**Algoritmusok és Adatszerkezetek Beadandó Dokumentáció**

Készítette: Magyar Viktor

Neptun kód: O9WEJJ

Feladat Sorszáma: 34

Feladat: Olimpiai Staféta

# Feladat Leírás

## Olimpiai staféta

Az olimpiai lángot egy kiindulási városból a cél városba kell eljuttatni. A két város távolsága K kilométer. Sok futó jelentkezett, mindegyikről tudjuk, hogy hányadik kilométertől hányadik kilométerig vállalja a futást. Ha egy futó az x kilométertől az y kilométerig vállalja a futást, akkor minden olyan futó át tudja venni tőle a lángot, aki olyan z kilométertől vállalja a futást, amire x≤z≤y.

Készíts programot, amely kiszámítja, hogy legkevesebb hány futó kell ahhoz, hogy a láng eljusson a cél városig!

## Bemenet

A standard bemenet első sorában a két város távolsága (10≤K≤1000) és a jelentkezett futók száma (2≤N≤20 000) van. A további N sor mindegyike két egész számot tartalmaz (0≤I<E≤K), ami azt jelenti, hogy egy futó az I-edik kilométertől az E-edik kilométerig vállalja a láng továbbítását.

## Kimenet

A standard kimenet első sorába a láng célba juttatásához minimálisan szükséges futók M számát kell írni! A második sor pontosan M számot tartalmazzon (egy-egy szóközzel elválasztva), azon futók sorszámait, akik teljesítik a feladatot: a felsorolásban a j-edik futó a j+1-edik futónak adja át a lángot! Több megoldás esetén bármelyik megadható. Ha a láng nem juttatható el a cél városig a jelentkezett futókkal, akkor a kimenet első és egyetlen sorába 0-t kell írni!

# Algoritmus Magyarázat

## Terv

Rendezzük sorba a futókat az indulási hely szerint! Ezután fontos, hogy az első kiválasztott futó az lesz, aki a 0 kilométertől a legmesszebbre képes menni, tehát először ki kell választani a maximális távolságra érkező futót, azok közül, akik a 0 kilométertől futnak.

Az utoljára kiválasztott futó érkezési helyéig válasszuk ki azt a futót, aki a legmesszebb vinné a lángot! Ha a következő futó már később indul, mint az aktuális futó befejezné a futást és a legmesszebb menő futó áttudja venni a lángot, akkor a legmesszebb menőnek kell átadnia a lángot.

## Algoritmus

**Itt feltesszük, hogy a Futok tömb már rendezve van az Indulási hely szerint.**

**FutoVallalas** = **Rekord**(ID **x** I **x** E,ID,I,E: **Egész**); -> **MultiHalmaz**

**Kivalogatas**(Futok: Tömb(1..\*: FutoVallalas), db: Egész, Key: Tömb(1..\*: Egész), Value: Tömb(1..\*: Egész)):

**Változók**

lm: **Egész**

db:=0; Key(db):=lm; Value(db):=Futok[lm].ID; lm:=MaxKereses(Futok);

**Ciklus** i=lm+1**-től** (Hossz(Futok)-1**)-ig**

**Ha** Futok[i].E>Futok[lm].E

**akkor** lm:=i;

**Ha** Futok[i+1].I>Futok[Key[db]].E **és** Futok[Key[db]].E >= Futok[lm].I **és nem** Key.Contains(lm)

**akkor** db:=db+1; Key[db] := lm; Value[db] := Futok[lm].ID;

**Ciklus vége**

**Eljárás vége**.

**Megjegyzés:** Ha a Futok[Key[db]].E < K, akkor nincs megoldás, mert nem tudják a jelentkezők elvinni a célvárosig a lángot.

**MaxKereses**(Futok: Tömb(1..\*: FutoVallalas)): **Egész**

**Változók**

max: **Egész**;

maxindex: **Egész**;

i: **Egész**;

max := Futok[0].E;

maxindex:= 0;

i:= 1;

**Ciklus amíg** i<Hossz(Futok) és Futok[i].I = 0

**Ha** Futok[i].E > max

**akkor** max := Futok[i].E; maxindex := i;

i:= i+1;

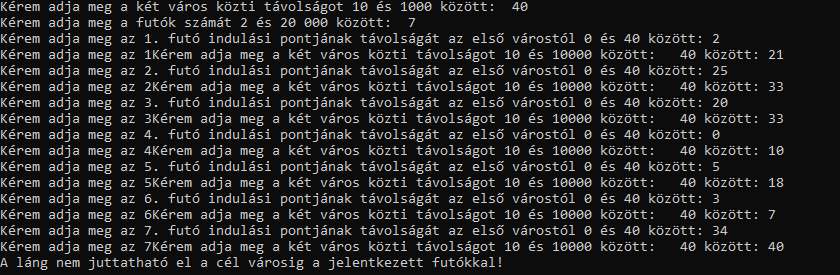
**Ciklus vége**

MaxKereses:= maxindex;

**Eljárás vége.**

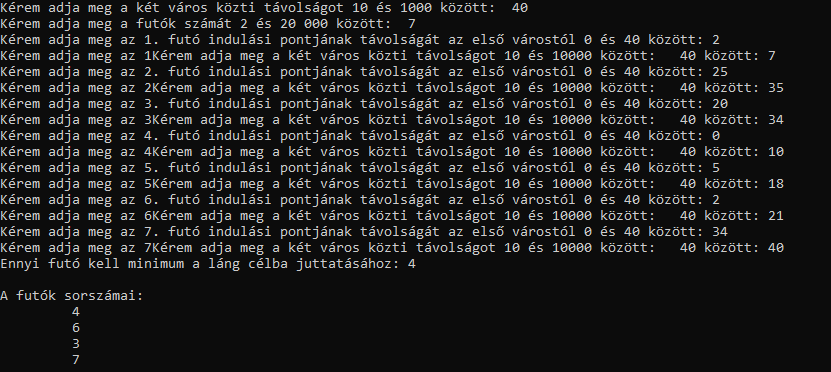
# Tesztesetek

A **TesztBemenetekEsKimenetek** mappában találhatóak a teszt bemenetek és kimenetek. A minta be- és kimenetekhez még hozzáírtam hármat, amiknek a futását bemutatom a felhasználóbarát verzión.

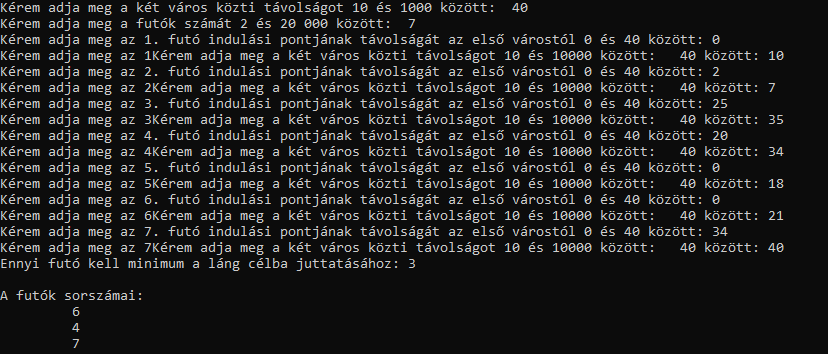
**Bemenet:** be3.txt

**Kimenet:** ki3.txt (A 0 kimenet azt jelenti, hogy a láng nem juttatható el a cél városig.)

**Bemenet:** be4.txt

**Kimenet:** ki4.txt

**Bemenet:** be5.txt

**Kimenet:** ki5.txt