Függvények növekedési korlátainak jellemzése

A jellemzés jól bevált eszközei az Ω, O, Θ, o és ω jelölések. Mivel az igények általában nemnegatívak, ezért az alábbi meghatározásokban mindenütt feltesszük, hogy az f és g függvények a pozitív egészek halmazán vannak értelmezve, az $f(n)$ és $g(n)$ függvényértékek pedig nemnegatív valós számok.

Az O jelöléssel aszimptotikus felső, az Ω jelöléssel aszimptotikus alsó korlátot tudunk adni, míg a Θ jelöléssel pontosan meg tudjuk adni a vizsgált függvény aszimptotikus viselkedését.

$O(g(n))$ ***(nagy ordó)*** azon $f(n)$ függvények halmazát jelenti, amelyekhez léteznek olyan c pozitív valós és n_0 pozitív egész állandók, hogy

$$\text{ha } n \geq n_0, \text{ akkor } f(n) \leq cg(n) \ . \tag{1.1}$$

Tömören:

$$O(g(n)) = \{f(n) \mid \exists c \in \mathbb{R}^+ \ \exists n_0 \in Z^+ : n \geq n_0 \rightarrow f(n) \leq cg(n)\}.$$

$\Omega(g(n))$ ***(nagy omega)*** azon $f(n)$ függvények halmazát jelenti, amelyekhez léteznek olyan c pozitív valós és n_0 pozitív egész állandók, hogy

$$\text{ha } n \geq n_0, \text{ akkor } f(n) \geq cg(n) \ . \tag{1.2}$$

$\Theta(g(n))$ ***(nagy teta)*** azon $f(n)$ függvények halmazát jelenti, amelyekhez léteznek olyan pozitív valós c_1, c_2 és pozitív egész n_0 állandók, hogy

$$\text{ha } n \geq n_0, \quad \text{akkor } c_1 g(n) \leq f(n) \leq c_2 g(n) \ . \tag{1.3}$$

Az elemzésekben arra törekszünk, hogy Θ típusú becsléseket adjunk, amihez azonos argumentumfüggvényt tartalmazó O és Ω típusú becslésekre van szükség. Ha egy becslésben hangsúlyozni akarjuk, hogy a becslés nem éles, akkor hasznos a o és a ω jelölés.

$o(g(n))$ ***(kis ordó)*** azon $f(n)$ függvények halmazát jelenti, melyekre teljesül, hogy minden pozitív valós c állandóhoz megadható egy pozitív egész n_0 úgy, hogy

$$\text{ha } \ n \geq n_0, \quad \text{akkor } f(n) \leq cg(n) \ . \tag{1.4}$$

Tömören:

$$o(g(n)) = \{f(n) \mid \forall c \in \mathbb{R}^+ \ \exists n_0 \in Z^+ : n \geq n_0 \rightarrow f(n) \leq cg(n)\} \ .$$

$\omega(g(n))$ ***(kis omega)*** azon $f(n)$ függvények halmazát jelenti, melyekre teljesül, hogy minden pozitív valós c állandóhoz megadható egy pozitív egész n_0 úgy, hogy

$$\text{ha } n \geq n_0, \text{ akkor } f(n) \geq cg(n) \ . \tag{1.5}$$

Ha $f(n) = o(g(n))$, akkor

$$\lim_{n \to \infty} \frac{f(n)}{g(n)} = 0 \ , \tag{1.6}$$

Sorszám	Növekedési korlát képlettel	Korlát típusa szóval
1	$\Theta(1)$	konstans
2	$\Theta(\lg^{\star} n)$	majdnem konstans
3	$o(\lg n)$	szublogaritmikus
4	$\Theta(\lg n)$	logaritmikus
5	$\Theta(\lg^{\Theta(1)} n)$	polilogaritmikus
6	$\omega(\lg n)$	szuperlogaritmikus
7	$o(n)$	szublineáris
8	$\Theta(n)$	lineáris
9	$\omega(n)$	szuperlineáris
10	$\Theta(n^2)$	négyzetes
11	$\Theta(n^3)$	köbös
12	$o(n^{\Theta(1)})$	szubpolinomiális
13	$\Theta(n^{\Theta(1)})$	polinomiális
14	$\omega(n^{\Theta(1)})$	szuperpolinomiális

1.1. táblázat. Függvények növekedésének leggyakoribb korlátai.

és ha $f(n) = \omega(g(n))$, akkor

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty \,. \tag{1.7}$$

A logaritmikus tényezőket gyakran elhanyagolják. Azt mondjuk, hogy $f(n) = \widetilde{O}(g(n))$ ***gyenge ordo*** ha léteznek olyan c pozitív valós, k és n_0 pozitív egész állandók úgy, hogy, ha $n \geq n_0$, akkor $0 \leq f(n) \leq cg(n)\lg^k(n)$.

Tömören:

$$\widetilde{O}(g(n)) = \{f(n) \mid \exists c \in \mathbb{R}^+ \; \exists n_0 \in Z^+ : n \geq n_0 \rightarrow f(n) \leq cg(n)\} \,.$$

Hasonlóképpen definiálhatók a $\widetilde{\Omega}(g(n))$ és $\widetilde{\Theta}(g(n))$ jelölések is.

Az 1.1. táblázatban összefoglaltuk a függvények növekedésével kapcsolatban leggyakrabban használt jelöléseket és kifejezéseket. A táblázatban szereplő korlátok tetszőleges erőforrással kapcsolatban használhatók – elemzéseinkben azonban elsősorban lépésszámokra vonatkoznak.

A táblázatban a *szub* kifejezést *kisebb,* a *szuper* kifejezést pedig *nagyobb* értelemben használtuk. Érdemes megemlíteni, hogy szokás a *kisebb vagy egyenlő,* illetve a *nagyobb vagy egyenlő* értelmű használat is.

A könyvben a szuperpolinomiális kifejezés szinonimájaként fogjuk használni az *exponenciális* jelzőt. Ezzel az ***exponenciális futási időt*** a matematikában szokásosnál tágabban definiáltuk: ha egy algoritmus futási ideje felülről nem korlátozható polinommal, akkor exponenciálisnak nevezzük.

1.1. Hatékonysági mértékek

Az algoritmusok elemzése során az igényelt erőforrások mennyiségét ***abszolút*** és ***relatív*** mennyiségekkel jellemezzük.

Ezeknek a mennyiségeknek a célszerű megválasztása attól is függ, hogy az algoritmus ***konkrét gépen*** vagy ***absztrakt gépen*** (számítási modellen) fut.

Jelöljük $W(n,\pi,\mathcal{A})$-vel, illetve $W(n,\pi,p,\mathcal{P})$-vel azt az időt (azoknak a lépéseknek a számát), amely alatt a π problémát az $\mathcal{A}$ soros, illetve a $\mathcal{P}$ párhuzamos algoritmus – (utóbbi p processzort felhasználva) – n méretű feladatokra legrosszabb esetben megoldja.

Hasonlóképpen jelöljük $B(n,\pi,\mathcal{A})$-vel, illetve $B(n,\pi,p,\mathcal{P})$-vel azt az időt (azoknak a lépéseknek a számát), amely alatt a π problémát az $\mathcal{A}$ soros, illetve a $\mathcal{P}$ párhuzamos algoritmus (utóbbi p processzort felhasználva) – n méretű feladatokra legjobb esetben megoldja.

Jelöljük $N(n,\pi)$-vel, illetve $N(n,\pi,p)$-vel azoknak a lépéseknek a számát, amelyekre az n méretű π feladat megoldásához mindenképpen szüksége van bármely soros, illetve bármely párhuzamos algoritmusnak – utóbbinak akkor, ha legfeljebb p processzort vehet igénybe.

Tegyük fel, hogy minden n-re adott a π feladat n méretű konkrét előfordulásainak $D(n,\pi)$ eloszlásfüggvénye. Ekkor legyen $E(n,\pi,\mathcal{A})$, illetve $E(n,\pi,p,\mathcal{P})$ annak az időnek a várható értéke, amennyi alatt a π problémát n méretű feladatokra megoldja az $\mathcal{A}$ soros, illetve a $\mathcal{P}$ párhuzamos algoritmus – utóbbi p processzort felhasználva.

A tankönyvekben az elemzések során gyakori az a feltételezés, hogy az azonos méretű bemenetek előfordulási valószínűsége azonos. Ilyenkor ***átlagos futási időről*** beszélünk, amit $A(n,\mathcal{A})$-val jelölünk.

A W, B, N, E és A jellemzők függnek attól a számítási modelltől is, amelyen az algoritmust megvalósítjuk. Az egyszerűség kedvéért feltesszük, az algoritmus egyúttal a számítási modellt is egyértelműen meghatározza.

Amennyiben a szövegkörnyezet alapján egyértelmű, hogy melyik problémáról van szó, akkor a jelölésekből π-t elhagyjuk.

Ezek között a jellemzők között érvényesek az

$$\begin{aligned} N(n) &\leq B(n,\mathcal{A}) && (1.8)\\ &\leq E(n,\mathcal{A}) && (1.9)\\ &\leq W(n,\mathcal{A}) && (1.10) \end{aligned}$$

egyenlőtlenségek. Hasonlóképpen a párhuzamos algoritmusok jellemző adataira az

$$\begin{aligned} N(n,p) &\leq B(n,\mathcal{P},p) && (1.11)\\ &\leq E(n,\mathcal{P},p) && (1.12)\\ &\leq W(n,\mathcal{P},p) && (1.13) \end{aligned}$$

egyenlőtlenségek teljesülnek. Az átlagos futási időre pedig a

$$\begin{aligned} B(n,\mathcal{A}) &\leq A(n,\mathcal{A}) && (1.14)\\ &\leq W(n,\mathcal{A}), && (1.15) \end{aligned}$$

illetve

$$B(n,\mathcal{P},p) \leq A(n,\mathcal{P},p) \tag{1.16}$$
$$\leq W(n,\mathcal{P},p) \tag{1.17}$$

áll fenn.

Hangsúlyozzuk, hogy ezek a jelölések nem csak futási időre, hanem az algoritmusok más jellemzőire – például memóriaigény, küldött üzenetek száma – is alkalmazhatók.

Ezután relatív jellemzőket, úgynevezett ***hatékonysági mértékeket*** definiálunk.

A gyorsítás (vagy relatív lépésszám) azt mutatja, hányszor kisebb a párhuzamos algoritmus futási ideje a soros algoritmus futási idejénél.

A $\mathcal{P}$ párhuzamos algoritmusnak az $\mathcal{A}$ soros algoritmusra vonatkozó ***gyorsítása***

$$g(n,\mathcal{A},\mathcal{P}) = \frac{W(n,\mathcal{A})}{W(n,p,\mathcal{P})}. \tag{1.18}$$

Ha a gyorsítás egyenesen arányos a felhasznált processzorok számával, akkor lineáris gyorsításról beszélünk. Ha a $\mathcal{P}$ párhuzamos és az $\mathcal{A}$ soros algoritmusra

$$\frac{W(n,\mathcal{A})}{W(n,p,\mathcal{P})} = \Theta(p), \tag{1.19}$$

akkor $\mathcal{P}$ A-ra vonatkozó gyorsítása ***lineáris.***

Ha a $\mathcal{P}$ párhuzamos és az $\mathcal{A}$ soros algoritmusra

$$\frac{W(n,\mathcal{A})}{W(n,p,\mathcal{P})} = o(p), \tag{1.20}$$

akkor $\mathcal{P}$ A-ra vonatkozó gyorsítása ***szublineáris.***

Ha a $\mathcal{P}$ párhuzamos és az $\mathcal{A}$ soros algoritmusra

$$\frac{W(n,\mathcal{A})}{W(n,p,\mathcal{P})} = \omega(p), \tag{1.21}$$

akkor $\mathcal{P}$ A-ra vonatkozó gyorsítása *szuperlineáris.*

A párhuzamos algoritmusok esetében fontos jellemző az $m(n,p,\mathcal{P})$ ***munka,*** amit a futási idő és a processzorszám szorzatával definiálunk. Akkor is ez a szokásos definíció, ha a processzorok egy része csak a futási idő egy részében dolgozik:

$$m(n,p,\mathcal{P}) = pW(n,p,\mathcal{P}). \tag{1.22}$$

Érdemes hangsúlyozni, hogy – különösen az aszinkron algoritmusok esetében – a ténylegesen elvégzett lépések száma lényegesen kevesebb lehet, mint amit a fenti (1.22) képlet szerint kapunk.

Egy $\mathcal{P}$ párhuzamos algoritmusnak az ugyanazon problémát megoldó soros algoritmusra vonatkozó $h(n,p,\mathcal{P},\mathcal{A})$ ***hatékonysága*** a két algoritmus munkájának hányadosa:

$$h(n,p,\mathcal{P},\mathcal{A}) = \frac{W(n,A)}{pW(n,p,\mathcal{P})}. \tag{1.23}$$

Abban a természetes esetben, amikor a párhuzamos algoritmus munkája legalább akkora, mint a soros algoritmusé, a hatékonyság nulla és egy közötti érték – és a viszonylag

nagy érték a kedvező.

Központi fogalom a párhuzamos algoritmusok elemzésével kapcsolatban a munkahatékonyság és munkaoptimalitás. Ha a $\mathcal{P}$ párhuzamos és $\mathcal{A}$ soros algoritmusra

$$pW(n,p,\mathcal{P}) = \widetilde{O}(W(n,\mathcal{A})), \tag{1.24}$$

akkor $\mathcal{P}$ $\mathcal{A}$-ra nézve ***munkahatékony.***

Ez a meghatározás egyenértékű a

$$\frac{pW(n,p,\mathcal{P})}{W(n,\mathcal{A})} = O(\lg^k n) \tag{1.25}$$

egyenlőség előírásával. Eszerint egy párhuzamos algoritmus csak akkor munkahatékony, ha van olyan k érték, hogy a párhuzamos algoritmus munkája nagyságrendileg nem nagyobb, mint a soros algoritmus munkájának $(\lg^k n)$-szerese.

Ha a $\mathcal{P}$ párhuzamos és $\mathcal{A}$ soros algoritmusra

$$pW(n,p,\mathcal{P}) = O(W(n,\mathcal{A})), \tag{1.26}$$

akkor $\mathcal{P}$ $\mathcal{A}$-ra nézve ***munkaoptimális.***

Ez a meghatározás egyenértékű a

$$\frac{pW(n,p,\mathcal{P})}{W(n,\mathcal{A})} = O(1) \tag{1.27}$$

egyenlőség előírásával. Eszerint egy párhuzamos algoritmus csak akkor munkaoptimális, ha van olyan k érték, hogy a párhuzamos algoritmus munkája nagyságrendileg nem nagyobb, mint a soros algoritmus munkája.

Egy párhuzamos algoritmus csak akkor munkahatékony, ha a gyorsítása legalább lineáris. Egy munkahatékony párhuzamos algoritmus hatékonysága $\Theta(1)$.

Ha egy algoritmus egy feladat megoldásához adott erőforrásból csak $O(N(n))$ mennyiséget használ fel, akkor az algoritmust az adott erőforrásra, számítási modellre (és processzorszámra) nézve ***aszimptotikusan optimálisnak*** nevezzük.

Ha egy $\mathcal{A}$ ($\mathcal{P}$) algoritmus egy feladat megoldásához adott erőforrásból a bemenet minden lehetséges $n \geq 1$ mérete esetében csak az okvetlenül szükséges $N(n,\mathcal{A})$ – illetve $(N(n,p,\mathcal{A}))$ – mennyiséget használja fel, azaz

$$W(n,\mathcal{A}) = N(n,\mathcal{A}), \tag{1.28}$$

illetve

$$W(n,p,\mathcal{P}) = N(n,p,\mathcal{P}), \tag{1.29}$$

akkor az algoritmust az adott erőforrásra (és az adott számítási modellre) nézve ***abszolút optimálisnak*** nevezzük és azt mondjuk, hogy a vizsgált feladat ***pontos bonyolultsága*** $N(n)$ $(N(n,p))$.

Két algoritmus összehasonlításakor a

$$W(n,\mathcal{A}) = \Theta(W(n,\mathcal{B})) \tag{1.30}$$

esetben azt mondjuk, hogy a $\mathcal{A}$ és $\mathcal{B}$ algoritmus futási idejének növekedési sebessége aszimptotikusan ***azonos nagyságrendű.***

Amikor két algoritmus futási idejét (például a legrosszabb esetben) összehasonlítjuk, akkor gyakran találunk olyan helyeket, melyeknél kisebb méretű bemenetekre az egyik, míg nagyobb méretű bemenetekre a másik algoritmus futási ideje kedvezőbb. A formális definíció a következő: ha a pozitív egész helyeken értelmezett $f(n)$ és $g(n)$ függvényekre, valamint a $v \geq 2$ pozitív egész számra teljesül, hogy

1. $f(v) = g(v)$;
2. $(f(v-1) - g(v-1))(f(v+1) - g(v+1)) < 0$,

akkor a v számot az $f(n)$ és $g(n)$ függvények ***váltási helyének*** nevezzük.

Például két mátrix szorzatának a definíció alapján történő és a Strassen-algoritmussal történő kiszámítását összehasonlítva (lásd például Cormen, Leiserson, Rivest és Stein többször idézett új könyvét) azt kapjuk, hogy kis mátrixok esetén a hagyományos módszer, míg nagy mátrixok esetén a Strassen-algoritmus az előnyösebb – egy váltási hely van, melynek értéke körülbelül 20.

Az idő mellett algoritmusok számos más erőforrást is használnak – például memóriát, üzeneteket. Utóbbira például $W_{\text{üzenet}}(n, p, \mathcal{P})$ módon hivatkozhatunk.