Bevezetés

Amiről beszélni fogok, az sem bonyolult, sem perlekedő nem lesz. JOHN L. AUSTIN

Ez a könyv *algoritmusokról* szól. Az algoritmus fogalmáról – azt gondoljuk – már minden olvasónknak van valamilyen képe. Pontos meghatározásra nem szeretnénk vállalkozni, elsősorban azért nem, mert a szó a szakmán belül is több különböző értelemben használatos. E változatok közös magjának érzékeltetésére olyan szinonimákat említhetünk, mint *eljárás*, *recept*, *módszer*. Jól meghatározott lépések egymásutánja, amelyek már elég pontosan, egyértelműen megfogalmazottak ahhoz, hogy gépiesen végrehajthatók legyenek.

Az „algoritmus" szó eredetéért a IX. századba és az Arab Kalifátus csillogó fővárosába, Bagdadba kell visszatekintenünk. Itt dolgozott az Al Khvarizmi (Khorezmből való) néven ismert Mohamed ibn Músza, aki több nevezetes tudományos könyvet írt. Ezek egyike (a latin fordításban fennmaradt *De Numero Indorum*) a decimális számokkal való hindu eredetű számolási eljárásokat írja le. Lényegében azokat, amelyeket az elemi iskolában tanultunk az egészekkel való alapműveletek elvégzésére. Elsősorban világos stílusának és a pontos, részletes indoklásoknak köszönhetően Al Khvarizmi könyvét évszázadokig forgatták a középkori Európában. Az algoritmus szó a szerző nevének a latin fordítások által eltorzított változata. Sokáig ezeket az indiai eredetű számolási szabályokat értették alatta.

A számítógépes algoritmus fogalma szorosan kapcsolódik a program fogalmához. Az egyik lehetséges megközelítés, amivel később részletesebben is megismerkedünk, lényegében azonosítja a kettőt. Az algoritmusok elmélete, ahogy ma állnak a dolgok, jóval kevesebbet vállal fel, mint amennyit az előbbi ambiciózus meghatározás sugall. Ez a terület főként a kisebb, építőkő jellegű problémákkal foglalkozik. Nemigen szokás algoritmusnak nevezni egy több tízezer utasításból álló adatbáziskezelő programot vagy annak logikai vázát.

Az algoritmika konstruktív iránya elsősorban akkora feladatokkal foglalkozik, melyek megoldása legfeljebb néhány száz C-sorban kódolható. A megoldást jelentő módszerek tervezésének nélkülözhetetlen része a módszerek elemzése. Az elemzés során arra keresünk választ, hogy algoritmusaink mennyire hatékonyak.

Az összetett, igazán nagyméretű feladatokkal a *szoftvertechnológia* (a szokásos angol szakkifejezésekkel *software engineering*, illetve *software technology*) foglalkozik. A problémák méretének növekedtével egészen más kérdések kerülnek előtérbe. Ilyenek például a tervezendő rendszer áttekinthetősége, felbonthatósága emberszabású részekre, az így kapott darabok összeillesztése, tesztelése, stb. Mi itt nem foglalkozunk ezekkel a kérdésekkel.

Algoritmusokkal és adatszerkezetekkel kapcsolatos ismeretekre, készségekre szüksége van mindenkinek, aki komolyan foglalkozik programozással és programok tervezésével. Ennek megfelelően kialakult egy eléggé letisztult törzsanyag, amit világszerte oktatnak a számítástechnikai, informatikai képzést nyújtó egyetemi szakokon. Elsődleges célunk volt ennek az anyagnak a feldolgozása.

A bevezető jellegű első fejezetben egyszerű példákon keresztül igyekeztünk érzékeltetni az algoritmusok tervezésének és elemzésének folyamatát. Ezek játékos, könnyen áttekinthető példák, amelyek segítségével az olvasó megismerkedhet a terület szemléletének, stílusának alapjaival. Ennek a résznek a fő célja az anyag befogadásához szükséges beállítottság kialakítása.

A 2-5. és részben a 6. fejezetben a ma már klasszikusnak számító alapvető adatszerkezetekkel és algoritmusaikkal foglalkozunk. A központi témák itt a rendezés és a keresés. A legérdekesebb rendező algoritmusok tárgyalása után a fajellegű, majd pedig a hash-elésen alapuló tárolási/keresési technikákról lesz szó. Ezt követően az információtömörítés két alapvető módszerét vesszük szemügyre (5. fejezet). A hatodik fejezet gráfokkal kapcsolatos algoritmusainak egy részét is szokás az adatszerkezetek témakörébe sorolni. Ilyenek a gyors bejáró módszerek és a rövid utak keresésére szolgáló eljárások. A gráfokról szóló részben olyan problémák is helyet kaptak, amelyek a kombinatorikus optimalizálás alapjaihoz tartoznak (hálózati folyamok, párosítások).

A hetedik fejezetet a Turing gépeknek szenteltük. Ennek az alapvető gépmodellnek az ismertetése után a kiszámíthatóság elemei következnek (rekurzivitás, eldönthetetlenség). Itt foglalkozunk a Kolmogorov-bonyolultság első tulajdonságaival, és itt kapott helyet a közvetlen elérésű gép definíciója is.

A nyolcadik fejezetbe kerültek az algoritmusok bonyolultságával kapcsolatos alapvető eredmények. Az idő- és tárkorlátokkal megadott nyelvosztályok bevezetése után az NP feladatosztály tárgyalására térünk. Komoly figyelmet szentelünk az osztály nehéz feladatainak, az NP-teljes nyelveknek.

A kilencedik fejezetben algoritmustervezési módszerekkel foglalkozunk. Olyan általános technikák ezek, amelyek feladatok széles körére alkalmazhatók. Példákon keresztül mutatunk be néhány ilyen megközelítést. Ezek a mohó módszer, az elágazás és korlátozás, a dinamikus programozás és a prekondícionálás. Fontosságuknak megfelelően kiemelten foglalkozunk a véletlent használó mód-

szerekkel.

Az utolsó fejezetben rövid ízelítőt adunk a kriptográfiából, az illetéktelen hozzáféréssel szemben biztonságos kommunikáció elméletéből. Ismertetjük a gyakorlatban is népszerű RSA kriptográfiai rendszert; ez lehetőséget ad több korábban tárgyalt fogalom, ismeret alkalmazására.

Az anyag feldolgozásakor építünk az elemi függvényekkel, gráfokkal és egészek oszthatóságával kapcsolatos alapvető tényekre. Feltételezzük még, hogy az olvasó ismeri egy általános célú programozási nyelv (mint pl. a Pascal vagy a C) utasításait.

A könyvben $\mathbb{Z}$ jelöli az egész számok és $\mathbb{R}$ a valós számok összességét. A nemnegatív egész, illetve valós számok halmazára a $\mathbb{Z}^+$, illetve $\mathbb{R}^+$ jelekkel hivatkozunk. Ha $f(x_1, x_2, \ldots, x_k)$ és $g(x_1, x_2, \ldots, x_k)$ az $(\mathbb{R}^+)^k$ egy részhalmazán értelmezett valós értékeket felvevő függvények, akkor $f = O(g)$ jelöli azt a tényt, hogy vannak olyan $c, n > 0$ állandók, hogy $|f(x_1, x_2, \ldots, x_k)| \leq c|g(x_1, x_2, \ldots, x_k)|$ teljesül, ha $x_i \geq n$ minden $i = 1, 2, \ldots, k$ esetén.

Legyenek $f(n)$ és $g(n)$ a pozitív egészeken értelmezett valós értékű függvények. Ekkor az $f = o(g)$ jelöléssel rövidítjük azt, hogy $f(n)/g(n) \to 0$, ha $n \to \infty$.

Gyakran fogunk *tömbökkel* dolgozni, mégpedig legtöbbször természetes számokkal indexelt tömbökkel. Például $A[i : j]$ jelenti az A elnevezésű tömbnek az i indexűtől a j indexű eleméig terjedő részét; $A[i]$ pedig a tömb i-indexű elemére utal.

Végezetül köszönetet mondunk mindazoknak, akik bátorításukkal, a kézirathoz fűzött megjegyzéseikkel, tanácsaikkal támogatták munkánkat. Különösen sokat köszönhetünk Babai Lászlónak, Bródy Ferencnek, Demetrovics Jánosnak, Dömötör Adrienne-nek, Elekes Györgynek, Friedl Katalinnak, Herczog Sándornénak, Kovács Annamáriának, Szabó Lászlónak, Vank Ágnesnek és Vámos Tibornak.

Budapest, 1998. július 20.

A szerzők