

Rónyai Lajos – Ivanyos Gábor – Szabó Réka

Algoritmusok

AZ INFORMATIKA ELMÉLETE

Rónyai Lajos
Ivanyos Gábor
Szabó Réka

Algoritmusok

© Ivanyos Gábor, Rónyai Lajos, Szabó Réka, Typotex, Budapest, 1998, 2020

Változatlan utánnyomás

Engedély nélkül semmilyen formában nem másolható!

Lektorálta: Elekes György

ISBN 978 963 493 101 0

ISSN 1787-3045

Kedves Olvasó!

Köszönjük, hogy kínálatunkból választott olvasnivalót!

Újabb kiadványainkról és akcióinkról a www.typotex.hu
és a facebook.com/typotexkiado oldalakon értesülhet.

Typotex Kiadó

Alapította Votisky Zsuzsa, 1989

A kiadó az 1795-ben alapított Magyar Könyvkiadók
és Könyvterjesztők Egyesülésének tagja.

Felelős kiadó: Németh Kinga

Főszerkesztő: Horváth Balázs

Szedés és tördelés: Ivanyos Gábor

A borítót készítette: Szalay Éva

Készült a Kódex Könyvgyártó Kft. nyomdájában

Felelős vezető: Marosi Attila

Tartalomjegyzék

Bevezetés	9
1. Algoritmikus problémák megoldása	12
1.1. A feladattól a modellig	14
1.1.1. Közlekedési lámpák ütemezése	14
1.1.2. Arthur király civilizációs törekvései	16
1.2. Algoritmusok	18
1.2.1. Szuperforrás keresése	18
1.2.2. Hosszú egészek párhuzamos összeadása	21
2. Rendezés	25
2.1. Keresés rendezett halmazban	27
2.2. Összehasonlítás alapú rendező módszerek	29
2.2.1. Buborék-rendezés	30
2.2.2. Beszúrásos rendezés	31
2.2.3. Egy alsó becslés	33
2.2.4. Összefésüléses rendezés	35
2.2.5. A kupac adatszerkezet és a kupacos rendezés	37
2.2.6. Gyorsrendezés	42
2.2.7. A k -adik elem kiválasztása	44
2.3. Kulcsmanipulációs rendezések	46
2.3.1. Ládarendezés (binsort)	47
2.3.2. Radix rendezés	48
2.4. A Batchter-féle páros-páratlan összefésülés	50
2.5. Külső táruk tartalmának rendezése	51
2.5.1. Összefésüléses rendezés külső tárukon	52
3. Keresőfák	57
3.1. Bináris fák	58

3.2.	Bináris keresőfák, naiv algoritmusok	60
3.3.	2-3-fák	64
3.4.	B -fák	69
3.5.	AVL -fák	71
3.6.	További megjegyzések kiegyensúlyozott fákról	77
3.7.	Egy önszervező megoldás: az S -fák	80
3.8.	Szófák	83
4.	Hash-elés és szekvenciális keresés	86
4.1.	A hash-elés alapjai	87
4.1.1.	Vödrös hash-elés	88
4.1.2.	Nyitott címezés	90
4.2.	Hash-függvények	95
4.3.	Hash-elés kontra keresőfák	98
4.4.	Szekvenciális keresés	99
5.	Információtömörítés	102
5.1.	A Huffman-kód	102
5.2.	A Lempel–Ziv–Welch-módszer	106
6.	Gráfalgoritmusok	110
6.1.	Bevezetés	110
6.1.1.	Alapfogalmak, jelölések	111
6.1.2.	Gráfok ábrázolásai	113
6.2.	A legrövidebb utak problémája (egy forrásból)	115
6.2.1.	Dijkstra módszere	116
6.2.2.	A Bellman—Ford-módszer	120
6.3.	Floyd módszere az összes csúcspár közötti távolság meghatározására	122
6.3.1.	Floyd módszere	123
6.3.2.	Tranzitív lezárás	125
6.3.3.	Egy alkalmazás: centrum keresése irányított gráfban . . .	126
6.4.	Mélységi bejárás	127
6.4.1.	Irányított gráfok mélységi bejárása	128
6.4.2.	Irányított körmentes gráfok (DAG-ok)	135
6.4.3.	Erősen összefüggő (erős) komponensek	141
6.4.4.	Irányítatlan gráfok mélységi bejárása	144
6.5.	A szélességi bejárás	146
6.6.	Minimális költségű feszítőfák	151
6.6.1.	Prim módszere	156
6.6.2.	Kruskal módszere	160

6.6.3.	Az UNIÓ-HOLVAN adatszerkezet	162
6.6.4.	Megjegyzések	166
6.7.	Maximális párosítás páros gráfokban	167
6.7.1.	A magyar módszer	168
6.8.	Maximális folyamatok hálózatokban	171
6.8.1.	Kapcsolat a minimális vágással: a Ford–Fulkerson-tétel . .	174
6.8.2.	A Ford–Fulkerson-algoritmus	178
6.8.3.	Edmonds–Karp és Dinic algoritmusai	179
6.8.4.	Alkalmazások	183
7.	Turing-gépek	190
7.1.	A Turing-gép fogalma	191
7.2.	Idő- és tárigény	197
7.3.	Néhány szimuláció	198
7.4.	A kiszámíthatóság alapfogalmai	202
7.5.	Az univerzális Turing-gép	205
7.6.	Alapvető kiszámíthatatlansági tételek	208
7.6.1.	A diagonális nyelv – egy nem rekurzív felsorolható nyelv	208
7.6.2.	Az univerzális nyelv – egy rekurzív felsorolható, de nem rekurzív nyelv	209
7.7.	Összefüggések a kiszámíthatósági fogalmak között	210
7.7.1.	Rekurzivitás és rekurzív felsorolhatóság	211
7.7.2.	Függvények és halmazok (nyelvek)	213
7.8.	További eldönthetetlen problémák	215
7.8.1.	A Megállási probléma (Halting problem)	216
7.8.2.	Hilbert 10. problémája	217
7.8.3.	A Dominóprobléma	222
7.8.4.	Post megfeleltetési problémája	226
7.8.5.	Egy nyitott kérdés: a kongruens számok felismerése . . .	227
7.9.	Kolmogorov-bonyolultság	229
7.10.	A közvetlen elérésű gép (RAM)	239
8.	Az NP nyelvosztály	246
8.1.	Idő- és tárkorlátok	247
8.2.	Tár-idő-tétel, nevezetes nyelvosztályok	250
8.3.	Nemdeterminisztikus Turing-gépek; az NP nyelvosztály	255
8.4.	Néhány NP-beli nyelv	262
8.4.1.	3 színnel színezhető gráfok	262
8.4.2.	Hamilton-körrel rendelkező gráfok	262
8.4.3.	Síkba rajzolható gráfok	263

8.4.4.	A prímszámok nyelve	265
8.4.5.	A felismerés és a keresés kapcsolata (prímtényező felbon- tás)	266
8.5.	Karp-redukció, NP-teljesség	268
8.6.	A SAT nyelv és a Cook–Levin-tétel	272
8.7.	További NP-teljes feladatok	275
8.7.1.	Konjunktív normálformájú formulák kielégítheósége és a 3-SAT	276
8.7.2.	3 színnel színezhető gráfok	278
8.7.3.	Maximális méretű független pontrendszer gráfokban . . .	280
8.7.4.	A 3 dimenziós házasság és az X3C feladat	282
8.7.5.	Hamilton-kört tartalmazó gráfok és az Utazó ügynök prob- léma	285
8.7.6.	A Hátizsák feladat és néhány más rokon probléma	289
8.7.7.	Lineáris programozás	292
8.7.8.	Minimális késésszámú ütemezés	296
9.	Néhány általános algoritmus-tervezési módszer	297
9.1.	Elágazás és korlátozás	299
9.2.	Dinamikus programozás	302
9.3.	Közelítő algoritmusok	305
9.4.	Véletlent használó módszerek	310
9.4.1.	Az RP nyelvosztály	314
9.4.2.	Prímtesztelés	316
9.4.3.	Nagy prímszám keresése	320
9.5.	Prekondicionálás	321
10.	Nyilvános kulcsú titkosítások	328
10.1.	Kriptográfia - a titkosítások tudománya	328
10.2.	Nyilvános kulcsú kriptográfia	331
10.3.	A Rivest–Shamir–Adleman- (RSA-) kód	332
	Tárgymutató	336

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 (Khorézmből 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, építő 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 algoritmusainkkal foglalkozunk. A központi témák itt a rendezés és a keresés. A legérdekesebb rendező algoritmusok tárgyalása után a fajlegű, 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 folyamatok, 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 prekondicionálás. Fontosságuknak megfelelően kiemelten foglalkozunk a véletlent használó mód-

szerekkel.

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

Az anyag feldolgozásakor építünk az elemi függvényekkel, gráfokkal és egészek oszthatóságával kapcsolatos alapveő 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 $f(x_1, x_2, \dots, x_k)$ és $g(x_1, x_2, \dots, x_k)$ az $(\mathbb{R}^+)^k$ egy részhalmazán értelmezett valós értékeket felveő függvények, akkor $f = O(g)$ jelöli azt a tényt, hogy vannak olyan $c, n > 0$ állandók, hogy $|f(x_1, x_2, \dots, x_k)| \leq c|g(x_1, x_2, \dots, x_k)|$ teljesül, ha $x_i \geq n$ minden $i = 1, 2, \dots, k$ esetén.

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

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 kéziratához fűzött megjegyzéseikkel, tanácsaikkal támogatták munkánkat. Különösen sokat köszönhetünk Babai Lászlónak, Bródy Ferencnek, Demetrovics Jánosnak, Dömötör Adrienne-nek, Elekes Györgynek, Friedl Katalinnak, Herczog Sándornak, Kovács Annamáriának, Szabó Lászlónak, Vank Ágnesnek és Vámos Tibornak.

Budapest, 1998. július 20.

A szerzők