

Algoritmuselmélet

Függvények nagyságrendje

Katona Gyula Y. / Vizer Máté

Számítástudományi és Információelméleti Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

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 alpműveletek végzéséről.

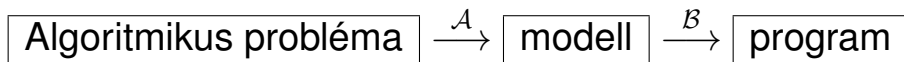
algorithmus \leftrightarrow számítógép program

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

Egy egyszerű algoritmus

Feladat

Input: Csupa különböző egész számokból álló $A[1 : n]$ tömb

Output: A tömb legkisebb elemének indexe és értéke.

- Egy lépés most a tömb két elemének összehasonlítása.
- Hány összehasonlítás kell a legrosszabb esetben?

Szöveges leírás:

- Végigmegyek a tömbön és közben megjegyzem, hogy mi volt az eddigi legkisebb érték.
- Minden lépésben az aktuális minimumot és az aktuális elemet hasonlítom össze.

Pszudokód:

```
min := A[1]; ind := 1
for i=2,..., n do
|   if A[i] < min then
|   |   min := A[i]; ind := i
|   end if
end for
return min, ind
```

Minimumkeresés

Állítás

Az algoritmus helyes kimenetet ad.

Bizonyítás.

Amint elér a minimumhoz, *min* értéke ez lesz. A további összehasonlításokkor ez már nem tud változni.

Állítás

Az algoritmus lépésszáma $n - 1$.

Bizonyítás.

A for ciklus $n - 1$ -szer fut le, minden alkalommal 1 összehasonlítást végez. **Ez bármilyen inputra így lesz.**

Minimumkeresés

Tétel

n elem közül a minimális kiválasztásához legrosszabb esetben $n - 1$ összehasonlítás kell.

Bizonyítás.

$n - 1$ összehasonlítás mindig elég: Az előbb mutattunk ilyen algoritmust.

$n - 1$ összehasonlításnál kevesebb nem mindig elég:

- Kezdetben az n elem bármelyike szóba jön, mint minimum.
- Egy összehasonlítás garantálja, hogy a nagyobbik már nem jön szóba. (De a kisebbikről nem ad új információt.)
- Csak akkor találtuk meg a minimumot, ha már csak 1 elem jön szóba.
- Ezért mindenképpen kell legalább $n - 1$ összehasonlítás.



Feladat

Input: Csupa különböző egész számokból álló $A[1 : n]$ tömb

Output: A nagyság szerint rendezett tömb.

- Egy lépés most a tömb két elemének összehasonlítása vagy felcserélése.
- Hány lépés kell a legrosszabb esetben?

Kiválasztásos rendezés

Szöveges leírás:

- Az i -edik körben megkeressük az $A[i : n]$ tömb minimumát
- A minimumot megcseréljük $A[i]$ -vel.

Pszudokód:

```
for  $i = 1, \dots, n - 1$  do  
     $(min, ind) := \text{MIN}(A[i : n])$   
    CSERE( $A[ind], A[i]$ )  
end for
```

Kiválasztásos rendezés

Állítás

Az algoritmus helyes kimenetet ad.

Bizonyítás.

Indukcióval: Tegyük fel, hogy az i -edik kör előtt az $A[1 : i - 1]$ a legkisebb $i - 1$ elemet tartalmazza. Ekkor az i -edik lépésben megkeressük az i -edik legkisebbet és az kerül $A[i]$ -be. Tehát igaz lesz, hogy az $A[1 : i]$ a legkisebb i elemet tartalmazza. □

Állítás

Az algoritmus lépésszáma legfeljebb $\frac{n^2}{2} + \frac{n}{2} - 1$, legalább $\frac{n^2}{2} - \frac{n}{2}$, legrosszabb esetben $\frac{n^2}{2} + \frac{n}{2} - 1$.

Bizonyítás.

A for ciklus $n - 1$ -szer fut le, az i -edik körben $n - i$ összehasonlítást és legfeljebb 1 cserét végez. **Az inputtól függ, hogy lesz-e csere.**

- Összehasonlítások száma: $n - 1 + n - 2 + \dots + 2 + 1 = \frac{n(n-1)}{2}$
- Cserék száma: $\leq n - 1$
- **Lépések száma, ha soha nincs csere:** $\frac{n(n-1)}{2} = \frac{n^2}{2} - \frac{n}{2}$
- **Lépések száma, ha mindig kell csere:** $\frac{n(n-1)}{2} + n - 1 = \frac{n^2}{2} + \frac{n}{2} - 1$



Kérdés

Hogyan tudjuk jól összehasonlítani a minimumkeresés és a kiválasztásos rendezés lépésszámát?

Milyen hatékony egy algoritmus?

- Egy algoritmus lépésszáma egy konkrét inputon egy konkrét egész szám lesz.
- De legtöbbször arra vagyunk kíváncsiak, hogyan függ a lépésszám az input méretétől? (nem a konkrét inputtól)
- Az input méretét legtöbbször n -nel jelöljük.
- A lépésszámának általában azt az $f(n)$ függvényt tekintjük, amire az algoritmus egy legfeljebb n méretű az inputon **legrosszabb esetben $f(n)$ lépést végez.**
- Legtöbbször csak f nagyságrendje 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.³

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

Az univerzum kora: $\approx 1.4 \cdot 10^{10}$ év

Függvények nagyságrendje

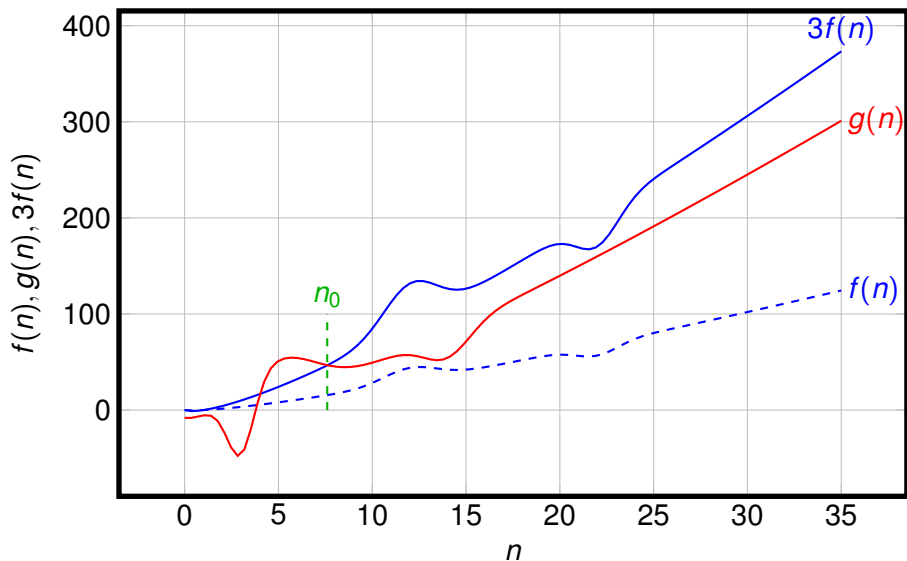
Definíció

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

Mostantól feltesszük, hogy a függvények (legalább nagy n -re) pozitívak.

Másik szokásos jelölés: $f(n) = O(g)$.

title



Példák

- $n - 1 \in O(n)$

Biz: $n - 1 \leq c \cdot n$ ha $n \geq 1$, $c = 1$

Vagyis a minimumkeresés lépésszáma $O(n)$.

- $100n + 300 \in O(n)$

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

- $5n^2 + 3n \in O(n^2)$

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

- $\frac{n^2}{2} + \frac{n}{2} - 1 \in O(n^2)$

Biz: $\frac{n^2}{2} + \frac{n}{2} - 1 \leq \frac{n^2}{2} + \frac{n}{2} \leq \frac{n^2}{2} + \frac{n^2}{2} = n^2 = c \cdot n^2$, ha $n \geq 1$, $c = 1$

Vagyis a kiválasztásos rendezés lépésszáma $O(n^2)$.

- $n^4 + 5n^3 \in O(n^5)$

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

Példák

- $n^{1000} \in O(2^n)$

Biz. érdeklődőknek: 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

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

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

ha $k \geq 10^6$.

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

Biz: Tudjuk, hogy $m^{1000} \in O(2^m)$, legyen $m = \log_2(n)$.

$$\implies \log_2^{1000}(n) \in O(2^{\log_2(n)}) = O(n)$$

- $2^n \in O(n!)$
- $n! \in O(n^n)$

Példák

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

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 $f \in \Omega(g)$ 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:

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

Függvények nagyságrendje

Definíció

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

Például:

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

Volt Prog1-en

Általános tételek

Tétel

Ha $f_1(n) \in O(g_1(n))$ és $f_2(n) \in O(g_2(n))$, akkor

- i) $f_1(n) + f_2(n) \in O(\max(g_1(n), g_2(n)))$,
- ii) $f_1(n)f_2(n) \in O(g_1(n)g_2(n))$.

Bizonyítás.

Gyakorlaton lesz.

Összefoglalás

$$O(\log n) \subset O(\log^2 n) \subset O(\log^{100} n) \subset O(n) \subset O(n^2) \subset O(n^3) \subset O(2^n) \subset \\ O(3^n) \subset O(n!) \subset O(n^n) \subset O(2^{2^n})$$

$$\Omega(\log n) \supset \Omega(n) \supset \Omega(n^2) \supset \Omega(n^3) \supset \Omega(2^n) \\ \supset \Omega(3^n) \supset \Omega(n!) \supset \Omega(n^n) \supset \Omega(2^{2^n})$$