare.

Restricții

- $1 \leq Q \leq 10^5$
- $1 \leq N \leq 10^{12}$
- Șirurile de caractere care apar în interogări au lungimea cel mult egală cu 26.
- Reprezentările DREI sunt numerotate în ordine crescătoare începând cu 1.

#	Punctaj	Restricții
1	10	Toate interogările sunt de tip 1
2	13	Toate interogările sunt de tip 2
3	20	Toate interogările sunt de tip 3
4	20	Toate interogările sunt de tip 4
5	12	Șirurile din interogări sau cele obținute ca rezultat al interogărilor de tip 3 conțin cel mult primele 10 litere din alfabet și $Q < 10000$.
6	25	Nu există restricții suplimentare.

Exemple

drei.in	drei.out
6	0
1 acb	1
1 abcd	0
1 xabby	-1
2 bac abc	cb
3 12	12
4 cb	

Problema Soldati

Fișier de intrare `soldati.in`
Fișier de ieșire `soldati.out`

Cei S soldați ai unei unități militare au un moment de relaxare și sunt răspândiți pe platou în poziții cunoscute. Evident, ei nu stau aliniați, în poziție de drepti. Platoul poate fi reprezentat printr-un tablou bidimensional cu n linii (numerotate de la 1 la n) și m coloane (numerotate de la 1 la m), o poziție în tablou fiind identificată prin indicele liniei și respectiv al coloanei. Când se anunță sosirea comandantului unității militare, cei S soldați trebuie să se alinieze. Alinierea presupune așezarea celor S soldații pe o aceeași linie L , pe coloane consecutive. Ca să nu se creeze haos, fiecare dintre ei se va deplasa numai pe orizontală și verticală. Dacă soldatul i ($1 \leq i \leq S$) se află inițial în poziția (L_i, C_i) iar la aliniere ajunge în poziția (Lin, Col) , distanța pe care el o parcurge va fi $|L_i - Lin| + |C_i - Col|$.

Cerință

Scrieți un program care să rezolve următoarele două cerințe:

- determină o linie Lin , astfel încât pentru alinierea soldaților pe pozițiile $1, 2, \dots, S$ de pe linia Lin , suma distanțelor parcurse de aceștia să fie minimă;
- determină o linie Lin și o coloană Col , astfel încât pentru alinierea soldaților pe linia Lin pe coloanele $Col, Col + 1, \dots, Col + S - 1$, suma distanțelor parcurse de aceștia să fie minimă.

Date de intrare

Fișierul de intrare `soldati.in` conține pe prima linie 4 valori naturale $C \ n \ m \ S$, separate prin câte un spațiu, reprezentând cerința care trebuie să fie rezolvată, numărul de linii, numărul de coloane, respectiv numărul de soldați. Următoarele S linii conțin pozițiile celor S soldați, câte un soldat pe o linie, pozițiile fiind descrise prin două numere naturale $L \ C$ separate printr-un spațiu, reprezentând linia, respectiv coloana.

Date de ieșire

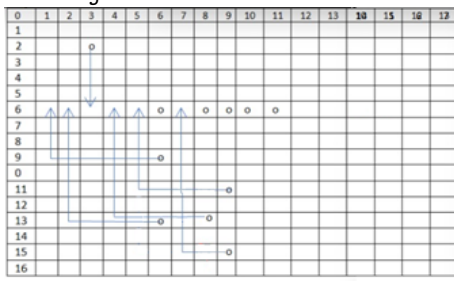
Fișierul de ieșire `soldati.out` va conține pe prima linie un număr natural $Dmin$ reprezentând distanța totală minimă parcursă pentru ca soldații să se alinieze conform cerinței C din fișierul de intrare. Dacă $C = 1$ pe cea de-a doua linie va fi scris un număr natural Lin reprezentând linia pe care se face alinierea. Dacă $C = 2$, pe cea de-a doua linie se vor scrie două numere naturale separate prin spațiu $Lin \ Col$, reprezentând linia pe care se va face alinierea, respectiv coloana de la care soldații încep să se alinieze. În cazul în care există mai multe linii pe care soldații se pot alinia parcurgând distanța totală minimă, se va afișa cea mai mică dintre ele. Pentru $C = 2$, în cazul în care există mai multe coloane Col începând cu care soldații se pot alinia pe linia Lin parcurgând distanța totală minimă, se va afișa cea mai mică dintre ele.

Restricții

- $2 \leq n, m \leq 10^6$
- $1 \leq S \leq \min(m, 10^5)$
- Pozițiile inițiale ale soldaților sunt distincte.

#	Punctaj	Restricții
1	5	Soldații se află inițial pe aceeași linie.
2	14	$2 \leq n, m \leq 1\,500$
3	12	$C = 1, 1\,500 < n, m \leq 10\,000$
4	33	$C = 1$, fără alte restricții
5	12	$C = 2, 1\,500 < n, m \leq 10\,000$
6	24	$C = 2$, fără alte restricții

Exemple

soldati.in	soldati.out	Explicații
1 16 17 11 2 3 6 6 6 8 6 9 6 10 6 11 9 6 11 9 13 8 13 6 15 9	54 6	Mai jos este reprezentat platoul și modul în care soldații se rearanjează 
2 16 17 11 2 3 6 6 6 8 6 9 6 10 6 11 9 6 11 9 13 8 13 6 15 9	46 6 3	2 3->6 3 4 pași pe coloana 3 6 6->6 4 2 pași pe linia 6 6 8->6 7 1 pas pe linia 6 6 9->6 9 0 pași (stă pe loc) 6 10->6 12 2 pași pe linia 6 6 11->6 13 2 pași pe linia 6 9 6->6 5 3 pași pe coloana 6 + 1 pas pe linia 6 11 9->6 10 5 pași pe coloana 9 + 1 pas pe linia 6 13 8->6 8 7 pași pe coloana 8 13 6->6 6 7 pași pe coloana 6 15 9->6 11 9 pași pe coloana 9 + 2 pași pe linia 6

Problema Tramvaie

Fișier de intrare `tramvaie.in`
Fișier de ieșire `tramvaie.out`

Un grup de olimpici plănuiesc să viziteze orașul Ortogonal situat în RUDA (Regatul Unit al celor Două Axe). Ori de câte ori planifică o călătorie, ei studiază hotelurile și restaurantele din oraș. Din experiențele trecute, ei știu că vor mânca foarte multă pizza și că întoarcerea de la restaurant la hotel va fi dificilă.

Prin urmare, pe harta orașului Ortogonal trasează un sistem de coordonate cartezian și identifică pe hartă hotelurile și restaurantele.

Străzile din orașul Ortogonal sunt fie paralele cu axa Ox , fie paralele cu axa Oy , distanța dintre oricare două străzi paralele consecutive fiind egală cu 1. Dacă ei se deplasează pe jos din punctul de coordonate (x_1, y_1) în punctul de coordonate (x_2, y_2) distanța parcursă va fi $|x_1 - x_2| + |y_1 - y_2|$. Însă atunci când ești ghiftuit cu pizza, vrei să mergi pe jos cât mai puțin. De aceea olimpicii noștri vor utiliza pentru deplasare tramvaiele. Liniile de tramvai se află pe unele dintre străzi, ca urmare și ele sunt fie paralele cu Ox , fie paralele cu Oy . În orașul Ortogonal nu există stații, tramvaiul poate fi luat de oriunde de pe o linie de tramvai și se poate coborî din tramvai oriunde pe linia de tramvai.

Cerință

Cunoscând amplasarea liniilor de tramvai în orașul Ortogonal, scrieți un program care să răspundă la Q întrebări de forma următoare: "Care este distanța minimă care trebuie să fie parcursă pe jos pentru a ne deplasa din punctul de coordonate (x_1, y_1) în punctul de coordonate (x_2, y_2) ?"

Date de intrare

Fișierul de intrare `tramvaie.in` conține pe prima linie numerele naturale N , M și Q , care reprezintă numărul de linii de tramvai paralele cu Ox , numărul de linii de tramvai paralele cu Oy , respectiv numărul de întrebări. Pe a doua linie se găsesc N numere întregi care reprezintă ordonatele la care se găsesc liniile de tramvai paralele cu Ox . Pe a treia linie se găsesc M numere întregi care reprezintă abscisele la care se găsesc liniile de tramvai paralele cu Oy . Pe următoarele Q linii se găsesc câte 4 numere întregi x_1 y_1 x_2 y_2 , reprezentând coordonatele celor două puncte pentru care trebuie să determinăm distanța minimă care trebuie să fie parcursă pe jos pentru a ajunge dintr-un punct în celălalt. Valorile scrise pe aceeași linie sunt separate prin câte un spațiu.

Date de ieșire

Fișierul de ieșire `tramvaie.out` va conține Q linii. Pe linia i se află un singur număr natural, reprezentând răspunsul la cea de a i -a întrebare din fișierul de intrare.

Restricții

- $1 \leq N, M, Q \leq 100\,000$
- Coordonatele liniilor de tramvai și ale punctelor între care ne deplasăm sunt numere întregi cuprinse între 0 și 10^9 .
- Nu vor exista două linii de tramvai paralele cu Ox cu aceeași ordonată.
- Nu vor exista două linii de tramvai paralele cu Oy cu aceeași abscisă.
- Este recomandat să nu utilizați x_1 sau y_1 ca nume de variabile.

#	Punctaj	Restricții
1	12	Toate numerele din fișierul de intrare sunt cel mult 1000
2	24	$N, M, Q \leq 1000$
3	7	$N = M = 1$
4	57	Fără restricții suplimentare

Exemple

tramvaie.in	tramvaie.out	Explicații
2 3 4	3	În figură este prezentat orașul:
3 12	3	
10 11 2	2	
1 1 14 14	2	
2 9 7 8		
8 2 12 4		
6 8 5 7		

Explicație

Între punctele roșii (prima pereche de puncte din fișierul de intrare) distanța minimă parcursă pe jos este 3. Aceasta se poate obține mergând spre est din punctul (1, 1) până la linia de tramvai paralelă cu Oy situată la abscisa 2 (distanța parcursă pe jos fiind 1), apoi cu tramvaie până la punctul de coordonate (14, 12) de unde se merge spre nord pe jos până la (14, 14).

Între punctele albastre distanța minimă este tot 3. Aceasta se poate obține mergând cu tramvaiul din punctul (2, 9) (care se află pe o linie de tramvai) până la punctul de coordonate (10, 8) și apoi pe jos spre vest până la punctul de coordonate (7, 8).

Pentru punctele mov distanța minimă este 2 și se poate obține mergând pe jos spre nord până la linia de tramvai paralelă cu Ox aflată la ordonata 3, apoi cu tramvaiul până la punctul de coordonate (12, 3) și apoi pe jos spre nord până la punctul de coordonate (12, 4).

Pentru punctele verzi, drumul în care se merge cel mai puțin pe jos este cel în care nu folosim niciun tramvai. Lungimea acestuia este $|6 - 5| + |8 - 7| = 2$.