Országos Programozó Verseny – Neumann János Egyetem GAMF Kar – 2025

3. forduló (online)

A feladatok megoldásának szabályai

- A 3. forduló négy feladatot tartalmaz és összesen 8 kérdést.
- Minden kérdésre egy nemnegatív egész szám vagy egy string a válasz (az adott kérdésnél szerepel, hogy melyik). Ezeket az eredményeket kell beküldeni. A csapatnak ügyelnie kell arra, hogy pontosan a kérdésnél leírt módon küldje be a választ.
- Beküldési határidő: ***2025. március 15.*** *(23:59)*
*- A megoldások beküldését az alábbi linken található űrlapon keresztül kell elvégezni (**kizárólag a versenyre nevezett csapatoknak**):*

https://forms.gle/rnFmP5bxMDBVBmvj6

- Minden nevezett csapat egyetlen alkalommal küldheti be (fordulónként) a megoldásait. Ha egy csapat többször is beküld megoldást, akkor a legkorábbit vesszük figyelembe a pontozásnál. Tehát csak akkor érdemes a megoldásokat beküldeni, ha valamennyi kérdésre megvan a válasz, vagy a csapat már nem tud vagy nem akar több feladatot megoldani. Kérjük, hogy nevezésnél megadott csapatnéven (karakterhelyesen) küldjék be a válaszokat! Ha a csapat elfelejtette a csapatnevét, akkor a nevezéskor kapott visszaigazoló e-mailben azt megtalálja.
- A megoldások részleteit nem kell beküldeni, csak a kérdésekre adott válaszokat (nemnegatív egész számok vagy stringek).
- A forduló helyes megoldásait és a csapatok pontszámait minden forduló lezárta után ismertetjük.

A 3. forduló feladatai

1. feladat

Egy cégnél futószalagon, egymás után érkeznek az elszállításra váró üres dobozok egy robothoz. Három különböző típusú doboz van, *A*, *B* és *C* jelű. Mindhárom típus felül nyitott kocka alakú, az *A* a legnagyobb és a *C* a legkisebb méretű, ezért megfelelő sorrendben egymásba rakhatóak (az *A* típusúba egy *B* vagy egy *C* típusú, a *B* típusúba pedig egy *C* típusú doboz rakható).
A pakolást egy robot végzi. A szállítószalagon érkező dobozokat az érkezés sorrendjében fogja meg, és a kialakított 1., 2. stb. csomagolóhelyre (vagy a már azon lévő dobozba) teszi. A robot szabadon választhat a már megkezdett csomagolóhelyek és az első szabad hely között. A dobozt mindig oda teszi, ahol a már megkezdett csomagolóhelyek közül először belefér (a csomagolóhelyek sorrendjében), ha nincs ilyen, akkor a következő szabad csomagolóhelyre teszi.
Például: A futószalagon először egy *A* típusú doboz érkezik, ezt az 1. csomaglóhelyre helyezi, majd ezután sorban a következő dobozok érkeznek: *BAC*. A *B* jelűt belehelyezheti az első csomagolóhelyen lévő *A* típusú dobozba, azután az *A* típusút a 2. csomagolóhelyre, majd a C típusút az első helyen lévő AB-be teszi.

1. cs. hely	2. cs. hely	3. cs. hely	4. cs. hely	5. cs. hely	6. cs. hely
ABC	A				

Ha az ezutáni sorrend *ACBCCA*, akkor végül a következő elhelyezés alakul ki:

1. cs. hely	2. cs. hely	3. cs. hely	4. cs. hely	5. cs. hely	6. cs. hely
ABC	AC	ABC	C	A	

a) Amikor a robot jelzi, hogy egy csomagolóhelyre már biztosan nem rakható új doboz, akkor a csomagolóhelyen lévő dobozokat befóliázzák, Amikor a futószalagról elfogynak a dobozok, akkor a nem befejezett csomagokat is befóliázzák.
A *dobozok.txt* állományban a futószalagra érkező dobozok sorrendje található egyetlen, csak *A*, *B* és *C* karaktereket tartalmazó sztringben.
Válaszként az ebből a dobozsorozatból a fenti szabályok szerint elkészíthető fóliázott egységek minimális számát kell megadni (a csomagolóhelyek száma nem korlátozott). (8 pont)

b) A költséghatékonyság növelésére – a robot mozgási idejének csökkentésére – a vállalat korlátozni szeretné a használandó csomagolóhelyek számát, ezért elemezni szeretné, hogy egy-egy dobozsorozat esetén mennyi a csomagolóhely-igény maximuma. Nyitott egy csomagolóhely, ha még lehetséges újabb dobozt helyezni rá vagy a rajta lévő dobozokba. A lefóliázandó (tovább nem bővíthető) csomagokat a csomagolóhelyről elviszik, a nyitott csomagokat egy csomagolóhellyel előrébb mozgatja majd egy másik mozgó szalag (a sorrenden nem változtatva).
Válaszként adja meg a *dobozok.txt* állományban lévő dobozsorozatra a csomagolóhely-igény maximális értékét! (5 pont)

c) Az elemzéshez további szimuláció is készül. A tapasztalatok szerint mindhárom típusú doboz 4% hibával készül. A hibás *A* típusú dobozokba *B* típusú doboz nem, de *C* típusú rakható. A hibás *B* típusú dobozokba nem rakható *C* típusú. Másfajta problémát nem okoznak a hibás dobozok.
A *dobozok.txt* állományban lévő adatsor alapján készítsen szimulációt a fenti hibákra úgy, hogy az *A* típusúakból a 25.-ként érkezőt és innen minden 25.-et tekinti hibásnak, ugyanígy a *B* típusúak közül az 25.-et és innen minden 25.-ediket tekinti hibásnak.
Válaszként adja meg, hogy így mekkora a csomagolóhely-igény maximuma! (A b) feladatban vázolt csomagolási algoritmus alapján.) (8 pont)

2. feladat

(Az Euler Project egyik feladatötlete alapján)

Értelmezzük egy sorozat tagjainak kiszámítási módját a következő iterációval:
Ha a sorozat egy tagja az n (az n pozitív egész szám), akkor a következő tag legyen:

n/2, ha n páros
3n+1, ha az n páratlan.

Pl.: ha a sorozat első tagja a 20, akkor a sorozat tagjai rendre:

$$20 \rightarrow 10 \rightarrow 5 \rightarrow 16 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 1$$

A sorozat elemeinek felírását addig folytassuk, amíg el nem érjük az 1-et. (Egy sejtés, hogy bármely pozitív egész számról indulva, mindig elérjük az 1-et véges sok lépésben.)
A példában szereplő sorozatnak összesen 8 tagja van.

Az 1 000 000 alatti pozitív egész számok közül melyik számról indulva lesz legtöbb tagja a sorozatnak?
Válaszként a kezdőszámot kell megadni és tőle egy függőleges (|) vonallal elválasztva a sorozat tagjainak a számát. A fenti példa esetén ez 20|8. (3 pont)

3. feladat

(Az Euler Project egyik feladatötlete alapján)

Képezzük négy tetszőleges számjegy (nem feltétlenül különböző) összes lehetséges különböző permutációját (ha a 4 számjegy különböző, akkor 24 db van). Ha számjegyek között szerepel a 0, akkor ezek nem mindegyike lesz négyjegyű (a 0-val kezdődő számokat nem tekintjük négyjegyűnek). Ha számjegyek között vannak azonosak is, akkor a permutációk száma kevesebb.

Pl.: a számjegyek: 1; 2; 4; 9
A permutációk: 1249, 1294, 1429, 1492, 1924, 1942, 2149, 2194, 2419, 2491, 2914, 2941, 4129, 4192, 4219, 4291, 4912, 4921, 9124, 9142, 9214, 9241, 9412, 9421
Ezek között 6 db prímszám van: 1249, 1429, 4129, 4219, 9241, 9421

a) Ha tekintjük a tízes számrendszer számjegyei közül az összes lehetséges módon kiválasztott számnégyesek négyjegyű permutációit (a kiválasztott számnégyesek tartalmazhatnak azonos számjegyeket is), akkor hány olyan számnégyes választható ki, amelyek permutációi között legalább 6 prímszám van? (6 pont)

b) Az 1487, 4817 és 8147 ugyanazon számnégyes három permutációja és mindhárom prímszám. A három szám ugyanakkor egy számtani sorozatot is alkot (A differencia: 3330. Tehát 1487 + 3330 = 4817 és 4817 + 3330 = 8147.)

A négyjegyű számok összes lehetséges módon képzett négyjegyű permutációi között még egy számnégyes található, amely rendelkezik ugyanezen tulajdonságokkal. Tehát ugyanazon négy számjegy permutációi, prímek és a három szám számtani sorozatot alkot. Keresse meg ezt a 3 db négyjegyű számot (a fenti példától különbözőt)! Válaszként a három számot növekedő sorrendben egymás után írva kapott 12 jegyű számot küldje be! A fenti példa esetén ez 148748178147 lenne.

(7 pont)

4. feladat

(A Nemes Tihamér Verseny egy feladatötlete alapján)

Egy vírusos fertőzés terjedését modellezzük a feladatban. A vírus két ember találkozásával terjed és sajnos 100%-os hatékonyságú. Tehát ha két ember találkozik, akkor biztosan átadják a vírust egymásnak és vírushordozókká válnak. Egy elszigetelt populációban az emberek száma legfeljebb 300, a nevük helyett egy egyesével növekvő sorszámmal hivatkozunk rájuk, ami 1-től kezdődik. Bármelyik két ember legfeljebb egyszer találkozott egymással. A találkozások száma legfeljebb 2000. Az *elek.txt* fájl minden sorában egy-egy találkozást adtunk meg a két ember sorszámával szóközzel elválasztva (mindig a kisebb sorszám áll elől). A fájl a sor első adata, azon belül a sor második adata szerint növekvően rendezett. A fájlban a teljes populáció minden tagja szerepel legalább egyszer és minden találkozás csak egyszer jelenik meg.

Az 1. lépésben a 2 sorszámú ember volt a vírusgazda, tehát tőle indult a fertőzés. A 2. lépésben azok az emberek kapták el a vírust, akik találkoztak a 2. sorszámú emberrel. A 3. lépésben már azok is terjesztették a vírust, akik a második lépésben vírushordozóvá váltak, tehát mindazok megkapták, akik velük találkoztak. És így tovább. Azt tudjuk, hogy az 5. lépés befejezésével 183 vírushordozó volt a populációban.

a) A vírusfertőzés hatása csak korlátozott ideig, 8 lépésig tart. Pl. a 10. lépésben már nem vírushordozók azok, akik a 2. lépésben megfertőződtek (meggyógyultak, ők már nem terjesztik a vírust). Természetesen a vírus terjed tovább. Válaszként adja meg a 11. lépés után a vírushordozók számát figyelembe véve a közben meggyógyultakat is (akik már nem vírushordozók, és nem is fertőződnek újra)! (9 pont)

b) Hányadik lépés befejezésekor lesz először olyan helyzet, hogy egyetlen vírushordozó sincs a populációban? (5 pont)
(A kiinduló, teljesen vírusmentes állapotot nem tekintjük lépésnek)