

Tartalom

[Tartalom](#)

[Bevezető](#)

[Alapelvek](#)

[1. óra Játsszunk robotosát! \(Programozási alapfogalmak\)](#)

[Vajas-lekváros kenyér készítő robot](#)

[Bevezető](#)

[Szükséges eszközök:](#)

[Tervezett időtartam](#)

[Foglalkozás](#)

[Források](#)

[Karel, a Robot](#)

[Bevezető](#)

[Szükséges eszközök](#)

[Tervezett időtartam](#)

[Foglalkozás](#)

[Konklúzió](#)

[2. óra Fűnyíró robot - Járd be a táblát!](#)

[Bevezető](#)

[Szükséges eszközök:](#)

[Foglalkozás](#)

[Egy kis fejtagítás](#)

[Vigyázz, kerítés! - feltételes utasítás](#)

[Bevezető](#)

[Szükséges eszközök](#)

[Foglalkozás](#)

[Feltételes ismétlés](#)

[Konklúzió](#)

[Források](#)

[3. óra Psszt! Tudsz titkot tartani? \(Titkosítás\)](#)

[Bevezető](#)

[Szükséges eszközök:](#)

[Feladat](#)

[Caesar-féle titkosítás \(ABC eltolás\)](#)

[Titkosítás publikus kulccsal](#)

[Érdekességek](#)

[Források](#)

4. óra Hány ujj van a számítógépnek? (Bináris számok)

Szükséges eszközök:

Bevezető

Áttérés egyik számrendszerből a másikba

Foglalkozás

Feladatok

Születésnapok kettes számrendszerben

Számok felsorolása

Tervezett időtartam

Foglalkozás

Bináris számolás az ujjaink segítségével

Küldjünk e-mailt kettes számrendszerben!

Tervezett idő

Érdekességek

CD működése

Morze kódok

Hármas számrendszer

Források

5. óra Rajzoljunk, hol a hiba? (Képkalkotás, hibafelismerés)

Szükséges eszközök:

Bevezető

Képbábrázolás (15 perc)

Fax gép (20 perc)

Hibafelismerés (20 perc)

Források

6. óra Hol lehet...? (Keresési algoritmusok)

Bevezető

Szükséges eszközök:

Lineáris keresés

Bináris keresés (számok keresése)

Bináris keresés (szótár verzió)

Hash keresés

Források

7. óra Sorakozó!!! (Rendezési algoritmusok)

Bevezető

Szükséges eszközök

Kiválasztó rendezés

Buborék rendezés

Gyorsrendezés

Források

8. óra Merre menjek? (Rendezési hálók, útvonalválasztás)

Rendezési hálók

Bevezető

Szükséges eszközök

Források

Útvonalválasztás és holtpont

Bevezető

Szükséges eszközök

Forrás

9. óra Debuggolás

Debuggolás

Szükséges eszközök

Foglalkozás

10 óra. Évzáró utolsó óra

Bevezető

A tananyag célja, hogy a gyerekeket megismertessük a számítástudomány (<https://hu.wikipedia.org/wiki/Számítástudomány>), algoritmikus gondolkodás és programozás alapfogalmaival. A tananyag nagy része a CS Unplugged (<http://csunplugged.org/>) foglalkozásainak itt-ott átalakított verzióiból épül fel, illetve a Karel keretrendszerből ([https://en.wikipedia.org/wiki/Karel_\(programming_language\)](https://en.wikipedia.org/wiki/Karel_(programming_language))) kialakított foglalkozásokra épül.

A tananyag olyan algoritmusokat, fogalmakat foglal magában, amit középiskolában vagy egyetemen tanítanak, de valójában egyszerűek és játékosan előadva egy 9-10 éves gyerek is könnyen megérti. A fő cél, hogy bevezessük a gyerekeket a számítógépek működésének varázslatos világába, hátha később kedvet kapnak a programozáshoz.

Ezen kívül jó lenne, ha a jövő generációja nem úgy tekintene a körülötte megjelenő számítógépekre és robotokra, mint egy érthetetlen, misztikus világra, amit csak a különcök érthetnek meg. Úgy gondoljuk, hogy a jövő számítógép-felhasználóinak már programozási tudással is rendelkezniük kell majd, hogy még hatékonyabban végezhesék a munkájukat.

Az oktatást 10 alkalomra kell tagolni és 6-10 gyerekkel érdemes játszani.

A tananyag az "open source" szoftverek mintájára teljesen publikus és mindenki által szabadon használható (License: <https://creativecommons.org/licenses/by-nc-sa/3.0/> Creative Commons BY-NC-SA)

- Bárki taníthatja, akár pénzt is kérhet érte (a tanításért, nem a tananyagért), de fel kell valahol tüntetnie, hogy ebből a tananyagból tanít.

- A tananyagot klónozni és módosítani is szabad, de akkor fel kell tüntetni, hogy ebből az anyagból származik és figyelembe kell venni az Alapelveket.

A foglalkozásokat két tanárnak érdemes tartania, hogy tudjanak segíteni a gyerekeknek, ha kell.

Alapelvek

- A tananyagot számítógép nélkül lehessen tanítani, hogy azok számára is elérhető legyen, akik nem tudnak számítógéphez jutni. A számítástudománnyal való ismerkedést számítógép nélkül is el lehet kezdeni.
- Igyekezzünk megmagyarázni a gyerekeknek, hogy egy adott téma vagy foglalkozás hogyan kapcsolódik a való világban található megoldásokhoz.
- A tananyag oktatásához ne kelljen komoly eszközöket vásárolni, vagyis olcsó legyen az oktatása.
- Általában a gyerekek ugyanazon a feladaton dolgozzanak közösen, vagy kevés számú csoportban, hogy a csapatmunkát fejlesszük, valamint ne maradjon ki senki a játékból. A közös feladatmegoldás azért is fontos, mert fejleszti a gyerekek kooperációs képességét, a tanár pedig könnyen tudja ellenőrizni, hogy ki mit csinál
- A foglalkozások során a versenyzést csak mindenki által jól elsajátított játékoknál alkalmazzunk, hogy ne legyen senkinek kudarcélménye.
- A játékok legyenek viccesek és érdekesek, motiválóak. Nem még egy matekszakkört vagy sakkszakkört akarunk, hanem inkább figyelemfelkeltés, ismerkedés, szórakozás a cél.
- Igyekezzünk olyan foglalkozásokat kitalálni, amik megmozgatják a gyerekeket, interaktívak, hogy minél gyakrabban keljenek fel a padból.
- Amennyiben lehetséges vezessük rá őket, hogy ők találják ki a következő lépést. Támogassuk a kérdezést, ötletelést, hogy szabadon engedhessék a fantáziájukat. Ennek érdekében megengedett a tanár számára az improvizáció. Például ha a gyerekek előállnak egy jó ötlettel, hogy mit hogyan jelöljünk, akkor nem kell ragaszkodni az eredeti elképzeléshez, lehet azt a jelölést is használni, amit ők találtak ki.
- A tanár is vegyen részt a játékokban.
- Jó lenne ha az egész nem lenne túl "fiús". Ha a lányok is szeretnék.

1. óra Játsszunk robotosat! (Programozási alapfogalmak)

Vajas-lekváros kenyér készítő robot

Bevezető

Ezen a foglalkozáson egy Robot programozásának nehézségeit demonstráljuk egy olyan robot segítségével, aki vajas-lekváros kenyeret készít.

Szükséges eszközök:

- Egy teli lezárt lekvárosüveg (200 Ft)
- Egy adag vaj vagy margarin lezárt dobozban (200 Ft)
- Kés, kanál
- Egy csomag toast kenyér. (150 Ft)
- Tányér
- Papírtörő

Tervezett időtartam

20 perc

Foglalkozás

A foglalkozás során az egyik tanár lesz a robot, aki csak néhány utasítást ismer. A gyerekek feladata, hogy irányítsák a robotot úgy, hogy az egy vagy több vajas-lekváros kenyeret készítsen.

A gyerekek egyenként adnak utasításokat a tanár robotnak. A robot mindig az utolsó kiadott utasítás utáni állapotban marad.

A tanár robotnak azt kell tennie, hogy a túl bonyolult utasításokra azt mondja, hogy "nem tudom kiszámítani". Ezen kívül minden utasítást szó szerint kell értelmezni, kihasználva azt, hogy a gyerekek nem elég precízen fogalmazznak. A túl általánosan megfogalmazott utasításokat a robot túlságosan is szó szerint értelmezi. Ez számos mulatságos jelenethez vezet. Mindeközben a gyerekek megtanulják, hogy egy robotot nem is olyan könnyű irányítani, nagyon precíznek kell lenni és közben még jól is szórakoznak.

- Például, amikor egy gyerek arra kéri a robotot, hogy vegye kezébe a (nem túl éles) kést, akkor a kést ne a nyelénél fogja meg, vagy ne azzal a kezével fogja meg, amivel értelmes elvégezni ezt a feladatot.
- Ha arra kéri a robotot, hogy vegyen ki kenyeret, akkor vegye ki az összes kenyeret.
- Ha arra kéri, hogy tegye le a kenyeret, akkor tegye a földre.
- Ha arra kéri, hogy nyissa ki a lekváros üveget, akkor próbálja egy kézzel tenni ezt, vagy próbálja az asztalhoz ütögetni az üveget (nem tudja, hogy kell kinyitni).

Érdeemes addig folytatni a játékot, amíg tényleg sikerül elkészíteni a vajas-lekváros kenyeret.

A végén meg kell beszélni a gyerekekkel, hogy tulajdonképpen mi történt, miért nem tudta végrehajtani az utasításokat a robot. Mire volt szükség ahhoz, hogy jól működjön a robot? A tanár kérdezze meg a gyerekeket, hogy mi a robot, és mi a különbség egy robot és egy számítógép között (minden robot agya egy/vagy több számítógép).

Itt egy kicsit el lehet mondani, hogy a Robot egy olyan gép, aminek vannak érzékelői és képes mozgatni tárgyakat, vagy karjait, vagy képes mozogni. Az egészet számítógép irányítja.

Lehet beszélni az ipari robotokról akik ennél a robotnál butábbak. Vannak humanoid robotok is már.

Források

<http://www.bu.edu/gk12/tyler/lessons/PBJ/PBJ.htm>

<https://www.youtube.com/watch?v=leBEFaVHIE>

<http://csunplugged.org/harold-the-robot-2/>

Karel, a Robot

Bevezető

Ezen a foglalkozáson Karel robot világát mutatjuk meg a gyerekeknek, és egyszerűbb feladatokat oldunk meg ebben a világban. Karel egy olyan Robot, ami egy négyzetrácsos világban mozog és olyan egyszerű utasításokat ismer, hogy "Lépj előre", "Fordulj jobbra", stb...

Szükséges eszközök

- Egy 6x6-es négyzetrácsos világ (a start és cél négyzet), ahol egy cella akkora, hogy ha két gyerek egymás melletti cellákban megáll, akkor elférnek. A négyzetrácsot kartonlapokból, ragasztószalagból, vagy 100 db A4-es papírral is megoldható más egyéb módon lehet megalkotni. Mi 36 db A4-es papírral oldottuk meg, amiket csak egymás mellé leraktunk.
- Tábla, amire lehet írni.
- Kréta
- Kis szivacs labda

Tervezett időtartam

30 perc

Foglalkozás

Legyél robot! Kiválasztunk egy önkéntest (ha sokan jelentkeznek, akkor egy labda odadobásával jelezhetjük, hogy kit választottunk). A kiválasztott gyerek felveszi a **két utasításos robot** szerepét. A két utasítást felírjuk a táblára: Lépj előre; Fordulj jobbra. Az a feladat, hogy a robotot egy start mezőről (rá kell írni a négyzetre) egy cél mezőre irányítsák a gyerekek közösen (a cél mező a start mezőtől balra legyen). Az utasítás-sorozatot felírjuk (az egyik tanár ír a másik a gyerekekkel kommunikál) a táblára egymás alá, és később hivatkozhatunk erre, mint **programra**.

A programot le is futtathatjuk újra a gyerekekkel. Valamint ha nem az optimális megoldást találták meg, akkor javíthatunk rajta.

Első körben szóban adjuk az utasításokat a robotnak. Először mindenki egy utasítást ad, azután lehet két-három utasítást is adni egyszerre.

A játék során el kell jutniuk a gyerekeknek arra a pontra, hogy hiányzik a "Fordulj balra" utasítás. Itt fontos tisztázni, hogy a robotba az alaputasítások bele vannak drótozva. Az új utasítás definíciója a program része.

```
definiáld fordulj_balra:  
    fordulj_jobbra  
    fordulj_jobbra
```

fordulj_jobbra

Itt érdemes rövidítéseket alkalmazni, hogy gyorsabban írjuk a programot a táblára. Pl: FJ, LE, FB.

Az új utasítással csak a program leírása lesz rövidebb, de nem fog új utasítást tudni a robot. A balra fordulás utasítás végrehajtásánál a robot ugyanúgy háromszor fordul jobbra, mint addig. Le kell futtatni egy olyan programot ahol van balra fordulás, hogy lássák, hogy a robot háromszor jobbra fordul ekkor is.

Kérdezzük meg a gyerekeket, hogy milyen új utasítást tudnak még kitalálni (lépj jobbra, lépj balra, lépj hátra, hátra arc)

Konklúzió

- A robotok alapvetően buták
- A programozók teszik okossá
- Precíz utasításokat kell adni
- A robotok alaputasításokat ismernek
- A program utasítások sorozata
- Új utasításokat is tudunk készíteni

2. óra Fűnyíró robot - Járd be a táblát!

Bevezető

A foglalkozás célja, hogy megtanítsuk a gyerekeknek a **ismétlési** vezérlési szerkezetet. Sokszor egy programban ugyanazok az utasítások vagy utasítássorozatok ismétlődnek. Ahhoz, hogy a program ne legyen túl hosszú, az ismétlődő utasításokat helyettesítjük egy olyan új utasítással, amibe beleírjuk, hogy hányszor ismétlődjön egy adott utasítássorozat.

Szükséges eszközök:

- Egy 5x5-ös négyzetrácsos világ, ahol egy cella akkora, hogy ha két gyerek egymás melletti cellákban megáll, akkor elférnek. A négyzetrácsot kartonlapokból, ragasztószalagból, A4-es lapok, vagy más egyéb módon lehet megvalósítani. Mi A4-es papírlapokat tettünk le a földre.
- Tábla, amire lehet írni.
- Kavicsokat esetleg vihetik a fű megjelölésre, illetve ha papírt használunk akkor az egyik oldalát megjelölhetjük (zöld filc).

Foglalkozás

Alaputasítások: Lépj előre, fordulj jobbra, fordulj balra. Kezdetnek ezeket felírjuk a táblára. Rövidítéseket is lehet használni.

Ezen a foglalkozáson a gyerekek egy fűnyíró robotot programoznak. A négyzetrács világ egy kert lesz.

A robot feladat a teljes kert bejárása (a négyzetrács széle kerítéssel van körülvéve), abból a célból, hogy a robot lenyírja a fűvet a kertben.

Először a gyerekekkel közösen felírjuk a programot, ami bejárja a területet (nem feltétlenül szisztematikusan). A Robot a bal alsó sarokból indul és észak felé néz.

Ahhoz, hogy a teljes táblát bejárja a robot, sok lépésre van szükség. Az első sor felírása után felvetjük a kérdést, hogy hogyan lehetne a továbbiakban ezt rövidebben felírni (a programozók lusták !!!) (esetleg felhívjuk a figyelmet az ismétlődésekre).

Ekkor megmutatjuk, hogy milyen jelölést alkalmazunk a ismétlés vezérlési szerkezetre, és közösen felírjuk az új programot (két sorra), ami most sokkal rövidebb lesz.

```
fordulj jobbra
ismételd (5x):
    lépj előre
fordulj balra
lépj előre
```

```
fordulj balra
ismételd (5x):
    lépj előre
```

Ha kész a program, akkor lefuttatjuk, úgy, hogy a gyerekek vagy robot megjelölik (pl: felszedik/megfordítják a lapokat, letesznek egy kavicsot, stb), hogy hol jártak már.

Tipp: Ezek a programok már bonyolultabbak és érdemes a táblán mutatni, hogy melyik utasításnál tartunk. Az egyik gyereket megkérhetjük erre, de adjunk a kezébe egy mutató eszközt.

Hívjuk fel a figyelmet arra, hogy az `ismételd` utasítás az őt követő bennebb írt utasításokat fogja ismételni. Gyakorlásképpen kérdezzük meg hogy mi a különbség a két kódrészlet között:

```
ismételd (5x):
    lépj előre
fordulj balra
```

```
ismételd (5x):
    lépj előre
    fordulj balra
```

Egy kis fejtágítás

Ezen a ponton történhet a számítástechnikai alapfogalmak ismertetése.

Számítógépes programok futnak mindenhol.

A számítógépes programok körülvesznek bennünket. Ezek vezérlik a legtöbb eszközt és kűtyűt, programozók által írt utasításokat követnek.

Mobiltelefonok: Programok teszik lehetővé a telefonhívást vagy az sms-küldést is. Ha felhívunk valakit, program keresi meg a telefonszámát.

Szoftver: Mindazt, amit egy számítógéppel meg tudunk csinálni – mint például az internet böngészése, a szövegszerkesztés, a játszás vagy a zenehallgatás – programozók által megírt program teszi lehetővé.

Autók: Sok autóban számítógép figyeli a sebességet, a hőmérsékletet vagy az üzemanyagszintet. A biztonság érdekében a számítógép a fék működésében is szerepet játszik.

Vigyázz, kerítés! - feltételes utasítás

Bevezető

Ezen a foglalkozáson az a cél, hogy a gyerekek megismerjék a feltételes utasítást. A robotok szembesülnek a külvilág tulajdonságaival, és ennek megfelelően más és más utasításokat hajtanak végre.

Szükséges eszközök

- Egy 4x5-es négyzetrácsos világ (ha a bal alsó sarokban állok, akkor felfelé 4 sor van és jobbra 5 oszlop), ahol egy cella akkora, hogy ha két gyerek egymás melletti cellákban megáll, akkor elférnek. A négyzetrácsot kartonlapokból, ragasztószalagból, A4-es lapok, postit vagy más egyéb módon lehet megvalósítani.
- Tábla, amire lehet írni.
- Kavicsokat esetleg vihetik a fű megjelölésre, illetve ha papírt használunk akkor az egyik oldalát megjelölhetjük (zöld filc).

Foglalkozás

Ha a kertünk mérete csökken (a tábla méretét csökkentjük 4x5-re, egy oszloppal kevesebb van), ugyanez a fenti program működőképes-e? Hajtsuk végre az előző foglalkozásban felírt programot a kisebb udvarban. Ilyenkor a pálya szélén túlhaladva a robot hibára fut (a tanár viccesen érzékelteti ezt).

Megkérdezhetjük a gyerekeket, hogy itt mi történt és miért?

Valójában a robot vak. Néhány utasítást (pálya nélkül) a robot (gyerek) bekötött szemmel is végrehajthat, annak szemléltetésére, hogy a robot nem rendelkezik érzékelőkkel. Mit kell érzékelnie a robotnak, ha általánossá szeretnénk tenni a programunkat, hogy bármilyen kert esetén le tudja nyírni a fűvet? (távolságmérő, kamera)

Felruházzuk a robotot azzal a tulajdonsággal, hogy tudja érzékelni az előtte lévő kerítést. Ennek ismeretében hogyan lehetne módosítani az előző programot úgy, hogy bármilyen kert esetén működőképes legyen? (ki lehet próbálni az ötleteket). A gyerekekkel közösen ötletelve kitalálhatjuk, hogy mit csináljon a robot ha a kerítéshez ér.

Első verzióban: Ha kerítés van előtte, akkor forduljon balra (a robot induló pozícióját be kell állítani). Ezzel megismerkednek a gyerekek a feltételes utasítás fogalmával (fel lehet vetni, hogy mi történik, ha a robot kerítés esetén megfordul).

Írhatunk egy olyan programot, ami 18x ismétli azt, hogy ha kerítés van előtte, akkor balra fordul, különben előre lép. Így a robot a kerítés mentén körbejárja a kertet. A Robot a bal alsó sarokból indul és észak felé néz.

Kert körbejárása

```
fordulj_jobbra
```

```
ismételd (18x):
```

```
  ha (#):
```

```
    fordulj_balra
```

```
  különben:
```

```
    lépj_előre
```

Második verzióban: Bevezetjük azt, hogy tudja érzékelni, hogy előtte le van vágva a fű.

Ez abban segít, hogy spirál alakban járjuk be a kertet. Amikor körbe ér a robot, akkor is elfordul amikor olyan hely van előtte, ahol már lenyírta a fűvet (jelöljük ismét a lenyírt fűvet, lásd fenn). A kérdés, hogy hányszor kell ismételnie ezt a folyamatot. Itt kerül bevezetésre a végtelen ismétlés. A robot a kert közepéig jut, mert körben mindenhol levágta a fűvet. Itt körbe-körbe forog.

Vicces jelenet, ha itt “várunk egy hétig”, amíg újra kinő a fű, ekkor ismét el tud indulni a robot.

Kert bejárása spirálban

```
fordulj_jobbra
ismételd (örökké):
    ha (fű van előtted):
        lépj_előre
    különben:
        fordulj_balra
```

Feltételes ismétlés

Megpróbálhatjuk a fűnyíró robotot arra rávenni, hogy soronként nyírja a fűvet. Ezt általában mi is így csináljuk. A jobb alsó sarokban indulunk, viszont nem tudhatjuk előre, hogy hányat kell lépni a kerítésig. Itt használható a feltételes ismétlés. Nem egy fix darabszámú ismételünk valamilyen utasítást, hanem addig ismételjük amíg egy utasítás teljesül. Például egy sort így mehetünk előre.

```
ismételd_ha (nincs előtted kerítés):
    lépj_előre
```

Ezután egy sorral feljebb megyünk és balra fordulunk a második sor elejére

```
fordulj_balra
lépj_előre
fordulj_balra
```

Majd megint végigmegyünk a soron, de most balra

```
ismételd_ha (nincs előtted kerítés):
    lépj_előre
```

Majd megint megfordulunk

```
fordulj_jobbra
lépj_előre
fordulj_balra
```

Most ott vagyunk ahonnan elindultunk, csak egy sorral feljebb, ha ismételtük ezt a két sor bejárását, akkor végigjutunk a kerten.

Kert soronkénti bejárása (kert végén hibára fut)

```
fordulj_balra
```

```
ismételd (örökké):
    ismételd_ha (nincs előtted kerítés):
        lépj_előre
    fordulj balra
    lépj előre
    fordulj balra
ismételd_ha (nincs előtted kerítés):
    lépj_előre
    fordulj jobbra
    lépj előre
    fordulj jobbra
```

Sajnos ez a program se tökéletes még, mert nem tudjuk előre, hogy hány sor van a kertben. Ehhez az, kell, hogy a forduláskor vizsgáljuk, hogy a kerítéshez értünk-e. Ha igen, akkor leállunk. Itt a teljes program:

Kert soronkénti bejárása leállással

```
ismételd (örökké):
    ismételd_ha (nincs előtted kerítés):
        lépj_előre
    fordulj balra
    ha (#):
        állj le
    különben:
        lépj előre
    fordulj balra
ismételd_ha (nincs előtted kerítés):
    lépj_előre
    fordulj jobbra
    ha (#):
        állj le
    különben:
        lépj előre
    fordulj jobbra
```

Konkluzió

- a fűnyírás nehéz
- szükség van ismétlésre (rövidebb programok)
- kellenek érzékelők és feltételes utasítás (hogy megváltozott körülmények között is jól működjön a program)
- feltételes ismétlés is hasznos (néha nem tudjuk előre, hogy hányszor kell ismételni valamit)
- hasznos, hogy ki tudjunk lépni az ismétlésből

Források

[https://en.wikipedia.org/wiki/Karel_\(programming_language\)](https://en.wikipedia.org/wiki/Karel_(programming_language))

3. óra Psszt! Tudsz titkot tartani? (Titkosítás)

Bevezető

A titkosítás arra való, hogy emberek között üzeneteket küldjünk úgy, hogy a küldőn és a fogadón kívül mások ne értsék meg az eredeti üzenetet. A számítógépek világában sokszor kódolt csatornán kommunikál két gép. Erre azért van szükség, mert az interneten az üzenetek több gépen keresztül jutnak el a feladótól a címzettig. Titkosítás nélkül bárki elolvashatja az üzeneteinket.

Szükséges eszközök:

- Elegendő csoki (vagy cukorka) minden gyerek részére (a tanároknak is)
- Egy doboz, amit lakattal lehet lezárni (Olyan doboz kell, amire két lakatot is rá lehet tenni)
- Két lakat két kulccsal (a lakatok különbözőek legyenek)
- Papír cetlik.
- Minden gyereknek: Betű-kód összerendelés (ABC betűihez egy szám rendelve)
- Mágnes a táblára tűzéshez

Feladat

Ahhoz, hogy demonstráljuk a gyerekeknek a titkosítás alapjait egy egyszerű példát lehet alkalmazni.

1. Válasszunk egy önkéntesen jelentkező gyereket azzal, hogy csokit kap. Ezután hívjuk ki és adjuk oda neki az egyik lakat kulcsát úgy, hogy a többi gyerek ne lássa, és megmondjuk neki, hogy a kulcsot csak akkor vegye elő, amikor szólunk.
2. Válasszunk még 3 önkéntes gyereket.
3. Állítsuk a gyerekeket sorba. Az egyik oldalon a tanár áll aki a küldő. A másik oldalon a csokit kapó gyerek, ő a fogadó. A három gyereket a küldő és a fogadó közé kell állítani sorba.
4. Ezután elmondjuk a gyerekeknek, hogy most a küldő tanár el fogja küldeni a csokit a gyerekeken keresztül a fogadónak, és hogy a küldő meg a fogadó nagyon messze vannak egymástól (interneten küldjük az üzenetet). Felmerül a kérdés, hogy meg lehet-e bízni a három továbbító gyerekben? Erről meg lehet kérdezni a gyerekeket. A végkövetkeztetés az legyen, hogy nem.

5. Kérdezzük meg a gyerekeket, hogyan juttassuk el biztonságosan a csokit a fogadónak? Megoldásként egy dobozt veszünk elő, amibe bezárjuk a csokit és a dobozt egy lakattal lezárjuk.
6. Viszont ha a lezárt dobozt most eljuttatjuk a fogadónak, akkor ő nem tudja kinyitni, ezért ez a módszer csak akkor működik, ha a küldőnek és a fogadónak is van egy ugyanolyan kulcsa. Ennél a módszernél a küldőnek és a fogadónak legalább egyszer találkozniuk kell (hogyan átadják egymásnak a kulcsot).
7. Most már szólhatunk a fogadó gyerekeknek, hogy vegye elő a kulcsát és nyissa ki a dobozt. Ekkor megeheti a csokit és a többi gyereknek is adunk egy csokit.

A következő játékban ezt fogjuk eljátszani konkrét üzenettel és a Cézár-féle titkosítással.

Caesar-féle titkosítás (ABC eltolás)

Az egyik legegyszerűbb titkosítás a Caesar-féle titkosítás, aminek az a lényege, hogy az abc betűit eltoljuk valamennyi betűvel a kódolt üzenetben.

1. Az asztalnál mutassuk meg az sorszámokkal ellátott ABC-t és magyarázzuk el, hogy hogyan működik. Tehát a kódolandó betű melletti számot növeljük az eltolás mértékével, majd a mellette levő betű lesz a kódolt betű. A végén az alsó sort az elején fejezzük be.
2. Ezután kérdezzük meg a gyerekeket, hogy mondjanak egy egyszerű szót vagy mondatot. Ezt felírjuk a táblára, és alá írjuk a kódolt szöveget.
3. A következő lépésben a dekódolást is bemutathatjuk úgy, hogy mi találunk ki egy szót amit kódolunk és a gyerekeknek kell közösen dekódolni. Itt fel lehet hívni a figyelmet, hogy itt valójában keresést végzünk az alsó sorban és, hogy ha sorba lennének rendezve a betűk akkor könnyebb keresni. Meg lehet kérdezni a gyerekeket, hogy valójában mi itt a kulcs és mi a doboz, mit jelent a bezárni és kinyitni? (A kulcs a titkosítás típusa: Caesar, és az eltolás mértéke: 3)
4. Esetleg megkérdezhetjük a gyerekeket, hogy van-e ötletük arra, hogy hogyan lehetne még titkosítani üzeneteket, milyen ABC hozzárendelést lehetne még alkalmazni? Rávezetjük őket arra, hogy véletlenszerű hozzárendelés lehetséges.
5. A következő lépésben két csoportot alakíthatunk a gyerekekből, és az egyik csoport titkos üzenetet küldhet a másik csoportnak úgy, hogy Caesar-kódolást alkalmaznak egy általuk választott eltolással. Meg kell mondaniuk a kódolásnál alkalmazott eltolást.

Titkosítás publikus kulccsal

1. Ismét felállítjuk a gyerekeket úgy, hogy van egy küldő, egy fogadó és három továbbító gyerek. Ismét csokit szeretnénk továbbítani. Átismételjük az Titkosítás alapjaiban, hogy hogyan tudjuk elküldeni a csokit a dobozzal és a lakattal anélkül, hogy ki tudná nyitni a közvetítő emberek.
2. Ezután azzal nehezítjük a feladatot, hogy azt mondjuk, hogy olyan embernek szeretnénk küldeni a csokit, akivel sose találkoztunk. Vajon, hogyan lehet megoldani a problémát akkor, ha a küldő és fogadó sohasem találkozott.

3. Az egyik ötlet, hogy a kulcsot is elküldjük, vagy először a kulcsot, utána a dobozt küldjük el, de ezek a megoldások nem jók, mert a továbbbítók megtarthatják a dobozt is meg a kulcsot is anélkül, hogy a fogadó vagy a küldő tudná, hogy ez történt.
4. Látványosan adunk a fogadó gyerekeknek egy lakatot kulccsal (a küldőnek már van) és a gyerekek próbálják meg kitalálni, hogyan lehetséges a biztonságos küldés ezek után.
5. A megoldás az, hogy a fogadó rátesz egy saját lakatot a dobozra és visszaküldi a küldőnek. Ezután a küldő leveszi a saját lakatját és elküldi a fogadónak. Amikor a fogadó másodszor is megkapja a dobozt, akkor már ki tudja nyitni a saját kulcsával. Ismét oszthatunk csokit, vagy eleve ne csokival játszunk.
6. (Ez a módszer se tökéletes, mert a továbbbító gyerekek is rátehetnek egy lakatot a dobozra, úgy, hogy a fogadó nem tud róla - man in the middle attack), de ezzel most nem foglalkozunk.
7. Ezt az egészet el lehet játszani Caesar-kódolással is úgy, hogy két csoportot alakítunk ki, akik tetszőleges eltolás számmal dolgozva megvalósítják a fentieket. Először az egyik csapat küld egy kódolt üzenetet, amit a másik csapat megint kódol, és visszaküldi az első csapatnak. Ezután az első csapat a saját kódolása szerint dekódolja az üzenetet, majd elküldi a második csapatnak, akik szintén a saját kódolásuk szerint dekódolják az üzenetet. A két csapat nem ismeri a másik csapat Caesar kódolásának eltolási mértékét.

Érdekességek

Érdekességképpen beszélhetünk különböző biztonsági/titkosítási rendszerekről, azonosítási formákról, hackerekről.

PI. bankkártyák működése, többfaktoros autentikáció, híres hacker támadások, veszélyek a hálózaton.

Források

<http://csunplugged.org/public-key-encryption/>

<http://www.rejtjelezo.hu/monoalfabetikus-abc-eltolas-ismerteto>

<http://www.counton.org/explorer/codebreaking/transposition-ciphers.php>

4. óra Hány ujja van a számítógépnek? (Bináris számok)

Szükséges eszközök:

- Hat nagy bináris kártyából álló (lapok, 1,2,4,8,16, 32 pöttyel) készlet
- 10 készlet kis méretű kártyára (szintén pöttyökkel, lásd. 13. oldal)
- Kinyomtatott angol abc bináris kód táblázat
- (Morze táblázat)

Bevezető

Először felvázoljuk a gyerekeknek, hogy a számítógépek csak két számjegyet használnak, és ezekkel is lehet számolni. Ezt fogjuk most megtanulni.

Áttérés egyik számrendszerből a másikba

Tervezett időtartam

30 perc

Foglalkozás

Ezen foglalkozás során ismerkedhetnek meg a bináris számokkal. Először el kell mondani, hogy a kettes számrendszerbeli számok csak 0-ból és 1-esből állnak, valamint nagyon hasonlóak a 10-es számrendszerbeli számokhoz.

Először a táblát két részre osztjuk egy függőleges vonallal. A bal oldalon a 10-es a jobb oldalon a 2-es számrendszerbeli világot ábrázoltuk. Elmondjuk a gyerekeknek, hogy a Földön azért használják a tízes számrendszert mert 10 ujjunk van, ezért azon a bolygón ahol a földön kívülieknek csak két ujja van, ott biztos 2-es számrendszerben számolnak. Ezután megmutatjuk, hogy egy táblára felírt számból, hogyan lehet kiszámolni a valódi értékeket a 10-es számrendszerben. A helyiértékeket a számok alá írjuk majd szorzás (helyi értékek és valódi értékek szorzata) és összeadás segítségével megkapjuk a szám értékét. Ezután ugyanezt megcsináljuk a 2-es számrendszer oldalán is.

A visszafelé alakítást, vagyis, hogy egy számot, hogyan lehet kettes szám rendszerben leírni azt a pénzek segítségével magyaráztuk el, vagyis, hogy úgy képzeljék el, mintha a helyi értékek pénzek lennének és az lenne a feladat, hogy felírják sorban, hogy miből mennyi kellett. Igazából a pénzzel való számolás egy kicsit olyan, mint egy számrendszer ahol a helyiértékek a lehetséges érmék.

Ezután kihívunk 5 gyereket, és sorban odaadjuk nekik a nagyobb méretű kártyákat (1,2,4,8,16 pöttyel). Velük demonstráljuk a számolási technikát.

Minden pöttyös lap 1-esnek és minden üres lap 0-nak felel meg. Egy bináris szám értéke ilyen módon a pöttyök számával lesz egyenlő.

Kérjük meg a gyerekeket, hogy rakjanak ki 01001-et. Melyik ez a szám a tízes számrendszerben? (9) Mi a 17 a kettes számrendszerben? (010001) Próbáljunk ki még néhányat, amíg meg nem értik a koncepciót. Ha például a 17-et szeretnénk "kirakni", akkor először megkeressük a 17-hez legközelebbi, de annál kevesebb pöttyel rendelkező kártyát. Ez a 16, maradt 1, ami jobbról a legelső kártya. Ezeket úgy fordítjuk, hogy a pöttyök láthatóak legyenek, az összes többi kártyát lefelé fordítjuk. Ezt az átalakítást addig gyakoroltassuk, amíg nem megy mindenkinek.

Kérdezzük meg a gyerekeket, hogy miért nevezik ezeket bináris számoknak. A "bi-" előtag a bicikli, bikini szavakban is megtalálható.

Feladatok

Kiosztunk a gyerekeknek a kis kártyákat, úgy, hogy párokban tudjanak dolgozni (minden párnak 5 kis kártya).

Születésnapok kettes számrendszerben

Írják le a gyerekek (párban dolgozva) kettes számrendszerben, hogy a hónap melyik napján születtek. Ehhez minden gyereknek kell rendelkeznie egy 5 kártyából álló készlettel. Mindenki kirakja a saját készletével a születése napját kettes számrendszerben, majd egy papíron átadja a számot a társának, aki visszafejti a születési napot. Azután ugyanezt a fordított irányba is eljátszhatják.

Számok felsorolása

Tervezett időtartam

10 perc

Foglalkozás

Közösen a gyerekekkel felírjuk az első 8 számot kettes számrendszerben. Megkérdezzük a gyerekeket, hogy hogy lehet ezt megtenni. Fontos, hogy megértsék a számok képzésének szabályszerűségét. Össze vethetjük ezt a 10-es számrendszerbeli számok képzésével.

A táblára ezt írjuk fel:

0	=	0
1	=	1
10	=	2
11	=	3
100	=	4

101 = 5
110 = 6
111 = 7
1000 = 8

Bevezetjük a bit fogalmát, ami egységnyi információ, nulla vagy egy értéket vehet fel. Például az 1000 bináris szám 4 biten ábrázolható. Ezen a ponton a shiftelést vagyis a kettővel való osztást és szorzást is meg lehet említeni.

Bináris számolás az ujjaink segítségével

Tanítsuk meg a gyerekeket az ujjaink segítségével binárisan számolni. Ezt mindenki egyszerre csinálhatja.

Kérdezzük meg, hogy meddig tudnak elszámolni az ujjakon?

Hogyan tudnának binárisan számolni a kezük segítségével?

Meddig tudnak binárisan számolni az ujjaink segítségével egy kezükön?

Itt elmondjuk, hogy a számítástechnikában az ujjakat biteknek nevezzük. 8 bit az egy byte.

Küldjünk e-mailt kettes számrendszerben!

Tervezett idő

15 perc

Az internetre telefonos modemmel kapcsolódó számítógépek a kettes számrendszert használják üzenetek küldésére. Mégpedig csipogással. A magas hangon csipogás az egyesnek felel meg, a mélyebb pedig nullának. Ezek nagyon gyorsan követik egymást, olyan gyorsan, hogy ebből mi mindössze folyamatos, szörnyű recsegést hallunk. Egy ilyen hangot lejátszhatunk a telefonunkon (<https://www.youtube.com/watch?v=gsNaR6FRuO0>)

Kihívunk 5 gyereket, és a kezükbe adjuk a nagy kártyákat a pöttyök számának növekvő sorrendjében. Az a feladat, hogy üzenetet küldjünk a gyerekeken keresztül, mint ahogy a modem is teszi.

Meg lehet kérdezni a gyerekeket, hogy szerintük hogyan tudunk üzenetet küldeni bináris számokkal. A megoldás, hogy a betűkhöz számokat rendelünk. Egy táblázatot kifüggeszthetünk a táblára, amely az egymáshoz rendelt betűket és számokat tartalmazza (A=1, B=2, ...stb).

Először az egyik tanár küld egy előre megbeszélte üzenetet (pl: szia), úgy hogy elcsipogja egyenként az üzenet betűinek bináris számkódját.

s = 19 = 10011
z = 26 = 11010
i = 9 = 01001

$$a = 1 = 00001$$

Minden csipogásnál az egyik gyerek fejére koppint, hogy tudja, hogy most neki szól a mély vagy magas hang.

Ezután a másik tanár egy üzenetet (hali) küld csipogással.

$$h = 8 = 01000$$

$$a = 1 = 00001$$

$$l = 12 = 01100$$

$$i = 9 = 01001$$

A gyerekek a csipogásnak megfelelően (alacsony hang = 0, magas hang=1) állítják a kezükben levő kártyákat, így kialakítva a bináris számot. A padban ülők segítenek a megfelelő betű megtalálásában, amit a fogadó a tanár a táblára ír.

Érdekességek

CD működése

A CD az információt bináris formában tárolja. A CD-n található árkok és az árkok hiánya segítségével kódoljuk a számokat, amiket számítógép olvas be.

Morze kódok

A bináris számok előtt más kódolást is használtak. Például morzejeleket. Itt is két jel volt, ti és tá. Ezekkel kódolták le az ABC-t.

Példaképpen felírhatjuk néhány betű morzejelét a táblára, vagy a kinyomtatott morze táblázatot kitűzzük a táblára.

Megmutatjuk az SOS-t.

Elmeséljük annak a katonának a történetét aki pislogással küldött morze üzenetet a feletteseinek, amikor fogságba esett.

Hármas számrendszer

Megkérdezhetjük a gyerekeket, hogy hány számjegy van a hármas számrendszerben.

Megmutathatjuk, hogy hármas számrendszerben (illetve kettes és tizesben is) ugyanazzal a technikával lehet a számokat leírni.

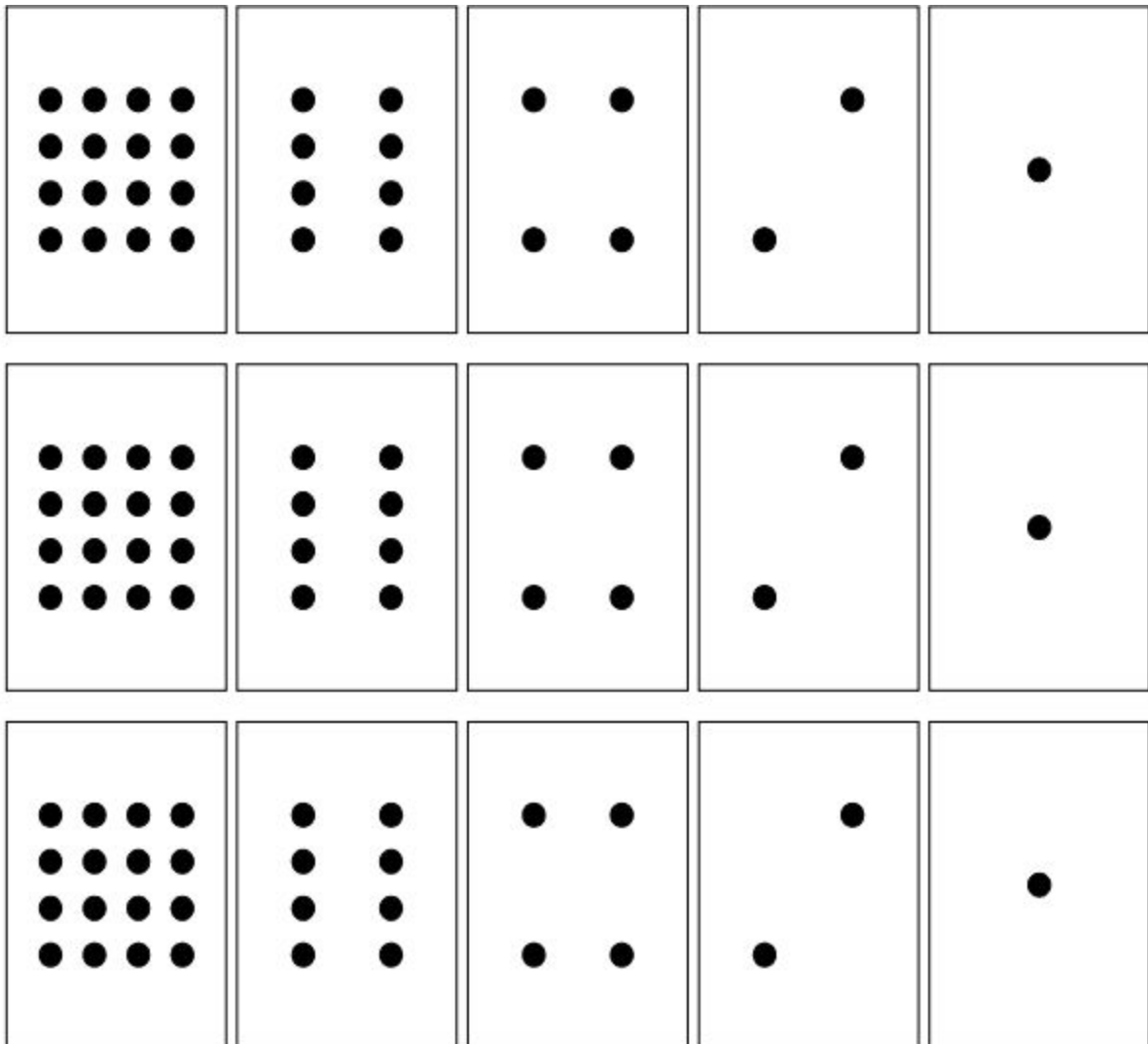
Felrajzolunk 5 négyzetet a táblára, amelyek a helyi értékeket ábrázolják. Ezekbe tudunk számokat írni.

Növeljük a számjegyet a legjobboldali helyiértéken. Ha növelés során kifogytunk a számjegyekből akkor váltsunk nullára és a tőle balra található helyiértéken növeljünk a számjegyeket, ha ott is kifogytunk a számjegyekből akkor váltsunk nullára, és a tőle balra

található helyiértéken növeljük addig, amíg nem jutunk el olyan helyiértékhez, ahol nem kell nullázni.

Ezután folytassuk megint a fentieket a legjobboldali helyiértéken.

Ha az idő megengedi, akkor tegyünk említést a 16-os számrendszerről. Valamint a 60-as számrendszerről is beszélhetünk (másodperc, perc).



Források

<http://csunplugged.org/binary-numbers/>

<http://csunplugged.org/wp-content/uploads/2014/12/1-binary-HU.pdf>

5. óra Rajzoljunk, hol a hiba? (Képképzés, hibafelismerés)

Szükséges eszközök:

- gyurmaragasztó (mágneskártya, kockás tábla és kréta) (36 mágnes 2500 Ft)
- kockás tábla
- kréta
- egy kép különböző nagyításai.

Bevezető

A számítógépek a rajzokat, fényképeket és más képeket pusztán számokat használva tárolják. A következő foglalkozás bemutatja, hogyan tudják ezt csinálni.

Képbábrázolás (15 perc)

Megkérdezzük a gyerekeket, hogy vajon a számítógépek (telefonok, tv-k) hogyan ábrázolják a képeket, amikor csak számokat nyelvének értnek.

Mutassunk be a gyerekeknek egy nyomtatott fényképrészletet különböző nagyításait. A legnagyobb nagyításban látszik, hogy a kép kis négyzetekből áll össze. Ezeket pixeleknek nevezzük (picture element).

Hogyan tudjuk ábrázolni a képet ha fekete fehér pixeleket használunk?

Ennek magyarázatához rajzoljuk fel az alábbi ábrát (a számok nélkül) soronként (táblára rajzoljuk vagy mágneses kártyákkal rakjuk ki).

	■	■	■		1, 3, 1
				■	4, 1
	■	■	■	■	1, 4
■				■	0, 1, 3, 1
■				■	0, 1, 3, 1
	■	■	■	■	1, 4

Egy pixel ábrázolásához egy bit-re van szükség. Vagy fehér vagy fekete a pixel (0 vagy 1). Bináris számokkal is ábrázolhatnánk a sorokat, de itt más módszert alkalmazunk.

A fenti ábrát soronként úgy tudjuk számokkal ábrázolni, hogy leírjuk, hogy egy színből hány összefüggő pixel van egy sorban. Fontos, hogy mindig a fehérrel kezdünk (ezért kezdődik a negyedik és ötödik sor nullával).

A fenti ábra melletti számokat kezdjük el felírni. Ebbe vonjuk be a gyerekeket is. Kérdezzük meg, hogy szerintük a színes képeket, hogyan lehetne ábrázolni. A színes képekhez a szín kódokat is el kell tárolni a hosszak mellett. Például az egyes a piros, kettes a sárga, stb...

Ez hasonlít a színező számokkal játékhhoz, csak itt négyzetrácsba helyezték a mezőket.

Fax gép (20 perc)

Először mondjuk el a gyerekeknek, hogy mi a fax gép, hogyan működik. Egy fax gép segítségével képeket küldhetünk tőlünk távol levő helyre telefonvonal segítségével. Az egyik oldalon egy küldő gép leolvassa a képet, és számokká alakítja. A fogadó oldalon a fax gép fogadja a számokat, és kinyomtatja a számoknak megfelelő képet.

Ezt úgy játsszuk el, hogy kiválasztunk egy küldő és egy fogadó gyereket. A többi gyerek viszi az információt. A számokat szállító gyerekek nem láthatják, hogy milyen képet szeretnének küldeni. Minden gyerek egy számot visz. Az első szám azt mondja meg, hogy egy sorban hány négyzet lehet maximum. Ezt fel kell rajzolni a fogadó gyerekeknek. A küldő gyerek egy papírról olvassa be a számokat, a fogadó gyerek egy táblára rajzolja a négyzeteket.

Konklúzió: ez a reprezentáció azért jó, mert kevesebb adatot kell átküldeni a telefon vonalon, mint ha minden pixelt sorban átküldtünk volna. Fekete fehér képen esetén bináris számokat is átküldhetnénk.

A telefonvonal zajos is lehet, ezért néha rosszul mennek át a számok. Ennek kivédésére használjuk a hibafelismerési módszereket. A zajt a faxos játék során "szimulálhatjuk" azzal, hogy próbáljuk megzavarni a gyerekeket.

Hibafelismerés (20 perc)

A következő foglalkozáson azt demonstráljuk, hogy hogyan veszik észre a számítógépek, ha egy képben hiba van, ha például a küldött képben valami megváltozott.

Ezt a játékot fekete-fehér mágneskártyákkal egy mágnes táblán vagy asztalon lehet játszani.

A játékhoz két tanárra van szükség: egy felügyelő tanár és egy hibafelismerő tanár.

A hibajavító tanár forduljon háttal, vagy menjen ki a teremből.

Két diák a táblára rakjon fel véletlenszerűen fekete-fehér kártyákat (egyik fele fekete, másik fehér, vagy két külön kártyacsomag) 5x5-ös négyzetrácsba. Amikor a kártyák felkerültek a helyükre, akkor a felügyelő tanár hozzáad még egy oszlopot jobbról és egy sort alul a négyzetrácshoz, úgy, hogy az 6x6-os legyen.

Ekkor bejön a hibafelismerő tanár, és memorizálja 5 másodpercig az állást, majd ismét kimegy/elfordul.

Egy gyerek fordítson meg egy kártyát ezután.

A hibafelismerő tanár ekkor kitalálja, hogy melyik kártya fordult meg.

Mi a trükk?

A felügyelő tanár úgy rakja fel az utolsó sort és oszlopot, hogy minden sorban és oszlopban páros számú fekete négyzet legyen. Így a hibajavító tanárnak csak meg kell számolni, hogy melyik az a sor és oszlop, ahol páratlan számú fekete négyzet van. A kettő találkozásánál lesz a megfordított kártya.

A játékot megismételhetjük többször is (a kártya megfordításától kezdődően), miközben a gyerekek próbálják kitalálni, hogy mi trükk. Ne felejtjük viszont minden alkalommal a megfordított kártyát visszafordítani.

A hibajavítási módszereket széles körben a számítástechnikában. Például: crc, paritás bit.

Források

<http://csunplugged.org/image-representation/>

<http://csunplugged.org/wp-content/uploads/2014/12/2-image-compression-HU.pdf>

<http://csunplugged.org/error-detection/>

<http://csunplugged.org/wp-content/uploads/2014/12/4-error-detection-HU.pdf>

6. óra Hol lehet...? (Keresési algoritmusok)

Bevezető

A számítógépektől gyakran várjuk, hogy információkat találjanak meg nagy adatgyűjteményekben. Ezt gyors és hatékony módon kell elvégezniük.

Ez a foglalkozás három különböző keresési módszert mutat be:

- lineáris keresés
- bináris keresés
- hashelés.

Szükséges eszközök:

- 10-30 papírlap
- 3-5 szótár
- Telefonregiszter.
- Cukorka

Lineáris keresés

1. Válasszunk ki önkéntes alapon egy gyereket, aki a keresést végzi. A maradék 9 gyereket állítsuk fel szemben a kereső gyerekekkel.

Minden gyerek kapjon egy lapot, rajta egy számmal egy és 100 között (véletlenszerű sorrendben) összehajtott papíron, hogy a gyerekek se lássák a számokat. A gyerekek nem tudják, hogy mik a számok. Válasszunk egy számot amit keresni kell. Érdemes ezt a számot a lehetséges számok hátsó negyedéből választani.

2. Adjunk a kereső gyerekeknek egy tárolót négy vagy öt cukorkával (ezt az összes keresési feladatban alkalmazzuk). A feladata az lesz, hogy megtalálja egy adott számot. „Fizetnie” kell, hogy megnézzen egy adott kártyát. Minden egyes választásnál húzzunk egy vonalat a táblára, így számolva a találgatásokat. Ha megtalálja a megfelelő számot, mielőtt felhasználná az összes cukorkát, megtarthatja a maradékot.

3. Ismételjük meg még egyszer a keresést egy másik gyerekekkel, ha szükséges.

4. Kérdezzük meg, hogy mi volt a stratégiájuk? Tárgyaljuk meg a legjobb és legrosszabb esetet. Hányszor kell találgatni a legjobb és legrosszabb esetben? A lineáris keresésnél sorban kell megnézni a számokat. A legrosszabb eset az, ha végig kell nézni az összes számot. A legjobb eset ha már elsőre eltaláljuk a megfelelő számot.

5. Magyarázzuk el, hogy miért lineáris keresésnek nevezik ezt a fajta keresést. Kérdezzük meg, hogy melyik a jobb módszer, ha sorban választunk, vagy ha összevissza? A sorban keresésnek az az előnye, hogy biztosan nem hagyunk ki semmit.

Mondjanak példát ilyen keresésre a való életből.

Bináris keresés (számok keresése)

1. Kérdezzük meg, hogy hogyan lehetne javítani a keresésen? Próbáljuk őket rávezetni arra, hogy ha a számok rendezve vannak, akkor könnyebb lenne a keresés.
2. Válasszunk megint egy önkéntes kereső gyereket, majd állítsuk fel a maradék 9 gyereket úgy, hogy a számokat sorban kapják meg. Válasszunk egy számot, amit keresni kell. Érdemes úgy felírni a számokat, hogy csak a végén találjuk meg a keresett számot.
3. Mondjuk a kereső gyereknek, hogy a középső gyereket válassza (számot). Ezután próbáljuk rávezetni a gyerekeket arra, hogy mi lesz a következő jó választás annak fényében, hogy rendezve vannak a számok. Az első szám kiválasztása után azok a gyerekek, akiknél biztosan nincs a keresett szám, leülhetnek. A maradék 4 gyerekből megint a középsőt választva leültethetünk további 2 vagy 3 gyereket.
4. Vegyük észre, hogy mennyivel kevesebb próbálkozás is elég volt a keresett szám megtalálásához. Mondjuk el, hogy miért hívják ez a fajta keresést binárisnak (bicikli, bikini)

Bináris keresés (szótár verzió)

Ehhez a foglalkozáshoz érdemes az elején néhány szót ejteni a szótárról, hogy mire valók és hogyan épülnek fel.

1. Vegyük elő a szótárakat és szervezzük csoportokba a gyerekeket, hogy minden 2 gyereknek legyen egy szótára.
2. A feladat az, hogy a gyerekek megtaláljanak egy adott szót (plate) a szótárakban. Ez a szó a lap tetején legyen megtalálható. Az oldalon belül már ne kelljen keresni. Minden csoport írja fel, hogy hány lapot kellett megnézni. Nem szabad lapozgatni, hanem fel kell ütni a könyvet egy adott helyen. Érdemes felírni, hogy melyik oldalon ütötték fel a könyvet vagy az ujjukat be kell tenni, hogy ne keveredjenek bele, hogy hol tartanak.
3. Kérdezzük meg, hogy ha lapon belül kellene keresni, akkor hogy keresnének? Bináris vagy lineáris keresést alkalmaznának-e?

Hash keresés

1. Vegyünk kilenc lapot, amire kilenc szót írunk fel úgy, hogy három különböző betűvel kezdődjenek. Ezután osszuk a gyerekeket három csoportba. Minden csoportot egy betűvel jelölünk és a gyerekeknek úgy kell beállni, hogy a megkapott szóval a megfelelő kezdőbetűt jelölő csoportba álljanak be (nem rendezve)
2. A kereső gyereknek el lehet mondani, hogy hogyan rendeződtek a gyerekek és el kell végezze a keresést.
3. Megkérdezhetjük a gyerekeket, hogy hogyan lehetne javítani a keresésen (rendezni a csoportokon belül).
4. Ezután kérjük meg a gyerekeket, hogy most a szavak utolsó betűje alapján csoportosuljanak. Majd ismét végezzük el a keresést. Érdemes beállítani úgy a szavakat, hogy most a csoportokban ne egyenletesen legyenek a szavak.

5. Kérdezzük meg a gyerekeket, hogy mi a legjobb és legrosszabb eset. Szerezzünk egy olyan telefonregisztert, ahol az egyes kezdőbetűket jelölő lapok különböző színűek. Ezzel jól el lehet magyarázni, hogy mire jó a hash-elés. A szótárban való lapozgatás a bináris és hash keresés kombinációja.

Források

<http://csunplugged.org/searching-algorithms/>

7. óra Sorakozó!!! (Rendezési algoritmusok)

Bevezető

A számítógépeket gyakran használják arra, hogy listákat sorba rendezzenek a lista elemeinek bizonyos tulajdonságai alapján. Például neveket ábécé szerint, találkozók vagy e-maileket dátum szerint vagy egyéb elemeket szám szerint. A listák rendezése azért jó, mert egy rendezett listában gyorsabban megtaláljuk, amit keresünk, és a szélsőértékeket könnyű észrevenni. Ha egy dolgot eredményeit sorrendbe tesszük, a legmagasabb és legalacsonyabb érték nyilvánvaló lesz. Ha rossz módszert választunk, hosszú időbe is beletelhet, míg sorba rendezünk egy hosszú listát, még gyors számítógépen is. Szerencsére sok gyors módszert ismerünk a rendezésre. Ezen a foglalkozáson a gyerekek megismerkedhetnek több rendezési módszerrel, és megfigyelhetik, hogyan tud egy intelligens módszer sokkal gyorsabban elvégezni egy feladatot, mint egy közönséges eljárás.

Szükséges eszközök

- 10-30 számozott papírlap (pl. album matrica)

A rendezési algoritmusokkal valamilyen tulajdonság szerint sorba tudunk rendezni elemeket. Ahhoz, hogy ezt demonstrálni tudjuk, az elemek a gyerekek lesznek és a gyerekek tulajdonságait használjuk összehasonlításra.

1. Első körben talán legegyszerűbben a magasság szerinti rendezéssel lehetne foglalkozni. Először megkérjük a gyereket, hogy álljanak sorba véletlenszerűen. Ezután megkérjük, hogy most álljanak sorba magasság szerint, a legalacsonyabbtól a legmagasabbig, balról jobbra. Valószínű, hogy ezt gyorsan meg tudják csinálni, ha van torna sor.
2. Most kérdezzük meg, hogy hogyan csinálták, mi volt a stratégiájuk?

Kiválasztó rendezés

A kiválasztó rendezés esetén kiválasztjuk valamely tulajdonság szerint a legkisebb gyereket és egy külön sorba állítjuk, majd kiválasztjuk a következő legkisebb gyereket és így tovább.

1. Minden gyereknek a kezébe adunk egy papírt amin egy szám van 1 és 100 között. A papírt nem mutathatják meg egymásnak csak összehasonlításkor. Majd véletlenszerűen sorba állítjuk őket. A feladat, hogy a kapott számok szerint álljanak sorba.
2. Először a legszélső gyerek odaáll a mellette levő elé, és összehasonlítják a számaikat. Akié a kisebb, az megy tovább a következő gyerekekhez. A végén a legkisebb számot fogó gyereket egy új sorba állítjuk.
3. Ezt a folyamatot addig ismételjük amíg minden gyerek átkerül a másik sorba.
4. A táblára mindig húzzunk egy vonalat, ha volt egy összehasonlítás. Minden iterációnál összeszámoljuk az összehasonlítások számát. Pár iteráció után már rá lehet őket vezetni a szabályra, mivel ki tudjuk számolni, hogy összesen hány összehasonlításra volt szükség. Egyre csökken az összehasonlítások száma.
5. Ezt a rendezést összevethetjük a torna sorba való rendezéssel. Kérdezzük meg a gyerekeket, hogy miért tudtak (ha) gyorsabban felállni magasság szerint.

Buborék rendezés

Ezt a rendezést sok gyerekkel is meg lehet csinálni, de mondjuk 6 és 10 gyerekkel tudna működni.

1. Minden gyereknek a kezébe adunk egy papírt amin egy szám van 1 és 100 között. A papírjaikat nem mutathatják meg egymásnak, csak az összehasonlítások alkalmával. Ezután véletlenszerűen sorba állítjuk őket. A feladat, hogy a kapott számok szerint álljanak sorba.
2. Balról jobbra haladva párokban helyet cserél két gyerek, úgy hogy a bal oldalon a kisebb, a jobb oldalon a nagyobb számot tulajdonoló gyerek legyen. Tehát először az első gyereket a másodikkal hasonlítjuk össze majd a másodikat a harmadikkal, harmadikat a negyedikkel, stb.. egész végig.
3. Ez egészet addig folytatjuk amíg lesz egy olyan kör amikor senki se cserél helyet.
4. Közben számoljuk meg, hogy hány összehasonlítás kellett elvégezni, mint ahogy azt a kiválasztó rendezésnél tettük.
5. A gyerekeknek el lehet mondani, hogy azért hívják ezt buborékrendezésnek, mert úgy kerülnek a helyükre, mint ahogy a buborékok mennek felfelé az üveg aljáról.
6. Megkérdezzük a gyerekeket, hogy hogy lehetne javítani az algoritmuson. Ez elég nehéz, de talán rá lehet őket vezetni, ha még egyszer megcsináljuk úgy a feladatot, hogy lássák a számokat. Ugyanis nem kell egészen végig csinálni a soron az összehasonlítást, csak a legelső nem jó helyen álló gyereket.
7. Kérdezzük meg a gyerekeket, hogy mi a legjobb és a legrosszabb eset. A legrosszabb eset az, amikor az elemek fordított sorrendben vannak rendezve.

Gyorsrendezés

1. Minden gyereknek a kezébe adunk egy papírt, amin egy szám van 1 és 100 között. Majd véletlenszerűen sorba állítjuk őket. A feladat, hogy a kapott számok szerint álljanak sorba. A papíron levő számokat nem mutathatják meg egymásnak, csak összehasonlításkor.
2. Kiválasztunk egy gyereket majd sorban minden gyerek összehasonlítja a számát a kiválasztott gyerekekkel. A kisebb számúak balra, a nagyobb számúak jobbra állnak. Érdekes úgy választani a gyereket, hogy a száma kb. a sor közepére essen.
3. Ezután az így keletkezett két csoportra ismét megcsináljuk ezt a kiválasztást és két csoportba osztást. Vigyázni kell, hogy a csoportok mindvégig egymás mellé kerüljenek, hogy a végén rendezett legyen a sor. Közben számolni kell az összehasonlításokat.
4. Ha megvan a rendezés, akkor meg lehet kérdezni, hogy mi a legjobb és mi a legrosszabb eset. A legrosszabb eset az, ha sorban választjuk ki a számokat és mindig az egyik csoport egy elemű.
5. Ismét összevethetjük az összehasonlítások számát a többi rendezéssel.

Források

<http://csunplugged.org/sorting-algorithms/>

8. óra Merre menjek? (Rendezési hálók, útvonalválasztás)

Rendezési hálók

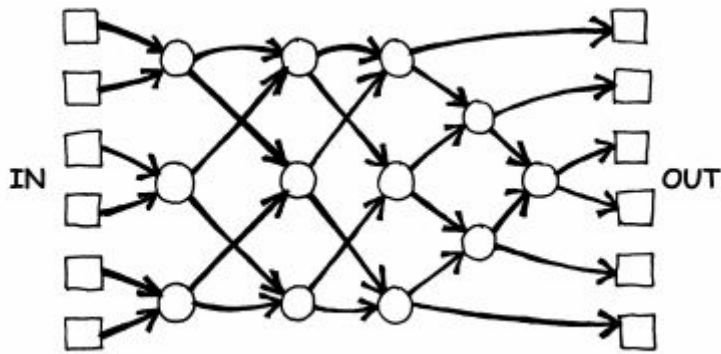
Bevezető

Annak ellenére, hogy a számítógépek gyorsak, sokszor ez se elég gyors, ezért több számítógépet vetnek be a rendezéshez. Ezt mutatják a rendezési hálózatok. Ez a módszer azt mutatja be, hogyan lehet sok számítógépet használni ahhoz, hogy rendezzünk egy listát.

Szükséges eszközök

- 30x30 cm-es négyzet és kör alakú lapok
- madzagok
- hálózati ábrák
- 6-10 gyerek

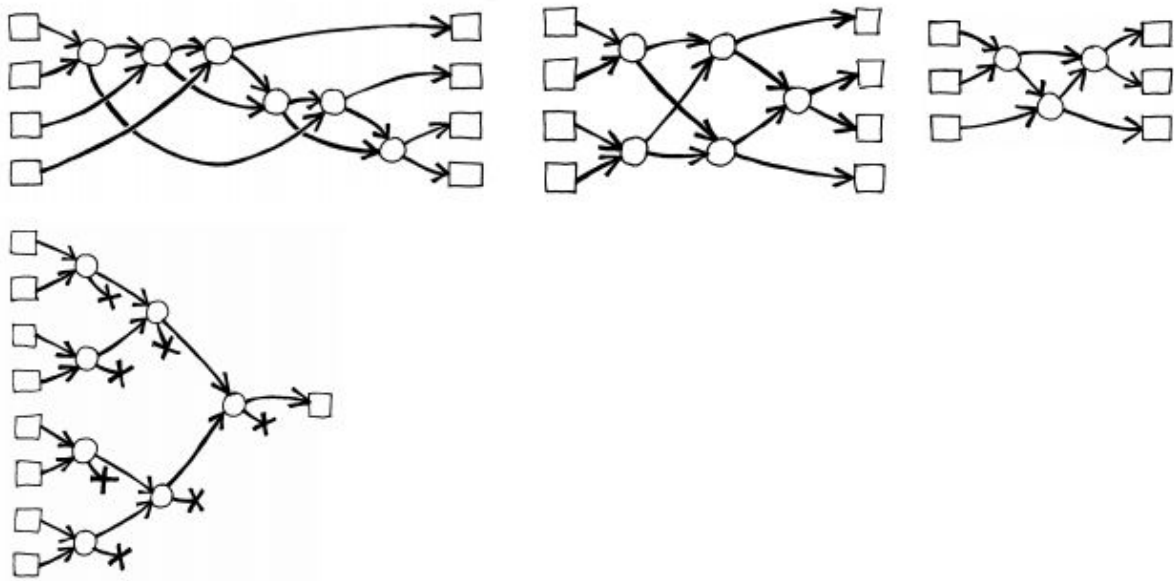
- stopperóra



1. Készítsük el a fenti hálózatot a földön papírlapokból és madzagokból.
2. Állítsunk fel 6 gyereket a fenti ábra szerint a bal oldalra a négyzetekre, kezükben számokkal. A sorrendjük véletlenszerű legyen. Megmutathatják a számokat azoknak a társaiknak, akik nem játszanak.
3. Ezután mindenki előrelép a körökre. Amikor két gyerek kerül egy körre, ott összehasonlítják a kezükben tartott számokat és akié a nagyobb az balra lép, aki pedig kisebb az jobbra. Fontos, hogy senkit se szabad hátrahagyni.
4. Addig ismételik a gyerekek az összehasonlítást amíg el nem érnek a jobb oldalon található négyzetekre. Ekkor fel kell mutatniuk a számaikat és ha mindent jól csináltak akkor rendezett sorrendben lesznek.
5. Kérdezzük meg a gyerekeket, hogy mi fog történni ha fordítva lépnek tovább a gyerekek, mi lesz a végeredmény?

Ezután rendezhetünk versenyt is az idővel, vagy ha van elég gyerek, akkor két csoporttal. A következő körben próbálkozhatunk nagyon nagy számokkal vagy a keresztnevek abc szerinti rendezésével. A keresztneves összehasonlítás azért jó, mert nem egyforma gyorsak lesznek a gyerekek, ezért be kell várniuk egymást.

Ezután kísérletezzünk más hálókkal is. A lenti ábrán van néhány példa. Jó, ha a gyerekek próbálnak kitalálni hálózatokat. Először három gyerekhez, majd négyhez találhatunk ki hálókat.



Próbáljuk meg kitalálni, hogy a mitől jó egy háló. Valójában az kell hozzá, hogy egy gyerek, minden végállapotba el tudjon jutni.

A végén pedig kitalálhatunk olyan hálót, ami a minimum és maximum számokat adja megoldásnak. Utóbbihoz az kell, hogy kisebb számok kiessenek.

Források

<http://csunplugged.org/sorting-networks/>
https://en.wikipedia.org/wiki/Sorting_network

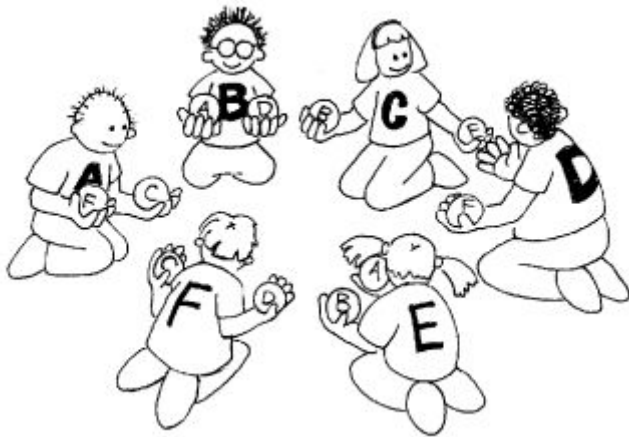
Útvonalválasztás és holtpont

Bevezető

Amikor sok ember van, akik ugyanazt az erőforrást használják (mint az autók az utakon vagy az interneten áthaladó üzenetek), holtpontok (deadlock) alakulhatnak ki. Valamiféle kooperációra van szükség, hogy ez ne következzen be.

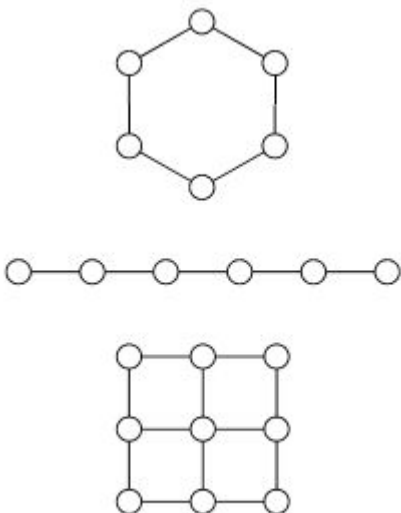
Szükséges eszközök

- 5 vagy több gyerek
- 10 db nagyobb papírlap, amire nagy számokat kell felírni, és a gyerekek nyakába lehet akasztani
- 2x10 darab kis kártya, amit a gyerekek egymásnak át tudnak adni, és elfér a kezükben. Minden szám két papírlapon szerepeljen 1-től 10-ig. Esetleg a papírlapok színesek is lehetnek. Minden nagy papírhoz tartozzon két kis papír ugyanolyan színben.



1. Ültessük vagy állítsuk a gyerekek körbe. Minden gyerek kapjon két kártyát véletlenszerűen. Ha 6 gyerek játszik, akkor az 1-től hatig szereplő kártyákat kell kiosztani. Valamelyik gyereknek csak egy kártyát osztunk ki.
2. A feladat az, hogy minden kis kártyát el kell juttatni a gazdájához, úgy, hogy csak a szomszédos üres kézbe lehet kártyát tenni.
3. Esetleg segíthetünk a gyerekeknek, ha nehezen boldogulnak. Illetve akik nem játszanak, azon is közbe szólhatnak. Először a mohó stratégiával kell próbálkozni, vagyis minden kártyát abba az irányba kell küldeni, ahol a gazdája van. Fontos hangsúlyozni, hogy a feladat összedolgozást igényel.

Ezután hozunk fel példát holtpontra. Például közlekedési dugók. Próbáljuk meg más halózzattal megoldani a feladatot.



Forrás

<http://csunplugged.org/routing-and-deadlock/>

9. óra Debuggolás

Debuggolás

Szükséges eszközök

20. db pohár
A4 papírok
ceruzák

Foglalkozás

A feladat során poharakat rakunk egymásra. Ez egy program végrehajtását jelenti. Először a poharak egymás tetején vannak fejjel lefelé egy tornyot alkotva. Ehhez képest a következő alaputasításokat használhatjuk:

- Egyet jobbra
- Fordíts
- Tedd le

pl: jobbra, tedd le, jobbra, jobbra, fordítsd, tedd le, fordítsd, jobbra, tedd le, jobbra, jobbra, tedd le.

1. Először 4 pohárral demonstráljuk, hogy hogyan működik az egész. Először egy alakzatot készítünk és a táblára írjuk nyilakkal a programot.
2. Ezután 2 darab 4 fős csoportot alkotunk. Az első feladat az lesz, hogy felírunk a táblára egy programot (6 pohárra) és végre kell hajtani:
 - a. jobbra, letesz, fordít, jobbra, letesz, jobbra, letesz, jobbra, jobbra, letesz, jobbra, fordít, letesz, jobbra, jobbra, jobbra, letesz.

A következő feladat egy sorjáték, két négyes csoportra osztjuk a gyerekeket. Tőlük távolabb felrakunk egy alakzatot poharakból. Két ugyanolyan alakzatot rakunk ki a két csapatnak. A feladat, hogy minden gyerek egy plussz utasítást írjon le egy papírra, ami az alakzat mellett található. Ehhez el kell olvasniuk az előző gyerek programját, fejben debugolni és hozzáírni egy új utasítást. A végén le kell futtatni a programokat, hogy lássuk, hogy helyes-e. Ha még van idő akkor a végén 6 poharas alakzattal is meg kell csinálni ezt.

10 óra. Évzáró utolsó óra

Ezen az órán nincs téma. Kötetlen beszélgetés lesz. A gyerekek kérdezhetnek minden félért, ami érdekli őket a számítógépekkel kapcsolatban vagy az elmúlt órákkal kapcsolatban. Mi is kérdezni fogunk. Jó lenne megtudni, hogy mi maradt meg nekik az órákból.

Megkérdezhetjük, hogy mi tetszett nekik a legjobban.

Esetleg lehetne nyomtatni egy papírt az órák anyagaival és pontozhatnák, hogy mi tetszett a legjobban.

Meg lehet azt is kérdezni, hogy melyik volt a legnehezebb anyag.

Az óra második részében egy igazi robotot hozunk be, amivel lehet egy kicsit játszani.

Esetleg adhatnánk nekik egy listát azon honlapokkal, ahol programozást tanulhatnak.