

Algoritmuselmélet

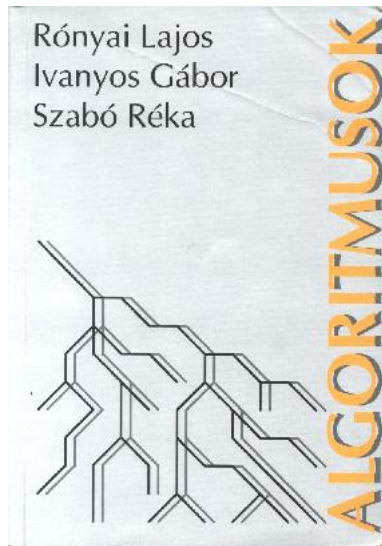
Függvények nagyságrendje, elágazás és korlátozás, dinamikus programozás

Katona Gyula Y.

Számítástudományi és Információelméleti Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

1. előadás

Források



- Rónyai Lajos–Ivanyos Gábor–Szabó Réka: **Algoritmusok**, TYPOT_EX, 1999
- Feladatgyűjtemény: <http://www.cs.bme.hu/~kiskat>
- Egyéb információk, hirdetések, fóliák ugyanitt.

Követelmények

- **ZH:** Egy évközi zárthelyi lesz, ezen lehet az aláírást megszerezni. A zh várhatóan 6 feladatból és egy ráadásból áll, mindegyik 10 pontot ér. Az elégségeshez 40%-os teljesítmény (azaz várhatóan minimum 24 pont) kell. A zh eredménye beszámít a végső jegybe. Az aláírásért az évközi zárthelyit legalább elégségesre meg kell írni.
- **PótZH:** Pótzárthelyi a megadott időpontban lesz, anyaga, szabályai, mint a zh-nál. Ennek eredménye felülírja a zh eredményét. Ha a zh elérte az elégséges szintet, de a pótzh nem, akkor az aláírás megmarad, de a pontszám az aláíráshoz szükséges minimumra (40%) csökken. A pótlási héten lesz még egy alkalom kizárólag azoknak, akik még nem szerezték meg az aláírást (anyaga ugyanaz, mint a zh anyaga).

Követelmények

- **Vizsga:** A vizsga is ugyanannyi feladatból áll, mint a zh, de természetesen az egész anyagból. Ha ezen nem sikerül elérni 40%-ot, a jegy elégtelen. Amennyiben legalább elégséges az eredmény, akkor a vizsgadolgozatot 60, a zh (illetve pótzh vagy pótpótzh) eredményét 40% súllyal számítjuk be a vizsgajegybe. Az így kapott legalább elégséges jegyen az eredményhirdetéskor lehetőség van szóbeli vizsgával egy jegyet módosítani (fel vagy le).

Algoritmus fogalma

- Egyelőre nem definiáljuk rendesen az algoritmus fogalmát.
- Eljárás, recept, módszer.
- Jól meghatározott lépések egymásutánja, amelyek már elég pontosan, egyértelműen megfogalmazottak ahhoz, hogy gépiesen végrehajthatók legyenek.

A szó eredete

Al Khvarizmi (Mohamed ibn Músza) bagdadi matematikus a IX. században könyvet írt az egészekkel való alapműveletek végzéséről.

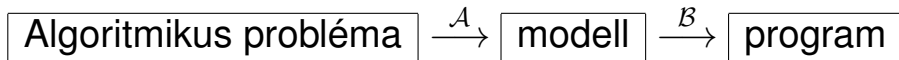
algorithmus \leftrightarrow számítógép program

valós probléma \implies absztrakt modell \implies **algorithmus** \implies program

Cél: feladatokra hatékony eljárás kidolgozása

Hatékony \implies gyors, kevés memória, kevés tárhely

Algoritmikus problémák megoldása



\mathcal{A} : pontosítás, egyszerűsítés, absztrakció, lényegtelen elemek kiszűrése, a lényeg kihámozása

Modell: sokféle lehet, elég tág, de elég egyszerű, formalizált, pontos

\mathcal{B} : hatékony algoritmus, bemenő adatok \rightarrow eredmény, érdemes foglalkozni a kapott algoritmus *elemzésével*, *értékelésével*, megvizsgálva, hogy a módszer mennyire hatékony

Példa: Buborék-rendezés leírása

Input: $A[1 : n]$ (rendezetlen) tömb

Ha valamely i -re $A[i] > A[i + 1]$, akkor a két cella tartalmát kicseréljük.

A tömb elejéről indulva, közben cserélgetve eljutunk a tömb végéig.

Ekkor a legnagyobb elem $A[n]$ -ben van. Ismételjük ezt az $A[1 : n - 1]$ tömbre, majd az $A[1 : n - 2]$ tömbre, stb.

Példa: Buborék-rendezés pszeudokódja

procedure buborék

(az $A[1 : n]$ tömböt növekvően (nem csökkenően) rendezi *)*

for ($j = n - 1, j > 0, j := j - 1$) **do**

for ($i = 1, i \leq j, i := i + 1$) **do**

 { ha $A[i + 1] < A[i]$, akkor cseréljük ki őket. }

Példa: Buborék-rendezés helyessége

Állítás

A külső ciklus egy-egy iterációja után, a belső ciklus lefutása után az $A[j + 1 : n]$ tömb már rendezett és az $A[j + 1 : n]$ tömb egyik eleme sem kisebb az $A[1 : j]$ tömb egyik eleménél sem.

Bizonyítás.

Indukcióval. Az elején nyilván OK.

A belső ciklus az $A[1 : j]$ legnagyobb elemét teszi az $A[j + 1]$ helyre, ez nem nagyobb, mint $A[j + 1 : n]$ többi eleme és nem kisebb, mint $A[1 : j]$ többi eleme. □

Állítás

Ha $j = 1$, akkor rendezett lesz a tömb az előző állítás miatt.

Példa: Buborék-rendezés hatékonysága

összehasonlítások száma: $n - 1 + n - 2 + \dots + 1 = \frac{n(n-1)}{2}$

cserék száma: $\leq \frac{n(n-1)}{2}$

Java animáció: Buborék rendezés

Video animáció: Buborék rendezés

Video tánc: Buborék rendezés

Java animáció: Buborék rendezés

Milyen hatékony egy algoritmus?

- Legtöbbször csak a lépésszám nagyságrendje érdekes. Hogyan függ a lépésszám az input méretétől?
- Az input méretét legtöbbször n -nel jelöljük.
- A lépésszám ennek egy f függvénye, azaz ha n méretű az input, akkor az algoritmus $f(n)$ lépést végez.
- Igazából az f függvény az érdekes.
- $100n$ vagy $101n$, általában mindegy
- n^2 vagy n^3 már sokszor nagy különbség, de néha mindegy
- n^2 vagy 2^n már mindig nagy különbség

Például

Kérdés

Egy 10^{10} művelet/mp sebességű számítógép mennyi ideig dolgozik, ha $f(n)$ műveletet kell végrehajtani n méretű bemenetre?

n	$f(n)$ n	n^2	$\log_{10} n$	2^n	$n!$
10	10^{-9}	10^{-8}	10^{-10}	$1,02 \cdot 10^{-7}$	$3,6 \cdot 10^{-4}$
10^2	10^{-8}	10^{-6}	$2 \cdot 10^{-10}$	$4 \cdot 10^{12}$ év	$2,9 \cdot 10^{140}$ év
10^6	10^{-4}	100	$6 \cdot 10^{-10}$	$3,1 \cdot 10^{301.012}$ év	$2,6 \cdot 10^{5.565.691}$ év
10^9	0,1	3,1 év	$9 \cdot 10^{-9}$	sok év	sok év

Függvények nagyságrendje

Definíció

Ha $f(n)$ és $g(n)$ az \mathbb{R}^+ egy részhalmazán értelmezett, valós értékeket felvevő függvények, akkor $f = O(g)$ jelöli azt a tényt, hogy vannak olyan $c, n_0 > 0$ állandók, hogy $|f(n)| \leq c|g(n)|$ teljesül, ha $n \geq n_0$.

Példák

- $100n + 300 = O(n)$

Biz: $100n + 300 \leq 100n + n \leq 101n \leq cn$, ha $n \geq 300$, $c = 101$

- $5n^2 + 3n = O(n^2)$

Biz: $5n^2 + 3n \leq 5n^2 + 3n^2 \leq 8n^2 \leq cn^2$, ha $n \geq 100$, $c = 8$

- $n^4 + 5n^3 = O(n^5)$

Biz: $n^4 + 5n^3 \leq 6n^4 \leq n^5 \leq cn^5$, ha $n \geq 6$, $c = 1$

Példák

- $n^{1000} = O(2^n)$

Biz: Teljes indukcióval, legyen $c = 1, n_0 = 10^6$.

$n = 10^6$ -re igaz, mert $10^{6000} \leq (2^4)^{6000} \leq 2^{10^6}$.

Tegyük fel, hogy k -ra igaz.

Felhasználjuk, hogy ha $k \geq 10^6$ akkor

$$\binom{1000}{i} \leq 1000^i = 1000 \cdot 1000^{i-1} \leq 1000^{i-1} \cdot 1000^{i-1} = \\ = (10^6)^{i-1} \leq k^{i-1}.$$

$$(k+1)^{1000} = k^{1000} + \dots + \binom{1000}{i} k^{1000-i} + \dots \leq k^{1000} + \dots + \\ k^{i-1} k^{1000-i} + \dots \leq k^{1000} + 1000k^{999} \leq 2 \cdot k^{1000} \leq 2 \cdot 2^k = 2^{k+1},$$

ha $k \geq 10^6$.

- $\log_2^{1000}(n) = O(n)$

Biz: Mivel a logaritmus függvény monoton nő, vehetjük a fentiek logaritmusát.

- $2^n = O(n!)$

- $n! = O(n^n)$

Példák

Igaz-e, hogy $n^2 = O(n)$?

Nem.

Biz: Indirekt, tegyük fel, hogy létezik olyan c, n_0 , hogy $n^2 \leq cn$ teljesül minden $n \geq n_0$ esetén.

Ekkor $n \leq c$ teljesül minden $n \geq n_0$ esetén, ami nyilván nem igaz, ha $n > c$.

Függvények nagyságrendje

Definíció

Ha $f(n)$ és $g(n)$ az \mathbb{R}^+ egy részhalmazán értelmezett, valós értékeket felvevő függvények, akkor $f = \Omega(g)$ jelöli azt a tényt, hogy vannak olyan $c, n_0 > 0$ állandók, hogy $|f(n)| \geq c|g(n)|$ teljesül, ha $n \geq n_0$.

Például:

- $100n - 300 = \Omega(n)$, hiszen $n > 300$, $c = 99$ -re teljesülnek a feltételek
- $5n^2 - 3n = \Omega(n^2)$
- $n^4 - 5n^3 = \Omega(n^4)$
- $2^n = \Omega(n^{1000})$

Függvények nagyságrendje

Definíció

Ha $f = O(g)$ és $f = \Omega(g)$ is teljesül, akkor $f = \Theta(g)$.

Például:

- $100n - 300 = \Theta(n)$
- $5n^2 - 3n = \Theta(n^2)$
- $n^4 - 5n^3 = \Theta(n^4)$
- $1000 \cdot 2^n = \Theta(2^n)$

Függvények nagyságrendje

Definíció

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

$$\frac{f(n)}{g(n)} \rightarrow 0, \text{ ha } n \rightarrow \infty.$$

Például:

- $100n + 300 = o(n^2)$, hiszen $\frac{100n+300}{n^2} \rightarrow 0$ ha $n \rightarrow \infty$
- $5n^2 + 3n = o(n^3)$
- $n^4 + 5n^3 = o(n^4 \log_2 n)$
- $n^{1000} = o(2^n)$