Számítógép nélküli feladatok

1. feladat: Sorrendbe rakás (50 pont)

Az alábbi ábrán egy sétapálya látható. A pályán a nyilaknak megfelelő irányba lehet haladni. A sétapálya egyes betűvel jelölt pontjaira ellenőröket szeretnénk kiküldeni, de csak olyan sorrendben, hogy egy adott pontra akkor mehet ellenőr, ha **minden** olyan pontra küldtünk már, amelyből erre a pontra el lehet jutni (például, ha A→B, B→C,B→D utak vannak, akkor először A-ba kell küldeni ellenőrt, utána B-be, utána pedig C és D sorrendje mindegy.



A. Add meg az ellenőrök kiküldésének összes lehetséges sorrendjét!

B..G. Add meg, hogy az A pontból indulva egy sétáló ember az egyes pontokra (B..G) hányféleképpen juthat el!

2. feladat: Társasjáték (70 pont)[[1]](#footnote-1)

Mint minden társasjátékban, az egyes mezőkre itt is csak bizonyos szabály szerint lehet ráugrani.

Ennél a játéknál minden mezőhöz tartozik egy szabály. Háromféle szabály van.

* nB: n mezőt ugrunk balra, 2B azt jelenti tehát, hogy 2 mezőt ugrunk balra:
* nJ: n mezőt ugrunk jobbra, 3J így azt jelenti, hogy 3 mezőt ugrunk jobbra:
* 0: nem lehet tovább ugrani.

A. Hányadik mezőn kell kezdeni ahhoz, hogy a játék végén minden mezőn egyszer járjunk? Rajzold az ábrára nyilakkal a bejárt mezők sorrendjét is!



B. Mely mezőkre nem lépünk, ha a legelső mezőn kezdjük a játékot?

C. Mely mezőkre nem lépünk, ha a legutolsó mezőn kezdjük a játékot?

D. Mely mezőkre nem lehet lépni sehonnan, ha az ötödik mezőn levő 3J helyére 1J-t írnánk?

E. Honnan lehetne eljutni a negyedik mezőre (ha nem onnan indulunk), ha az ötödik mezőn levő 3J helyére 2B-t írnánk; és mely mezőkön járnánk, ha olyan mezőről indulnánk, ahonnan nem lehet eljutni a negyedik mezőre?

3. feladat: Mit csinál (110 pont)

Az alábbi algoritmus egy N elemű, 1-től N-ig indexelt pozitív számokat tartalmazó X vektort dolgoz fel a K (K>0) változó segítségével, eredményét az U, V, L változókba írja. (X[N+1]-et segédelemként használja.)

Valami(K,N,X,L,U,V):
 U:=1; V:=1; S:=X[1]; X[N+1]:=K+1
 Ciklus amíg V≤N és S≠K
 Ha S<K akkor V:=V+1; S:=S+X[V]
 különben ha S>K akkor S:=S-X[U]; U:=U+1
 Ciklus vége
 L:=(S=K)
Eljárás vége.

A. Mi lesz L,U,V értéke, ha N=5, X=(3,4,7,2,1), K=9?

B. Mi lesz L,U,V értéke, ha N=6, X=(3,4,2,2,4,8), K=8?

C. Mi lesz L,U,V értéke, ha N=3, X=(1,4,2), K=6?

D. Mi lesz L,U,V értéke, ha N=5, X=(2,5,1,1,0), K=6?

E. Hogyan függ az S változó értéke a ciklusfeltétel vizsgálatakor az X vektortól?

F. Mi a feltétele annak, hogy az eljárás végén L értéke hamis legyen?

G. Fogalmazd meg általánosan, hogyan függ U és V értéke az X vektorban szereplő értékektől!

4. feladat: Automata (70 pont)[[2]](#footnote-2)

Egy automata kezdetben A állapotban van, jeleket olvas és a jelek hatására az állapota megváltozhat. Ha A állapotban a bemenetére 1-es jel érkezik, akkor marad A állapotban, ha 0 érkezik, akkor átkerül B állapotba. Ha B állapotban a bemenetére 0 érkezik, akkor marad B állapotban, ha pedig 1, akkor átkerül A állapotba.

Az automata az alábbi rajzzal ábrázolható:

![C:\Users\zsako.ZSAKOL\AppData\Local\Microsoft\Windows\INetCache\Content.Word\2011-0064_58_nyelvek_es_automatak[1].png]()

Például 110 hatására az automata a három jel olvasása után B állapotban lesz. Az első 1-es hatására A-ban marad, a második 1-es hatására is A-ban marad, a 0 hatására B állapotba lép.

A. Milyen állapotban lesz az automata az 11010 jelsorozat hatására? Add meg az egyes jelek utáni állapotot is!

B. Milyen állapotban lesz az automata a 0101010011 jelsorozat hatására? Add meg az egyes jelek utáni állapotot is!

C. Milyen jeleket kapott az automata, ha az A állapotból az egyes jelek után a B,B,A,B,A,A,B állapotokba került?

D. Milyen jelsorozatokra lesz az automata a legvégén B állapotban?

Számítógépes feladat – VÁLASZTHATÓ

5. feladat: Hajók a kikötőben (100 pont)

Egy tengeri kikötőbe hajók érkeznek, érkezési időpontjaikat (nap sorszám) ismerjük időrendben. Az érkezéseket M napon át figyelték.

Készíts programot, amely megadja:

1. az M nap alatti leghosszabb időszak hosszát, amikor nem érkezett hajó;
2. annak a napnak a sorszámát, amikor a legtöbb hajó érkezett (több megoldás esetén az elsőt);
3. a leghosszabb időszak első és utolsó napját, amikor minden nap érkezett hajó (több megoldás esetén az elsőt)!

A *standard bemenet* első sorában a hajók száma (1≤N≤10) és a napok száma (1≤M≤100) van. A következő N sorban egy-egy hajó érkezésének napja van (1≤Ei≤M).

A *standard kimenetre* három sort kell írni, az elsőbe a leghosszabb szakasz napjainak a számát, amikor nem érkezett hajó (0, ha nincs ilyen), a másodikba egy olyan nap sorszámát, amikor a legtöbb hajó érkezett (több megoldás esetén a legkorábbit), a harmadikba pedig a leghosszabb időszak első és utolsó napjának sorszámát, amikor minden nap érkezett hajó (több megoldás esetén a legkorábbit)!

Példa:

|  |  |
| --- | --- |
| Bemenet | Kimenet |
| 7 4536642424243 | 354242 43 |
| *A Neumann János Számítógép-tudományi Társaság az Eötvös Loránd Tudományegyetem Informatikai Kara közreműködésével a 2019/2020-as tanévre meghirdeti a Nemes Tihamér Online Programozási Versenyt (*[*http://tehetseg.inf.elte.hu/nemes-online/nemes\_aktualis.html*](http://tehetseg.inf.elte.hu/nemes-online/nemes_aktualis.html)*). A versenyt öt fordulóban rendezzük meg. Az egyes fordulókban a versenyzőknek 3-5 programozási feladatot kell megoldaniuk C++, Pascal, C#, Python, Visual BASIC vagy Java nyelven, a beküldött megoldásokat Linux környezetben fordítjuk újra és az így kapott programokat értékeljük. Azoknak is lehetőséget nyújt a versenyzésre, akik nemrég kezdték a programozás tanulást. A versenyre nevezni a verseny honlapján lehet, a szükséges adatok (név, település, iskola, évfolyam, email) megadásával a https://biro.inf.elte.hu/faces/versenyre.xhtml oldalon.* |

Számítógép nélküli feladat – VÁLASZTHATÓ

5. feladat: Tornádók (100 pont)

A tornádófigyelő szolgáltnak az elmúlt M (M>0) napban N (N>0) tornádót jelentettek be, időrendi sorrendben A T[1]..T[N] tömbelemek tartalmazzák az egyes tornádók napjai sorszámait.

Készítettünk egy programot, amely megadja:

1. azon napok számát, amikor nem volt tornádó (Adb);
2. a leghosszabb időszak napjai számát, amikor egyik napon sem volt tornádó (Bdb);
3. mennyi volt a legtöbb tornádó egy napon (Cdb);
4. a leghosszabb időszak első és utolsó napját, amikor minden nap volt tornádó (De,Du).

A megoldás sajnos hibás lett, keresd meg a hibákat az alábbi algoritmusban!

Tornádók:
 Adb:=T[1]-1
 Ciklus i=2-től N-ig
 Ha T[i]-T[i-1]≥0 akkor Adb:=Adb+T[i]-T[i-1]
 Ciklus vége
 Adb:=Adb+M-T[N]-1
 Bdb:=T[1]-1
 Ciklus i=2-től N-ig
 Ha T[i]-T[i-1]-1<Bdb akkor Bdb:=T[i-1]-T[i]-1
 Ciklus vége
 Ha M-T[N]>Bdb akkor Bdb:=M-T[N]
 c:=1; Cdb:=1
 Ciklus i=1-től N-ig
 Ha T[i]>T[i-1] akkor i:=c
 különben ha i-c+1>Cdb akkor Bdb:=i-c+1
 Ciklus vége
 De:=T[1]; Du:=T[1]; e:=De; u:=Du
 Ciklus i=2-től N-ig
 Ha T[i]=T[i-1]+1 akkor
 Ha u-e>De-Du akkor De:=e; Du:=u Elágazás vége
 e:=T[i]; u:=T[i]
 különben e:=T[i]
 Ciklus vége
 Ha u-e>Du-De akkor De:=e; Du:=u
Eljárás vége.

|  |
| --- |
| *A Neumann János Számítógép-tudományi Társaság az Eötvös Loránd Tudományegyetem Informatikai Kara közreműködésével a 2019/2020-as tanévre meghirdeti a Nemes Tihamér Online Programozási Versenyt (*[*http://tehetseg.inf.elte.hu/nemes-online/nemes\_aktualis.html*](http://tehetseg.inf.elte.hu/nemes-online/nemes_aktualis.html)*). A versenyt öt fordulóban rendezzük meg. Az egyes fordulókban a versenyzőknek 3-5 programozási feladatot kell megoldaniuk C++, Pascal, C#, Python, Visual BASIC vagy Java nyelven, a beküldött megoldásokat Linux környezetben fordítjuk újra és az így kapott programokat értékeljük. Azoknak is lehetőséget nyújt a versenyzésre, akik nemrég kezdték a programozás tanulást. A versenyre nevezni a verseny honlapján lehet, a szükséges adatok (név, település, iskola, évfolyam, email) megadásával a https://biro.inf.elte.hu/faces/versenyre.xhtml oldalon.* |

1. http://e-hod.elte.hu/archiv/feladatok/HOD\_osszes\_2018\_senior.pdf [↑](#footnote-ref-1)
2. A feladat Friedl Katalin, Csima Judit: Nyelvek és automaták című könyvéből származik: https://www.tankonyvtar.hu/hu/tartalom/tamop412A/2011-0064\_58\_nyelvek\_es\_automatak/ar01s02.html [↑](#footnote-ref-2)