

Informatika (verseny)feladatok matematikája

Szabó Zsanett

szabo.zsanett@inf.elte.hu

ELTE IK

Absztrakt. A programozás feladatok egy része matematika. A kérdés az, hogy mekkora ez a rész és milyen szintű, milyen jellegű matematikai ismeretek szükségesek egyes programozás feladatok megoldásához? Ebben a cikkben áttekintem, hogy az elmúlt évek programozás versenyein a „papíros” fordulók feladatai milyen jellegű és milyen szintű matematikai ismereteket vártak el a versenyző diákoktól. Mindezt összehasonlítottam a nemzetközi programozási versenyeken elvárt ismeretekkel, illetve a matematika órákon tanultakkal.

Kulcsszavak: programozás versenyek, versenyfeladatok, algoritmikus gondolkodás, matematika, gráfok, rekurzió, algoritmusok értelmezése

1. Bevezetés

A programozás versenyeken alapvetően a programozásé a főszerep annak rendje és módja szerint. Az első forduló azonban több verseny esetében is úgynevezett papíros fordulók, ahol a feladatokat számítógép nélkül, papíron kell megoldaniuk a versenyzőknek.

A papíros fordulók létrejöttének oka többek között az, hogy így több tanuló kap lehetőséget a versenyen való részvételre. A versenyt szervező iskolák számára könnyebbség, hogy nem szükséges technikai háttér az első fordulóhoz, a tanulóknak pedig könnyebbség, hogy nem kell attól tartaniuk, hogy egy kódolási hiba miatt nem lesz értékelhető a munkájuk.

A programozás versenyek papíros fordulóinak sajátossága, hogy a feladatok megoldásához nincs szükség konkrét programozási, kódolási tudásra. Ezek a feladatok leginkább az algoritmikus gondolkodást mérik. Megoldásukhoz azonban (ha nem is konkrétan kimondott, nevesített formában) szükség van bizonyos matematikai ismeretekre. A feladatok egy része ebből adódóan előkerülhetne akár matematika órán vagy szakkörön is.

2. Programozás versenyek

Az általános- és középiskolás tanulók többféle versenyen mérhetik össze a programozással kapcsolatos tudásukat. Ezek közül négy, országos szinten minden tanuló számára elérhető versennyel, illetve azok feladataival foglalkoztam.

Az egyik ilyen verseny a HÓDítsd meg a biteket! informatika verseny. Ez a 4-12. osztályos tanulóknak szóló verseny a nemzetközi BEBRAS-kezdeménnyezés, vagyis a Nemzetközi informatikai és számítógép-készség verseny (*International Contest on Informatics and Computer Fluency*) magyar partnere. A Dr. Valentina Dagiene litván professzor által létrehozott kezdeménnyezés célja, hogy megmutassa az informatika sokszínűségét, felhasználási lehetőségeit és területeit, hogy felkeltse a tanulók érdeklődését az informatika iránt, és feloldja az informatikával kapcsolatos félelmeket, negatív érzéseket. Mindezek mellett célja, hogy segítse az oktatásban résztvevőket minél színesebb, érdekesebb, motiválóbb informatikai gondolkodást segítő feladatokkal, lehetőségekkel [1]. Ezen az egyfordulós versenyen a tanulóknak számítógépen kell megválaszolniuk a feladatokat, amelyeknél négy válaszlehetőség közül kell kiválasztaniuk a helyeset. A feladatok nem kódolással, hanem algoritmikus és logikus gondolkodással kapcsolatosak.

Nem ritka, hogy ezen a versenyen egy-egy iskola vagy osztály minden tanulója indul, ez egy tömegversenynek mondható. Az e-HÓD verseny alkalmas arra, hogy az első pozitív élményt, motivációt megadja a diákok számára, ami után érdemes továbblépniük az érdeklődő tanulónak a többi programozással kapcsolatos verseny, például a Logo vagy a Nemes Tihamér versenyek irányába.

Egy másik verseny a Logo Országos Számítástechnikai Tanulmányi Verseny, ahol a 3-12. évfolyamos (korábban 3-10. évfolyamos) tanulók mérhetik össze tudásukat három, illetve az alsó tagozaton két fordulóban. Ezen a versenyen az első forduló két részből áll: egy számítógép nélküli és egy számítógépes feladatsorból. A számítógép nélküli feladatok egy részének megoldásához szükség van a Logo programozási nyelv ismeretére, de a feladatok többsége anélkül is megoldható. A számítógép nélküli feladatsorban a versenyzők 3-4 kisebb feladatot (algoritmus- vagy programrészletet, működési vázlatot) kapnak, és olyan kérdésekre kell válaszolniuk, mint pl. Mit rajzol? Milyen feltételek mellett működik? Mi hiányzik belőle? Mire használjuk a paramétereiket? Megoldja-e a kitűzött feladatot? [2].

A harmadik verseny, melynek feladatait vizsgáltam, az a Nemes Tihamér Nemzetközi Informatika Tanulmányi Verseny programozás kategóriája [3]. Ezen a versenyen 5-10. évfolyamos tanulók vehetnek részt. A verseny három fordulója közül az elsőben itt is számítógép nélkül kell megoldaniuk a feladatokat a versenyzőknek, illetve az 5-8. osztályosok az utolsó feladatnál választhatnak, hogy egy számítógépes vagy egy számítógép nélküli feladatot oldanak meg. A számítógép nélküli feladat általában egy algoritmus értelmezése és hibakeresés, a számítógépes feladatnál pedig kódolniuk kell. Ennél a versenyenél a számítógép nélküli feladatok egy része nem igényel különösebb programozási ismereteket, egy részükhöz viszont szükség van arra, hogy a versenyen résztvevő tanulók értelmezni tudjanak bizonyos algoritmusokat.

A negyedik programozáshoz kapcsolódó verseny az Országos Középiskolai Tanulmányi Verseny informatika tantárgyának II. kategóriája [4]. Itt a 11-12. évfolyamos tanulók mérhetik össze programozói tudásukat szintén három fordulóban. Ahogyan a Nemes Tihamér Informatika Tanulmányi Verseny programozás kategóriájában, úgy az első fordulóban itt is számítógép nélküli feladatmegoldást várnak el a tanulóktól. A feladatok összetétele ennél a versenyenél is hasonló az ott leírtakhoz. A feladatok egy része megoldható konkrét programozási tudás nélkül, algoritmikus gondolkodással, míg a feladatok másik részének megoldásához szükség van arra, hogy a versenyzők értelmezni tudjanak egy-egy algoritmust, és meg tudják mondani, hogy azok bizonyos bemenő adatokra milyen eredményt adnak.

3. Papíros fordulók

A számítógép nélküli, vagyis papíros fordulók a tanulók algoritmikus gondolkodását mérik, nem a kódolási képességüket. A tanulók egy része kimondottan örül annak, hogy bár programozás versenyen indulhatnak, mégsem kell ténylegesen programkódot írniuk. Nem kell attól tartaniuk, hogy egy hiba miatt nem fognak tudni működő programkódot írni, illetve attól, hogy nem fognak tudni értelmezni egy-egy hibaüzenetet vagy attól, hogy bár lefordul a programjuk, de az a futás során végtelen ciklusba kerül.

A tanulók (és felkészítő tanáraik) szempontjából a másik érv a papíros fordulók létjogosultsága mellett az, hogy a versenyek első fordulóra általában még a tanév első félévben sor kerül, addigra viszont az előzetes ismeretek nélkül az iskolába kerülő tanulók az alacsony óraszámok mellett nem tudnak megtanulni magabiztosan kódolni, az algoritmusok értelmezését viszont könnyebben el tudják sajátítani. A később megrendezésre kerülő második, illetve harmadik fordulóig pedig több lehetőségük van arra, hogy kódolásban is fejlődjenek.

A szervező iskolák számára könnyebbség, hogy ezekhez a számítógép nélküli, papíros fordulókhoz nem kell technikai háttérrel biztosítaniuk, így korlátozniuk sem kell a versenyen induló tanulók számát az informatikaterem befogadóképessége miatt.

A felsorolt előnyök mellett a versenyek papíros fordulói alkalmasak a tehetségek felismerésére, illetve a programozás tanulásának ösztönzésére is. Emiatt nem csak Magyarországon, hanem más országok programozás versenyein is vannak ilyen típusú feladatok.

4. A feladatok kategorizálása

A programozás versenyek kiírásaiban, illetve azok honlapjain szerepelnek rövid témafelsorolások a versenyzőktől elvárt ismeretekkel kapcsolatosan. Ezek azonban meglehetősen rövidek, szűkszavúak. Az OKTV általános követelményeit például három pontban sorolták fel, ezek közül pedig csak egy vonatkozik az első fordulóban előkerülő ismeretközökre: „rendszereszméletű feladatmegoldás, algoritmusok kidolgozása” [4]. A Nemes Tihamér és a Logo programozási versenyeknél ennél konkrétabb a versenyzőktől elvárt alapvető ismeretek listája, de itt is csak a nagyobb témakörök kerültek felsorolásra [2, 3].

A programozás versenyek számítógép nélküli versenyfeladatait vizsgálva úgy gondoltam, hogy a feladatok a nagy témakörökön belül vagy néhány esetben azoktól függetlenül egy másfajta szemlélet szerint is csoportosíthatók. Azt vettem észre, hogy időről időre hasonló feladattípusok térnek vissza a versenyfeladatok között, ezért a feladatok megoldásához kapcsolódó ismeretek, megoldási módszerek szerint csoportosítottam a Logo, a Nemes Tihamér és az OKTV versenyek számítógép nélküli feladatait.

A típusfeladatok időről-időre való visszatérése bizonyos szempontból nem meglepő. A középis-kolás korosztálynak még igénye van arra, hogy a versenyfeladatok valamilyen szinten ismerősek legyenek számukra, emiatt nem változhatnak meg gyökeresen a feladatok. Mindez pedig az első fordulók, vagyis a számítógép nélküli feladatok esetén még inkább igaz. Természetesen ez nem azt jelenti, hogy a korábbi évek feladatainak ismerete elegendő az új versenyfeladatok sikeres megoldásához. Az új feladatok némelyike – illetve az azokhoz szükséges megoldási út – ugyan több esetben hasonlíthat egy-egy korábbi feladathoz, de eléggé különbözőnek is egymástól ahhoz, hogy a korábbi feladat megoldása ne legyen egy az egyben átültethető az újra.

A feladatok kategorizálása során minden feladtnál feltüntettem a megoldásukhoz szükséges módszert, a feladattípust vagy a feladat megoldásához köthető ismeretet, fogalmat. Így a feladatok megoldásához szükséges ismeret szerint határoztam meg a kategóriákat. Külön jelöltem (a későbbi táblázatokban sötétkék háttérrel) azokat a feladatokat, amelyek megoldásánál konkrét algoritmizálási tudásra is szükség van. Ilyenek például azok a feladatok, ahol pszeudokódot kellett értelmezniük, futtatniuk a versenyzőknek, illetve az abban előforduló hibákat kellett kijavítaniuk. Megkülönböztettem azokat a feladatokat is (világoskék háttérrel kaptak a táblázatokban), amelyeknél szükség van ugyan valamennyi algoritmizálási képességre, de a feladat szövegében konkrétan leírásra vagy példán keresztül bemutatásra kerül az, hogy mit jelentenek az algoritmus egyes elemei.

A Logo Országos Számítástechnikai Tanulmányi Verseny előző négy tanévi számítógép nélküli feladatai az 1. táblázat szerinti besorolást kapták.

| 2017-2018 | | | |
|-----------|--|--|--|
| | Logo 5-6 | Logo 7-8 | Logo 9-12 |
| 1. | szabály értelmezése | szabály értelmezése | szabály értelmezése |
| 2. | algoritmus értelmezése, futtatása | algoritmus értelmezése, futtatása | algoritmus értelmezése, futtatása |
| 3. | út négyzetrácson | út négyzetrácson (45 fokkal) | algoritmus értelmezése, futtatása |
| 4. | | | út négyzetrácson (45 fokkal) |
| 2016-2017 | | | |
| | Logo 5-6 | Logo 7-8 | Logo 9-12 |
| 1. | szabály értelmezése, hibakeresés | szabály értelmezése, hibakeresés | szabály értelmezése, hibakeresés |
| 2. | algoritmus értelmezése, futtatása, hibajavítás | algoritmus értelmezése, futtatása, hibajavítás | algoritmus értelmezése, futtatása, hibajavítás |

| | | | |
|------------------|--|--|---|
| 3. | legkevesebb lépéses út keresése négyzet rácson | legkevesebb lépéses út keresése, szabály értelmezése | algoritmus értelmezése, futtatása |
| 4. | | | legkevesebb lépéses út keresése, szabály értelmezése |
| 2015-2016 | | | |
| | Logo 5-6 | Logo 7-8 | Logo 9-12 |
| 1. | algoritmus értelmezése, futtatása | algoritmus értelmezése, futtatása | algoritmus értelmezése, futtatása |
| 2. | algoritmus értelmezése, futtatása | algoritmus értelmezése, futtatása | algoritmus értelmezése, futtatása |
| 3. | navigáció négyzet rácson | navigáció négyzet rácson | navigáció négyzet rácson |
| 4. | | | algoritmus értelmezése, futtatása |
| 2014-2015 | | | |
| | Logo 5-6 | Logo 7-8 | Logo 9-10 |
| 1. | algoritmus értelmezése, futtatása | algoritmus értelmezése, futtatása | algoritmus értelmezése, futtatása |
| 2. | algoritmus értelmezése, futtatása | algoritmus értelmezése, futtatása | algoritmus értelmezése, futtatása |
| 3. | navigáció négyzet rácson | navigáció négyzet rácson | navigáció négyzet rácson, algoritmus értelmezése, futtatása |
| 4. | | | algoritmus értelmezése, futtatása |

1. táblázat: Logo Országos Számítástechnikai Tanulmányi Verseny feladatainak kategorizálása

A Logo verseny feladatai között a vizsgált tanévek mindegyikében előfordult valamilyen négyzet rácson történő navigációval kapcsolatos feladat, ami szorosan kötődik a Logo programozási nyelv sajátosságaihoz. Emellett szembevetendő, hogy az utóbbi két évben a korábbinál kevesebb olyan feladat volt mindegyik korosztályban, ahol szükség volt a Logo nyelv ismeretére vagy algoritmizálási ismeretekre.

A Nemes Tihamér Nemzetközi Informatika Tanulmányi Verseny és az Országos Középiskolai Tanulmányi Verseny programozás kategóriájának számítógép nélküli feladatait is hasonlóképp vizsgáltam. A két verseny feladatait egy táblázatban összegeztem (2. táblázat) a versenyek hasonlósága miatt, illetve amiatt, hogy bizonyos szempontból az OKTV a Nemes Tihamér verseny folytatása, feladataik többször ugyanazok vagy ugyanannak a feladatnak különböző variációi. Gyakran előfordul, hogy a különböző korosztályok feladata csak annyiban különbözik, hogy milyen mennyiségű vagy összetettségű bemeneti adat, illetve hány paraméter esetén kell megoldaniuk a feladatot a tanulóknak.

| | | | |
|------------------|-----------------------------------|-----------------------------------|-----------------------------------|
| 2018-2019 | | | |
| | Nemes 5-8 | Nemes 9-10 | OKTV |
| 1. | gráf (körmentesség) | gráf (körmentesség) | gráf (körmentesség) |
| 2. | algoritmus értelmezése | szabályból gráf (fa) | szabályból gráf (fa) |
| 3. | algoritmus értelmezése, futtatása | algoritmus értelmezése, futtatása | algoritmus értelmezése, futtatása |
| 4. | szabály értelmezése | szabály értelmezése | szabály értelmezése |
| 5. | gépes | szabály értelmezése | szabály értelmezése |
| 6. | hibakeresés algoritmusban | szabály értelmezése | szabály értelmezése |
| 7. | | algoritmus értelmezése | algoritmus értelmezése |
| 2017-2018 | | | |
| | Nemes 5-8 | Nemes 9-10 | OKTV |
| 1. | rekurzió, szabály alkalmazása | rekurzió, szabály alkalmazása | gráf (elvágó pontok) |
| 2. | rekurzió, utak száma | algoritmus futtatása, gráf | vizualizáció alapján algoritmus |
| 3. | algoritmus értelmezése, futtatása | gráf (elvágó élek) | algoritmus értelmezése, futtatása |
| 4. | titkosítás, kód helyreállítása | titkosítás, kód helyreállítása | hibakeresés algoritmusban |
| 5. | gépes | algoritmus értelmezése, futtatása | rekurzió, szabály alkalmazása |
| 6. | hibakeresés algoritmusban | rekurzió, szabály alkalmazása | titkosítás, kód helyreállítása |
| 7. | | algoritmus, ciklusból rekurzió | algoritmus, ciklusból rekurzió |

| 2016-2017 | | | |
|-----------|--------------------------------------|--|--------------------------------------|
| | Nemes 5-8 | Nemes 9-10 | OKTV |
| 1. | szabályból gráf (legrövidebb kör) | mohó stratégia | mohó stratégia |
| 2. | rekurzió, utak száma | rekurzió | rendezés |
| 3. | úthálózat | gráf (topologikus rendezés) | gráf (topologikus rendezés) |
| 4. | algoritmus értelmezése, futtatása | titkosítás | titkosítás |
| 5. | gépes | algoritmus értelmezése, futtatása | vizualizáció alapján algoritmus |
| 6. | hibakeresés algoritmusban | algoritmus értelmezése, futtatása | algoritmus értelmezése, futtatása |
| 7. | | szabály értelmezése | szabály értelmezése |
| 2015-2016 | | | |
| | Nemes 5-8 | Nemes 9-10 | OKTV |
| 1. | algoritmus értelmezése, futtatása | algoritmus értelmezése, futtatása, hibajavítás | hibakeresés algoritmusban |
| 2. | szabály értelmezése (véges automata) | algoritmus értelmezése, futtatása | algoritmus értelmezése, futtatása |
| 3. | szabály értelmezése (optimalizálás) | szabály értelmezése (véges automata) | szabály értelmezése (véges automata) |
| 4. | gépes | szabály értelmezése (optimalizálás) | szabály értelmezése (optimalizálás) |
| 5. | algoritmus kiegészítése | algoritmus értelmezése, futtatása | algoritmus értelmezése, futtatása |

2. táblázat: Nemes Tihamér Nemzetközi Informatika Tanulmányi Verseny és Országos Középiskolai Tanulmányi Verseny (programozás kategória) számítógép nélküli feladatainak kategorizálása

A feladatok között itt is vannak olyanok, amelyeknél fontos, hogy a versenyzők értsék a legtöbbször pszeudokódként megadott algoritmusokat, de a feladatok jelentős része, körülbelül fele ilyen jellegű ismeretek nélkül is megoldható. Az algoritmusok megértését igénylő feladatok száma közel állandónak mondható az előző négy tanév feladatsorai alapján. A feladatok száma viszont nőtt, így az algoritmusok megértését igénylő feladatok aránya összességében csökkent a Nemes Tihamér programozási verseny és az OKTV első fordulójának feladatsoraiban.

Az általános- és középiskolás korosztálynak szóló, már említett négy programozási versenyen a versenyzőktől elvárt ismeretek körét érdemes összehasonlítani a Nemzetközi Informatikai Diákolimpián (IOI) elvárt ismeretekkel, mivel a Nemes Tihamér verseny és az OKTV is az IOI válogatóversenyei közé tartozik. Az IOI-n elvárt ismereteket részletes Syllabus-ban foglalták össze [5]. Ebből kiderül, hogy bizonyos aritmetikai és geometriai ismeretek mellett fontos szerepet kapnak a gráfelmélet témakör egyes ismeretei, algoritmusai is a versenyfeladatokban. Így nem meglepő, hogy ezek a témakörök a Nemes Tihamér versenyen és az OKTV-n is előfordulnak.

5. Legjellemzőbb feladattípusok

A négy programozási verseny számítógép nélküli feladatai két nagy kategóriába sorolhatók. Az egyik kategóriához az algoritmusok értelmezését igénylő feladatok tartoznak, a másikhoz pedig a matematikai jellegű feladatok.

5.1. Algoritmus értelmezését igénylő feladatok

Az algoritmusok értelmezését igénylő feladatoknál három gyakori feladattípus van. Eszerint további három alkategóriát határoztam meg az algoritmusok értelmezését igénylő feladatok esetén:

- Mít csinál az algoritmus?
- hibakeresés
- algoritmus átalakítása

5.1.1. Mit csinál az algoritmus?

Ehhez az alkategóriához soroltam azokat a feladatokat, amelyeknél a versenyzőknek meg kell határozniuk azt, hogy mit csinál az adott algoritmus, illetve milyen eredményt kapunk, ha a megadott kiinduló adatok esetén végrehajtjuk az algoritmus lépéseit.

Ilyen feladat például a következő (Nemes, 2017-18, 5-8. osztály):

3. feladat: Mit csinál (30 pont)

Az alábbi algoritmus egy N elemű X vektort dolgoz fel, eredményét az N elemű Y vektorba írja.

Valami (N, X, Y) :

```
D[1..N] := 0
Ciklus i=1-től N-1-ig
  Ciklus j=i+1-től N-ig
    Ha  $X[i] > X[j]$  akkor  $D[i] := D[i] + 1$ 
      különben  $D[j] := D[j] + 1$ 
  Ciklus vége
Ciklus vége      {*}
Ciklus i=1-től N-ig
   $Y[D[i] + 1] := X[i]$ 
Ciklus vége      {**}
```

Eljárás vége.

A. Mi lesz a D vektorban $\{*\}$ -nál és az Y vektorban $\{**\}$ -nál, ha $N=6$, $X=(3,8,1,2,9,7)$?

B. Mi lesz a D vektorban $\{*\}$ -nál és az Y vektorban $\{**\}$ -nál, ha $N=9$, $X=(3,9,3,1,2,1,1,8,7)$?

C. Fogalmazd meg általánosan, mi lesz a D vektorban $\{*\}$ -nál és az Y vektorban $\{**\}$ -nál tetszőleges N és X esetén!

Egy másik példa ilyen típusú feladatra (Logo, 2017-18, 5-6. osztály):

2. feladat: Teknős Tádé (16 pont)

Teknős Tádé hatszögeket próbált rajzolni. Sikerült négyféle változatot készítenie és az elkészült rajzokat képként is kimentette, de elfelejtette, hogy melyik rajzot melyik programmal készítette. Segíts neki párosítani a rajzokat a programokkal!

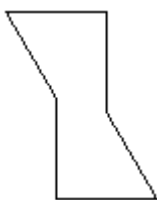
Melyik programhoz melyik ábra tartozik?

1. ismétlés 3 [előre 50 jobbra 30 előre 50 jobbra 90]

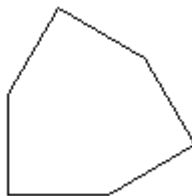
2. ismétlés 3 [előre 50 balra 30 előre 50 jobbra 150]

3. ismétlés 2 [előre 50 jobbra 30 előre 50 jobbra 60 előre 50 jobbra 90]

4. ismétlés 2 [előre 50 balra 30 előre 50 jobbra 120 előre 50 jobbra 90]



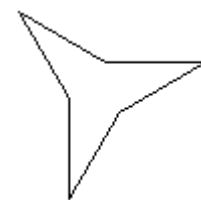
A



B



C



D

5.1.2. Hibakeresés

A másik, algoritmus értelmezését igénylő feladattípusnál a megadott algoritmusban kell megkeresniük a hibát a versenyzőknek. Ilyen feladat például a következő (OKTV, 2017-18, 11-12. osztály):

4. feladat: Összefésüléses rendezés (28 pont)

Az összefésüléses rendezés algoritmusa a következő elven működik:

- az egyelemű sorozat rendezett, nincs vele tennivaló;
- ha a sorozat több elemű, akkor
 - o középen két részre osztjuk;
 - o mindkét részt rendezzük az összefésüléses rendezés algoritmusával;
 - o végül a két kapott rendezett sorozatot összefésüljük.

Az alábbi algoritmus ezt csinálná (az X tömb. E. és U. eleme közötti részt rendezné), azonban sajnos hibák kerültek bele. Jelöld, melyek a hibák!

Rendez (E, U) :

```
Ha E<U akkor K:=(E+U)/2
                Rendez (E, K); Rendez (K, U)
                Összefésül (E, U, K)
```

Elágazás vége

Eljárás vége.

Összefésül (E, K, U) :

```
i:=E; j:=K+1; D:=E; Y:=X
Ciklus amíg i≤K és j≤U
    D:=D+1
    Ha Y[i]<Y[j] akkor X[D]:=Y[j]; i:=i+1
        különben X[D]:=Y[j]; j:=j-1
```

Ciklus vége

Ciklus amíg i≤U

```
D:=D+1; X[D]:=Y[i]; i:=i+1
```

Ciklus vége

Ciklus amíg j<U

```
D:=D+1; X[D]:=Y[j]; j:=j+1
```

Ciklus vége

Eljárás vége.

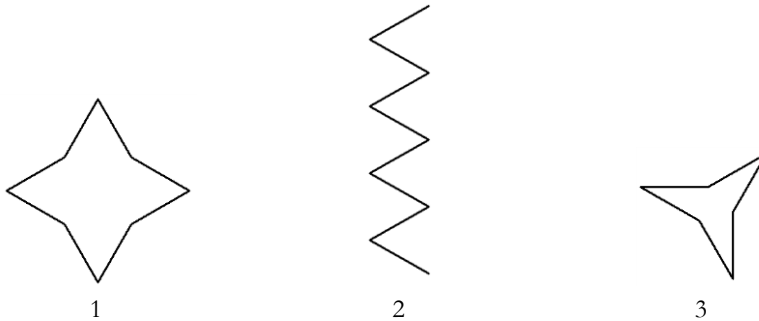
Ilyen hibakereséses, hibajavító feladatra a Logo verseny feladatai között is van példa (Logo, 2016-17, 5-6. osztály):

2. feladat: Teknős Tádé (15 pont)

Teknős Tádé szeret Logo programokat írni, viszont nem mindig tudja megkülönböztetni a bal és a jobb kezét. Ezért a Logo programjaiban időnként a BALRA helyett JOBBRA, a JOBBRA helyett pedig BALRA utasítást ír (de nem mindig).

1. ismétlés 4 [balra 60 előre 100 jobbra 120 előre 100 balra 30]
2. ismétlés 4 [jobbra 60 előre 100 jobbra 120 előre 100 balra 60]
3. ismétlés 3 [jobbra 60 előre 100 balra 150 előre 100 jobbra 30]

Ezekkel ezt a három ábrát szeretne volna rajzolni:



A. Mit rajzol Tádé három programja?

B. Hogyan kell kijavítani, hogy azt rajzolja, amit várt!

5.1.3. Algoritmus átalakítása

A harmadik, algoritmus értelmezését igénylő gyakori feladattípusnál a tanulóknak meglévő algoritmust kell átalakítaniuk bizonyos feltételek szerint.

Ilyen feladat volt például a következő (*Nemes, 2017-18, 9-10. osztály*), ahol a versenyzőknek meglévő algoritmusokat kellett rekurzívvá alakítaniuk. A feladat nem várja el a rekurzió előzetes ismeretét a tanulóktól. A feladat szövegében leírásra kerül, hogy mit jelent az, ha egy függvény rekurzív. Emellett pedig egy példán keresztül bemutatásra kerül az is, hogy hogyan lehet átírni egy ciklussal megírt algoritmust rekurzívvá. Ennek ellenére természetesen igaz az is, hogy a rekurzió ismerete, illetve az, ha a versenyző látott korábban ilyen jellegű feladatot, az nagyban hozzájárul a feladat könnyebb megértéséhez, illetve megoldásához.

7. feladat: A Rek bolygó programozói (26 pont)

Nemrég rendezték a programozók intergalaktikus találkozóját, ahol a földi programozók találkoztak a Rek bolygóról érkező kollégáikkal. A földiek meg akarták mutatni a kedvenc programjaikat, de kiderült, hogy a Rek bolygón nem ismerik a ciklusokat és az értékadást. Nincs is rá szükségük, mert mindent rekurzióval valósítanak meg (vagyis olyan függvényeket írnak, amik meghívják saját magukat). A függvények értéke náluk egyszerűen az utoljára kiszámolt kifejezés értéke lesz.

Egy programot már sikerült átírnia a tolmácsnak úgy, hogy az idegenek is értsék, de a többiben néhány helyen bizonytalan. Írd be a hiányzó kifejezéseket! Szerencsére már csak a függvény paramétereit, számokat és műveleteket kell használnod. A programok paramétereit mind pozitív egészek lehetnek.

| földi program | Rek program |
|---|---|
| <pre>faktoriális(N) : a := 1 Ciklus i=1-től N-ig a := a * i Ciklus vége faktoriális := a Függvény vége.</pre> | <pre>faktoriális(N) : ha N = 1 1 különben N * faktoriális(N-1) Függvény vége.</pre> |

Írd át az alábbi algoritmusokat a Rek bolygó nyelvére az alábbi kódrészek kiegészítésével!

| | |
|--|---|
| <pre> valami (F) : a := 1 b := 1 Ciklus i=3-tól F-ig a := a + b b := a - b Ciklus vége valami := a Függvény vége. </pre> | <pre> valami (F) : ha [] < 3 [] különben valami ([])+valami ([]) Függvény vége. </pre> |
| <pre> valami (A,B,C) : s := 0 Ciklus i=A-tól B-ig ha i mod C = 0 s := s + 1 Ciklus vége valami := s Függvény vége. </pre> | <pre> valami (A,B,C) : ha [] > [] [] különben ha A mod C = 0 []+valami ([], [], []) különben valami ([], [], []) Függvény vége. </pre> |
| <pre> valami (A) : c := 0 b := 1 Ciklus amíg b*b ≤ A c := c + b * b b := b + 1 Ciklus vége valami := c Függvény vége. </pre> | <pre> valami (A) : valami2 ([], 1, 0) Függvény vége. valami2 (A,B,C) : ha [] > [] [] különben valami2 ([], [], []) Függvény vége. </pre> |

5.2. Matematikai jellegű feladatok

Az algoritmusok értelmezését igénylő feladatok mellett a vizsgált versenyek számítógép nélküli feladatainak másik nagy csoportját képezik azok a feladatok, amelyek megoldásához bizonyos szintű matematikai ismeretek szükségesek. Ezek a feladatok jellegükből adódóan matematika órákon vagy szakkörön is előfordulhatnak. A matematikai jellegű kifejezés nem feltétlenül takar konkrét matematikai ismereteket. Azokat a feladatokat is ide soroltam, ahol a megoldáshoz logikus gondolkodásra, szabályalkalmazásra van szükség. Az ide csoportosított feladatok között vannak olyanok is, amelyek megoldhatóak matematikai ismeretek nélkül is, viszont a matematikai háttérismeret nagyban meg tudja könnyíteni, le tudja egyszerűsíteni a megoldás menetét.

5.2.1. Megadott szabály alkalmazása

A programozási versenyek számítógép nélküli feladatainál a matematikai jellegű feladatok első csoportját azok a feladatok alkotják, amelyekben egy-egy adott szabályt kell alkalmazni a megadott adatokra. Az ilyen feladatok a HÓDítsd meg a biteket! informatika verseny feladatai között is gyakoriak. Egy ezek közül például a következő (*e-bód, 2015., benjamin – közepes, kadét – könnyű, 2013-JP-02-B, Varázslatos alagút*):

A hódvasút kétféle alagutat használ:

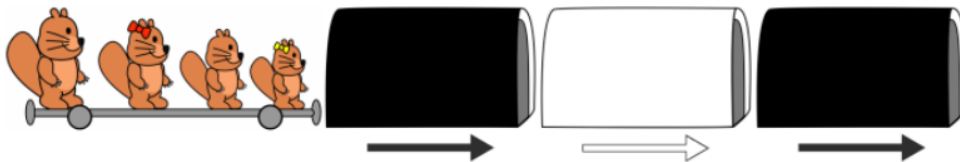
Ha a szerelvény egy fekete alagúton megy keresztül, az utasok fordított sorrendben jönnek ki a túloldalra.



Ha a szerelvény egy fehér alagúton megy keresztül, az első és a hátsó utas helyet cserél.



A mi szerelvényünk három alagúton megy keresztül:



Milyen sorrendben jönnek ki az utasok az utolsó alagútból?

| | | | |
|---|--|---|--|
| A | | C | |
| B | | D | |

Egy meghatározott szabály alkalmazását várta el a versenyzőktől ez a feladat is (Logo, 2017-18, 5-6. osztály):

1. feladat: 10x4-es kijelző (20 pont)

Van egy négyzetekből összeállított, 10 oszlopból és 4 sorból álló színes kijelzőnk. Ennek minden négyzetét különböző színűekre festhetjük egy speciális teknőc segítségével. A teknőcnek oszloponként egy listában kell megadnunk, hogy mely mezőket fesse ki és milyen színnel. A következő színek kódokat használhatjuk:

P: piros

Z: zöld

F: fekete

A szabály az, hogy a listában egy szín nem szerepelhet kétszer. A színek előtt egy számnak kell állnia, amelyet úgy kapunk, hogy az adott sorok előtt szereplő számokat összeadjuk. A fehér négyzeteket figyelmen kívül kell hagynunk.

Nézzük az alábbi ábra részletet.

Itt az első oszlopban a piros csak a 8-as sorszámú sorban szerepel, ezért ezt a számot leírjuk, majd mögé írjuk a színkódot. Az eredmény: 8P. A zöld mező előtti sorokban a 2 és 4 szerepel, vagyis ezek összegét (6), és a Z színkódot írjuk. Az eredmény: 6Z. A fehér négyzetet figyelmen kívül hagyjuk, vagyis az oszlopok kódjai:

oszlop kódja: 8P 6Z

oszlop kódja: 9F 2Z 4P

A színeket tetszőleges sorrendben is leírhattuk volna, vagyis az 1. oszlopnál a 6Z 8P is teljesen jó megoldás.

A fenti szabályok alapján határozd meg az oszlopok kódjait az alábbi ábrára, amely egy kigyós játék egy részletét ábrázolja:

| | | |
|---|---|---|
| | 1 | 2 |
| 1 | | F |
| 2 | Z | Z |
| 4 | Z | P |
| 8 | P | F |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|----|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 1 | P | P | | F | F | F | | | P | P |
| 2 | F | P | P | | | F | | P | P | |
| 4 | | | P | P | P | | | P | F | |
| 8 | F | F | | | P | P | P | P | F | F |

A feladatok között előfordulnak olyanok is, ahol nem a szabály alapján kell meghatározni azt, hogy mi lesz az eredmény, hanem a kezdeti érték és az eredmény alapján magát a szabályt kell kitalálniuk a tanulóknak. Ilyen jellegű feladat volt például a következő (*Nemes, 2016-17, 9-10. osztály*), ahol még annyival nehezítették a titkosítás szabályának kitalálását, hogy a számok és a kódolt változatuk nem ugyanabban a sorrendben voltak megadva, így a versenyzőknek több lehetőséget kellett vizsgálniuk.

4. feladat: Titkosítás (40 pont)

Egy titkosítási eljárás úgy működik, hogy a kapott számok kettes számrendszerbeli bitjeit összekeverik. A keverést egy vektor írja le, amely i . elemében megadjuk, hogy a titkos kód i . bitjét a szám hányadik bitjéből kell venni (balról 0-tól sorszámozva a biteket). Például, ha a keverést a (2,1,3,0) vektor írja le, akkor a 0000, 1100, 0111 számok titkos kódjai a 0000, 0101, 1110 számok lesznek.

Add meg a keverést leíró vektort, ha a számok és a kódjuk az alábbi (a számok és a kódok sorrendje nem feltétlenül azonos):

A. Számok: 0001, 0011, 0100. Kódjuk: 0010, 1010, 0100.

B. Számok: 0001, 0011, 0111. Kódjuk: 0010, 1010, 1110.

C. Számok: 0001, 0101, 1011, 0011. Kódjuk: 1110, 1001, 1000, 1100.

D. Számok: 0101, 0111, 1011. Kódjuk: 1010, 1011, 0111.

Az előző két feladat nem csak a szabályok alkalmazásáról szól, hanem akár egy titkosítás, titkosítás felé vezető feladatsor feladatai is lehetnének. A titkosítás és a titkosítás éppúgy informatikai probléma, ahogyan matematikai is. A versenyfeladatok között talán épp emiatt szerepeltek más olyan feladatok is, amelyek valamilyen módon ehhez a témakörhöz kapcsolódtak. Ilyen feladat volt például a következő, ahol a titkosítás hibalehetőségeivel, a kódolás és dekódolás hibáinak javításával kellett foglalkozniuk a tanulóknak (*Nemes, 2017-18, 5-8. osztály*).

4. feladat: Fényjelek (44 pont)

Ádám és Éva sötétedés után fényjelekkel kommunikálnak. Évának van egy zseblámpája, amivel piros, zöld és kék fényeket tud kiadni. Minden üzenetet azonos hosszú jelsorozattal kódolnak. Például előző héten a (piros, zöld, zöld) jelentette az igen szót, a (piros, zöld, piros) pedig a nem szót. Sajnos azonban azt tapasztalták, hogy a távolból néha nagyon nehéz megkülönböztetni a látott színt, így hibák fordulnak elő. Úgy döntöttek ezért, inkább hosszabb jelsorozatokat használnak, hogy néhány hiba esetén még a vevő ki tudja javítani a félrenézett színeket, ezzel az eredeti üzenetet helyreállítva. Ezt úgy teszik, hogy ha egy a kódtáblában nem található jelsorozatot észlelnek, akkor megkeresik azt a jelsorozatot a kódtáblában, ami a legkevesebb helyen különbözik tőle, és erre javítják.

A. A következő négy jelből álló jelsorozatokat használja Ádám és Éva: [P,Z,K,Z], [Z,K,Z,K], [K,P,P,Z], [Z,P,K,P]. Ádám a következő jelsorozatokat jegyzi fel: [K,Z,K,Z], [Z,P,K,Z], [Z,P,P,P], [P,Z,Z,Z], [K,P,P,K], [Z,K,Z,K]

A1. Helyreállítás segítségével határozd meg mi lehetett Éva üzenete!

A2. Legfeljebb hány színt nézhet félre Ádám jelsorozatanként, hogy biztosan helyes legyen a helyreállítás után az üzenet?

B. A következő öt jelből álló jelsorozatokat használja Ádám és Éva: [Z,K,P,K,K], [K,Z,Z,Z,P]. Ádám a következő jelsorozatokat jegyzi fel: [P,K,P,K,Z], [Z,K,P,K,K], [Z,Z,Z,K,P], [Z,Z,P,Z,K], [Z,K,Z,Z,P], [K,Z,P,P,P]

B1. Helyreállítás segítségével határozd meg mi lehetett Éva üzenete!

B2. Legfeljebb hány színt nézhet félre Ádám jelsorozatanként, hogy biztosan helyes legyen a helyreállítás után az üzenet?

B3. Ádám és Éva szeretne egy harmadik üzenetet felvenni a kódtáblájukba. Segíts nekik találni az új üzenethez egy új jelsorozatot úgy, hogy Ádám 2 hibát még biztosan javítani tudjon!

B4. Az előző feladatban talált jelsorozatot is a kódtáblához adva Ádám a következő jelsorozatokat észleli: [Z,Z,Z,K,K], [Z,P,P,K,Z], [P,K,Z,Z,P], [P,P,K,Z,K], [K,P,K,P,P], [K,Z,Z,Z,P]. Mi lehetett az eredeti üzenet?

5.2.2. Rekurzióval kapcsolatos feladatok

A matematikai jellegű feladatok második nagyobb kategóriáját a rekurzióval kapcsolatos versenyfeladatok alkotják. Már az algoritmusok értelmezését igénylő feladatok között is volt olyan, ahol előkerült a rekurzió, de a pszeudokód értelmezése nélkül megoldható, matematikai jellegű feladatok között is találunk olyanokat, amelyek rekurzióval oldhatók meg.

Ilyen feladat volt például a következő (*Nemes, 2017-18, 9-10. osztály*), ahol egy-egy fa kidöntése újabb és újabb fakidöntéseket eredményezett.

1. feladat: Favágás (20 pont)

Egy faszorba N fát ültettek balról jobbra, egy vonalba. Mindegyik fának ismerjük a bal szélső fáról vett távolságát és a magasságát. Ha egy fát kivágunk, akkor az a jobboldali szomszédja felé dől, s amelyik szomszédjára rádől, az is kidől. Az 1 távolságra levő fára az 1 magasságú fa nem dől még rá, a 2 magasságú viszont igen.

Az alábbi fák esetén add meg, hogy mely fákat kell kivágni, hogy az összes fa kidőljön! A számpárok első tagja a fa távolsága a bal szélső fától, a második pedig a magassága.

A. (0,6), (3,1), (5,2), (8,1), (15,10)

B. (0,3), (2,3), (4,1), (6,3), (8,2), (10,1)

C. (0,4), (3,4), (4,3), (6,1), (7,5), (9,2), (11,4), (12,2), (14,1), (15,1), (16,5)

Egy másik rekurzióhoz kötődő feladat (*Nemes, 2017-18, 5-8. osztály*) volt az, ahol egy különleges növényfaj egyedeinek száma attól függ, hogy a korábbi évben hány faj volt, és azok hány évesek voltak.

1. feladat: Növény (36 pont)

Egy különleges növényfajt fedeztek fel az egyenlítői dzsungelben. A növény 5 évig él, élete első három évében egy-egy magjából újabb növény kel ki (azaz pl. az első évben ültetett növény a 2., 3. és 4. évben hoz magot, amit újra elültetünk, az ötödik évben még él, a hatodikban pedig elpusztul). Beszereztünk egy egyéves növényt és elültettük egy arborétum üvegházába.

Töltsd ki az alábbi táblázatot, amiből kiderül, hogy a következő 10 évben hány új növény fog kikelni és melyik évben hány növényt láthatnak az üvegház látogatói!

| Év | 1. | 2. | 3. | 4. | 5. | 6. | 7. | 8. | 9. | 10. |
|---------------|----|----|----|----|----|----|----|----|----|-----|
| Új növény | 1 | | | | | | | | | |
| Összes növény | 1 | | | | | | | | | |

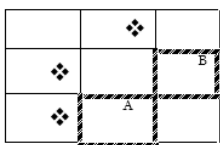
5.2.3. Kombinatorikai jellegű, útkeresős feladatok

Egy másik feladatcsoportot alkotnak a matematikához kapcsolódó, kombinatorikai jellegű, lehetséges utak számával kapcsolatos versenyfeladatok. Ezeknél a feladatoknál egyértelműen jobb helyzetbe kerülnek azok a tanulók, akik matematikai tanulmányaik során már találkoztak ilyen jellegű feladatokkal. A lehetséges utak számát a legtöbb esetben jól meg lehet határozni logikai úton is, de a matematikai tanulmányaikból ismert séma alkalmazásával egyszerűbben juthatnak el a versenyzők a jó megoldáshoz, ráadásul kisebb az esélye annak, hogy kihagynak bizonyos eseteket.

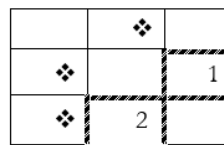
Ilyen feladat volt például a következő (*Nemes, 2017-18, 5-8. osztály*):

2. feladat: Kincsek (40 pont)

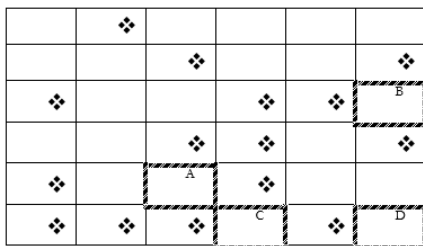
Egy jobbra-lefelé lejtő hegyoldalon kincseket helyeztünk el, amelyek egy részét egyetlen szánkóval szeretnénk összegyűjteni. A szánkóval a bal felső sarokból indulhatunk, és lejtő irányba (azaz vagy jobbra, vagy lefelé) haladhatunk. Amelyik mezőn átmegyünk, az ott levő kincset felvesszük. A hegyoldalon kijelöltek néhány gyűjtőpontot, a szánkóval valamelyikhez el kell jutnunk (és onnan tovább nem mehetünk).



A baloldali ábrán három kincset és két gyűjtő helyet látunk. A jobboldali ábrán látható, hogy melyik helyre maximum hány kincset vihetünk, ha a bal felső sarokból indulunk. Egy lehetséges út az alsó gyűjtő helyhez: le, le, jobbra; a jobboldali gyűjtő helyhez: le, jobbra, jobbra.



Add meg, hogy az alábbi ábrán szereplő kincsek és gyűjtőhelyek esetén melyikbe maximum hány kincs vihető, továbbá mindegyik gyűjtőhelyhez adj is meg egy ilyen utat!


























A Logo versenyen a vizsgált évek mindegyikében szerepelt olyan feladat a számítógép nélküli feladatok között, ahol egy robottal kellett bejárni egy megadott útvonalat vagy összegyűjteni valamit. Ezek a feladatok a Logo programozási nyelvhez állnak közel, de emellett a matematikához is kapcsolódnak, mert több esetben a lehető legkevesebb lépésből álló vagy valamilyen más szempontból legjobbnak számító útvonalat kell megkeresniük a versenyzőknek. Az ilyen, négyzet rácson való közlekedéssel kapcsolatos feladatok egyike például a következő (Logo, 2016-17, 5-6. osztály):

3. feladat: Piac (20 pont)

Egy piacon a bevásárlás megkönnyítéséhez hordár robotokat használnak, akik az utat is megmutatják egy-egy zöldséghez, gyümölcshöz. A robot az alábbi térkép szerint tud egy vagy több mezőnyit előre (E) vagy hátrafelé (H) mozogni, illetve szükség szerint tud 90 fokot jobbra (J) vagy balra (B) fordulni. A robot alapállapotban mindig a bejárati mezőn áll és az azon szereplő nyíl által meghatározott irányba néz.

A piacon csak a kijelölt útvonalakon szabad közlekedni (a vastag körvonalú mezőkre nem lehet lépni). Egy-egy zöldséget, gyümölcsöt egy vele szomszédos mezőn állva és a zöldséget, gyümölcsöt tartalmazó mező felé fordulva lehet megvásárolni. A zöldségeket, gyümölcsöket tartalmazó mezőkön keresztül menni nem lehet (hiszen az azokat tároló ládákon átmászni tilos).

| | | | | | | | | |
|---|---|---|--------------|--|--|---|--|--|
| | | | | | | | | szilva  |
| | banán  | citrom  | | reték  | répa  | kukorica  | | meggy  |
| | narancs  | ananasz  | | hagyma  | paradicsom  | | alma  | barack  |
| | | | | | | | | |
| szilva  | dinnye  | meggy  | | répa  | reték  | | szilva  | |
| barack  | | | ↑ BEMÁRAT | | saláta  | | körte  | alma  |

Add meg a következő feladatokhoz tartozó robotprogramot, amivel a robot a lehető legkevesebb utasítással megoldja a feladatot!

- Retket szeretnék vásárolni!
- Almát szeretnék vásárolni, de a hagyma illatát nem szeretem, nem szeretnék elmenni mellette (de a sarkánál el lehet menni)!
- Barackot, almát és meggyet szeretnék vásárolni valamilyen sorrendben!
- Az összes szilvát meg szeretném venni!

5.2.4. Mohó stratégia, optimalizálás

A vizsgált versenyek feladatai között előfordultak olyanok, amelyek mohó stratégiával kapcsolatosak, illetve olyanok is, amelyek optimalizálási problémát rejtenek magukban. Ahogyan az utak számának meghatározását elváró feladatoknál, úgy ezeknél a feladatoknál is jelentős előnyhöz jutnak azok a versenyzők, akik korábbi (matematikai vagy informatikai) tanulmányaik során találkoztak már ilyen jellegű problémákkal, és felismerik, hogy a versenyfeladatban is ugyanazt a megoldási stratégiát kell alkalmazniuk.

Az elmúlt néhány év feladatai között mohó stratégiával kapcsolatos feladat volt például a következő (*Nemes, 2016-17, 9-10. osztály*) a Nemes Tihámér versenyen a 9-10. osztályosok feladatsorában, illetve ugyanezen a feladatnak egy kicsit módosított változata a 11-12. évfolyamosoknál is előkerült ugyanebben az évben az OKTV számítógép nélküli feladatai között.

1. feladat: Fesztiválok (36 pont)

Magyarországon sok fesztivált rendeznek. Ismerjük mindegyik első és utolsó napja éven belüli sorszámát. A fesztiválok minden nap reggeltől éjfélig tartanak, s ha egy fesztivált meglátogatunk, akkor az elejétől a végéig ott kell lennünk.

Add meg, hogy az alábbi fesztiválok közül maximum hányat tudunk meglátogatni és adj is meg egy látogatási tervet (megtől meddig milyen sorszámú fesztiválon leszünk)! (Több megoldás esetén bármelyik megoldható.)

A. 6 fesztivál, 1: 2-3, 2: 2-4, 3: 5-7, 4: 3-4, 5: 2-2, 6: 1-2

B. 6 fesztivál, 1: 1-100, 2: 95-105, 3: 101-120, 4: 121-131, 5: 132-200, 6: 131-132

C. 8 fesztivál, 1: 1-10, 2: 2-6, 3: 3-7, 4: 3-3, 5: 13-13, 6: 12-13, 7: 10-11, 8: 7-9

Optimalizáláshoz kapcsolódó feladat már az 5-8. osztályosok feladatsorában is előfordul. Az ilyen feladatoknál a feladatok szövegében általában szerepel az optimális szó vagy az, hogy a legjobb vagy legkedvezőbb esetet kell megtalálni.

Ezek a feladatok könnyen érthetőek, megoldásuk azonban meglehetősen nehéz is lehet, ezért jól mérik a versenyző tanulók gondolkodási módszereit. A verseny jellegéből adódóan igaz, hogy a rövid megoldást igénylő feladatok esetében legtöbbször nem derül ki az, hogy pontosan milyen gondolatsor mentén jutott el a tanuló az általa helyesnek vélt megoldáshoz, hiszen ehhez a versenyzőknek a teljes gondolatmenetüket részletesen rögzíteniük kellene. Az első fordulót javító pedagógusnak viszont jó visszajelzést adhatnak az ilyen jellegű feladatok a tanulók gondolkodásával kapcsolatosan, mert a tanulók megoldásaikban azonban az esetek többségében egy-két gondolati elem felismerhető, így következtetni lehet a feladatmegoldás során használt módszerre is.

Optimalizálási problémával kapcsolatos feladat például a következő (*Nemes, 2015-16, 5-8. osztály*), ahol egy kétszemélyes játéknál kellett optimális stratégia alapján gondolkodniuk a versenyzőknek. Ennek a feladatnak a módosított, nehezített változatai az idősebb korosztályok feladatsoraiban is előkerültek.

3. feladat: Játék (40 pont)

Jancsi és Juliska a következő kétszemélyes játékot játsszák. Leraknak egymás mellé $2 \times K$ kupacban gyöngyöket. Felváltva lépnek, előbb mindig Jancsi, utána Juliska. A következő lépő elveheti a valamelyik szélső kupacban levő összes gyöngyöt. Maximum mennyi gyöngy lehet az alábbi esetekben Jancsié, ha feltételezzük, hogy Juliska is optimálisan játszik, azaz a lehető legtöbb gyöngyöt akarja megszerezni? Írd le a játék menetét is!

A. 20 10

B. 1 2 8 4

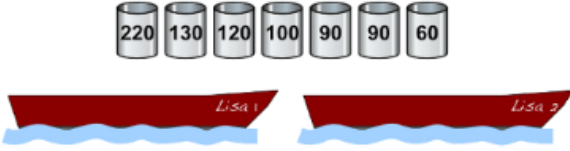
C. 1 7 8 4

- D. 1 2 1 2 1 2 1 2 1 2
 E. 1 2 1 2 1 2 2 1 2 1
 F. 1 5 3 8 4 7
 G. 5 1 3 4 8 7

Az ilyen feladatok könnyű érthetőségéből és gondolkodtató jellegéből adódóan a HÓDítsd meg a biteket! verseny feladatai között is gyakoriak azok, amelyeknél optimalizálni kell valamilyen szempont szerint. Ilyen feladatra példa a következő (*e-bód, 2015., kadét – nebéz, junior – közepes, 2014-DE-08, Lisák feltöltése*):

Bertalané és Barnabásé, a két halászá a „Lisa1” és „Lisa2” hajó - a két Lisa.

Bertalanak és Barnabásnak a két hajóval el kell szállítania pár hordó halat. A szállítást súly alapján fizetik.



Mindkét hajó legfeljebb 300kg-ig terhelhető.

Maximum mennyi halat tudnak egyszerre a két hajóval elszállítani?

| | |
|---|--------|
| A | 810 kg |
| B | 600 kg |
| C | 590 kg |
| D | 530 kg |

5.2.5. Gráfokkal kapcsolatos feladatok

A programozás versenyek matematikához is kapcsolódó számítógép nélküli feladatainak többsége a gráfok témaköréből kerül ki. A feladatok között előfordulnak olyanok, amelyek útkereséssel (legrövidebb, leghatékonyabb), elvágó élekkel vagy elvágó pontokkal, összefüggőséggel, körmentességgel vagy topologikus rendezéssel kapcsolatosak.

Fontos megemlíteni, hogy annak ellenére, hogy a programozás versenyek papíros versenyfeladatai között gyakran fordulnak elő gráfokkal foglalkozó feladatok, a tanulók a matematika órákon csak meglehetősen későn, 11. évfolyamon foglalkoznak ezzel a témakörrel, ráadásul akkor is csak röviden. Már az Euler-vonal, Euler-kör és a fagráfok is csak kiegészítő anyagként jelennek meg az egyik széles körben használt 11. osztályos tankönyvben [6]. A 11. osztályban elsajátítandó ismeretekon kívül az egyik új kiadású, 9. évfolyamosoknak szóló matematika tankönyvben [7] egyetlen anyagréssz erejéig ugyan előkerül a gráfok témaköre, de az ott előforduló ismeretanyag még kevés ahhoz, hogy a tanulók a gráfokhoz kapcsolódó versenyfeladatok megoldása során a matematika órán tanult gráfelméleti ismereteikre hagyatkozzanak.

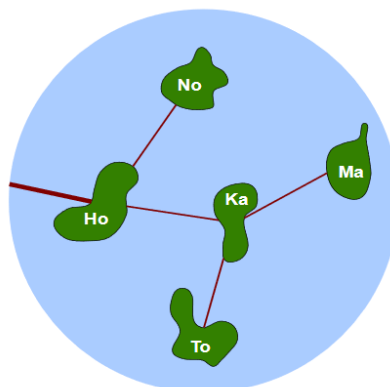
A programozás versenyek gráfokkal kapcsolatos feladatainak szövegében talán éppen a gráfokkal kapcsolatos matematikai ismeretek, illetve azok rendszerezettségének hiánya miatt általában nem is szerepel a gráf kifejezés. A tanulók számára így a gráf csak a feladatban leírt probléma ábrázolásának eszköze. A gráfokkal kapcsolatos feladatok a szövegkörnyezet miatt általában könnyen érthetőek, megoldásukhoz azonban precíz, szisztematikus és logikus gondolkodásra van szükség. A versenyzők ilyen módon a helyes megoldáshoz is el tudnak jutni gráfelméleti ismeretek nélkül is, mégis rengeteg lehetőség van a tanulók algoritmikus gondolkodásának fejlesztésére gráfelméleti ismereteik fejlesztésének segítségével.

Gráfokkal kapcsolatos versenyfeladat gyakran előfordul a Hódítsd meg a biteket! verseny feladatai között is. Ezekben nem kerül leírásra a gráf szó, helyette például szigetek közötti utakról, kapcsolatokról van szó, ahogyan a következő példában is (*e-bód, 2017., benjamin – nehéz, kadét – közepes, junior – könnyű, 2017-DE-06, Honomakato*), amelyben a gráf összefüggőségét kellett vizsgálniuk a versenyzőknek. Ezen a versenyen könnyebbséget jelenthet a tanulók számára az is, hogy a négy megadott válaszlehetőség közül kell kiválasztaniuk a megfelelőt. Emiatt nem szükséges minden lehetőséget megvizsgálniuk, hanem elegendő csupán a négy válaszlehetőséget végiggondolniuk, leellenőrizniük.

A Honomakato szigetcsoport öt szigetből áll: Ho, No, Ma, Ka és To.

A Ho fősziget üvegszálak kábellel csatlakozik az internethez. Ezen kívül optikai kábelek kötik össze a következő szigetpárokat: Ho és No, Ho és Ka, Ka és Ma, illetve Ka és To. Így minden kábel összeköttetésben áll a Ho szigettel és csatlakoztatva vannak az internethez.

Honomakato lakosai szeretnék stabilabbá tenni a kapcsolatot azáltal, hogy ha egy optikai kábel megsérülne, akkor is kapcsolatban maradjon minden sziget.



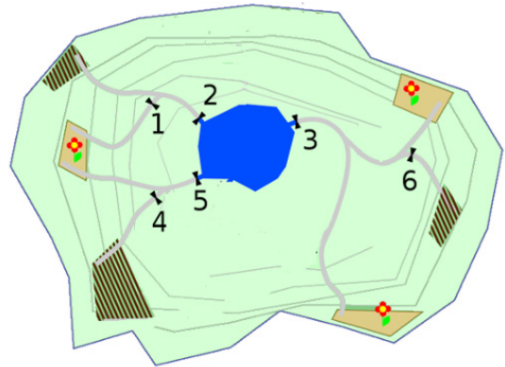
Az alábbi kábelpárok melyike teszi stabilabbá a kapcsolatot?

- A) A Ho-To és No-Ma kábelek.
- B) A Ho-To és Ma-To kábelek.
- C) A Ka-No és No-Ma kábelek.
- D) Két kábel nem elég a hálózat stabilá tételéhez.

Egy korábbi év gráfokhoz kötődő versenyfeladatában (*e-bód, 2015., kishód – könnyű, benjamin – könnyű, 2017-AT-03, Takarékos öntözés*) pedig a tóból a földekhez vezető lehetséges utakat kellett vizsgálni. Ennél a feladatnál is nagy könnyebbséget jelent a megoldás során az, hogy a négy válaszlehetőség közül kell választani.

A Nyírfa családnak van egy tava, a tó körül pedig földjei. A vizet a tóból csatornákon keresztül vezetik a földekre. Ehhez mindig a megfelelő zsilipeket nyitják ki vagy zárják le.

A Nyírfa család takarékosan bánik a tó vizével. Most csak a virággal beültetett földeket akarják megöntözni. Az üres földeknek száraznak kell maradniuk.



Segíts a Nyírfa családnak! Melyik zsilipeket nyissák meg?

| | |
|---|---|
| A | Az 1., a 3. és a 4. zsilip legyen nyitva. |
| B | A 2., a 3. és a 6. zsilip legyen nyitva. |
| C | Az 5. és a 3. zsilip legyen nyitva. |
| D | A 2., a 3. és az 5. zsilip legyen nyitva. |

A Nemes Tihamér programozási verseny korábbi gráfokkal kapcsolatos számítógép nélküli feladatai között is vannak a fentihez hasonló feladatok. A következőnél például bizonyos utak lezárásáról, a gráfban lévő elvágó élekről van szó. Lényeges különbség viszont az előző két példához képest a gráf megadásának módja. Az e-hód verseny példaként említett feladataiban a tanulók a gráf rajzát kapták meg, a Nemes Tihamér verseny feladatánál (*Nemes, 2017-18, 9-10. osztály*) viszont csak az élek listáját, vagyis itt már a gráf megfelelő ábrázolása is a versenyzők feladata. (Ez a Nemes Tihamér verseny és az OKTV gráfokkal kapcsolatos feladatainál általában így van.)

3. feladat: Városok (20 pont)

Ismerjük egy megye települései közötti utakat (a két település sorszámával, amelyeket összekötnék). Tudjuk, hogy el lehet jutni bármely településről bármely településre. Egyes utakat felújítás idejére lezárnak. Add meg, hogy melyek azok az utak, amelyek közül bármelyiket lezárva, nem lehet eljutni bármely településről bármely településre!

A. (5,2), (1,3), (1,4), (2,3), (5,6), (2,6), (3,4)

B. (9,5), (6,5), (10,6), (10,9), (3,2), (3,10), (8,3), (1,8), (7,8), (4,1), (4,7), (7,1)

Egy másik gráfokhoz kötődő feladat (*Nemes, 2016-17, 9-10. osztály*) a topologikus rendezéssel foglalkozott. Itt az élőlények táplálkozási kapcsolatainak átlátását megkönnyíti, a megoldáshoz vezető utat pedig lerövidíti, ha irányított gráf segítségével ábrázoljuk az összefüggéseket. A helyes megoldáshoz azonban más úton, gráfos ábrázolás nélkül, következtetés útján is eljuthatnak a tanulók.

3. feladat: Rendezés (31 pont)

Egy tóban sokféle élőlény él, melyekről tudjuk, hogy melyik melyiket eszi meg: egy számpár első tagja az evő, a második tagja pedig az általa megevett élőlény sorszáma). Add meg az alábbi táplálkozási kapcsolatok alapján egy olyan sorrendjüket, amiben minden párból előbb kell szerepelnie annak, aki eszik, annál, amit megeszik. (Több megoldás esetén bármelyiket.)

A. (5,2), (1,3), (1,4), (2,3)

B. (9,5), (6,5), (10,6), (10,9), (3,2), (3,10), (8,3), (1,8), (7,8), (4,1), (1,10)

A következő példaként említendő feladatnál (Nemes, 2016-17, 5-8. osztály) valamivel nehezebb észrevenni azt, hogy a megadott adatok gráf segítségével történő ábrázolása vezethet el könnyen a megoldáshoz. Ennél a feladatnál a gyanúsítottak állításai alapján érdemes elkészíteni egy gráfot, majd ebben a gráfban kell megkeresni a legrövidebb kört, mert a legrövidebb körhöz tartozó csúcsoknak megfelelő személyek alkotják a gyanúsítottak legkisebb létszámú csoportját.

1. feladat: Hazudósok (40 pont)

Egy bűncselekmény helyszínén N gyanúsított járt. A rendőrségi kikérdezésre kétféle választ adhattak:

- Együtt(i, j): az i . állítása szerint az i . és j . találkozott egymással a bűncselekmény helyszínén
- Előbb(i, j): az i . állítása szerint az i . előbb elment, mint a j . megérkezett

A gyanúsítottak állításai ellentmondóak, pontosan egy valaki hazudott. Add meg az alábbi állítások csoportjára a legkisebb létszámú gyanúsított csoportot, amiben biztosan van hazudós!

- A. Együtt(1,2), Előbb(2,4), Előbb(3,1), Előbb(4,3), Együtt(1,4)
 B. Együtt(1,2), Együtt(4,3), Előbb(3,2), Előbb(1,3)
 C. Előbb(4,2), Előbb(1,2), Előbb(3,4), Előbb(4,1), Előbb(2,3)

6. Összegzés

Összességében elmondható, hogy a programozás versenyek számítógép nélküli feladatainak jelentős részénél előkerülnek bizonyos matematikai ismeretek. Az ilyen feladatok megoldását általában megkönnyíti, leegyszerűsíti, ha a tanulók rendelkeznek a feladathoz kapcsolódó matematikai ismeretekkel, de általában szisztematikus próbálgatással, megfelelő következtetés útján is megoldhatóak ezek a feladatok.

A programozás versenyek feladataiban gyakran előforduló matematikához kapcsolódó feladattípusokat a megadott szabály alkalmazását igénylő, a kombinatorikai jellegű, a rekurzióval, mohó stratégiával vagy optimalizálással, illetve a gráfelmélettel kapcsolatos feladatok jelentik. A versenyfeladatok vizsgálata és kategorizálása alapján úgy gondolom, hogy a tanulók programozási versenyekre – főleg azok első fordulóra – való felkészítése során az algoritmizálási (pszeudokóddal, hibakereséssel, algoritmusok átalakításával kapcsolatos), programozási feladatok megoldása és gyakorlása mellett érdemes a feladatokban előforduló matematikai témakörökkel is megismertetni a tanulókat. A versenyen nagy előnyt jelenthet számukra, ha nem akkor találkoznak először például mohó stratégiára épülő vagy optimalizálással kapcsolatos feladattal. Ezek mellett pedig talán a legfontosabb az, hogy a verseny előtt lássanak gráfokhoz kötődő feladatokat, és bátran használják a gráfokat a szemléletes ábrázolás eszközeként bizonyos problémák esetén. Mindez már az általános iskolás korosztály esetében is hasznos, hiszen már a nekik szóló versenyfeladatokban is előfordulnak gráfokkal kapcsolatos feladatok. A középiskolásoknál pedig különösen fontosak lehetnek ezek a matematikai ismeretek a számítógép nélküli programozás feladatok megoldásánál, mert nekik általában nagyobb, összetettebb példákra kell megoldaniuk az ilyen feladatokat, ezek átlátása pedig jóval nehezebb a megfelelő matematikai háttérismeretek nélkül.

7. Hivatkozások

- [1] „e-hód - Hódítsd meg a biteket!,”
<http://e-hod.elte.hu/> utoljára megtekintve: 2018.11.20.
- [2] „Logo Országos Számítástechnikai Tanulmányi Verseny,”
<http://logo.inf.elte.hu/index.html> (utoljára megtekintve: 2018.11.20.)
- [3] „Nemes Tihamér Nemzetközi Informatikai Tanulmányi Verseny - Programozás kategória,”
<http://tehetseg.inf.elte.hu/nemes/index.html> (utoljára megtekintve: 2018.11.20.)
- [4] „A 2018/2019. tanévi OKTV-vel kapcsolatos tudnivalók,”
https://www.oktatas.hu/pub_bin/dload/kozoktatas/tanulmanyi_versenyekek/oktv/oktv2018_2019_vk/116_informatika_1819.pdf (utoljára megtekintve 2018.11.20.)
- [5] „International Olympiad in Informatics,”
<https://ioinformatics.org/files/oi-syllabus-2018.pdf> (utoljára megtekintve: 2018.11.20.)
- [6] J. Kosztolányi, I. Kovács, K. Pintér, J. Urbán és I. Vincze, Sokszínű matematika 11., Tizenötödik kiadás, Szeged: Mozaik Kiadó, 2017, pp. 38-62.
- [7] J. Kosztolányi, I. Kovács, K. Pintér, J. Urbán és I. Vincze, Sokszínű matematika 9., Hatodik kiadás, Szeged: Mozaik Kiadó, 2018, pp. 38-42.