Hallgatói teljesítményértékelés az algoritmikus gondolkodás tükrében

Pluhár Zsuzsa[1], Torma Hajnalka[2], Törley Gábor[3]

{[1]pluharzs, [2]thajni, [3]pezsgo}@inf.elte.hu
ELTE IK

Absztrakt. Az informatikai gondolkodás (computational thinking) egyik alapköve az algoritmikus gondolkodás. Ennek mérésére, fejlesztésére több irányzat is létezik. Egyik igen komoly és sokrétű elgondolás a számítógép nélkül, a gondolkodási sémákon keresztül történő megközelítés. Ilyen a nemzetközi Bebras kezdeményezés is. Kutatásunkban az Eötvös Loránd Tudományegyetem Informatika Karának angol nyelvű BSc-s hallgatóin keresztül arra kerestük a választ, hogy az első szemeszter tantárgyainak hatására mennyire változik a hallgatók algoritmikus gondolkodása, illetve az alapozó programozást támogató (Programming) kurzuson nyújtott teljesítmény milyen összefüggéseket mutat ezen változásokkal. A felméréshez a Bebras kezdeményezés feladatainak segítségével készítettük el felmérésünket, melynek alapjait és elsődleges eredményeit a cikkünkben foglaltuk össze.

Kulcsszavak: algoritmikus gondolkodás, programozás oktatás, felsőoktatás

1. Algoritmikus gondolkodás

Az informatikai gondolkodás (Computational Thinking) kifejezést Jeanette Wing[1] definiálta újjá és terjesztette el , majd 2010-ben újragondolta, és megalkotta a ma talán leggyakrabban idézett definíciót:

> „*Computational Thinking is the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent*" [2/1.o.]

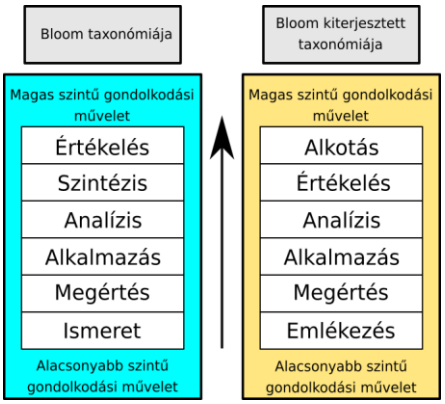
Wing és az őt követő kutatók definíciói ugyan nem egységesek, de az általuk megfogalmazott meghatározások sokszínűségében bizonyos területek egyaránt kiemelkednek. Ilyen a logikus gondolkodás, a problémamegoldás, az algoritmikus gondolkodás, az elemzés, rendszertervezés, általánosítás, és az egyik legfontosabb elem az absztrakció képessége, amely lehetővé teszi a komplex problémák megoldását is.

Selby[3] értelmezésében a következő képességek szükségesek az informatikai gondolkodáshoz: absztrakt fogalmakban gondolkodás, részekre bontás a gondolkodás során, algoritmikus gondolkodás, értékelésben való gondolkodás és általánosítás képessége a gondolkodás során. Selby[3] és munkatársainak munkássága azért is jelentős, mivel kimondták, hogy ahhoz, hogy értékelni lehessen, hogy mennyire fejlődött a tanulók informatikai gondolkodása, először fontos az informatikai gondolkodást alkotó elemek azonosítása.

1.1. Az algoritmikus gondolkodás és szintjei

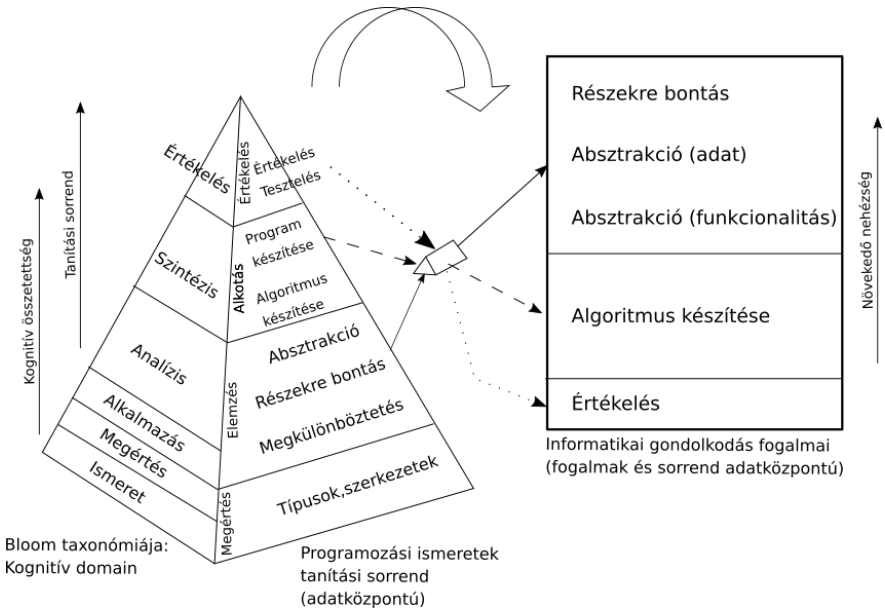
Az algoritmus egy probléma megoldásához vezető lépések sorozata. Az algoritmusok jelen vannak a hétköznapjainkban, de algoritmusokat használunk informatikai problémák megoldásakor is. Az algoritmikus gondolkodás kompetenciája egy fontos informatikai kompetencia, definiálására, szintekre bontására több törekvést ismerünk. Zsakó és Szlávi [4] szerint az algoritmikus gondolkodás szintjei az algoritmus felismerése, megértése, az algoritmus végrehajtása, az algoritmus elemzése, algoritmus alkotása és az algoritmus megvalósítása. Ez a felosztás összhangban van Bloom oktatási célokat osztályozó taxonómiájával [5], illetve a módosított Bloom-i taxonómiával (1. ábra). A kiter-

jesztett Bloom-i taxonómiában [6] a kognitív folyamatok dimenziója az emlékezés, megértés, alkalmazás, elemzés, értékelés és alkotás szintekből áll. Jelen munkánk során a Bloom kiterjesztett taxonómiájában található kognitív szinteket vettük alapul.



1. ábra: Bloom eredeti és kiterjesztett taxonómiája

Selby az informatikai gondolkodás fejlesztése kapcsán a programozás oktatásával is foglalkozott. A Bloom-i taxonómiát vetítette le a programozás és az informatikai gondolkodás elemeinek oktatására. Az általa készített ábrákon a tanulási sorrendre koncentrált, mivel lehetnek olyan elvárások, amelyeknek egy tanuló aktuális tudásállapotban nem képes megfelelni a kognitív összetettség miatt, ugyanakkor az informatikai gondolkodásban esetlegesen egyszerűbb lehet a felsőbb szintű művelet, mint egy alsóbb szintű. [7] A 2. ábra azt mutatja, hogyan viszonyulnak egymáshoz Bloom taxonómiájának kognitív szintjei, az (adatközpontú) programozásoktatás tanítási sorrendje, és az informatikai gondolkodás fogalmainak tanítási sorrendje. Az ábrán jól látható, hogy a programozás-oktatás és az informatikai gondolkodás fogalmainak tanítási sorrendje nem egyezik meg.



2. ábra: Bloom taxonómiája, a programozási ismeretek és informatikai gondolkodás fogalmainak tanítása közötti kapcsolat

2. Algoritmikus gondolkodás fejlesztése

Az informatikai gondolkodás, illetve ezen belül az algoritmikus gondolkodásának fejlesztésére több kezdeményezés is kialakult. Ezek közül három főbb csapásirányt különböztethetünk meg:

Az első valamely kiemelt részterület más ismeretkörbe való beépítésével, általában projektmunka keretében épít be bizonyos célzott tevékenységeket.

A második irányzatban az informatikai gondolkodást programozás oktatásával, szimulációs játékok alkalmazásával fejlesztik és vizsgálják. Ezek között találhatunk új fejlesztéseket játékok, keretrendszerek tervezésével és megvalósításával, vagy meglévő alkalmazások integrálását [pl. 8, 9, 10].

A harmadik megközelítés a programozástól, esetleg a számítógéptől is elszakadva konkrét aktivitásokon keresztül valósítja meg a fejlesztést [pl. 11, 12, 13, 14, 15, 16].

2.1. Hód

Ezt a harmadik megközelítést tartja szem előtt a nemzetközi Bebras kezdeményezés [13] és ennek magyar megvalósulása a Hód [14], melynek informatikai gondolkodással kapcsolatos fejlesztés melletti célja, hogy

- felkeltse az érdeklődést az informatika iránt;
- feloldja az informatikával kapcsolatos félelmeket, negatív érzéseket;
- megmutassa az informatika területének sokszínűségét, felhasználási lehetőségeit és területeit.

A kezdeményezés elsődleges terepe egy verseny, melyhez a feladatokat a nemzetközi csoport dolgozza ki - és melyek megoldásához csak strukturált és logikus gondolkodásra van szükség, semmilyen különleges informatikai tudás nem szükséges a megválaszolásukhoz. A feladatok érdekes problémákat mutatnak be. Nem tesztek, inkább szórakoztató gondolkodtató feladványok, melyek át- és továbbgondolásával új ismeretekre tehetnek szert a résztvevők, illetve meglévő ismereteiket mélyíthetik el.

A feladatok előkészítését, pontosítását a nemzetközi csapat egy műhelykonferencia keretein belül végzi, kiválogatva és módosítva a résztvevő országokból beküldött kérdéseket. Ezután az egyes országok honosítják a kérdéseket és a náluk megrendezett versenyhez testre szabják azokat.

Magyarországon az ELTE Informatika Karával 2011-ben csatlakoztunk a kezdeményezéshez, és azóta évről évre megszervezzük a megmérettetést.

A versenyen szereplő kérdések egy jelentős része algoritmikus gondolkodással kapcsolatos feladatokat rejt magában.

2.2. Programozás oktatása

Az ELTE Informatikai Karának angol nyelvű Computer Science BSc képzésében részt vevő hallgatók az első félév során vesznek részt a Programming elnevezésű kurzuson. A kurzus 2 óra előadást és 3 óra laborgyakorlatot foglal magában hetente. A tantárgy, a magyar nyelvű megfelelőjéhez hasonlóan, programozás-módszertani bevezetésként szolgál. A kurzus célja az alapvető problémamegoldó algoritmusok megismerése és használata különböző egyszerű és összetettebb adatszerkezetekkel. A megoldandó problémákra elsődlegesen egy algoritmus-leíró nyelven vár el megoldást, majd ezt követi az algoritmus kódolása C++ programozási nyelven. Olyan általános sémákat igyekszik a hallgatók számára elérhetővé tenni, amely segítheti őket a problémák megértésében, a problémák részekre bontásában, az algoritmusok megalkotásában, majd végül a programok elkészítésében.

A korábbi évek tapasztalata alapján az angol nyelvű BSc képzésben résztvevő hallgatók számára a kurzus nem egyforma nehézségű, sokaknak már az egyszerű algoritmusok megértése, felhasználása, majd később nehezebb feladatokban történő alkalmazása problémákat okoz. A bonyolultabb algoritmusok, illetve a komplexebb bemeneti adatstruktúrával rendelkező problémák pedig óriási

kihívást jelentenek számukra. Ezek a tapasztalatok visszavezetnek Selby [7] megállapításaihoz az informatikai, illetve algoritmikus gondolkodás szintjeinek és a tanítási nehézség kapcsolatáról (lásd újra 2. ábra).

3. Kutatási célok és eszközök

Kutatásunkban célul tűztük ki, hogy megvizsgáljuk, mutatkozik-e kapcsolat a hallgatók BSc képzésünkbe való belépésekor mutatott algoritmikus gondolkodási szintje, illetve a tárgy teljesítése/teljesíthetősége között.

Ha van valamilyen kapcsolat, akkor az informatikai gondolkodás mely dimenziói szükségesek a sikeres hallgatói előmenetelhez, illetve tapasztalható-e változás a hallgatók informatikai gondolkodásában az egyetemi tanulmányaik előrehaladtával. További vizsgálati kérdés, hogy milyen egyéb tényezők hatnak a teljesítési szintekre: kimutatható-e bármilyen kapcsolat az angol nyelv ismerete, az előtanulmányok, illetve a különböző gondolkodási szintek megléte között.

3.1. Kérdőív

A vizsgálathoz és a változások követéséhez elő- és utókérdőív változatot készítettünk. Az összekötésekhez a hallgatói azonosítókat használtuk, de a kérdőívben egyértelműen leírtuk, hogy azok eredménye semmilyen módon nem befolyásolja a kurzuseredményeiket.

Az előkérdőív tartalmazta a szükséges háttérváltozókra vonatkozó kérdéseinket. Önbevallás alapján felmértük, hogy a hallgatók milyennek értékelik a saját angol nyelvtudásukat, illetve, hogy milyen sokat/mélyen tanultak algoritmizálást, algoritmus-elméletet. Rákérdeztünk továbbá néhány programozási nyelv/környezet (C, C++, PHP, Javascript, HTML5/CSS, Java, Scratch) előismeretének mértékére. A nyelvek kiválasztásának fő szempontja az volt, hogy általában ezek a legelterjedtebb nyelvek a közoktatásban, valamint a fenti nyelvekkel fognak találkozni a hallgatók az egyetemi képzésük folyamán. Az önértékeléshez 5 fokú Likert-skálát állítottunk össze a választható fokozatokhoz. Emellett rákérdeztünk a nemükre és nemzetiségükre.

Az előkérdőív második felében és az utókérdőívben a Hód kérdésekből [17] válogattunk a kiterjesztett Bloom taxonómiának megfelelő szintenként 2-2 kérdést. A kiválasztás során arra törekedtünk, hogy az életkori és nehézségi paraméterek minél közelebb álljanak a résztvevőkhöz, az idősebb korosztálynak (10-12. osztály) megfelelő szintekről a nem egyértelműen nagyon könnyű feladatok közül kerültek ki a feladatok. Válogatási szempontot jelentett az is, hogy a különböző országokból, kultúrákból érkezők számára a feladatban szereplő szituációk, történetek ne jelentsenek előnyt vagy hátrányt a megoldás során.

Szint	Előkérdőív	Utókérdőív
Alkalmazás (applying)	Számfordító (2012-HU-01a) Pöttyök és parancsok (2013-SK-09)	Az utókérdőív kérdései még nem publikusak annak lefolytatásáig.
Analizálás (analyzing)	Alakzatjáték (2016-CA-09) Véletlen képek (2013-DE-02)	
Szintézis (evaluating, synthesis)	Játék a golyókkal (2016-IT-02B) Ebéd (2015-DE-06)	
Alkotás (Creating)	Fogpiszkálós (2017-HU-06) Kalózvadászat (2015-SI-07)	

1. táblázat: A kérdőívben szereplő feladatok és bloom-i kognitív domain szintjeik

A kérdéseket Google űrlapként készítettük el. Nem használtuk ki az általános kiértékeléseket, illetve a kitöltők nem kaptak visszajelzést a megoldásuk helyességéről.

Általánosan meghagytuk azt a lehetőséget, hogy négy lehetséges válasz közül kelljen kijelölniük az általuk helyesnek ítéltet. Ez több célt szolgált: az egyik, hogy minél pontosabban össze tudjuk majd vetni a feladatok megoldásainak sikerességét a Hód versenyeken elért eredményekkel. Az egyes „rossz" megoldások megadásánál törekedtünk arra, hogy téves gondolkodási folyamatokat ábrázoljanak, és így ezeket is ki tudjuk értékelni, nyomon tudjuk követni. Továbbá a nyelvi nehézségeket a megfogalmazások területén és az elgépelések lehetőségét is igyekeztünk így a minimálisra csökkenteni.

4. Elsődleges eredmények

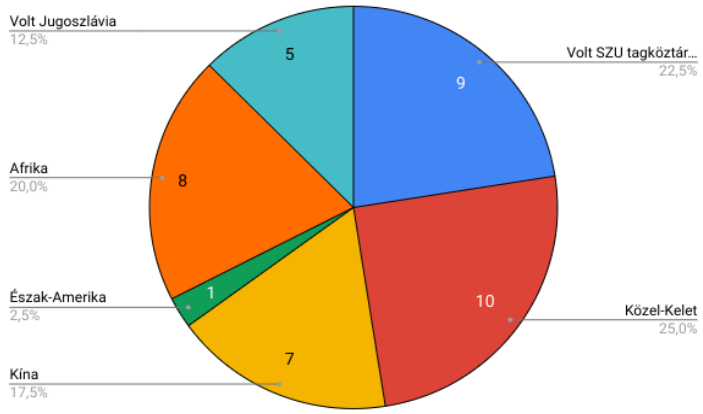
A kutatás jelenlegi fázisában az előkérdőívek kitöltésén vagyunk túl. Az ehhez köthető statisztikai eredményeket foglaljuk össze a következőkben.

Az előkérdőív kitöltéséhez nem szabtunk időkorlátot és önálló munkaként, úgymond bárhonnan kitölthették a kérdőívek a szemeszter első 3 hetében.

4.1. Minta

Az előkérdőívet a Programming alapozó kurzuson résztvevő hallgatók 60%-a (N=42) töltötte ki. A külföldi hallgatói arányainkkal megegyezően 12 nő és 30 férfi.

A hallgatók nemzetiségének eloszlását az 3. ábra mutatja.

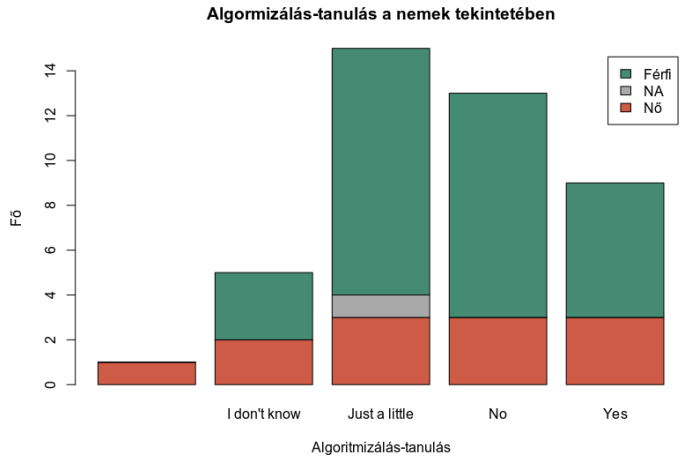


3. ábra: A mintában szereplő hallgatók nemzetiségei

4.2. Háttérváltozók alakulása

Angol tudását tekintve a hallgatók 7%-a erősebb alapfoknak, 32%-a közepesnek, 59%-a jónak értékelte azt.

A megkérdezett hallgatók mindössze 21%-a állította magáról, hogy tanult korábban algoritmusokról, és 41%-uk azt, hogy nem tudja vagy egyértelműen nincs előismerete a témát illetően (4. ábra). Kimondhatjuk, hogy a programozás elméleti alapjainak tanulása a megkérdezett hallgatók jelentős többsége számára új tananyag.

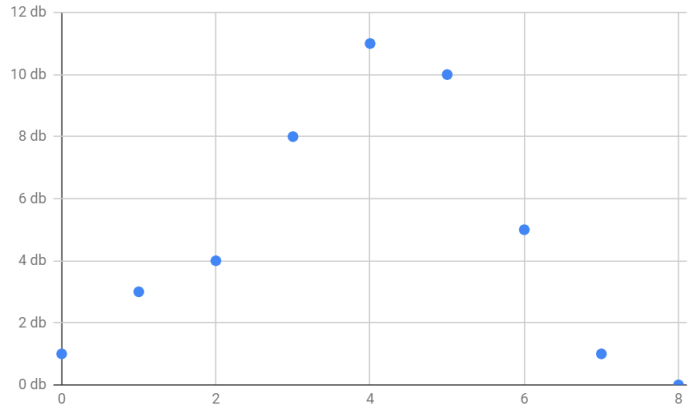


4. ábra: A hallgatók válaszai a korábbi algoritmizálás-tanulásról a nemek eloszlásában

A programozási nyelvek és környezetek területén a C++ nyelv ismerete volt a hallgatók számára a legerőteljesebb: 30%-uk vallotta, hogy segítséggel képes egyszerű programot írni benne, és 25%-uk, hogy képes önállóan programot is írni ezen a nyelven, de csupán 5%-uk nevezte magát képzettnek. A C és a HTML nyelvek kerültek a második legismertebb kategóriába úgy, hogy a hallgatók 30%-a nem is hallott róluk és 33%-uk éppen csak említés szintjén. A legismeretlenebb nyelv a PHP volt, a hallgatók 84%-a nem is hallott róla.

4.3. Hód kérdések eredményei

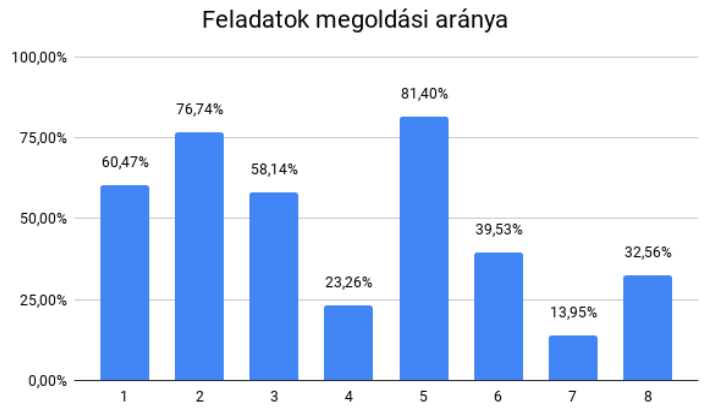
A hallgatóknak minden alkategóriából kettő, azaz összesen nyolc feladatot kellett megoldaniuk. Helyes válasz esetén ezekre 1-1 pontot adtunk, így maximum nyolc pontot lehetett elérni. A hallgatók átlagos pontszáma 3,86 volt. A hallgatók 72%-a csak 2, 3 vagy 4 pontot szerzett.



5. ábra: Hallgatók által elért pontok számának alakulása

A feladatok nehézség szerint ugyan közel azonosként kerültek kiválasztásra, de az elért eredményeket tekintve (ld. 5. ábra) megállapíthatjuk, hogy a 4. (2013-DE-02 Véletlen képek) és a 7. (2017-HU-06 Fogpiszkálós) feladat nehezebbnek bizonyult. Ezeket a feladatokat hallgatók kevesebb, mint negyede tudta megválaszolni. Mindkét feladat esetében kiemelhető egy jellemzően választott (hallgatók 41%-a), helytelen megoldási stratégiát mutató helytelen válasz. A „Véletlen képek” feladatban

egy teljesen egyértelműen megfelelő válasz került kiválasztásra. Ennek oka lehet az, hogy a feladat egy tagadással kérdezett rá a megoldásra („melyik ábra NEM nyomtatható…”), amit a hallgatók nem vettek figyelembe.

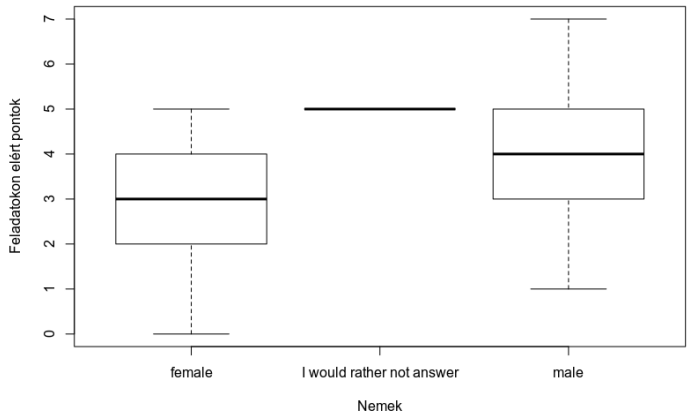


6. ábra: Az egyes feladatok megválaszolásának alakulása

Az 5. feladat kivételével egyértelműen meghatározható, hogy az algoritmikus gondolkodás komplexebb szintjeihez tartozó feladatok általában nehezebbeknek bizonyultak.

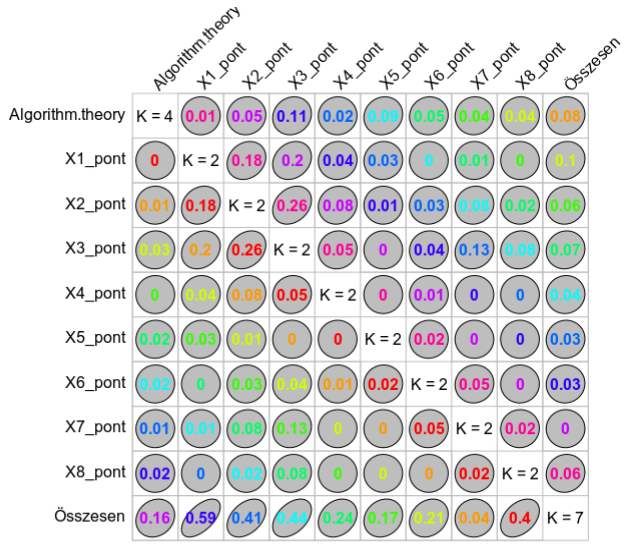
4.4. Összefüggések

Az előkérdőívben szereplő háttérváltozók és az algoritmikus gondolkodást vizsgáló kérdések közötti összefüggés-vizsgálatok csak a nem esetében mutattak gyenge összefüggést (lásd 7. ábra). A férfiak jobb eredményt értek el, mint a nők (p=0.03, átlagok különbsége: 1,16).



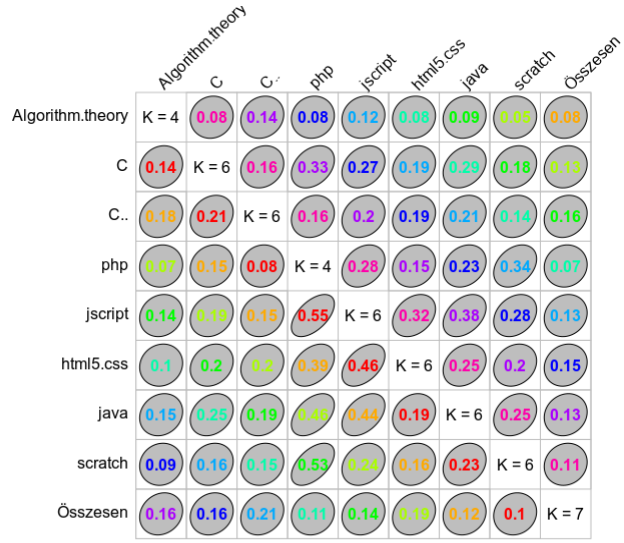
7. ábra: A nemek és az elért összpontszámok összefüggése

Az algoritmizálás korábbi tanulására adott válaszok nem állnak összefüggésben a kapott pontszámokkal (8. ábra).



8. ábra: Az algoritmus-tanulás és az elért összpontszámok összefüggése (Goodman-Kruskal tau számítás)

A korábban tanult nyelvekre adott válaszok nem állnak összefüggésben azzal, hogy valaki tanult-e algoritmus-elméletet, illetve hány pontot ért el az algoritmikus gondolkodás szintjét mérő feladatokon (9. ábra).



9. ábra: Az algoritmus-tanulás, a tanult nyelvek és az elért összpontszámok összefüggése (Goodman-Kruskal tau számítás)

Összességében elmondható, hogy az előkérdőív alapján a vizsgált változók és az elért pontszám között összefüggés nem mutatható ki.

5. Konklúzió

A felmérésünk célja az volt, hogy képet alkothassunk az ELTE Informatikai Karának Programming kurzusára járó hallgatók előképzettségéről, és a belépéskori algoritmikus gondolkodás szintjéről. Az előkérdőív alapján a vizsgált változók és az algoritmikus gondolkodást felmérő feladatokra kapott pontszám között összefüggést nem találtunk.

Jövőbeli terveink szerint összevetjük az algoritmikus gondolkodást felmérő feladatokon elért pontszámokat a tanulók kurzusteljesítési adataival. Továbbá az utókérdőív kitöltését követően elemzéseink kiterjeszthetőek lesznek, mélyebb összefüggés vizsgálatokat is el tudunk majd végezni. A jelenlegi felmérés lezárásával az eredmények összehasonlíthatóak lesznek a Hód verseny eredményeivel.

A későbbiekben érdemesnek tartjuk a vizsgálatunkat több kérdéssel is kiegészíteni az egyes algoritmikus gondolkodási kategóriákban – ezzel erősítve a kapott eredményeket. Illetve a kapott mutatók tekintetében kimondottan algoritmikus gondolkodásra irányuló tevékenységekkel bővíteni a programozó képzésünket.

Irodalom

1. Wing, J. (2006). Computational thinking. Commun. ACM, 49, 33-35.
2. Wing, J. (2011). Research Notebook: Computational Thinking - What and Why? The Link. Pittsburgh, PA: Carneige Mellon. - elérhető: https://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf (utoljára megtekintve: 2018.10.15.)
3. Selby C. C. (2013) Computational Thinking: The Developing Defenition. Submitted for ItiCSE Conference 2013. - elérhető: http://people.cs.vt.edu/~kafura/CS6604/Papers/CT-Developing-Definition.pdf (utoljára megtekintve: 2018.10.15.)
4. Zsakó, L., Szlávi P. (2010): Informatikai kompetenciák: Algoritmikus gondolkodás. InfoDidact 2010. - elérhető: https://people.inf.elte.hu/szlavi/InfoDidact10/Manuscripts/ZsL_SzP.htm (utoljára megtekintve: 2018.11.07.)
5. Bloom, B.S., Krathwohl, D. R. (1956) Taxonomy of Educational Objectives: The Classification of Educational Goals, by a committee of college and university examiners. Handbook I: Cognitive Domain. NY, NY: Longmans, Green
6. Anderson, L. W., Krathwohl, D. R., et al (Eds..) (2001) A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives. Allyn & Bacon. Boston, MA (Pearson Education Group)
7. Selby C. C. (2015) Relationships: computational thinking, pedagogy of programming, and Bloom's Taxonomy. In Proceedings of the Workshop in Primary and Secondary Computing Education (WiPSCE '15). ACM, New York, NY, USA, 80-87. DOI=http://dx.doi.org/10.1145/2818314.2818315
8. Brennan, K, Resnick, M. (2012) New frameworks for studying and assessing the development of computational thinking, AREA.
9. Brennan, K. (2011) Creative computing: A design-based introduction to computational thinking. – elérhető: http://scratched.media.mit.edu/sites/default/files/CurriculumGuide-v20110923.pdf (utoljára megtekintve: 2016. 10. 25.)
10. Aiken, J. M., Caballero, M. D., Douglas, S. S., Burk, J. B., Scanlon, E. M, Thoms, B. és Schatz, M. F. (2012) Understanding Student Computational Thinking with Computational Modeling, PERC Proceedings.

11. Bell, T., Witten, I. H., Fellows, M. (2010) Computer Sciene Unplugged. – elérhető: http://csunplugged.org/books (utoljára megtekintve: 2016. 10. 25.)
12. Dagiene, V. (2006) Information technology contests – introduction to computer science in a attractive way, Informatics in Education, 5. 1.s., 37–46.
13. Cartelli, A., Dagiene, A., Futschek, G. (2010) Bebras Contest and Digital Competence Assessment: Analysis of Frameworks. International Journal of Digital Literacy and Digital Competence. Január-Március. 24-39.
14. Pluhár, Z., Gellér, B. (2018) International Informatic Challenge in Hungary. In: Teaching and Learning in a Digital World : Proceedings of the 20th International Conference on Interactive Collaborative Learning. Berlin, Germany: Springer, .pp. 425-435.
15. CS Unplugged weboldal. Elérhető: http://csunplugged.org (utoljára megtekintve: 2018.11.10.)
16. Computer Science for Fun weboldal. Elérhető: http://cs4fun.org (utoljára megtekintve: 2018.11.10.)
17. Magyar e-hód feladatok archívuma: http://e-hod.elte.hu/archivum (utoljára megtekintve:2018.11.10.)