

Problema Acoperire

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

Florice! vrea să facă cât mai mulți bani. Ca să aibă suficienți bani să-și poată cumpăra un apartament, are de rezolvat o problemă care se poate modela astfel: El are N intervale inițiale, date prin capetele lor. Florice! mai trebuie să creeze intervale noi, denumite intervale de acoperire.

Un interval inițial se consideră acoperit dacă există cel puțin un interval de acoperire cu intersecția dintre intervalul de acoperire și cel inițial de lungime cel puțin jumătate din lungimea intervalului inițial. De exemplu, intervalul $[0, 4]$ este acoperit de mulțimea de intervale $\{[2, 5], [-1, 1]\}$, dar nu este acoperit de mulțimea de intervale $\{[0, 1], [2, 3], [3, 4]\}$.

Cerință

Prietenul său, Ted, îi spune că are nevoie de mai multe provocări în viață să fie mai fericit, și îi pune Q întrebări de forma: *“Dacă ai voie să creezi cel mult K intervale de acoperire, care ar fi lungimea minimă a celui mai lung interval de acoperire astfel încât toate intervalele inițiale să fie acoperite? Și dacă poți, care este soluția minimă lexicografică? O soluție este minimă lexicografică dacă este minimă întâi după numărul intervalelor de acoperire, iar după aceea comparând intervalele după capetele de stânga și de dreapta, ordonând intervalele după capetele din stânga.”*

Date de intrare

Prima linie a fișierului de intrare `acoperire.in` conține numărul N . Următoarele N linii conțin câte două numere naturale S_i și D_i , unde numerele de pe linia $i + 1$ reprezintă capetele stânga și dreapta ale celui de-al i -lea interval. Urmează o linie cu numărul Q , iar pe următoarele Q linii se află câte un număr K , care reprezintă câte o întrebare a lui Ted, unde vrem să aflăm lungimea minimă a celui mai lung interval de acoperire, dacă putem crea cel mult K intervale de acoperire.

Date de ieșire

Fișierul de ieșire `acoperire.out` trebuie să conțină Q răspunsuri în ordine. Pentru o întrebare cu K dat, pe prima linie trebuie afișată lungimea celui mai lung interval de acoperire în condițiile cerute, pe următoarea linie trebuie să fie numărul de intervale de acoperire folosite (cel mult K), iar pe următoarele linii capetele stânga și dreapta ale intervalelor, ordonate după capătul stânga.

Dacă o valoare este număr întreg, trebuie afișată ca număr întreg, altfel trebuie afișată ca număr real cu fix o zecimală.

Restricții

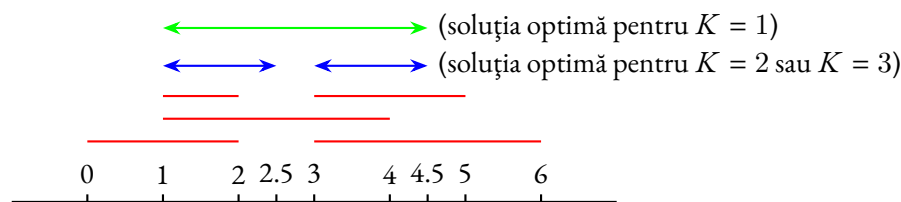
- $1 \leq N \leq 10^5$
- $0 \leq S_i < D_i \leq 10^8$
- $1 \leq Q \leq 20$
- $1 \leq K \leq N$ și suma tuturor K este cel mult 10^5 .
- Se acordă puncte pentru un test doar dacă toate lungimile minime ale celui mai lung interval de acoperire sunt aflate corect. Dacă o soluție nu este corectă sau minimă lexicografică se va acorda totuși 75% din punctajul unei întrebări. Toate întrebările unui test au aceeași valoare ca punctaj.
- Dacă vrei să afișezi doar lungimea, puteți afișa 0 ca numărul intervalelor din soluție.

#	Punctaj	Restricții
1	10	$N = 1$
2	10	$N = 2$ și intervalele inițiale sunt disjuncte.
3	20	$Q = K = 1$
4	20	Intervalele inițiale sunt disjuncte, oricare două.
5	40	Fără restricții suplimentare.

Exemple

acoperire.in	acoperire.out
5	3.5
0 2	1
1 4	1 4.5
1 2	1.5
3 5	2
3 6	1 2.5
3	4 5.5
1	1.5
2	2
3	1 2.5
	3 4.5

Fișierul de intrare conține 5 intervale inițiale, ilustrate cu roșu mai jos.



Pentru $K=1$ avem un singur interval de acoperire, de lungime 3.5, adică $[1, 4.5]$, și soluția este minimă lexicografic.

Pentru $K=2$, răspunsul din exemplul de mai sus reprezintă o soluție cu cel mai lung interval de acoperire de lungime 1.5. Soluția primește doar 75% din punctajul celei de-a doua întrebări, pentru că nu este minimă lexicografic: înlocuirea intervalului $[4, 5.5]$ cu $[3, 4.5]$ reprezintă o soluție mai mică lexicografic.

Pentru $K=3$ soluția este minimă lexicografic, deoarece sunt necesare doar două intervale de acoperire.

Soluția obține $(100+75+100)/3 = 91.66\%$ din punctajul testului.

Epilog

“În final, Floricel și-a dat seama că lucrurile materiale sunt necesare dar nu suficiente pentru împlinire în viață. Acum că nu se mai ia după ce vede la alții, nu mai urmărește fake lifestyle, cringe și propagandă pe social media, iar scopurile spre care tinde sunt alese de el însuși, pentru împlinirea sa și a celor din jur. Acum e cu adevărat fericit.”

Problema Birocratie

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

Boris este un afacerist de succes, având contracte care aduc pe de o parte venituri (dobânzi, comisioane etc.) dar și obligații la taxe (impozite, rate etc.).

Boris s-a hotărât să viziteze o clădire de birouri. Clădirea are un singur nivel în care birourile sunt lipite unele de altele formând un caroiac pătratic de dimensiune $N \times N$. Planul birourilor se poate reprezenta ca o matrice pătratică, unde birourile sunt elemente din matrice cu liniile și coloanele numerotate de la 1 la N . Mai exact, la linia i , coloana j se găsește biroul (i, j) .

Boris va intra în clădire prin biroul $(1, 1)$ și va trece printr-o serie de birouri. Traseul se va termina în biroul (N, N) . La trecerea dintr-un birou în altul se permite:

- pe același rând doar de la stânga la dreapta: $(i, j) \rightarrow (i, j + 1)$;
- pe aceeași coloană doar de sus în jos: $(i, j) \rightarrow (i + 1, j)$;
- în sens diagonal în următoarele două direcții:
 - $(i, j) \rightarrow (i + 1, j - 1)$, dar și
 - $(i, j) \rightarrow (i - 1, j + 1)$.

Totuși, **Boris nu se va mai întoarce niciodată într-un birou din care a ieșit.**

De câte ori Boris vizitează un birou (i, j) el încheie un contract în valoare de $B(i, j)$ lei. Dacă $B(i, j) > 0$, atunci el primește $B(i, j)$ lei, iar dacă $B(i, j) < 0$, el plătește $-B(i, j)$ lei. Scopul lui Boris este acela de a ieși din clădire cu un câștig total maxim. Câștigul se definește ca fiind totalul de bani primiți minus totalul de bani plătiți pe traseu.

Atenție! Câștigul poate să fie și negativ dacă Boris plătește mai mult decât primește.

Cerință

Cunoscând planul birourilor și valorile $B(i, j)$ pentru $1 \leq i, j \leq N$ care îl așteaptă pe Boris în fiecare birou, ajutați-l să calculeze câștigul maxim pe care îl poate avea la ieșirea din clădire.

Date de intrare

Fișierul de intrare `birocratie.in` conține pe prima linie numărul N , iar pe următoarele N linii câte N numere întregi separate prin spații, reprezentând valorile $B(i, j)$ pentru $1 \leq i, j \leq N$.

Date de ieșire

Fișierul de ieșire `birocratie.out` va conține o singură linie pe care se află un singur număr întreg reprezentând câștigul maxim posibil.

Restricții

- $5 \leq N \leq 1\,000$
- $-1\,000 \leq B(i, j) \leq 1\,000$ pentru $1 \leq i, j \leq N$.

#	Punctaj	Restricții
1	12	B are toate elementele pozitive, $N \leq 300$.
2	12	B are toate elementele egale și negative, $N \leq 300$.
3	15	Pe fiecare diagonală paralelă cu diagonală secundară elementele din B sunt egale, $N \leq 300$.
4	13	Elementele de pe chenarul lui B sunt negative, iar celelalte elemente sunt pozitive, $N \leq 300$.
5	13	Toate elementele din B sunt egale în valoare absolută (modul), elementele de pe chenar sunt pozitive, iar celelalte elemente sunt negative, $N \leq 300$.
6	16	$N \leq 300$
7	19	Fără restricții suplimentare.

Mai sus prin *chenarul* lui B ne referim la elementele care se află pe prima/ultima linie și prima/ultima coloană.

Exemple

birocratie.in	birocratie.out	Explicații
5 1 2 5 8 2 1 3 -10 2 1 0 9 1 -7 3 -2 3 4 -1 2 3 -4 2 3 1	42	Câștigul maxim este 42, obținut adunând elementele evidențiate în traseul ilustrat mai jos.

1	2	5	8	2
1	3	-10	2	1
0	9	1	-7	3
-2	3	4	-1	2
3	-4	2	3	1

Problema Nestemate

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

Dominic este un alchimist renumit pentru experimentele sale cu pietre prețioase. De-a lungul carierei sale a reușit să strângă o colecție de N nestemate pe care le-a numerotat de la 1 la N . Conform studiilor sale, aspectul fiecărei nestemate este caracterizat prin trei întregi X, Y și Z reprezentând culoarea, claritatea și strălucirea acesteia.

Dominic a descoperit o metodă secretă prin care poate face o nestemată din colecția sa să capete aspectul unei alte nestemate din colecție. Metoda are însă o slăbiciune, reușind dacă și numai dacă cel puțin una din valorile primei nestemate este egală cu cel puțin una dintre valorile celeilalte nestemate, dar este irelevant dacă proprietatea pe care o reprezintă cele două valori coincide. De exemplu, nestemata $(1, 3, 4)$ poate fi transformată în nestemata $(3, 2, 2)$ deoarece ambele au una din proprietăți egală cu 3.

Curioasă să-i testeze abilitățile de alchimist, Regina a ales două pietre din colecție numerotate cu A și B și i-a ordonat să facă piatra A să capete aspectul pietrei B . Dominic trebuie să îndeplinească sarcina aplicând transformări succesive asupra pietrei A , căreia îi va schimba proprietățile de fiecare dată. Deoarece regina pare un pic nerăbdătoare, el speră să reușească folosind un număr cât mai mic de transformări. Dominic vă roagă pe voi să aflați dacă porunca reginei poate fi realizată, iar în caz afirmativ, care este numărul minim de transformări necesare în acest sens.

Atenție! În cazul în care procesul este posibil, aspectele intermediare ale pietrei A până când ajunge să arate ca B vor corespunde altor pietre din colecție.

Cerință

Se dau numărul de teste T și pentru fiecare test N , A și B și proprietăților celor N nestemate din colecția lui Dominic. Se cere să se afle numărul minim de transformări necesare (dacă este posibil).

Date de intrare

Prima linie a fișierului de intrare `nestemate.in` conține T , numărul de teste. Pentru fiecare test regăsim pe prima linie numărul întreg N , pe a doua linie numerele întregi A și B , iar pe următoarele N linii câte trei valori întregi X, Y și Z descriind nestematele.

Date de ieșire

Fișierul de ieșire `nestemate.out` trebuie să conțină T linii, câte una pentru fiecare test dat, conținând numărul minim de transformări necesare sau -1 dacă procesul nu poate fi realizat.

Restricții

- $1 \leq T \leq 5$
- $2 \leq N \leq 100\,000$
- $1 \leq A, B \leq N$ și $A \neq B$.
- $1 \leq X, Y, Z \leq 500\,000$
- $1 \leq \text{Suma valorilor } N \text{ din toate cele } T \text{ teste} \leq 300\,000$
- Se garantează că este necesară cel puțin o transformare (pietrele A și B nu au același aspect).

#	Punctaj	Restricții
1	11	Se garantează că, dacă există soluție, se poate ajunge la rezultat prin exact o transformare și $N \leq 10$.
2	13	Se garantează că, dacă există soluție, se poate ajunge la rezultat prin maxim două transformări și $N \leq 10$.
3	16	Se garantează că, dacă există soluție, se poate ajunge la rezultat prin maxim trei transformări și $N \leq 10$.
4	10	$N \leq 10$
5	10	$N \leq 1000$
6	13	Se garantează că o valoare apare în configurația a maxim 3 nestemate distincte.
7	27	Fără restricții suplimentare.

Exemple

nestemate.in	nestemate.out	Explicații
2	2	Pentru primul test se
4	-1	folosește nestemata 4 așa cum
1 2		este reprezentat în desenul
2 1 1		de mai jos pentru a
5 3 6		transforma piatra 1 în piatra
4 3 5		2.
3 2 7		Pentru al doilea test
4		procesul de transformare nu
1 3		poate fi realizat, deci se
2 1 1		afișează -1.
2 2 2		
4 3 5		
2 2 7		



Transformarea $(2, 1, 1) \rightarrow (3, 2, 7)$ este validă deoarece $2 \in \{2, 1, 1\} \cap \{3, 2, 7\}$.

Transformarea $(3, 2, 7) \rightarrow (5, 3, 6)$ este validă deoarece $3 \in \{3, 2, 7\} \cap \{5, 3, 6\}$.