

# A Nemes Tihamér Programozási verseny témaköreiről készült syllabus

Nikházy László

laszlo.nikhazy@gmail.com

ELTE IK

**Absztrakt.** A Nemzetközi Informatikai Diákolimpia (IOI) feladatait egy nemzetközileg elfogadott útmutató alapján tűzik ki. Ez az IOI Syllabus, amely tartalmazza, hogy milyen témakörök lehetnek, illetve nem lehetnek a versenyen. A Syllabus nagyon sokat segít a versenyre való felkészülésben is. Hasznos lenne egy ilyen dokumentum a Nemes Tihamér NITV Programozás kategóriájában mindhárom korcsoportnak. A korábbi évek feladatait rendszerezve javaslatot teszünk a „Nemes Tihamér Syllabus” tartalmára, és azt közzétesszük a szakmai közösség számára megvitatás céljából – abban bízva, hogy a 2020/2021-es tanévre kialakul hivatalos formában is. Részletezzük, hogy milyen módszerrel készült a jelenlegi javaslat, és statisztikák alapján elemezzük a korábbi versenyek anyagát.

**Kulcsszavak:** programozási verseny, algoritmusok, adatszerkezetek, versenyfelkészülés, syllabus

## 1. Bevezetés

Jelen cikk arról szól, hogy mi alapján készült a bemutatott Nemes Tihamér Syllabus (továbbiakban NT Syllabus) jelenlegi verziója, ami pillanatnyilag egy javaslat. Ez a dokumentum elérhető nyilvánosan [1]. A GitHub platform lehetőséget ad bárki számára, hogy hozzászóljon a tartalmához, és célunk, hogy a feladatkitűzőkkel és a felkészítő tanárokkal való eszmecsere útján, szakmai konszenzussal alakuljon ki a következő évre érvényes verzió.

Az NT Syllabus az IOI Syllabus [2] mintájára készül. Ha eléri célját, egy hivatalos dokumentum alakul ki belőle, amelynek aktuális verzióját a versenybizottság hagyja jóvá és teszi közzé minden évben. Mivel az informatika, és ezen belül a versenyprogramozás is folyamatosan fejlődik, ezt követni szeretnénk, vagyis az évek során a dokumentum változhat, fejlődhet. Minden évben a dokumentum gazdája felelős a dokumentum frissítéséért, amelynek a fent említett konszenzussal kell megvalósulnia.

A cikk felépítése a következő. A 2. fejezetben megfogalmazzuk, hogy milyen célok betöltésére hoztuk létre az NT Syllabus-t. A 3. fejezetben részletezzük, hogy milyen kategóriákat alkalmaz a dokumentum, amelyek szükségesek a további megértéshez. A 4. fejezetben leírjuk, hogy milyen módszert alkalmazva jött létre a jelenlegi verzió. Az 5. fejezetben bemutatjuk a választott formátumot, és egy részletet a dokumentumból a példaként a tartalmára is. A 6. fejezetben a Syllabus írása során elemzett feladatminta segítségével fogalmazzunk meg észrevételeket az elmúlt évek versenyfordulóinak témaköreivel kapcsolatban.

## 2. Célok

Az NT Syllabus-nak három fő célja van:

- Meghatározza a versenyre szükséges előzetes tudást.
- Segít a versenyzők felkészülésében, a tanároknak a felkészítés tervezésében.
- A feladat kitűzőknek támpontot ad, hogy egy-egy feladat melyik kategóriában lehet.

A fenti célok elérése érdekében az NT Syllabus-ban megtalálható minden olyan téma (algoritmusok, adatszerkezetek, módszerek, nyelvi eszközök), amely egyáltalán szóba jöhet a

versenyen, és ezeket kategóriákba soroljuk. A kategóriák tükrözik azt, hogy milyen módon fordulhat elő egy témakör, ezt a következő fejezet részletezi.

Az NT Syllabus jelen verziója kifejezetten a 2. és 3. forduló (gépes) feladatairól szól. Az első fordulóban szélesebb körből fordulhatnak elő feladatok, mert ott elvárt egy-egy (esetleg ismeretlen) számítástechnikai témakör alapszintű megértése a feladatleírás alapján.

### 3. Témák kategorizálása

Az NT Syllabus az IOI Syllabus-hoz hasonló kategóriákat alkalmaz, az alábbiakban felsoroljuk, majd bővebben magyarázzuk őket. A verseny három korcsoportjában különböző kategóriába sorolhatók a témakörök.

Jelölés	Kategória
✓	Lehet, megkötések nélkül
✓📄	Lehet, de magyarázandó
✓📄	Lehet, de a feladatleírásban nem
?	Hatáskörön kívüli
✗❓	Nem lehet, de nyitott a megvitatásra
✗	Kizárt

1. táblázat: Témakörök kategóriái

- ✓ **Lehet, megkötések nélkül**  
Ebben a kategóriában lévő témakörök előzetes tudásnak tekintendők. A versenyzőktől elvárt az ismeretük. A feladatleírásban magyarázat nélkül előfordulhatnak.
- ✓📄 **Lehet, de magyarázandó**  
A versenyzőknek ismerniük kell ezt a témakört, de amikor feladatleírásban szerepel, akkor definiálni kell. Általában akkor használjuk ezt a besorolást, ha egy ✓ témát többféleképpen is lehet értelmezni.
- ✓📄 **Lehet, de a feladatleírásban nem**  
Olyan témakörök sorolhatók ide, amelyeknek ismeretére a versenyzőknek a feladatmegoldás során van szüksége. Tehát a feladat szövegében nem szerepelhetnek.  
Egy ilyen téma például az *Aszimptotikus felső becslés a komplexitásra*. Ez egy központi téma a verseny kapcsán, hiszen egy helyes program az algoritmus komplexitása alapján kap pontot. Mégsem fordulhat elő a feladatleírásban a komplexitás fogalma, a precíz definíció ismerete nem is elvárt.
- ? **Hatáskörön kívül**  
Bármilyen téma, amit nem említ a Syllabus, ebbe a kategóriába esik. A versenyzőktől nem elvárt, hogy ismerjék ezeket. A legtöbb versenyfeladat nem kapcsolódik ilyen témakörökhöz. De ez nem zárja ki, hogy a versenyen legyen olyan feladat, ami ilyen témakörhöz kapcsolódik. A versenybizottság kitzúzhat ilyen feladatot a verseny színesítése érdekében, de ebben az esetben is megoldhatónak kell lennie a fenti kategóriákba eső ismeretekkel.

### **✖** Nem lehet, de nyitott a megvitatásra

A Syllabus az évek során változik, fejlődik a versennyel együtt, így lehetőség adódik arra, hogy egy (valamely korcsoportban) kizárt téma később bekerüljön a verseny anyagába. Általában ezek a témakörök kapcsolódnak megengedett témakörökhöz, és nehéz határt szabni. Ebben az esetben a megvitatásra nyitott kategóriába kerülnek ezek a témakörök, ezzel is biztatva a szakmai közösséget a visszajelzésre a kérdést illetően. Ha bekerül az idők során egy témakör, akkor a kizárt kategóriából először ebbe kell kerülnie.

### **X** Kizárt

Ebbe a kategóriákba főként nehéz algoritmikus módszerek, bonyolult matematikai fogalmak tartoznak. Garantált, hogy nem lesz olyan feladat a versenyen, amelynek megoldásához *elengedhetetlen* egy ilyen témakör ismerete. Szintén kívánatos, hogy ezeknek a témáknak az ismerete ne nyújtson utat egyszerűbb, vagy több pontot érő megoldáshoz. Ugyanakkor előfordulhat, hogy egy feladat témaköre kapcsolódik egy ilyen témához, de ennek ismerete nem segít lényegesen a megoldásban.

## **4. Elemzési módszer**

A következőkben leírjuk, hogy az NT Syllabus jelenlegi, első javaslat verziója milyen módszerrel alakult ki. A dokumentum alapját az IOI Syllabus képezi, viszont ezt jelentősen testre szabtuk a magyar versenyekre. A kategóriák megegyeznek az ottaniakkal.

A témakörök halmazának kialakításakor felsoroltuk az IOI Syllabus témaköreit (magyarra fordítva), és kiszűrtük belőlük azokat, amelyek túl messze állnak a magyar versenyek hatáskörétől, valamint azokat, amelyek készségekre (például hibakeresés) és eszközök használatára (például fejlesztőkörnyezet) vonatkoznak. Tehát az NT Syllabus az IOI Syllabus-hoz képest erősen rövidített, az áttekinthetőség kedvéért. Főként a konkrét tárgyi ismereteket és módszereket sorolja fel. Emellett a témaköröket kiegészítettük olyan elemekkel, amelyeket a magyar versenyeken gyakran előfordulnak, viszont az IOI Syllabus nem tér ki rájuk specifikusan, nem szerepel benne külön témakörként. Néhány esetben az IOI Syllabus egy-egy témakörét fel kellett bontanunk többre.

A fenti lépés után a témaköröket kategóriákba kellett sorolnunk mindhárom korcsoportra külön-külön. Ehhez az utóbbi évekből feladataiból egy reprezentatív mintát vettünk, korcsoportonként legalább 50 feladatot (a harmadik, OKTV korcsoport esetében 80 feladatot). Készítettünk egy hatalmas megosztott táblázatot [3], amelynek oszlopaiban minden azonosított témakör szerepel, a soraiban pedig a vizsgált feladatok. Minden feladatot elemeztünk a feladatleírás és a helyes megoldások szempontjából, és bejelöltük, hogy mely ismeretek szükségesek hozzá. Ebben nagy segítségünkre voltak tapasztalt versenyző diákok. A táblázat kitöltése közben is azonosítottunk még újabb témaköröket, amelyeket hozzávettünk az NT Syllabus felsorolásához.

A táblázatban kialakult statisztika és a versenykiírás [4] alapján minden témakörre eldöntöttük, hogy az egyes korcsoportokban mely kategóriába tartozzon. Hangsúlyozzuk, hogy ez csak egy javaslat, egy kiindulás a szakmai közösség számára. Az „igazi” változat hosszas egyeztetések után alakul ki, a versenybizottsággal, a felkészítő tanárokkal, illetve akár versenyzőkkel. Végül a versenybizottság dönti el, hogy tényleg felhasználja-e ezt a dokumentumot a verseny hivatalos dokumentumaként.

## **5. Formátum, példa**

Az alábbi képen látható egy részlet az NT Syllabus jelenlegi (2019. 11. 16.) változatából. Táblázatos formában jelenítjük meg, hogy az egyes témakörök a három korcsoportban milyen kategóriába tartoznak. Ezzel áttekinthető az is, ahogyan nehezedik a verseny az idősebbek számára. Kialakítható belőle egy „tanterv” a versenyfelkészítésre vonatkozóan.

## AL3a. Algoritmusok

NT1	NT2	OKTV	Leírás
✓	✓	✓	Egyszerű számelméleti algoritmusok: számrendszer átváltás, Euklideszi algoritmus (LNKO-ra), prímteszt $O(\sqrt{n})$ osztókereséssel, Eratoszteni szita, prímfelbontás (osztókereséssel vagy szitával)
x	x	✓	Gyors hatványozás (négyzetre emelésekkel)
x	x	✓	Műveletek tetszőlegesen nagy egész számokkal (összeadás, kivonás, szorzás)
✓	✓	✓	Egyszerű programozási tételek tömbökön: összegzés, megszámlálás, keresés, minimum/maximum, kiválogatás
✓	✓	✓	Programozási tételek összeépítése, pl. feltételes maximum
✓	✓	✓	Rendezett sorozatok összefésülése, metszet, unió
✓	✓	✓	Egyszerű string algoritmusok (pl. minta keresése naív módszerrel)
✓	✓	✓	$O(n^2)$ rendezések (buborék, leszámlláló, minimum-kiválasztásos)
x	✓	✓	Gyorsrendezés (quicksort), <i>NT1-ben a sort() függvény használata elvárt, de az algoritmus ismerete nem</i>
x	✓	✓	$O(n \log n)$ rendezések (kupac, összefésüléses)
x	x	✓	Lineáris idejű rendezések (láda/vödör rendezés, radix rendezés) <sup>3</sup>
x	✓	✓	Rekurzív fabejárás
x	✓	✓	Rendezett fák bejárásai (pre, in- és post-order)
x	✓	✓	Szélességi és mélységi gráfbejárás
x	✓	✓	Összefüggő komponensek meghatározása
x	✓	✓	A mélységi feszítőfa alkalmazásai, például topologikus sorrend
x	✓	✓	Irányított körmentes gráf emeletekre bontása

1. ábra: Részlet az NT Syllabus tartalmából

## 6. Elemzési eredmények, észrevételek

Egy nagyon hasznos kimenete az elemzésnek a feladatokból készült megosztott táblázat [3], amely nem csak statisztikaként szolgál, az egyes témakörökre való célzott felkészülést is segíti, hiszen lehet a témakörhöz tartozó feladatokat választani ez alapján. Tervezzük, hogy bővítjük az elemzett feladatok körét még további, korábbi versenyfordulókkal. A táblázatból kiolvasható statisztika alapján a következő fejezetekben megmutatjuk, hogy korcsoportonként melyik az a néhány témakör, ami a legtöbb feladatban szerepel. Hangsúlyozzuk, hogy az alábbi táblázatok csak néhány kiemelt témát mutatnak, a teljes statisztika az online táblázatban [3] látható.

A táblázatokban feltüntetjük, hogy hozzávetőlegesen a feladatok mekkora hányadához szükséges egy-egy témakör, kerekített értékekkel, hiszen a minta méretéből adódóan 4-5 % pontatlanság

mindenképp előfordulhat. Megjegyzés ezzel kapcsolatban, hogy minden feladatnál a legfontosabb módszereket, algoritmusokat, adatszerkezeteket jelöltük meg, amelyek lényegében a feladat témájának tekinthetők. Bizonyos alapvető algoritmusok (például maximumkiválasztás) nagyon sok más, bonyolultabb algoritmusnak és módszernek velejárói, azoknál nem jelöltük be őket. Ezeket az alapvető algoritmusokat akkor jelöltük meg, ha a feladat erről szól, vagy legalábbis a megoldásnak egy nagyon lényeges hányadát képezi ez.

## 6.1. Az 1. korcsoport leggyakoribb témakörei

Az alábbi táblázatból látható, hogy az 5-8. osztályosoknál lényegében nem szerepelnek összetettebb, előre elsajátítandó „tankönyvi” algoritmusok. A feladatok mintegy 65%-a megoldható az úgynevezett programozási tételek [5] alkalmazásával. Ezek egyszerű algoritmus sablonok, amelyeket a diákok sok esetben a tudatosításuk nélkül is képesek alkalmazni. A feladatok nagyjából 35%-ához kell valamilyen haladóbb algoritmikus stratégia használata. Ezek rendszerint valamilyen komolyabb ötletet igénylő, nehezebb feladatok. Mivel fordulónként 3-4 feladat van, ezért a versenyzők számíthatnak arra, hogy a második fordulóban egy, a döntőben egy vagy két ilyen feladat lesz. Az utóbbi években a mezőny fejlődése miatt a döntőben már általában négy feladat van, ez esetben átlagosan két feladat várható ebből a típusból.

Adatszerkezetek között is döntő többségben csak nagyon egyszerűek szerepelnek a versenyen, természetesen ide tartozik a tömb, ritkább esetben a kétdimenziós tömb. Némileg alulreprezentált a fontosságához képest a karakterlánc (string), amely nem is feltétlenül van minden évben. Egy fontos kiemelendő viszont a hisztogram, másnéven (a szerző által bevezetve) „számláló tömb”, amikor egy tömb elemeiben számoljuk meg, hogy az egyes értékek hányszor szerepelnek egy sorozatban.

Algoritmikus stratégiák	
Mohó algoritmusok	15%
Két mutató technika	10%
Dinamikus programozás	5%
Szimuláció	5%
Adatszerkezetek	
Tömb (egy dimenziós)	55%
Hisztogram, számláló tömb	20%
Karakterlánc (string)	15%
Két dimenziós tömb	10%
Algoritmusok	
Minimum-, maximumkiválasztás	30%
Programozási tételek összeépítése	15%
Keresés, eldöntés	10%
Számelméleti algoritmusok	10%
Matematika	
Számrendszerek, mértékegység átváltások	20%
Műveletek maradékokkal	20%
Legnagyobb közös osztó	10%

2. táblázat: Az 1. korcsoport leggyakoribb témakörei

A matematika témaköréből kitüntetett figyelmet érdemel a mértékegység átváltások, illetve számrendszerek témaköre. (Azért soroljuk őket egybe, mert szinte ugyanazok az eszközök kellene

hozzájuk, a mértékek is egyfajta, vegyes alapú számrendszernek tekinthetők.) A versenybizottság előszeretettel tűz ki ilyen feladatokat, minden évben várható egy vagy kettő. A maradékos műveletek is nagyon fontosak, egyrészt a mértékegységekhez, számrendszerekhez is kell, másrészt szintén nem elhanyagolható mértékben vannak számelméleti algoritmusok a versenyen (például legnagyobb közös osztó).

## 6.2. A 2. korcsoport leggyakoribb témakörei

A 9-10. osztályosok esetében a legfontosabb különbséget a dinamikus programozás „térhódítása” és a gráfalgoritmusok megjelenése okozza. Természetesen a többi témakörből (például mohó algoritmusokból) is jóval nehezebb feladatok vannak, mint az 1. kategóriában. Az alábbi táblázatban megjelenített fa-, és gráfbejárási algoritmusok részesevése között némi átfedés van (mert bizonyos feladatoknak többféle egyenrangú megoldása van), de a táblázat áttekintésével kijelenthetjük, hogy a feladatok legalább egyharmada kapcsolódik a gráfokhoz. A gyakorlatban azt jelenti ez, hogy a második fordulóban egy, a harmadikban két ilyen feladatra lehet számítani. Egy másik nagyon fontos fejlemény, hogy a rendezés használatának mindenképp a versenyző eszköztárában kell lennie.

Az adatszerkezetekben való eltolódás is a fent említett okok miatt következik be. A gráfok szomszédsági listás reprezentálásához szükséges a nyújtózkodó tömb (C++ vector), illetve néhány egyéb helyzetben is hasznos. A karaktorsorozatoknak változatlanul nem nagy a jelenléte, de mivel fordulónként 5-6 feladat van, ezért minden évben lehet számítani string-es feladatra.

A gráfok mellett megjelenik még egy nehéz matematikai téma, a kombinatorika. Ebbe nem értjük bele a gráfelméletet, vagyis döntő többségében leszámításokhoz köthető feladatokról van szó. Ezzel párhuzamosan az 1. korcsoportban gyakori mértékegység átváltások sokkal ritkábban fordulnak elő.

Algoritmikus stratégiák	
Dinamikus programozás	20%
Két mutató technika	15%
Mohó algoritmusok	15%
Rekurzív kiszámítás	5%
Adatszerkezetek	
Tömb (egy dimenziós)	50%
Nyújtózkodó tömb	30%
Gráfrepresentációk: szomszédsági lista	30%
Két dimenziós tömb	15%
Karakterlánc (string)	10%
Algoritmusok	
Mélyégi gráfbejárás	20%
Minimum-, maximumkiválasztás	15%
Rekurzív fabejárás	10%
Rendezés	10%
Szélességi gráfbejárás	10%
Matematika	
Írányítatlan, súlyozatlan gráf	10%
Írányított gráf	10%
Fa gráf	10%
Kombinatorika	10%

3. táblázat: A 2. korcsoport leggyakoribb témakörei

## 6.2. A 3. korcsoport, vagyis az Informatika OKTV leggyakoribb témakörei

A 11-12. osztályosok versenyében egy nagyon fontos és nehéz új témakör a geometriai algoritmusok. A statisztikából arra következtethetünk, hogy nagyjából évenként egy ilyen feladat van. Szintén fontos, hangsúlyos témakör a haladó gráfalgoritmusok, amibe sok algoritmust beleértünk, többek között a Dijkstra-algoritmust legrövidebb utak keresésére, Kosaraju algoritmusát erősen összefüggő komponensek meghatározására, Tarjan algoritmusát elvágó pontok és élek keresésére, Euler-séta keresését stb. Az optimalizálási problémák bináris kereséssel való megoldása is szignifikánsan jelenik meg. A statisztika alapján az látható, hogy alig van olyan feladat, amelynek megoldásához nem szükséges valamilyen összetettebb „tankönyvi” algoritmus, vagy pedig valamilyen algoritmikus stratégia használata.

Nagyjából minden feladathoz kell egy- vagy kétdimenziós tömb, vagy nyújtózkodó tömb. Fontos újdonság az előző korcsoportokhoz képest a prioritási sor gyakori alkalmazása. Feltüntettük még azt is, hogy sok feladathoz szükséges valamilyen egyszerűbb rekordot használni (C++ struct, pair stb.)

A gráfok között az irányított gráfok gyakoribbak, mint a fiatalabb korosztályban, és a súlyozott gráfok is szignifikánsan részben fordulnak elő. A kombinatorika változatlanul jelen van, évenként átlagban egy feladatban. Újdonságként jelenik meg a matematikai témakörök között a korábban említett mélyebb geometriai ismeretek, ami elsősorban koordináta-geometriát jelent.

Algoritmikus stratégiák	
Mohó algoritmusok	25%
Dinamikus programozás	20%
Geometria: sóprés, görgetés	10%
Rekurzív kiszámítás	5%
Szimuláció	5%
Bináris keresés	5%
Adatszerkezetek	
Tömb (egy dimenziós)	60%
Nyújtózkodó tömb	35%
Gráfrepresentációk: szomszédsági lista	35%
Két dimenziós tömb	20%
Egyszerű rekord	15%
Prioritási sor	10%
Algoritmusok	
Mélyégi gráfbejárás és alkalmazásai	20%
Rendezés	20%
Szélességi gráfbejárás	15%
Haladó gráfalgoritmusok	10%
Matematika	
Irányított gráf	15%
Irányítatlan, súlyozatlan gráf	10%
Kombinatorika	10%
Geometriai fogalmak és számítások	10%
Fa gráf	10%
Súlyozott gráf	5%

4. táblázat: Az Informatika OKTV (programozás kategória) leggyakoribb témakörei

## 7. Köszönetnyilvánítás

A cikk és a Nemes Tihamér Syllabus elkészítéséhez nagyban hozzájárult a már sokszor hivatkozott táblázat az elmúlt évek feladatairól [3]. Ennek elkészítésében számos szakköri és különórai tanítványom segített: Bata Zsombor, Bukva Dávid, Csontos Dávid, Czanik Pál, Deák Bence, Fadgyas Péter, Gyimesi Péter, Harmati László, Hervay Bence, Horcsin Bálint, Horváth Boldizsár, Juhász-Molnár Erik, Kerekes Boldizsár, Máté Lőrinc, Molnár Bálint, Nagy Nándor, Noszály Áron, Reimann Kristóf, Sárvári Tibor, Soós Máté, Szabó Kornél, Székely Milán, Szilágyi Balázs, Tiszay Dávid, Tóth Balázs, Várkonyi Zsombor. Nekik ezúton is szeretném megköszönni a munkájukat.

Az elkészült Nemes Tihamér Syllabus javaslat pedig reményeim szerint nagy hasznára lesz a jövőben mindenkinek, aki valamilyen módon érintett a versenyben, és emeli a verseny minőségét és színvonalát.



## Irodalom

1. Nikházy László: *NT-Syllabus*, GitHub repository.  
<https://github.com/niklaci/NT-Syllabus> (utoljára megtekintve: 2019.11.16.)
2. M. Forišek: *IOI.Syllabus*. (2018)  
<https://ioinformatics.org/files/ioi-syllabus-2018.pdf> (utoljára megtekintve: 2019.11.16.)
3. *NT feladatok témakörökkel*, Google táblázat.  
<http://bit.ly/nt-feladatok> (utoljára megtekintve: 2019.11.16.)
4. *A Nemes Tihamér NITV tárgya, követelményei*.  
<http://nemes.inf.elte.hu/index.html>
5. Szlávi Péter, Zsakó László: *Módszeres programozás: programozási tételék*. Mikrológia 19, ELTE IK (2008)