

# Programozási versenyekre készülés online lehetőségei

Erdősné Németh Ágnes

erdosne@blg.hu

ELTE IK

Batthyány Lajos Gimnázium

**Absztrakt.** A programozás tanításánál és a különböző programozási versenyekre való készülésnél a jól megválasztott, széles körben elérhető online gyakorlófelületek és versenyek tanár és diák számára is segítséget, változatosságot jelenthetnek. Az interneten elérhető lehetőségeket különböző szempontok szerint vizsgálhatjuk. Osztályozhatjuk az on-line forrásokat általánosan: kik vehetnek részt rajta, milyen nyelveken írhatnak programot, milyen gyakran van verseny, milyen hosszú, milyen nyelven érhető el. Megvizsgálhatjuk azt, hogy az adott forrás milyen típusú problémákkal, milyen szinten foglalkozik, milyen előzetes tudással lehet használni. Csoportosíthatjuk aszerint, hogy adnak-e segítséget a tanárnak ahhoz, hogy a diákjainak feladatokat jelöljön ki, megítélje ezek nehézségét, kapjon ötletet a megoldáshoz, lássa a gyerekek eredményeit. Besorolhatjuk különböző csoportokba az online versenyeket aszerint, hogy a diákok látják-e egymás megoldását, a memória- és futásidőlimitnek megfelelő algoritmus ismertté válik-e. Cikkemben a legerjedtebb nagy portálokat jellemzem és csoportosítom a fenti szempontok szerint.

## 1. Bevezetés

A középiskolás diákok számára rendezett nemzeti és nemzetközi informatikai versenyek nagy része - az IOI modellt követve - a diákok algoritmikus programozási képességeit méri és hasonlítja össze. Ez azt jelenti, hogy a gyerekeknek egy algoritmikus problémát kell megoldaniuk és legtöbbször implementálniuk valamilyen programozási nyelven. A megírt programot automatikus teszteléssel értékelik: bizonyos bemenő adatokra ellenőrzik, hogy a program ugyanazokat a kimenő adatokat adja-e, mint a mintamegoldás. Az automatikus értékelésnek vannak előnyei és hátrányai is a kézi javításhoz képest. Sokkal objektívebb eredményt tud adni, de kis hiba is a megoldás hibáságát okozhatja. A mintamegoldás sem biztos, hogy szisztematikusan feltérképezi az összes lehetséges esetet, nem a feladat problémájára adott kóddal is lehet (részlegesen) helyes eredményt elérni. Az értékelés nagymértékben az előre elkészített teszteseteken múlik. Az automatikus rendszer nem értékeli a kód szépségét, egyszerűségét, nem lehet feltérképezni vele a gondolatok nagyszerűségét, az algoritmus fantasztikusságát, főleg ha a kódolásba akár csak egy kis hiba is csúszott.

A diákokat az eredményes versenyzéshez először meg kell tanítani algoritmikusan gondolkodni, majd egyszerű programot írni bizonyos típusú alapproblémákra valamilyen programozási nyelven. Majd meg kell ismertetni őket a saját szintjükön használható, általános algoritmusokkal és azokat jól be kell gyakorolni.

Ezután fel kell készíteni arra, hogy egy adott feladatra koncentrálni megtalálják a lehető legjobb modellt és algoritmust, ami eleget tesz a hatékonysági megfontolásoknak. A kitalált algoritmust hibamentesen, a memóriakorlátokat is figyelembe véve le kell tudniuk kódolni valamilyen programozási nyelven a verseny szűkre szabott ideje alatt.

A versenyekre való felkészítésben a tanári instrukciókat, előadásokat az egyéni gyakorlások követik. Az otthoni és iskolai feladatmegoldás hatékonyabbá tehető az online gyakorlási lehetőségek kihasználásával, az online versenyeken való részvétellel a felkészítési-felkészülési folyamat részeként. A felkészítők munkájának kiegészítéseképpen, az internet adta lehetőségeket kihasználva, a diákok széles körből választhatnak-kaphatnak megoldandó feladatokat. A felkészítőknek ismerni kell a lehetőségeket, s mindenkinek az éppen aktuális tudásához, felkészültségéhez illő, motiváló feladatokat, online versenyt kell tudni ajánlani.

A cikkben az online versenyek Pohl (2006) [1] és Combéfis at al.(2014) [2] osztályozását kiegészítem néhány új szemponttal, amik - a lehetőségeket felkészítőként nézve - fontosak lehetnek. Tanárként nyomon kell követnem, hogy melyik feladatokkal, milyen eredménnyel foglalkozott a diák. Tudnom kell válogatni a feladatok közül mind nehézségben, mind megoldási módszer tekintetében. Olyan gyakorlási lehetőséget kell mutatnom, amit a Cuba-Ricardo (2014) [3] osztályozása szerinti saját tanulási stílusában használhat a diák a feladatok megoldása során.

S nem utolsósorban az online versenyek a másokkal való kommunikációra, ismerkedésre is használhatóak a blogokon, fórumokon keresztül, kitérve a világot, a sikeres munka örömét adva.

Az algoritmikus programozás típusú informatikai versenyekre készítő online lehetőségek közül néhányat az előzőleg feltérképezett szempontok alapján osztályozok és értékelek, kitérve a versenyzési és a tanulási lehetőségekre is.

## **2. Programozás tanítása**

Az informatikai versenyeken való részvételhez nagy szükség van jó minőségű oktatásra és rengeteg gyakorlási lehetőségre. Tanár és diák álma, hogy mindig pontosan annyi információt kapjon a diák, amennyire szüksége van. Mindig pontosan olyan feladaton kelljen dolgoznia, ami már épp egy kis erőfeszítést igényel tőle, de nem megoldhatatlan számára. Fontos lenne, hogy mindig a saját tempójában tudjon haladni, azaz a tananyag pontosan rá legyen szabva. Tizenöt-harminc fős osztályteremben, de még a kisebb létszámú szakkörökön is a gyerekek lakhely és kor szerint vannak válogatva, meglehetősen nehézé téve a tanár munkáját, hogy hatékonyan, testre szabottan taníthasson minden gyereket a csoportban, bármilyen tantárgyról is legyen szó.

Az algoritmikus gondolkodás és a programozás tanításánál nagyon hatékonyan lehet használni és kihasználni a számítógép adta lehetőségeket. Mivel az algoritmikus gondolkodás fejlődése egyénenként nagyon különböző, így a tanári instrukciók és a számítógépes gyakorlás kombinációja hatékonyabbá, testre szabottabbá teheti a folyamatot.

Cél, hogy a gyerekek a saját tempójukban, saját igényeikre szabva gyakorolhassanak és haladhassanak az anyagban. Csak akkor léphessenek tovább, amikor már éppen jól begyakorolták az aktuális tudnivalót. Egyénektől és anyagrésztől függően ehhez több-kevesebb gyakorló feladatra van szükség. Fontos, hogy a diákok a tanulási folyamatban sok visszajelzést kapjanak, hibáikra rögtön fény derüljön. Szükséges, hogy a tanulási folyamatban mindvégig motiváltak legyenek, legyen sikerélményük. Ha megoldottak egy feladatot, annak a megoldása elemezhető legyen. Lehessen segítséget kérni a tanártól vagy a többi diaktárstól, kiterjesztve ezt az igényt arra, hogy az éppen ugyanazt a feladatot megoldóval meg lehessen vitatni a megoldást és az elkövetett hibákat.

A fenti igények kiszolgálására kerestem online számítógépes támogatást, s meglehetősen sok lehetőséget találtam az interneten. Attól függően, hogy mi a célja a tanulásnak és hogy milyen szinten van a diák tudása, más és más online gyakorlási lehetőség és versenylehetőség létezik.

Az online versenyek és gyakorlófelületek használatának fő céljai és hatásai:

- az automatizált lehetőségeken keresztül kiegészítik-helyettesítik a tanár munkáját,
- nagy létszámú résztvevőt hatékonyan tudnak kiszolgálni,
- segítik a programozás tanítását,
- gyakorlási lehetőséget adnak,
- válogatott, kipróbált, jó minőségű gyakorló feladatokat tartalmaznak,
- automatikus javítást és korrekt visszajelzést adnak,
- meg lehet nézni mások megoldásait, kódjait,
- fejlesztik a programozási és algoritmizálási képességeket,
- fejlesztik az önismeretet,
- rövid versenyeken keresztül az idővel való gazdálkodást tanítják,
- online kapcsolatokat tesznek lehetővé egymástól távol élő, de ugyanolyan érdeklődésű emberek közt.

A programozási versenyekre való felkészítéshez a fentieknek megfelelő gyakorló és versenyfelületeket kerestem. Az így talált és a saját gyakorlatomban már használt online rendszerek közül néhányat részletesen vizsgálok a cikk további részében.

Először POHL (2006)[1] és COMBEFIS at al.(2014) [2] osztályozási szempontjait veszem sorra, majd ezeket a szempontokat kiegészítem néhány újjal. Az így kapott szempontok alapján egy jellemzést adok az általam vizsgált rendszerekről. Mindegyiknél kitérek arra, hogy mikor, milyen tudásszinten, milyen célra használható az adott felület.

### **3. Online programozási versenyek és gyakorlófelületek osztályozásának szempontjai**

Ebben a fejezetben Combefis at al (2014) [2] nyomán összeírom az online versenyek osztályozási szempontjait. A továbbiakban ezek a kritériumok szolgálnak a versenyek összehasonlítására. Ezek a szempontok könnyebbé teszik a választást akkor, amikor versenyzési lehetőséget keresünk a különböző tudásszinten levő diákjaink számára. Az általános szempontokat Pohl (2006) [1], a megoldandó feladatokkal kapcsolatos szempontokat Combefis at al(2014) [2] munkája nyomán írtam össze. Ezeket kiegészítettem a saját, segítő tanár nézőpontjából írott szempontokkal és a gyakorlási lehetőségek használatával kapcsolatos, illetve az online felületen levő oktatóanyagokkal kapcsolatos szempontokkal.

### 3.1. Általános szempontok

Az első kritérium (I1), hogy az adott verseny feladatait egyedül kell megoldania a versenyzőnek vagy engedélyezve van a csoportmunka.

A második szempont (I2), hogy a versenyt korhoz, nemhez vagy tanulmányi évhez, iskolai osztályhoz kötik-e.

A harmadik szempont (I3) a versenyen elfogadott programozási nyelvek listája.

A negyedik általános szempont (I4) a verseny időpontja és a verseny hossza. A verseny időpontja lehet kötött vagy egy bizonyos időszakon belül szabadon választható – tekintettel az időzóna eltérésekre. A verseny hossza az az időintervallum, ami alatt a versenyzők megismerhetik a feladatokat, megoldást adhatnak rá és azt beküldhetik. Lehetnek rövid és hosszú versenyek. Rövid verseny, amikor pár óra alatt (3-5 óra) megadott számú feladatot kell megoldani, míg hosszú versenyen erre több nap, hét is rendelkezésre áll.

Az ötödik általános szempont (I5), hogy milyen gyakran rendezik az adott versenyt. Nyílt versenyeknek nevezzük azokat a versenyeket, amik állandóan elérhetőek, s a verseny akkor kezdődik, amikor a felhasználó úgy dönt, hogy most kezd neki. Ezek a nyílt versenyek használhatóak gyakorlási lehetőségek is.

A hatodik szempont (I6) ebben a körben az, hogyan számolják ki a versenyző által elért pontszámot a beküldött megoldásai alapján, s az hogyan befolyásolja a többi versenyző közt elért helyezését. Itt figyelhetjük meg, hogy számít-e a megoldott kód beküldésének ideje, hibás kód feltöltéséért jár-e pontlevonás.

Az általános szempontokat kiegészítem egy újabbal (I7), hogy egy adott versenyen elért eredmény beleszámít-e egy, több versenyből és gyakorlási lehetőségből álló versenysorozatba és befolyásolja-e az illető helyezését. Létezik-e egy általános rangsor az adott versenyen. E szempont alapján vannak olyan versenyek, amik csak egy adott alkalomra szóló rangsort építenek, míg mások a folyamatos munkával elért eredményt is külön rangsorolják. Ez utóbbiaknál a versenyző folyamatos munkája, fejlődése is nyomon követhető, ami meg is jelenik az adott honlapon.

Nyolcadik szempontként (I8) vizsgálnám, hogy a feladatok szövege milyen nyelven érhető el.

### 3.2. Megoldandó feladatokkal kapcsolatos szempontok

Ebben a csoportban az első szempont (T1) az, hogy a versenyzőnek milyen típusú megoldást kell készítenie. Ez lehet egy program forráskódja, egy futtatható állomány, lehet egy szöveges dokumentum, amit a versenyző egy saját gépen futtatott programmal állított elő, de lehet egy adott szó, szám vagy intervallum is, amit a felületen egy adott helyre kell beírni.

A második szempont (T2), hogy milyen típusú a feladat: egy adott specifikációjú függvényt kell írni, vagy modellezni és megoldani kell egy adott problémát, mesterséges intelligenciát kell tervezni, ami például egy kétszemélyes játékot játszik az értékelő rendszer megfelelő programja ellen.

A harmadik szempont tartalmazza (T3) azokat a korlátozásokat, amiknek a beküldött programnak meg kell felelnie: futási idő limit, használható tárigény limit, feltölthető megoldások száma, a feltölthető program mérete.

A negyedik szempont (T4), hogy feltöltés után a versenyző milyen visszajelzést kap a munkájáról. Lehet, hogy semmit, lehet, hogy egy hibaleírást (fordítási hiba, futási hiba, rossz válasz, memóriatúllépés, futási idő túllépése, ...)

Az ötödik szempont (T5), hogy a versenyt részekre osztották-e például résztvevők nemzeti-sége, kora vagy tudásszintje alapján. Ha az adott verseny csak bizonyos nemzetiségi feltételeknek, vagy iskolai osztályba járóknak van versenyszerűen meghirdetve, míg mások esetleg részt vehetnek rajt rangsorolás nélkül. Például a nemzeti diákolimpiai válogatóversenyeken általában csak az adott ország diákjainak szól a válogatás, s őket külön is rangsorolják.

Hatodik szempontnak (T6) itt felvenném, hogy a verseny után elérhető-e valamilyen formában a használandó megoldás, algoritmus leírása, illetve elérhető-e egy vagy több mintaprogram, például a versenyzők által feltöltött megoldások.

Egy verseny ajánlhatóságához nagymértékben hozzátartozik, hogy mennyire kiszámítható a feladatok nehézségi szintje (T7), témakörei, a használandó algoritmusok köre. Ugyanis ezektől függ, hogy egy adott versenyen való részvételre biztatok-e tanárként gyereket vagy sem.

### 3.3. Szempontok gyakorlás céljára

Tanári szempontból az előző felsorolásban nagyon sok fontos szempont előkerült már, de van néhány, ami még fontos lehet, ha az online felületet programozás tanítására és gyakorlásra használhatjuk.

Az első szempont, hogy a weblap ad-e további gyakorlási lehetőséget (E1), akár a már lezajlott versenyek feladataival, akár egyéb feladatokkal.

Második szempont (E2), hogy a gyakorlófeladatok kategorizálva vannak-e téma illetve nehézségi szint szerint.

Harmadik szempont (E3), hogy beküldhetőek-e online értékelésre a feladatmegoldások.

Negyedik szempont (E4), hogy valamikor elérhetőek-e a tesztadatok, akár csak offline értékeléshez, akár a megoldás beküldése után.

Ötödik szempont (E5), hogy van-e az adott oldalon oktatóanyag, hozzá válogatott feladatokkal. Használható-e az oldal egyéni tanulásra.

És gyakorlófelületként nézve sok szempont újra előkerül:

Ilyenek (E6) a gyakorlásként elérhető feladatok korlátai., a feladatok nyelve, a használható programozási nyelvek (ha mások, mint a versenyek nyelve), a visszajelzés minősége, a használandó algoritmus leírása, mintamegoldás elérhetősége.

A hetedik szempont (E7) az, hogy tanárként látszik-e a diákjaim által elért eredmény, követhető-e a fejlődésük, ellenőrizhető-e a kódjuk.

Online versenyeken attól függően, hogy mi a célja a versengésnek, más-más szempontra kell fókuszálni. Ugyanígy, ha gyakorlófelületként használjuk az adott portálokat, akkor is más-más kritériumot kell nézni.

Például:

- ha a csapatmunkára vagyunk kíváncsiak, akkor az I1 kritérium értékét kell megnézni,

- ha diákjaink csak bizonyos nyelveken értenek, akkor az I8 szempont meghatározó,
- ha csak egy programozási nyelven tudnak programozni, akkor mindenképpen szükséges az I3 értéke,
- új programozási nyelv tanulásakor a I3 értéke fontos a gyakorlófeladatok keresésekor,
- az I2 szempont besorolását mindenképpen ellenőrizni kell, ha komolyan akarunk versenyezni egy adott versenyen, mind kor, mind tudásszint, mind nemzetiség szempontból,
- E2 nagymértékben megkönnyítheti a tanár munkáját, amikor diákjainak feladatot ad, s a feladtválogatáshoz az E6 is segíthet.

## 4. Online versenyek osztályozása

Az általam vizsgált online versenyek és gyakorlási lehetőségek mindegyike algoritmusorientált és feladatot kell megoldani rajtuk: konkrét algoritmust kell (kreatívan) alkalmazni, majd azt egy programba kódolni (T2). A versenyzőnek egyedül kell dolgoznia a megoldáson (I1).

Nyilvánosan elérhetőek az interneten, egy regisztráció után szabadon használhatóak a világ bármely részéről. Csak olyanok kerültek a listára, amiknél az értékelés automatikus. Minden általam kategorizált versenyt és portált elsősorban középiskolások számára készítették (I2).

### 4.1. Project Euler

A Project Euler honlapján olyan érdekes matematikai és számítógép programozási feladatokat találunk, amiket számítógép segítségével, kis programokat írva oldhatóak meg, végiggondolva a bennük rejlő matematikai tudnivalókat.

A megoldás egy érték beírása a honlapon (T1), a szabadon választott nyelven (I3) írt programot a felhasználó a saját gépén futtatva jut eredményhez. Klasszikus értelemben nem verseny (I4, I5), hanem gyakorlófelület. Nincs pontszám (I6, I7), az előrehaladást a megoldott problémák számasságával jellemzik. A problémák az angolon kívül még kínaiul, németül, koreaiul, románul, oroszul és spanyolul is elérhetőek (I8), megoldhatóak.

A megírt programra semmiféle korlátozás nincs (T3), hiszen az a felhasználó saját gépén, akár órákon át futva is adhat eredményt. Rossz eredmény feltöltése után csak egy egyszerű hibaüzenet jön, hogy az eredmény helytelen (T5), egy piros X kíséretében. Helyes eredménynél egy zöld pipa a jutalom. Semmilyen további segítség, minta, instrukció nem érhető el (T6).

Az egész honlap egy gyakorlás (E1), a feladatok nincsenek kategorizálva (E2). Következtetni lehet a helyes megoldások számából a feladat nehézségére. Viszont a helyes megoldás beküldése után egy nagyon részletes, programkóddal és matematikai számításokkal alátámasztott indoklás-hoz juthatunk (E6).

Az oldalon 486 feladat található. Részletes statisztikákat nézhetünk a felhasználókról és a feladatokról.

Diákjainkat egy lapra fel tudjuk venni barátként, láthatjuk, hogy melyik feladatokat oldották meg (E7), s akár helyi versenyt hirdethetünk közöttük.

Kiválóan alkalmas a gyorsabban haladók lekötésére, a motiváció fenntartására, érdekes problémákkal való találkozásra, az érdeklődés felkeltésére.

## 4.2. Croatian Open Competition in Informatics

A horvátok felkészítési versenye a diákolimpiai válogatóra, verseny formájában. A versenyen résztvevőkre nincs semmiféle korlátozás (I2), az utolsó, válogató fordulón a horvát versenyzőket külön is kezelik (T5).

Használható programozási nyelvek (I3): Python, Pascal, C, C++ és Java.

Évente 6+2 vagy 7+1 forduló van, havonta, előre meghirdetett időpontban. Minden verseny háromórás (I4, I5). Minden forduló hat feladatot tartalmaz, a regionális és az országos diákolimpiai válogató három feladatot.

Minden feladatnál szerepel, hogy milyen tesztadatokra mennyi pontot érhet a helyes megoldás. A beküldés időpontja a verseny időtartama alatt nem befolyásolja a helyezést (I6). A versenyeknek nincs összesített eredménye (I7). Angolul és horvátul érhető el (I8).

A versenyzőnek forráskódot kell feltöltenie a szerverre (T1), melyet a rendszer azonnal, csak a feladatoknál megadott adatokra ellenőriz. Minden feladatra egyedileg van memória és futási idő limit (T3). Egy feladatra akárhányszor tölthető fel megoldás. Fordítási vagy futási hiba esetén nem fogadja el a megoldást (T4). Mivel ilyenkor csak kis adatokra tesztel, így memória és futási idő limit nem fordul elő.

Nagyon hasznos, hogy a verseny után (T6) elérhetőek az értékelések, a többiek forrásfájljai, az értékeléshez használt tesztállományok és a megoldás algoritmusának leírása.

Megbízhatóan lehet tudni (T7), hogy az első két feladat csak alapalgoritmust használ ötletesen, a második kettőben dinamikus programozás, mohó algoritmus, gráfeljárások szerepelhetnek nagyon kreatív módon, érdekesen és az utolsó két feladat diákolimpia szintű. Azaz nagyon széles a feladatok nehézsége, a kezdőknek is örömet okoz az első két feladat megoldása, míg a diákolimpiára készülőknél is munkás végigcsinálni.

Később elérhetőek az előző évek feladatsorai, tesztfájljai, a beküldött megoldások és a hivatalos megoldás algoritmusának leírása. Sajnos a verseny végeztével nem lehet online feltölteni a feladatokat, viszont elemezhetőek a tesztadatok, s a saját számítógépen tesztelhető is a megoldás.

A versenyt kilencedik éve rendezik, így már kb. 370 érdekes, jól használható, megbízható feladatot érhető el, tesztadatokkal és megoldási útmutatóval együtt, angolul.

## 4.3. Sphere online judge

Lengyel feladatarchívum, online értékelő és versenyhelyszín. Versenyeket semmilyen feltételhez nem kötik (I2). A legszélesebb spektrumot adja az elfogadott programozási nyelvek tekintetében (I3). A legtöbb feladat a következő ötven programozási nyelv valamelyikén oldható meg: ADA, DOC, ASM, AWK, BASH, BF, C, C#, C++ 4.0.0-8, C++ 4.3.2, C99 strict, CLPS, CLOJ, LISP sbcl, LISP clisp, D, ERLang, F#, FORTGO, HASKell, ICON, ICK, JAR, JAVA, JS, LUA, NEM, NICE, NODEJS, CAML, PAS fpc, PAS gpc, PDF, PERL, PERL 6, PHP, PIKE, PS, PRLGPYTH 2.7, PYPY, PYTH 3.2.3, PYTH 3.2.3 n, PY3.4, RUBY, SCALA, SCM qobi, SCM guile, SED, ST, TCL, TECS, TEXT, WSPC.

Van három nyílt verseny, régi feladatokból (I4, I5). Sajnos az utóbbi években nem indítottak új online versenyeket, így a honlap gyakorlási célokat szolgál. A versenyzők akármikor beküld-

hetnek a versenyekre és a problémákra is megoldást. Mindenkinek megnézhető a profilja, ahol az általa megoldott feladatokból alakul ki egy rangsor (I6, I7).

A feladatok szövege az angolon kívül örményül, lengyelül, portugálul és vietnámiul érhető el (I8).

Megoldásként program forráskódját (T1) kell feltölteni vagy az online felületen szerkeszteni és beküldeni. Minden feladat végén megtalálható a feladatra vonatkozó programkód, memória és futási idő korlát (T3). Beküldés után részletes hibajelzést kapunk (T4). A rendszer az összes tesztadatra rögtön teszteli a megoldást (E3), így a teljes pontszám látható. 5167 feladatnál elérhető a megoldás forráskódja (T6), míg a honlapon levő 21581 feladat nagy többségének nem. Minden feladat után fórum üzemel, ahol a felhasználók több-kevesebb részletességgel kitérgetik az adott feladatot.

Középiskolás, diákolimpiás, illetve egyszerű problémákat tartalmazó nyílt versenyt hirdettek a honlapon. Ez a kategorizálás segít a feladatok témájának és nehézségének megítélésében (T7). Új eszközként egy problémaosztályozó eszköz (SPOJ Problem Classification Tool) is elérhető az oldalon, mellyel adott megoldási módszerhez tartozó feladatokat lehet keresni. A feladatokat négy kategóriába is besorolták (E2), illetve a nehézséget abból is meg lehet állapítani, hogy milyen nagy versenyről származik a feladat. A feladatok elve és a tesztadatok fájljai nem érhetőek el (E4). Versenyként névre lehet keresni az eredménylistában (E7).

Ha feladatsorokba tudjuk rendezni a gyerekeknek a megoldandó problémákat, akkor jól használható gyakorlásra.

#### 4.4. Timus Online Judge

A legnagyobb orosz programozási feladatbank, automatikus értékelő rendszerrel. Az uráli régió összes versenyének és IOI felkészítőjének a feladatait gyűjtötték egy helyre. A régióban zajló versenyeken online módon részt lehet venni bárkinek (I2) a honlapon keresztül. A C/C++, Pascal, Java, C#, Go, Python, Ruby, Haskell és Scala programozási nyelvek valamelyikén lehet feltölteni a megoldást (I3). Rövid, három-öt órás versenyeket rendeznek (I4), havi rendszerességgel (I5). A feladatokra a megoldás módszere szerint előre megadott pontot lehet kapni, a beküldési idő versenytől függően számíthat (I6). A rangsort a megoldott feladatok pontszámainak összege alapján számolják (I7), így alakul ki egy ranglista. A feladatok szöveg angolul és oroszul érhető el (I8).

A versenyeken résztvevőknek forráskódot (T1) kell feltölteni, minden feladatnál adottak a memória és futási idő korlátok (T3). Feltöltés után részletes visszajelzést és pontszámot kap a versenyző (T4). Az éles versenyeken a helyben versenyzőket máshogyan is rangsorolják (T5). Sajnos a megoldás és az algoritmus nem érhető el a verseny után (T6).

A versenyek után is feltölthetőek a megoldások, folyamatos online gyakorlási lehetőséget adva (E1). A gyakorlófeladatokat versenyek szerint és megoldási módszer szerint is kategorizálják, s minden feladathoz egy nehézségfaktort is számolnak (E2). A feladatok nehézsége a megoldók számából és az új felhasználók megoldásának idejéből számolódik minden éjjel. Folyamatosan beküldhetőek a feladatok online értékelésre (E3), viszont a tesztadatok nem érhetőek el (E4). Nincs oktatóanyag a honlapon, viszont a jó kategorizálás miatt egyéni tanulásra is használható (E5). A versenyzők felhasználónevét keresve részletes elemzést kapunk a megoldott feladatról (E7).



A honlapon 1035 nehéz, IOI szintű, megoldási módszerekre és nehézségre is kategorizált érdekes feladat található.

#### 4.5. USA Computing Olympiad

Az Egyesült Államok informatikai diákolimpiára készítő portálja és válogatóversenye, minden évben hat versennyel, melyek három nehézségi szinten zajlanak. A legjobban teljesítő 24 amerikai diák vehet részt ezután a felkészítő táborban, s közülük a legjobb négyet választják ki a diákolimpiára.

A verseny nyílt, mindenki bronz fokozaton kezd, majd ha egy versenyen elért 500 pontot, akkor juthat a következő szintre (I2). A versenyen C, C++, JAVA, Pascal és Python nyelven lehet feltölteni megoldást (I3). A versenyek havonta következnek (I5), alkalmanként 3-5 feladatot kell megoldani 3-5 órás időtartam alatt, három napos időablakban.(I4). A beküldött megoldást értékelik, az azonos pontszámot elért versenyzők ugyanolyan helyezést kapnak (I6). Az egyes versenyeken elért pontok befolyásolják a következő versenyt, a pontszámok a honlapon lévő rangsorhoz összeadódnak (I7). A feladatok nagy része angolul, oroszul, spanyolul, franciául, grúzul, vietnámiul, törökül, bolgárul is elérhető (I8).

A versenyzőknek programkódot kell feltölteni (T1), s tesztetenként 1 másodperces futási idő limittel kell futnia (T3). Feltöltés után csak hibajelzést küld a rendszer (T4), míg részletes elemzést a felhasználó a verseny után kap. Külön értékeli a legjobb amerikai megoldókat (T5). A verseny után viszont letölthetőek a tesztesetek és a részletes elemzése a megoldásnak, C++ programkóddal. Az egyes szintekhez tartozó feladatok nehézsége jól meghatározott (T7).

A honlapon több száz órányi, ingyenesen elérhető oktatóanyag van, online gyakorlási lehetőséggel. A tréninganyagok egy része angol, arab, spanyol, görög, magyar, perzsa, orosz, vietnami nyelven is elérhető. Az anyagok egy gyakorlott feladatmegoldó szintről indulnak, s csak rendszeres olvasással és feladatmegoldással lehet bennük továbbhaladni. A gyakorlófeladatok jól kategorizáltak (E2), beküldhetőek értékelésre a megoldások (E3). Nagyon részletes, tesztesetekre lebontott értékelést ad vissza a rendszer (E4). Részletes analízist kapunk, C++ mintaprogrammal. Az oldal jól használható egyéni tanulásra (E5). A diákok tevékenysége nem látszik (E7).

Erős haladó szintről indulva nagyon jól használható tanulásra, gyakorlásra, versenyzésre.

#### 4.6. UVa Online Judge

Az UVa Online Judge rengeteg latin-amerikai, spanyol és ázsiai és nyugat-európai versenynek ad online helyszínt. Több ezer feladatot tartalmaz, s nagyon részletes, jól kidolgozott felkészítő anyagot, rengeteg gyakorló feladattal.

Az egyes versenyeknek megvannak a maguk pontosan körülírt szabályai, így nem lehet őket pontosan kategorizálni.

Közös, hogy C, JAVA, C++, Pascal nyelven egy felületre lehet beírni a programkódot (T1, I3). Havonta néhány rövid (3-5 órás), nyílt versenyre kerül sor (I4, I5). A versenyzők pontszámát a honlapon adott rangsorpontjához adják (I6). A feladatokat a honlapon csak angolul találtam meg (I8).

A szokásos memória és futási korlátoknak kell megfelelnie a programoknak (T3), viszont a versenyző az eddigi portáloknál tapasztaltnál sokkal részletesebb hibajelzést kap a feltöltött

megoldásáról (T4). Minden verseny nyílt, de az adott helyen lakókat külön is rangsorolják (T5). Az egyes versenyfeladatok részletes megoldása nem érhető el (T6).

A felhasználók részletes tevékenységprofilja megtalálható, a megoldott feladatok listájával együtt (E7).

Több programozás könyv gyakorlófeladatai vannak fenn a honlapon, így az online értékelés után a feladatok és módszerek magyarázata valószínűleg ezen könyvekben érhetőek el.

Több ezer feladat található itt, online értékeléssel, versenyek és könyvek szerint sorba rendezve.

#### 4.7. CodeChef

Az indiai Codechef az egyik legnagyobb az algoritmusok, programok és programozási versenyek világában. Minden hónap egy hosszú, kéthetes versennyel kezdődik, majd a hónap közepén egy rövid, háromórás megmérettetéssel folytatódik (I3, I4). Ezen kívül helyet ad az elsősorban ázsiai regionális programozási versenyeknek. Lehet rajta zártkörű és nyílt versenyt is rendezni (I2). Már több, mint 400 versenyt bonyolított 2009 óta.

A versenyek megszabhatják az elfogadott programozási nyelveket, de a portál képes a következő nyelveken megírt kódok kezelésére: ADA, ASM, BASH, BF, C, C99 strict, CAML, CLOJ, CLPS, CPP 4.3.2, CPP 4.8.1, CPP11, CS2, D, ERLang, FORTran, FS, GO, HASK, ICK, ICON, JAR, JAVA, JS, LISP clisp, LISP sbcl, LUA, NEM, NICE, NODEJS, PAS fpc, PAS gpc, PERL, PHP, PIKE, PRLG, PYTH, PYTH 3.1.2, RUBY, SCALA, SCM guile, SCM qobi, ST, TCL, TEXT, WSPC.

A versenyeken a helyezés kiszámításakor számít, hogy mennyi idő alatt készült el az adott program (I6). A versenyzők pontszáma az általuk megoldott feladatok pontszáma alapján alakul ki (I7) a globális rangsorhoz. Minden verseny elérhető mandarin kínai nyelven, angolul és oroszul (I8).

A versenyeken forráskódot kell feltölteni (T2), s feladattól függ a futási és memórialimit (T3). Feltöltés után a versenyző egyszerű hibatípus jelzést kap (T4), pontszámmal együtt. Verseny után az összes versenyző kódja elérhető (T6).

A weblapon minden versenyfeladat megjelenik gyakorlófeladatként is (E1). A gyakorlófeladatok nehézségi szint szerint csoportosítva találhatóak a honlapon: könnyű, közepes, nehéz, kihívás, extrém és iskolai feladatok csoportokba (E2). A gyakorló feladatok beküldhetőek online értékelésre (E3). A tesztesetek nem érhetőek el (E4), viszont minden feltöltött megoldás kódja látszik (E5), így az oldal egyéni használatra is alkalmas. Részletes hibajegyzék, tesztesetekkel, segítséggel, mintakóddal érhető el a feladat megoldása után. Az egyéni tanulást a WIKI-be rendezett tutorialok széles választéka, a blogok és a fórumok is segítik. Az online tevékenységről, a megoldott feladatokról felhasználónév alapján részletes jelentés érhető el (E7).

A leggyorsabban növekvő portál, rengeteg versennyel, lehetőséggel a programozás tanuláshoz.

## 5. Összefoglalás

A programozás tanítása során használt különböző szintű algoritmusok gyakorlásához rengeteg feladat található online felületeken, gyakorlási ciklusokba rendezhetően, időnkénti motiváló versenyekre szabdalhatóan. A tudásszintnek megfelelően választott gyakorlási és versenyzési lehetőségek a diákokat motiválttá teszik, a tanár munkáját segítik. Az automatikus értékelés, a korrekt, jó minőségű visszajelzés révén tehermentesítik a felkészítőt.

Az online felületek és versenyek használata különböző tanulási-tanítási tevékenységek kiegészítéseképpen (fejlesztik az algoritmizálási és programozási képességeket, az önértékelést, sikerélményt adnak és örömet okoznak, így) jól használhatóak az oktatás során.

## 6. Irodalom

1. Pohl, W. (2006). Computer science contests for secondary school students: approaches to classification. *Informatics in Education*, 2006, Vol 5. (1), 125–132.
2. S. Combéfis, J. Wautelet (2014). Programming Trainings and Informatics Teaching Through Online Contests. *Olympiads in Informatics*, 2014, Vol. 8, 21–34.
3. G. Cuba-Ricardo, P.A. Leyva-Figueroa, L.L. Mendoza-Tauler (2014). Learning Strategies of Informatics Contestants. *Olympiads in Informatics*, 2014, Vol. 8, 35–48.
4. A. Charguéraud, M. Hiron (2008). Teaching Algorithmics for Informatics Olympiads: The French Method. *Olympiads in Informatics*, 2008, Vol. 2, 48–63.
5. Forišek, M. (2013). Pushing the boundary of programming contests. *Olympiads in Informatics*, 2013, Vol. 7, 23–35.
6. M. Hiron, L. Février (2012). A Self-Paced Learning Platform to Teach Programming and Algorithms. *Olympiads in Informatics*, 2012, Vol. 6, 69–85.
7. CodeChef home page.  
<http://www.codechef.com/>
8. Croatian Open Competition in Informatics, Internet online contest series.  
<http://hsin.hr/coci>
9. IOI home page.  
<http://www.ioinformatics.org/>
10. Project Euler home page.  
<https://projecteuler.net/>
11. Sphere online judge home page.  
<http://spoj.pl/>
12. Timus Online Judge home page.  
<http://acm.timus.ru/>
13. UVa Online Judge home page.  
<http://uva.onlinejudge.org>
14. USA Computing Olympiad home page.  
<http://www.usaco.org/>