

# A gráf fogalom fejlődése az algoritmusok tanulásának folyamán

Erdősné Németh Ágnes

erdosne@blg.hu

ELTE IK Doktori Iskola

Batthyány Lajos Gimnázium, Nagykanizsa

**Absztrakt.** Didaktikailag érdekes kérdés, hogy az informatikai és a kapcsolódó matematikai fogalmak a diákok gondolkodásában hogyan fejlődnek, egy-egy témakör kapcsán mikor és milyen ismereteket lehet tanítani. A diákok a gráfokat már egészen korán elkezdik használni matematikaórákon, majd újra és újra, egyre magasabb szinten kerül elő matematikából és az informatikai algoritmusok tanulása során – gráfeljárásokként és reprezentációs eszközként is. Cikkemben a gráfokkal kapcsolatos fogalmak és eljárások előfordulását és azok használatának lehetőségeit veszem sorra a gyerekek gondolkodási módszereinek fejlesztése során.

**Kulcsszavak:** gráfok tanítása, modellezés, gráfalgoritmusok

## 1. Bevezetés

Didaktikailag és metodikailag fontos kérdés, hogy az egyes informatikai fogalmak és eljárások mely életkorban kerülhetnek elő, a tanulás folyamatában és a gondolkodás fejlettségének milyen szintjén bukkannak fel és hogyan fejlődnek. Cikkemben a programozás egyik kulcsfontosságú adatszerkezetének és leghatékonyabb modellező eszközének, a gráfoknak a fogalmi és használati fejlődését veszem szemügyre.

A való életben nagyon sok probléma van, amiket természetes módon, az eredeti tulajdonságai alapján gráfokkal le tudunk írni: úthálózatok, közösségi hálózatok, emberek közti rokonsági kapcsolatok, vezetékhálózatok, szállítási láncok. Vannak olyan problémák is, ahol a gráfok nem ilyen természetes módon, nyilvánvalóan jelennek meg a feladatok megfogalmazása során, hanem az egyszerűsítés után, a modellezés során használjuk őket: térképszínezés megoldására, különböző elemekből álló sorozatok darabszámának kiszámolására, pakolási és sorba rendezési problémák megoldása során és játékok állapotleírására. A gráfok használata nagyon egyszerű, szemléletes eszköz a különböző matematikai problémák modellezésére is.

Cikkemben azt vizsgálom, hogy a gráfokkal kapcsolatos egyes fogalmak mikor kerülnek elő a gyerekek számára a matematikai tanulmányok folyamán, illetve a számítógépes gondolkodás elsajátítása közben az informatikaórákon.

A tehetséges diákokkal való foglalkozás a gondolkodási módszereik fejlesztését, a számukra új módszerek tanulását jelenti, azon módszereket, amelyeket kreatívan alkalmazva tudnak bonyolultabb feladatokat megoldani. Nem új algoritmusokat és új eljárásokat mutatok be, hanem feltérképezem, hogy mikor és hol fordulnak elő a diákok tanulmányainak folyamán a gráfok [17]. Vizsgálom, hogy az informatikából tehetséges, programozást tanuló diákok elméjében hogyan fejlődnek a gráfokkal kapcsolatos fogalmak, mikor, milyen gráfeljárásokat lehet bevezetni számukra, mely életkorban mit fognak tudni kreatívan alkalmazni, milyen sorrendben lehet nekik a különböző módszereket, eljárásokat megtanítani.

## 2. Gráfok a matematika és informatika kerettantervben

A matematika nemzeti alaptantervet és a hivatalos kerettanterveket [6] átnézve feltűnik, hogy meglepően kevésszer fordul elő a gráf fogalma, bár minden évfolyamnál szerepel egy „Gondolkodási módszerek, halmazok, matematikai logika, kombinatorika, gráfok” tematikus egység.

A gráf szó az alsó tagozaton csak a címben szerepel, a matematika tankönyvekben viszont elég gyakran találkozunk gráfos ábrákkal: szemléltetésre használják és feladatok megfogalmazására fejtörőknél és logikai feladatoknál. A Kenguru Matematikai Tesztversenyen és a Zrínyi Ilona Matematikai Tesztverseny feladatai közt is minden feladatsorban feltűnik néhány gráfos feladat.

Sajnos alsó tagozaton informatikai kerettantervről nem beszélhetünk.

Az 5-6. osztályos matematika kerettantervben a tematikus egység címéből is kikopott a gráf fogalma, így nem meglepő, hogy a tartalmi összegzésben sincs benn. A 7-8. osztályban a kiválasztásos feladatok szemléltetésére és a kombinatorikai feladatokban az esetek rendszerezésére használják a fákat, a fogalom megnevezése nélkül, matematikai modellként.

A felső tagozatos informatika kerettantervben a „Problémamegoldás informatikai eszközökkel és módszerekkel” tematikus egység részeként a folyamatábrák értelmezésének tanításakor és a többi tantárggyal való kapcsolódási pontként bukkan fel a gráf fogalma: a matematikában a diákok a gráf szemléletes fogalmával és egyszerű alkalmazásaival találkoznak, míg az etika tantárgy keretein belül a gráfokkal lerajzolt modellek értelmezésekor kerül elő a fogalom.

A 9-10. osztályos matematika kerettanterv is hangsúlyozza a gráfok eszközjellegű használatát kiválasztási és sorba rendezési feladatok és más problémák megoldásának ábrázolása során, a gondolatmenet leírására és szemléltetésére. Megnevezi a gráf részeit (csúcs, él, foksám) és egyszerű hálózatok lerajzolását is elvárja. A matematika tankönyvekben egyetlen száraz lecke szól a gráfokról, tele definícióval. Az emelt szintű tananyagban ezeken kívül szerepelnek még az egyszerű gráf, fa, kör, út, teljes gráf, összefüggő gráf és a komplementer gráf fogalmak is, bizonyítások szemléltetésére, modellezésre.

A 11-12-es kerettantervben a gráfokkal kapcsolatos ismeretek alkalmazása, bővítése, konkrét példák alapján gráfokkal kapcsolatos állítások megfogalmazása bukkan fel, de csak a foksám összege és az élek száma közötti összefüggés bizonyítása szerepel. Elvárás a sorba rendezési és kiválasztási problémák felismerése és a megoldás gondolatmenetének szemléltetése gráffal. Az emelt szintű kerettantervben a gráf éleinek és a csúcsainak száma közötti összefüggés, az Euler-séta és a Hamilton-kör bizonyításai szerepelnek az előbbieken kívül.

A középiskolás informatika kerettantervben nincs gráfokkal kapcsolatos új ismeret megfogalmazva, a matematikával való kapcsolódási pontként elvárt a rendszerezést segítő eszközök – fadiagram, útdiagram, táblázatok – használata és készítése, illetve a gráf szemléletes fogalma, egyszerű alkalmazásai.

A matematika érettségi feladatokban középszinten az első részben olyan feladatok fordulnak elő, amik ellenőrzik a gráffal kapcsolatos alapfogalmak ismeretét (csúcs, él, foksám, összefüggő, fa), míg a második részben a kombinatorikai feladatok modellezésére használhatóak az ezzel kapcsolatos ismeretek. Az emelt szintű feladatsorokban is szemléltetésre, modellezésre, a gondolatmenet illusztrálására tudják használni a diákok a gráfokkal kapcsolatos ismereteiket.

Az informatika érettségien sem középszinten, az alkalmazói feladatokban, sem emelt szinten a programozás feladatban nem fordult elő még gráf – még a megformázandó szövegben, a lerajzolandó ábrák közt sem.

A matematikaversenyekre felkészítő anyagokban és a matematikaversenyek anyagai között viszont rengeteg példát találhatunk gráfok használatára.

A tehetséges gyerekekkel való foglalkozás során alapesetben csak a matematikából használat közben megismert szemléletes fogalomra és a modellezés során használt ismeretekre támaszkodhatunk. A matematikából is versenyző gyerekeknél a matematikaversenyre felkészítés közben megismert fogalmak is jól használhatóak, míg az ottani bizonyítások csak ritkán tartalmaznak algoritmizálható gondolatmenetet, megoldást.

### 3. Számítógépes gondolkodás – gráfok az e-hód feladatokban

Az e-hód verseny feladatai [8] a számítógépes gondolkodás fejlesztésére és a meglévő ismeretek ellenőrzésére használhatók [9]. Az 5-6. osztályosok feladatai közt csak nehéz megjelölésű gráfos feladatok vannak, jelezve azt, hogy nem megszokott, általánosan használt eszközt kell kezelniük a gyerekeknek (Legrövidebb út, Barlangászok, Jelzőtűz, Sok barát). Ezekben a feladatokban a lerajzolt, irányítatlan gráfokon valaminek a megszámlálását kéri a feladat. A 7-8-sok számára is közepes és nehéz gráfos feladatok fordulnak elő (közepesek a kisebbek előbb felsorolt feladatai, nehezebbek az Utcakövek, Hód úr bevásárol, Dobókocka, Hobbit hód, Mozi, Nyaklánc). A nehezebb feladatok már irányított gráfokon valaminek a megszámlálását, útkeresést vagy összehangot várnak el. Az általános iskolások számára adott feladatokban a gráf ábrája a feladatleírás része.

A középiskolásoknak szánt feladatoknál a gráfokkal való modellezés, magának a gráfnak a lerajzolása is a gyerekek dolga, irányított és irányítatlan gráfok esetén is. A máshol nem tanult vagy csak emelt szinten szereplő fogalmak ezekben a feladatokban pontos definíció nélkül, szemléletesen kerülnek elő (izomorf, összefüggő, súlyozott, irányított gráf fogalmak). Itt is a közepes és nehéz feladatok közt szerepelnek (A bűvész, Szó-zűrzavar, Találkozás, Szomszéd-ság, Drága hidak, Folyóellenőrzés).

Az e-hód feladatai is a matematikából már használt és ismert témáknál, illetve az egyéb gondolkodtató, logikai játékoknál maradnak: kombinatorika, színezések és útválasztások.

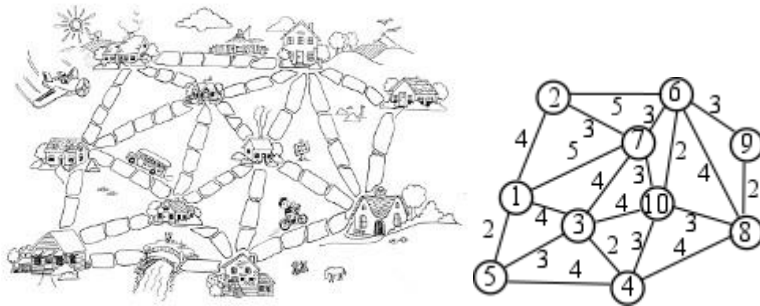
Az e-hód feladatok nagyon hasznosak, tágitják az informatikával a továbbiakban nem foglalkozó gyerekek látókörét, bepillantást engednek nekik a nehezebb feladatok világába.

### 4. Gondolkodásfejlesztés számítógép nélkül - CSUnplugged

A számítógép nélküli informatika világában (CSUnplugged) [7] a programozáshoz használt haladó algoritmusok szemléltetéséhez, az új algoritmusok bevezetéséhez találunk öt különböző játékot a gráfok témakörben. Ezek használata motivál, érdekesebbé teszi egy-egy új probléma felvetését, az azzal való megismerkedést. Az adott eljárás bevezetése előtt használhatóak, mindegyikben a modellalkotás eszköze a gráffal való ábrázolás. Mindegyikhez tartozik játékleírás, feladatlap, módszertani útmutató, a feladat továbbgondolása és további források az információszerezésre. Minden feladat után találhatunk további modelleket és online játékokat, illetve forrásokat a problémával való komolyabb megismerkedéshez. Ezen források közt már egyetemi jegyzetek, könyvek és tudományos cikkek is vannak, komoly matematikai apparátussal leírva.

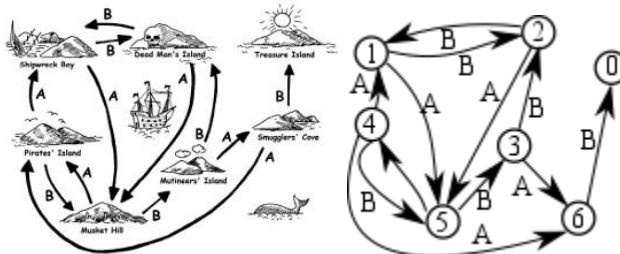
Az Activity 9: The Muddy City a minimális feszítőfák (Minimal Spanning Tree) bevezetésekor használható, a Kruskal algoritmus felfedeztetésére és megértetésére hasznos. A feladat két adott pont között a legrövidebb út megkeresése: a gráfot modellezésre használjuk, és minimális feszítőfát keresünk benne. A további lehetőségek közt Scratch modellt és online játékot is találhatunk.

lunk a közismerten utazó ügynök problémának nevezett feladatra, míg a további forrásokban a pontos matematikai algoritmus leírásig is eljuthatunk.



1. ábra: Muddy City feladat [7] és modellje

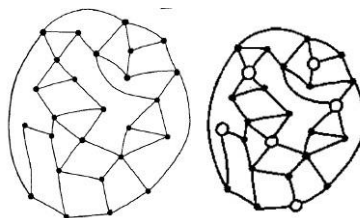
Az Activity 12: Treasure Hunt a véges állapotú automaták (Finite State Automata) világába visz a modellként használt irányított gráffal történő ábrázoláson keresztül.



2. ábra: Treasure Hunt feladat [7] és modellje

Az Activity 14: The Poor Cartographer (Graph Coloring) feladat a térképek színezéséhez, konkrétan a négyszintűtel megfogalmazásához tartalmaz feladatokat, sok-sok egyéb, gráfokkal kapcsolatos további forrásra vezető linkekkel.

Az Activity 15: Tourist Town a meghatározó halmaz (Dominating Set) megkeresésére, rejtvények megfogalmazására jó példa. A feladatban a gráf minden pontja a meghatározó halmaz elemeitől maximum egy élen elérhető.



3. ábra: Tourist Town feladat modellje és megoldása[7]

A feladat szemléletes példa az egy irányban könnyen meghatározható függvényekre: könnyű a feladat feltételeinek megfelelő gráfot és abban meghatározó csúcsalmazt létrehozni, nehéz meghatározni a meghatározó halmazt. Egy megfejtésről viszont könnyű ellenőrizni, hogy az helyes vagy sem.

A feladat a gráfok összefüggésének vizsgálatával kapcsolatos eljárások előtt használható. Könnyű rajta bemutatni azt is, hogy ha egy feladatnak nem egyértelmű a megoldása, akkor a tesztelő-rendszer a feltételeknek való megfelelést bizonyos esetekben könnyen tudja ellenőrizni.

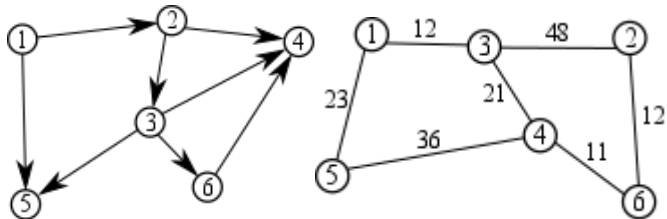
Az Activity 16: Ice roads a Steiner-féle fakeresés (Steiner trees), egy speciális minimális feszítőfa kereső eljárás felfedezését segíti. Ennek a használata a programozásban már az OKTV szint felett van.

## 5. Gráfok alkalmazásainak helye a programozás tanulásában

A programozás tanulása közben a gráfok bevezetése előtt a diákoknak jól kell ismerni az alapvető adattípusokat (egész, valós, logikai, karakter), egy egyszerű adatszerkezet változatait (vektor, tömb, szöveg) és a programozás alapeljárait (eldöntés, megszámlálás, kiválasztás, kiválogatás, keresés, minimum- és maximumszámítás). Ezen adatszerkezeteket és programozási tételeket algoritmus-leíró nyelven és valamilyen programozási nyelven is biztosan, hibátlanul kell tudniuk kódolni. A számítógéppel való kommunikáción keresztül kell tudniuk adatokat beolvasni és kiíratni valamilyen formában (képernyőre, konzolra, fájlba). Tudniuk kell meghatározni a használandó adattípusokat és megtalálni a megfelelő tárolási struktúrát, illetve megfogalmazni, hogy melyik alapeljárást vagy annak módosított verzióját kell használni a feladatmegoldás folyamán.

Egy diák - ha szerencséje van és már általános iskolában tanul programozást, akkor - ezeket az ismereteket a felső tagozat végére megszerzi. Ha először középiskolában találkozik a programozással, akkor megfelelő motivációval és segítséggel egy fél év alatt el tudja sajátítani, akár kilencedik elején. A kevésbé motiváltak számára ezen ismeretek megtanulása az emelt szintű érettségi idejére történik meg, de ők a középiskolai tanulmányaik során valószínűleg nem találkoznak már más, bonyolultabb algoritmusokkal. A tehetséggondozó órákon a gráfok bevezetése a felső tagozat végén vagy kilencedikben történhet.

Első lépésként az eddig szemléletesen használt gráfok alapfogalmakat néhány rajz segítségével lehet megnevezni, a fogalmakat pontosítani:



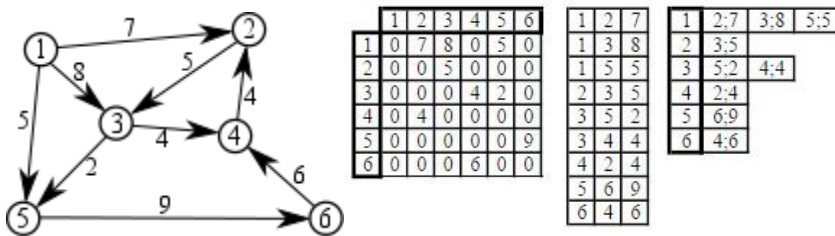
4. ábra: Irányított, súlyozatlan gráf; irányítatlan, súlyozott gráf

- csúcsok, élek
- irányított és irányítatlan gráf
- súlyozott és súlyozatlan élek
- fokszám, irányított gráfban befok és kifok.

Az alapfogalmak tisztázásával párhuzamosan lehet bevezetni a gráfok tárolásának első lehetőségeit:

- csúcsmátrix
- éllista
- csúcslista.

Érdemes először papíron lerajzolt vagy vetítőn kivetített gráfot ábrázoló rajz alapján kézzel felépíteni az egyes tárolási lehetőségeket, és fordítva: a csúcsmátrix, éllista vagy csúcslista alapján felrajzoltatni a gráfot.



5. ábra: irányított és súlyozott gráf, csúcsmátrix, éllista, csúcslista

Az egyes tárolási módszereknél érdemes megbeszélni a többi alapfogalom jelentését és kiszámítási módját.

A gráfok tárolási formájának eldöntésekor lehet és érdemes - a tanulmányok folyamán először - megvizsgálni a tárolással és algoritmus hatékonysággal összefüggő problémákat: elfér vagy sem az adott adatszerkezettel a memóriában, könnyű beszúrni/módosítani/törölni adatot vagy sem, melyik műveletre van vagy van leggyakrabban szükség. Fontos megbeszélni, hogy mindig az adott feladat dönti el, hogy mikor melyik tárolási módszert alkalmazzuk.

Az alapfogalmak gyakorlására MESTER [18] feladatbank Haladó szint, Gráfok, elemi feladatok feladatai használhatóak (Állatkert, Falvak, Ember, Rémhír).

Az ezekkel a fogalmakkal kapcsolatos feladatok a Nemes Tihámér Országos Informatika Versenyen az 1. korcsoport feladatai között és a 2. korcsoport megyei fordulós feladatai közt szerepelnek [1,2,3,4,5].

A következő fogalmak elméleti tisztázására is sor kerülhet, de az ezekkel kapcsolatos gráfos algoritmusok már bonyolultabbak, a gráfbejárások után kerülnek sorra.

- út, kör
- fa, gyökér, levél
- összefüggő gráf, izolált pont, összefüggő komponensek
- teljes gráf, komplementer gráf
- irányított gráfban forrás, szuperforrás, nyelő, szupernyelő.

Az előző fogalmak és algoritmusok megismerése után más témakörökkel folytatandó a tehetőségi diákoknak az algoritmusokkal való ismerkedés.

## 6. Floyd-Warshall algoritmus és variációi

A rekurzióval és a dinamikus programozással kapcsolatos ismeretek tanítása után újra egy csepp kitérőt tehetünk a gráfok világába és egy klasszikus algoritmust és annak variációit vehetjük elő [20]. Eredetileg két változata van a Floyd-Warshall algoritmusnak:

- súlyozatlan gráfban minden csúcspárról eldöntendő, hogy létezik-e közöttük út
- súlyozott gráfban minden csúcspárról kiszámítandó a köztük levő minimális út hossza

Az eljárás működésének első látásra kicsit bonyolult a megértése, bár a dinamikus programozás elvének megtanulása után a diákok már elég rutinosak egy rekurzív formula megértéséhez. A kódolása viszont nagyon egyszerű: ha csúcsmátrixszal tároljuk a gráfot, akkor a fenti feladatok megoldásának megfelelő mátrix előállítására három egymásba ágyazott ciklussal történik. A cikkben leírt változatai [20] az eredeti feladatnak – maximin, minimax, megbízhatóság-

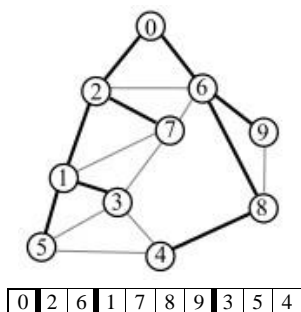
sabb út, minimin, maximax - érdekessé teszik ennek az algoritmusnak a használatát, még ha első látásra nem is tűnik olyan hatékonyak az  $O(n^3)$  miatt.

A MESTER [18] feladatbankban a Haladó szint, Gráfok – legrövidebb utak feladatai közt találunk hozzá gyakorlófeladatokat (Szállítás, Vám, Túra, Kastély).

## 7. Gráfbejárások használata

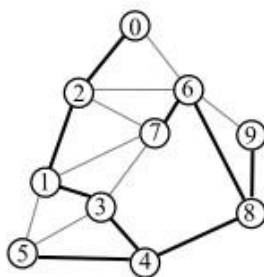
A gráfbejárások tanulása előtt a diákoknak meg kell ismernie a sor és a verem adatszerkezeteket és az azokat működtető eljárásokat. Hibamentesen kell tudniuk használni a rekurzió elvén működő eljárásokat. [10,11,12,13,14,15,19]

A tanulás folyamán a következő, gráfokkal kapcsolatos eljárás a gráf szélességi bejárása és a szélességi feszítőfa használatával megoldható feladatok. Ezeknél sor adatszerkezetet kell használni. A szélességi bejárás megértése és hibamentes alkalmazása sok gyakorlást igényel, változatos feladatokon. Bőséges feladatanyagot találunk a MESTER [18] feladatbank Haladó szint, Gráfok, szélességi bejárás témakörében (Futár, Falvakban üzletek, Modul), több nehézségi szinten, újra és újra elővéve a módszert.



6. ábra: Szélességi bejárás és szélességi feszítőfa irányítatlan, súlyozatlan gráfon, sor

A verem és a rekurzió fogalmának elsajátítása után következhet a mélységi bejárás illetve a mélységi feszítőfákat használó óriási feladattömeg megoldása. A mélységi bejárás megértése és megfelelő használata újabb kihívást jelent a középiskolások számára. A MESTER [18] feladatbank erre is egy teljes fejezetet szentel.



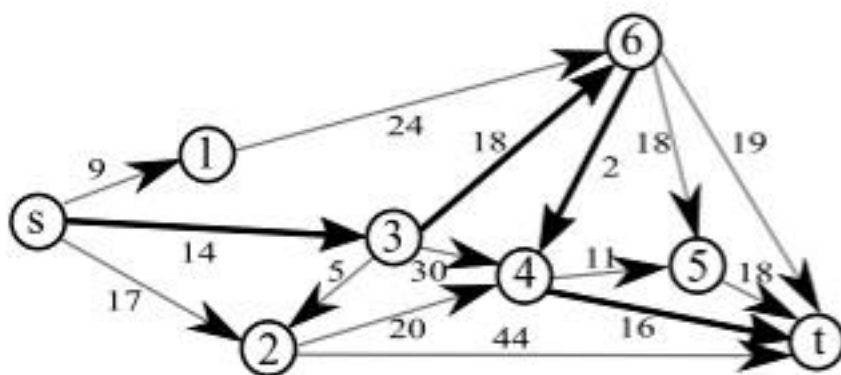
7. ábra: Mélységi bejárás és mélységi feszítőfa irányítatlan, súlyozatlan gráfon

Az így elsajátított gráfbejárások használhatóak az olyan feladatok megoldása során is, amikor a feladat leírásából nem következik, hogy gráfmodellel lerajzolható. Nem is biztos, hogy ábrázoljuk a teljes gráfot, csak az egyes lépésekben szükséges részét számoljuk ki. Ezek a feladatok a gráffal való modellezés után gráfbejárással oldhatók meg (robotos és labirintus példák).

Gráfbejárásokkal és egyszerű feszítőfákkal kapcsolatos feladatok a Nemes Tihamér Versenyen a 2. korcsoport országos fordulójában és az OKTV második és harmadik fordulójában fordulnak elő [1,2,3,4,5].

A szélességi bejárás használatának egy következő szintje a prioritási sor adatszerkezet használatával történő kódolás. Ehhez a használt programozási nyelvnek lehetővé kell tennie az összetett adatszerkezet használatát.

Súlyozatlan gráfban a legrövidebb út két adott pont között a szélességi bejárással található meg. Súlyozott gráfban újabb eljárás fogalmazódik meg: a Dijkstra algoritmus. Ennek tanítása előtt hatékony rendezési eljárásra és a mohó paradigma elsajátítására van szükség. Feladatokat a MESTER [18.] feladatbank Haladó szint, Gráfok, legrövidebb utak témakörében találunk (Tűzoltó, Repülőút). A halmaz, a kupac illetve a prioritási sor adatszerkezet megismerése után az algoritmus újra elővehető, újrakódoltatható nehezebb feladatokon.



8. ábra: Dijkstra algoritmus

A topologikus rendezéssel megoldható feladatok is a gráfbejárásokon alapulnak, példákat a MESTER [18] feladatbank Haladó szint, Gráfok, körmentes gráfok témánál találunk.

A gráfok bejárásával kapcsolatos algoritmusokat a diákok gondolkodásának fejlődésével többször is elő kell venni, ciklikusan, egyre nehezedő szinten.

## 8. Feszítőfák haladó feladatai

A kétféle gráfbejárással a feszítőfákról szóló feladatok egy része is megoldható. A következő eljárások már csak az OKTV-n fordulnak elő. Megtanulásuk előtt szükség van hatékony sorba-rendezési algoritmusok megtanulására. Az összetett adatszerkezeteket is tudó programnyelv használata közben felfedezhetjük a prioritási soron kívül a halmaz és az 'unió-holvan' adatszerkezetet is. Ezekkel nyílik lehetőség két különböző, súlyozott gráfban minimális költségű feszítőfát meghatározó algoritmus megtanulására, melyeket más-más típusú feladatokban használhatunk:

- Prim eljárás: mindig a következő minimális költségű élet húzza be a már összefüggő komponenshez.
- Kruskal algoritmus: az éleket, hosszúság szerint növekvő sorrendben veszi sorra. Nem összefüggő gráfban az eljárás lefutása után rögtön megkapjuk a feszítő erdőt, így az összefüggő komponensek keresésére is használható.



## 9. További kutatás

Az eddig leírt gráf modellek és a gráfokkal kapcsolatos algoritmusok képezik a magyar versenyrendszerben a Nemes Tihamér Országos Informatikai Verseny és az Országos Középiskolai Tanulmányi Versenyen (OKTV-n) programozás kategóriájában előforduló feladatok alapjait. [1,2,3,4,5]

A nemzetközi versenyekre készüléskor, a válogatóversenyeken és az informatikai olimpiákon használt algoritmusok feltérképezése, az azokkal kapcsolatos gondolkodás fejlesztése egy következő kutatás és cikk témáját képezi.

## Irodalom

1. Programozási feladatok I-II., Kossuth Kiadó, Budapest, 1997
2. Programozási versenyfeladatok tára (1985-1994), NJSzT, Budapest, 2002
3. Programozási versenyfeladatok tára (1995-1999), NJSzT, Budapest, 2005
4. Programozási versenyfeladatok tára (2000-2004), NJSzT, Budapest, 2007
5. NJSZT Nemes Tihamér Országos Informatikai Tanulmányi Verseny – Programozás kategória, archívuma <http://nemes.inf.elte.hu/>
6. 51/2012. (XII. 21.) számú EMMI rendelet 1., 2. és 3. melléklete Módosítva a 34/2014. (IV. 29.) EMMI rendelet 2., 3., 4. mellékletének megfelelően Matematika és Informatika kerettantervek <http://kerettanterv.ofi.hu/> Letöltve 2016.10.
7. CSUnplugged\_OS\_2015\_v3.1, <http://csunplugged.org> Letöltve: 2016.10.
8. Bebras–International Contest on Informatics and Computer Fluency (2007-2015) <http://bebras.org>; <http://www.beaver-comp.org.uk/>; <http://informatik-biber.de/> <http://e-hod.elte.hu>
9. Dagiene, V., Stupuriene, G. (2016) Bebras - a Sustainable Community Building Model for the Concept based Learning of Informatics and Computational Thinking. Informatics in Education, VOL. 15, No 1, PP. 25-44.
10. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C. (2001). Introduction to Algorithms. MIT Press, 2nd edition
11. Dasgupta, S., Papadimitriou, C.H., Vazirani, U.V. (2006). Algorithms. McGraw-Hill
12. Halim, S., Halim, F. (2014) Competitive Programming 3. The New Lower Bound of Programming Contests, <https://cpbook.net/>
13. Kleinberg, J., Tardos, É. (2006). Algorithm Design. Addison-Wesley
14. Sedgewick, R., Wayne, K. (2011). Algorithms, Fourth Edition. Addison-Wesley.
15. Skiena, S.S. (2008). The Algorithm Design Manual. Springer-Verlag, 2nd edition
16. Zsakó, L.: Variations for spanning trees, Annales Mathematicae et Informaticae 33 (2006) pp. 151-165.
17. Szlávi, P., Zsakó, L.: Informatika oktatása TÁMOP-4.1.2 A1 és A2 könyvei, ELTE IK, 2012 [http://www.tankonyvtar.hu/hu/tartalom/tamop412A/2011-0052\\_34\\_informatika\\_oktatasa/adatok.html](http://www.tankonyvtar.hu/hu/tartalom/tamop412A/2011-0052_34_informatika_oktatasa/adatok.html)
18. Mester feladatértékelő feladatai: <https://mester.inf.elte.hu>
19. Rónyai, L., Ivanyos, G., Szabó, R. (1999) Algoritmusok, Typotex
20. Horváth, Gy., Horváth, Gy., Zsakó, L.: Variations on a classic task XXIXth DIDMATTECH (2016) pp. 72-78.