# Leghosszabb béke

Ismerjük N egymást követő naphoz az ezeken a napokon zajlott összes háború kezdetének és végének napsorszámait. A napokat egytől sorszámozzuk.

Készíts programot, amely megadja a leghosszabb békés időszak hosszát és kezdetét!

## Bemenet

A *standard bemenet* első sorában a napok száma (1≤N≤100000), valamint a háborúk száma (1≤M≤100000) van. A következő M sorban egy-egy háború első és utolsó napjának sorszáma van (1≤Elsői≤Utolsói≤N).

## Kimenet

A *standard kimenet* első sorába a leghosszabb béke napjainak számát és első napjának sorszámát kell írni! Több azonos hosszú békés időszak esetén a legkorábban kezdődőt kell kiírni! Ha egyáltalán nem volt békés időszak, akkor egyetlen -1-et kell kiírni!

## Példa

|  |  |
| --- | --- |
| Bemenet | Kimenet |
| 50 6 3 20 45 46 28 30 12 20 40 48 8 12 | 9 31  Magyarázat: a békés időszakok: 1-2, 21-27, 31-39, valamint 49-50. |

## Korlátok

Időlimit: 0.2 mp.

Memórialimit: 32 MB

# Legtöbbször szomszédok

Egy díjugrató versenyen összesen N versenyző vesz részt. A versenyzők rajtszámok szerinti sorrendben egymás után következnek és rögtön kapnak is egy pontszámot a teljesítményükre. Így a rangsor minden versenyző ugratása után változik.

Ha egy versenyző már megkapta a pontszámát, akkor onnan kezdve minden ugratás után a rangsorban előtte álló (azaz nála több pontot elért) versenyzők közül a legkevesebb pontot szerzőre azt mondjuk, hogy *közvetlenül* előtte áll a versenyben. Hasonlóan, a nála kevesebb pontot szerzett versenyzők közül aki a legtöbb pontot gyűjtötte, az *közvetlenül* utána áll. Holtverseny esetén többen is lehetnek, akik egyszerre állnak közvetlenül egy versenyző előtt, illetve után. A vele holtversenyben levők sem előtte, sem utána nincsenek.

A híres indián vezér dédunokája, Őrült Ló is elindult a versenyen. Készíts programot, amely megadja, hogy ki állt a legtöbb ugratás után közvetlenül Őrült Ló előtt, illetve után!

## Bemenet

A *standard bemenet* első sorában a díjugratók száma (1≤N≤100000) és Őrült Ló rajtszáma (1≤M≤N) van. A következő sorban az egyes versenyzők pontszáma található (1≤Pi≤10000), rajtszám szerinti sorrendben.

## Kimenet

A *standard kimenet* első sorába azt kell írni, hogy a verseny során ki állt legtöbbször közvetlenül Őrült Ló előtt és hány ugrást követően! A második sorba azt kell írni, hogy a verseny során ki állt legtöbbször közvetlenül Őrült Ló után és hány ugrást követően!

Több megoldás esetén a legkisebb rajtszámút kell kiírni! Ha valamelyikre nincs megoldás, akkor abba a sorba egyetlen -1-et kell kiírni!

## Példa

|  |  |
| --- | --- |
| Bemenet | Kimenet |
| 10 3 8 2 **6** 3 9 4 6 8 1 5 | 1 8 6 4 |
| Magyarázat: Őrült Ló a 3-as rajtszámú versenyző. Az 1. versenyző először a 3. versenyző ugratása után állt közvetlenül a 3. előtt és ezután végig ott volt, azaz a 10. után is – ez összesen 8 alkalom. A 6. versenyző a 6. versenyző ugratását követően állt először közvetlenül a 3. után, utoljára pedig a 9. ugratásnál, ez összesen 4 alkalom. | |

## Korlátok

Időlimit: 0.2 mp.

Memórialimit: 32 MB

# Túra

Peti és az apukája egy közeli erdőben szokott túrázni. Az erdőben tisztások találhatók, melyek között ösvények futnak. Bármelyik tisztásról bármelyik másikra pontosan egyféle képpen juthatnak el az ösvényeken keresztül. Petiék meghatározták az egyes tisztások szépségét. Egy túra során elindulnak egy tisztásról, majd ösvényeken keresztül egy másik tisztásra sétálnak, minden tisztást legfeljebb egyszer érintve. A túra szépségértéke az útközben érintett tisztásokhoz tartozó szépségek összege. A túra akkor szép, ha az értéke nemnegatív.

Írj programot, amely eldönti, hogy minden lehetséges túra szép-e az erdőben! Ha nem, akkor egy nem szép túrát is meg kell adni!

## Bemenet

A standard bemenet első sorában a tesztesetek száma (1≤T≤10) található. Ezt T teszteset leírása követi. Minden teszteset első sorában a tisztások száma található (1≤N≤100000). A második sorban a tisztások szépségei vannak (ti=1, ha szép, ti=-1, ha nemszép). A következő N-1 sor mindegyikében két-két tisztás sorszáma van, amiket ösvény köt össze (1≤ai≠bi≤N). Az egy bemenetben található tesztesetekre az a tisztások száma legfeljebb 100000.

## Kimenet

A standard kimenetre mind a T tesztesethez egy-egy válaszblokk kerüljön! Egy válaszblokk első sorában IGEN álljon, ha minden túra szép az erdőben! Ha nem minden túra szép, akkor az első sorba NEM kerüljön! Ekkor a második sorba egy nem szép túra által érintett tisztások számát kell írni, a harmadik sorban pedig a nem szép túra által érintett tisztások legyenek felsorolva, a túrabeli sorrendjükben! Több nem szép túra esetén bármelyik megadható.

## Példa

|  |  |
| --- | --- |
| Bemenet | Kimenet |
| 2 4 -1 1 -1 -1 1 4 1 2 1 3 5 -1 1 1 -1 1 2 3 4 5 | NEM 3 4 1 3 IGEN |

## Korlátok

Időlimit: 0.4 s

Memórialimit: 64 MB

## Pontozás

A pontszám 8%-a szerezhető olyan tesztekre, ahol N≤10.

A pontszám további 24%-a szerezhető olyan tesztekre, ahol N≤1000.

# Vásárlás

Két rivális áruházlánc, A és B ugyanazt az N féle terméket árulja. Minden termékből egy darabot szeretnénk vásárolni. Az A áruházlánc korlátozásokat vezetett be: ha náluk vásároljuk meg az L és R sorszámú termékeket (L≤R), akkor minden közbülső sorszámú terméket (L≤i≤R) is náluk kell, hogy megvásároljuk.

Írj programot, ami meghatározza, hogy minimálisan mennyi pénzt kell költenünk a termékek megvásárlásához!

## Bemenet

A standard bemenet első sorában a termékek száma található (1≤N≤100000). A második sorban az A áruházlánc termékei árai találhatók (1≤ai≤10000). A harmadik sorban a B áruházlánc termékei árai találhatók (1≤bi≤10000).

**Kimenet**

A standard kimenetre a minimális összköltséget kell írni!

**Példa**

|  |  |
| --- | --- |
| Bemenet | Kimenet |
| 4 5 6 7 8 10 1 2 20 | 21 |

## Korlátok

Időlimit: 0.35 s

Memórialimit: 64 MB

## Pontozás

A pontszám 10%-a szerezhető olyan tesztekre, ahol N≤10.

A pontszám további 10%-a szerezhető olyan tesztekre, ahol N≤100.

A pontszám további 30%-a szerezhető olyan tesztekre, ahol N≤2000.