

Név: Vida Levente

Neptunkód: DLNINP

*Feladat a biroból (lehet képernyőkép is)*

## Első útszakasz többszöri felújítással

Egy  $N+1$  kilométer hosszú utat rossz állapota miatt szakaszonként újra aszfaltoznak. Az aszfaltozást  $M$  alkalommal végzik. Minden alkalomról tudjuk, hogy melyik kilométerszelvénytől ( $K_i$ ) melyik kilométerszelvényig ( $V_i$ ) terjedő szakaszt aszfaltoznak (azaz a 2 7 számpár azt jelzi, hogy a 2-3 km-es szakasztól a 7-8 km-es szakaszig történet a felújítás, a 0 N pedig hogy a teljes utat a 0 km-kőtől az N km-nél kezdődő és  $N+1$  km-nél végződő szakaszig, azaz a teljes utat).

Írj programot, amely megadja az első 10 kilométer hosszú szakasz kezdetét, amelyen legalább háromszor volt felújítás!

### Bemenet

A *standard bemenet* első sorában az út hossza ( $1 \leq N \leq 10\,000$ ) és a felújítások  $M$  száma ( $1 \leq M \leq 100$ ) van, alatta soronként egy-egy felújítás adatai ( $0 \leq K_i < V_i \leq N$ ).

### Kimenet

A *standard kimenet* egyetlen sorába azon legelső útszakasz kezdetét kell írni, ahonnan az első 10 kilométer hosszú szakaszon legalább háromszor volt felújítás! A -1 értéket kell kiírni, ha nem volt ilyen útszakasz.

### Példa

Bemenet

100 5

0 10

0 5

15 95

12 37

13 24

Kimenet

15

<b>Specifikáció</b>	<b>max: 20 pont</b>
<p><u>Specifikáció</u></p> <p> <math>X = \text{Rekord}(\text{kezdo} \in \mathbb{N}, \text{veg} \in \mathbb{N})</math>  <math>B_e: n \in \mathbb{N}, m \in \mathbb{N}, \text{ut} \in X[m]</math>  <math>K_i: \text{db} \in \mathbb{N}, \text{van} \in \mathbb{L}, \text{min} \in \mathbb{U}</math>  <math>E_f: 1 \leq n \leq 10000, 1 \leq m \leq 100, \forall i \in [1..m]: (0 \leq \text{ut}[i].\text{kezdo} \leq \text{ut}[i].\text{veg} \leq n)</math>  <math>U_f: (\text{van}, \text{min}) = \text{MIN}(i := 1..m, \text{ut}[i].\text{kezdo}, \text{DARAB}(j := 1..m, \text{ut}[i].\text{kezdo} \geq \text{ut}[j].\text{kezdo} \wedge \text{ut}[i].\text{kezdo} + 9 \leq \text{ut}[j].\text{veg} \geq 3))</math> </p>	<p>Itt megalkotva, vagy másolva más környezetből vagy kézzel írt megoldás fényképe</p>
<b>Sablon + forrásmegjelölés</b>	<b>max: 5 pont</b>
<p><u>Forrásmegjelölés</u></p> <div style="display: flex; justify-content: space-between;"> <div style="width: 45%;"> <p><b>Feltételes maximumkeresés sablon</b></p> <p><b>Feladat</b> Adott az egész számok egy <math>[e..u]</math> intervalluma, egy <math>f[e..u] \rightarrow \mathbb{H}</math> függvény és egy <math>T[e..u] \rightarrow \text{Logikai feltétel}</math>. A <math>\mathbb{H}</math> halmaz elemein értelmezett egy teljes rendezési reláció. Határozzuk meg, hogy az <math>[e..u]</math> intervallum <math>T</math> feltételt kielégítő elemei közül az <math>f</math> függvény hol veszi fel a <b>legnagyobb értéket</b>, és mondjuk meg, mekkora ez az érték!</p> <p><b>Specifikáció és algoritmus:</b>  <math>B_e: e \in \mathbb{Z}, u \in \mathbb{Z}</math>  <math>K_i: \text{van} \in \mathbb{L}, \text{maxindez}, \text{maxerteH}</math>  <math>E_f: -</math>  <math>U_f: \text{van} = \exists i \in [e..u]: (T(i))</math> és  <math>\text{van} \rightarrow (\text{maxinde}[e..u])</math> és  <math>\text{maxerte} = f(\text{maxinde})</math> és <math>T(\text{maxinde})</math> és  <math>\forall i \in [e..u]: (T(\text{maxinde}) \rightarrow \text{maxerte} \geq f(i))</math></p> <p>Rövidítve:  <math>U_f: (\text{van}, \text{maxinde}, \text{maxerte}) = \text{MAX}(i = e..u, f(i), T(i))</math></p> </div> <div style="width: 45%;"> <p><b>Megszámolás sablon</b></p> <p><b>Feladat</b> Adott az egész számok egy <math>[e..u]</math> intervalluma és egy <math>T[e..u] \rightarrow \text{Logikai feltétel}</math>. Határozzuk meg, hogy az <math>[e..u]</math> intervallumon a <math>T</math> feltétel <b>hányszor</b> veszi fel az igaz értéket!</p> <p><b>Specifikáció</b>  <math>B_e: e \in \mathbb{Z}, u \in \mathbb{Z}</math>  <math>K_i: \text{db} \in \mathbb{N}</math>  <math>E_f: -</math>  <math>U_f: \text{db} = \text{SZUMMA}(i = e..u, 1, T(i))</math></p> <p>Rövidítve:  <math>U_f: \text{db} = \text{DARAB}(i = e..u, T(i))</math></p> </div> </div> <div style="margin-top: 10px;"> <p>forrásfájl: programozasi_minta1b.pdf oldal: 6, 9</p> <p>link:  <a href="https://canvas.elte.hu/courses/38943/files/2451951?module_item_id=587641">https://canvas.elte.hu/courses/38943/files/2451951?module_item_id=587641</a> </p> </div>	<p>Lehet képernyőkép vagy másolat az előadás anyagából.</p> <p>Forrásmegjelölés : a forrásfájl neve (publikálás útvonala) és oldalszám/dia sorszáma.</p>

Visszavezetési táblázat (megfeleltetés)	max: 5 pont																																
<div>Visszavezetés</div> <div>1. Tétel: <u>Megszámlálás</u></div> <div><math>i := e..u \sim j := 1..m</math><math display="block">T(i) \sim \text{út}[i].\text{bezd} \geq \text{út}[j].\text{bezd} \wedge \text{út}[i].\text{bezd} + 9 \leq \text{út}[j].\text{vég}</math><math display="block">db \sim db</math></div> <div>2. Tétel: <u>Feltételes minimumberesés</u></div> <div><math>i := e..u \sim i := 1..m</math><math display="block">f(i) \sim \text{út}[i].\text{bezd}</math><math display="block">T(i) \sim db \geq 3</math></div>	<div>A specifikáció és a minta közötti kapcsolat leírása</div>																																
Algoritmus	max: 20 pont																																
<div>Algoritmus</div> <table><tr><td colspan="4"><math>i := 1..m, \text{min} = -1, \text{van} := \text{hamis}</math></td></tr><tr><td colspan="4"><math>db = 0</math></td></tr><tr><td colspan="4"><math>j := 1..m</math></td></tr><tr><td colspan="4"><math>\text{út}[i].\text{bezd} \geq \text{út}[j].\text{bezd} \wedge \text{út}[i].\text{bezd} + 9 \leq \text{út}[j].\text{vég}</math></td></tr><tr><td colspan="4"><math>db := db + 1</math></td></tr><tr><td><math>db &lt; 3</math></td><td colspan="2"><math>db \geq 3 \wedge \text{van}</math></td><td><math>db \geq 3 \wedge \neg \text{van}</math></td></tr><tr><td>—</td><td><math>\text{út}[i].\text{bezd} &lt; \text{min}</math></td><td colspan="2"><math>\text{van} := \text{igaz}</math></td></tr><tr><td></td><td><math>\text{min} := \text{út}[i].\text{bezd}</math></td><td>—</td><td><math>\text{min} := \text{út}[i].\text{bezd}</math></td></tr></table>	$i := 1..m, \text{min} = -1, \text{van} := \text{hamis}$				$db = 0$				$j := 1..m$				$\text{út}[i].\text{bezd} \geq \text{út}[j].\text{bezd} \wedge \text{út}[i].\text{bezd} + 9 \leq \text{út}[j].\text{vég}$				$db := db + 1$				$db < 3$	$db \geq 3 \wedge \text{van}$		$db \geq 3 \wedge \neg \text{van}$	—	$\text{út}[i].\text{bezd} < \text{min}$	$\text{van} := \text{igaz}$			$\text{min} := \text{út}[i].\text{bezd}$	—	$\text{min} := \text{út}[i].\text{bezd}$	<div>Itt megalkotva vagy képernyőkép vagy kézzel írt megoldás fényképe</div>
$i := 1..m, \text{min} = -1, \text{van} := \text{hamis}$																																	
$db = 0$																																	
$j := 1..m$																																	
$\text{út}[i].\text{bezd} \geq \text{út}[j].\text{bezd} \wedge \text{út}[i].\text{bezd} + 9 \leq \text{út}[j].\text{vég}$																																	
$db := db + 1$																																	
$db < 3$	$db \geq 3 \wedge \text{van}$		$db \geq 3 \wedge \neg \text{van}$																														
—	$\text{út}[i].\text{bezd} < \text{min}$	$\text{van} := \text{igaz}$																															
	$\text{min} := \text{út}[i].\text{bezd}$	—	$\text{min} := \text{út}[i].\text{bezd}$																														

**Kód forrása**

**max 11 pont**

biro.elte.hu téma: Programozás 2. beadandó  
 feladat: Első útszakasz többszöri felújítással  
 beadás (feltöltés) sorszáma: 7.  
 pontszám: 100

## Kód

```
using System;
namespace felujitas
{
    internal class Program
    {
        struct public int kezd, veg;

        static void Main(string[] args)
        {
            // Beolvasás
            int n, m;
            do
            {
                string[] firstRow = Console.ReadLine().Split(' ');
                n = int.Parse(firstRow[0]);
                m = int.Parse(firstRow[1]);
            } while (!(1 <= n && n <= 10000 && 1 <= m && m <= 100));

            x[] ut = new x[m];
            for (int i = 0; i < m; i++)
            {
                do
                {
                    string[] row = Console.ReadLine().Split(' ');
                    ut[i].kezd = int.Parse(row[0]);
                    ut[i].veg = int.Parse(row[1]);
                } while (!(0 <= ut[i].kezd && ut[i].kezd < ut[i].veg && ut[i].veg <=
n));
            }
            // Feladat
            int min = -1;
            bool van = false;
            for (int i = 0; i < m; i++)
            {
                int db = 0;
                for (int j = 0; j < m; j++)
                {
                    if (ut[i].kezd >= ut[j].kezd && ut[i].kezd + 9 <= ut[j].veg)
                        db++;
                }

                if (db >= 3 && van == false)
                {
                    van = true;
                    min = ut[i].kezd;
                }
                else if (db >= 3 && van == true)
                {
                    if (ut[i].kezd < min)
                        min = ut[i].kezd;
                }
            }
            // Kimenet
            Console.WriteLine(min);
        }
    }
}
```

## Teszt

max 3 pont

	Saját 1.	Saját 2.	Saját 3
Bemenet:	20 4 0 3 3 18 2 17 4 15	15 3 0 5 5 10 10 15	30 3 0 29 3 17 1 15
Kimenet:	4	-1	3