Eötvös Loránd Tudományegyetem
Természettudományi Kar

Cserháti Enikő
Matematia BSc,
Alkalmazott matematikus szakirány

Ritka mátrixok

Szakdolgozat

Témavezető:
Ágoston István
egyetemi docens

Budapest, 2016.

Tartalomjegyzék

Köszönetnyilvánítás

Ezúton szeretnék köszönetet mondani témavezetőmnek Ágoston Istvánnak, aki mindig türelmesen és segítőkészen állt hozzám, tanácsaival végig segítette a munkámat.

1. Bevezetés

A mátrixok fogalmával legelőször a lineáris egyenletrendszerek tárgyalásánál találkoztam, majd a későbbi tanulmányaim során rendszeresen felbukkantak pl. a lineáris algebrában (mint liníaris tarnszformációk), a geometriában (kúpszeletek leírásánál), az analízisben (gondoljunk pl. a Jacobi-mátrixra), a kombinatorikában (gráfok szomszédsági mátrixaként) stb.

A szakdolgozatom célja az volt, hogy bemutassak egy speciális mátrixosztályt, az ún. ritka mátrixokat, ismertessek egy-két velük kapcsolatos problémát, majd a gyors mátrixszorzás algoritmusain keresztül ismerkedjünk meg az osztály "különlegességeivel".

2. Ritka mátrixok: elemi észrevételek

2.1. Mik is a ritka mátrixok?

Legelőször nézzük mik azok a ritka mártixok.

2.1.1. Definíció: *Ritka mátrixnak nevezzünk egy olyan mátrixot, amely nagyon "sok" nulla elemet tartalmaz.*

Láthatjuk, hogy ez nem igazi definició, viszonyítás kérdése az, hogy mit mondunk soknak. Ha például a mátrix elemeinek egynegyede nem nulla, akkor egy 2×2-es mátrixok esetén csak egy darab nem nulla eleme van, egy 4×4-es mátrixnak 4 darab nem nulla eleme van, de már egy 100×100-as méretű mátrixnál ebben az esetben 2500 nem nulla elemet fog tartalmazni. Ha pl. (M_n) egy olyan mátrixsorozat, melyben az n-edik tag $n \times n$-es, és $f(M_n)$ jelöli a nem 0 elemek számát M_n-be, $g(M_n)$ pedig az összes elemét akkor az előbbi esetben-amikor a nem 0 elemek száma $\frac{1}{4}$, azt kapjuk, hogy

$$\frac{f(M_n)}{g(M_n)} = \frac{\frac{1}{4}n^2}{n^2} = \frac{1}{4}.$$

Ilyenkor tehát a 0 elem száma nő ugyan a sorozatban, de az összes elemhez viszonyított számok aránya állandó.

2.1.2. Feladat: *Mit kapunk, ha a mátrixsorozatok elemei $n \times n$-es diagonális, illetve tridiagonális mátrixok?*

Megoldás: A diagonális mátrixnál csak a főátlóban vannak nem nulla elemek, ez n-et jelent, így

$$\frac{f(M_n)}{g(M_n)} = \frac{n}{n^2} = \frac{1}{n} \to 0, \text{ ha } n \to \infty.$$

A tridiagonálisnál, pedig a főátlóban szintén n elem van a mellete lévő két átlóban pedig $n-1$, tehát

$$\frac{f(M_n)}{g(M_n)} = \frac{2 \cdot (n-1) + n}{n^2} = \frac{3n-2}{n^2} = \frac{3 - \frac{2}{n}}{n} \to 0, \text{ ha } n \to \infty.$$

Ezeknél a tridiagonális mátrixoknál látható, hogy ahogy növeljük a méretüket, egyre kisebb arányban lesznek a nem nulla elemek, egyre nagyobb lesz a "ritkaságuk", hiszen a diagonális mátrixnál pl. egy 5×5-ös mátrix 5 nem nulla elemet fog tartalmazni, míg egy 100×100-as 100 nem nulla elemet fog tartalmazni. És ugyanígy a tridiagonális mátrixnál egy 100×100-as mátrix 298 darab, míg egy 1000×1000-es méretű 2998 darab nem nulla elemt fog tartalmazni. Itt aszimptotikusan azt tudnánk mondani, hogy a két fajta mátrix nem nulla elemeinek száma $\approx O(n)$ és így (az előbbi jelöléssel)

$$\frac{f(M_n)}{g(M_n)} \to 0.$$

Tehát önmagában egy mátrixnál nem érdemes arról beszélni, hogy ritka-e vagy sem, inkább mátrixsorozatoknál van értelme a kérdésnek. Így ritka mátrixok szorozatáról akkor beszélünk, ha a nem nulla elemek aránya tart a 0-hoz. Ritka mátrixok sorozatára természetes példát kapunk, ha olyan gráfok növő sorozatára nézzük a szomszédsági mátrixát melynek foka egy bizonyos korlát alatt marad.

2.2. Ritka mátrixok szorzata és inverze

Naiv módon azt gondolnánk, hogy ha sok nem nulla elemünk van két ritka mátrixban, akkor a szorzatuk is sok nem nulla elemet fog tartalmazni.

2.2.1. Példa:

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 2 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 & 3 \end{bmatrix} \text{ és } B = \begin{bmatrix} 0 & 3 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Ekkor:

$$AB = \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 0 & 8 & 0 \\ 0 & 0 & 3 & 0 \end{bmatrix}$$

Viszont ez nem igaz minden mátrixra, ugyanis megadhatunk olyan ritka mátrixokat, amelyek sorozata már nem ritka: pl. ha az egyik mátrixnak van egy csupa nem nulla oszlopa a másik mátrixnak pedig csupa nem nulla sora:

2.2.2. Példa:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \text{ és } B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Ekkor:

$$AB = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

Vajon mi a helyzet a ritka mátrixok inverzével? Általában egy ritka mátrix inverze lehet sűrű és egy sűrű mátrix inverze is lehet ritka.
Nézzünk itt is két példát:

2.2.3. Példa:

$$\begin{bmatrix} 1 & 1 & \dots & 0 & 0 \\ 0 & 1 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \\ 0 & & \dots & 1 & 1 \\ 0 & 0 & \dots & & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & -1 & 1 & -1 & \dots \\ 0 & 1 & -1 & \dots & \dots \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & 1 & -1 \\ 0 & \dots & \dots & 0 & 1 \end{bmatrix}$$

2.2.4. Példa:

$$\begin{bmatrix} 1 & 1 & \dots & 1 \\ 0 & 1 & \dots & 1 \\ \vdots & & & \vdots \\ 0 & \dots & 1 & 1 \\ 0 & 0 & \dots & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & -1 & 0 & \dots & 0 \\ 0 & 1 & -1 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \\ 0 & \dots & 0 & 1 & -1 \\ 0 & \dots & \dots & 0 & 1 \end{bmatrix}$$

Pl. C.M. da Fonseca:[F07] a tridiagonális mátrix sajátértékeivel foglalkozik, és a következő tételt mondja ki:

2.2.5. Tétel: *Legyen*

$$C = \begin{bmatrix} a-b & a-b & a-b & \dots & a-b \\ a-b & 2a-b & 2a-b & \dots & 2a-b \\ a-b & 2a-b & 3a-b & \dots & 3a-b \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a-b & 2a-b & 3a-b & \dots & na-b \end{bmatrix}, \text{ ahol } a > 0, a \neq b.$$

Ekkor

$$D = \begin{bmatrix} 1+\frac{a}{a-b} & -1 & 0 & \dots & \dots & \dots \\ -1 & 2 & -1 & 0 & \dots & \dots \\ 0 & -1 & 2 & -1 & 0 & \dots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & & 0 & -1 & 2 & -1 \\ 0 & \dots & \dots & 0 & -1 & 2 \end{bmatrix}, \text{ ahol } a > 0, a \neq b \text{ ez}$$

$(1+a) \cdot C$ *inverze.*

Ez a példa is mutatja, hogy sűrű mátrix inverze lehet ritka, ami pl. a sajátértékek kiszámolásánál is lehet hasznos, hiszen ha A sajátértéke λ, akkor A^{-1}-nak sajátértéke $\frac{1}{\lambda}$. S mint a fönti tridiagonális mátrixok esetén, előfordulhat, hogy a ritka mátrix sajátértékeit könnyebben kiszámolhatjuk.

2.3. Ritka mátrixok tárolása

A fönti elemi észrevételeken túl a ritka mátrixok sajátos, gyakran pozitív tulajdonsága előkerül pl. a numerikus eljárásoknál. Hogy egy példát említsünk, reguláris mátrixok QR felbontásának a számolásánál a Givens-forgatások módszere különösen ritka mátrixoknál hatékony. Több ilyen alkalmazás is találhatók pl. a [FH13] könyvben. Mi a továbbiakban egyetlen speciális kérdéssel foglalkozunk még (a mátrixszorzás előtt), nevezetesen a ritka mátrixok tárolásával.

Ha ugyanis a mátrixunk m nem nulla elemet tartalmaz, akkor naiv módon minden elemnél ha az értékét és a kordinátáit $(a_{i,j}, i, j)$; eltároljuk, ennek $3m$

(azaz $O(m)$) a tárolási igénye, szemben az egyébként szükséges $n \cdot k$-val, ahol $n \times k$ a mátrix mérete. Ha tehát $\frac{m}{n \cdot k} \to 0$, ekkor sokat spórolunk.

2.3.1. Példa: *Legyen*

$$A = \begin{bmatrix} 0 & 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 15 \\ 0 & 0 & 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Ekkor A tárolása: (5,1,3),(1,3,2),(15,4,6),(4,5,5).

Egy másik tárolási mód bizonyos esetekben még ennél is kevesebb tárhellyel oldja meg a problémát. Itt is 3 tömbbe tároljuk el az adatokat, de nem mindegyik tömb lesz m elemű.

Az első E tömbbe tároljuk a mátrix nem nulla elemeinek értékeit a bal felső saroktól indulva végig menve a sorokon a jobb alsó sarokig. A második E_C tömbe az E tömb értékeihez tartozó oszlopindexeket tesszük a harmadik tömb i.eleme pedig az i.sor első nem nulla elemének E-beli indexe lesz, végül pedig ennek a tömbnek az utolsó elemébe a (nem nulla elemek száma+1)-et írjuk.

2.3.2. Példa:

$$B = \begin{bmatrix} 5 & 0 & 0 & 3 \\ 0 & 1 & 2 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 4 \end{bmatrix}$$

Ekkor:

$E = [5, 3, 1, 2, 3, 4]$

$E_C = [1, 4, 2, 3, 3, 4]$

$E_R = [1, 3, 5, 6, 7]$

Láthatjuk, hogy az E és E_C tömbökbe m darab, míg az E_R tömbbe $n+1$ darab elemet tárolunk el, ahol a mátrixunk $n \times k$-as méretű. Ez összesen $(2m + n + 1)$.

Az is látszik, hogy akkor éri meg jobban ezt a tárolási módszert használni, ha $n + 1 < m$. A **2.3.2. Példa**-ban szereplő mátrix 6 nem nulla elemet tartalmaz, és 4 sora van, így ennek a tárolása: $2 \cdot 6 + 5 = 17$ tárhelyet igényel. Tehát jobban járunk ezzel a módszerrel, hiszen a naiv módszer $6 \cdot 3 = 18$ tárhelyet igényel.

Ezzel szemben a **2.3.1. Példa**-nál a naiv módszerrel járunk jobban, mivel az A mátrix 4 nem nulla elemet és 6 sort tartalmaz, így $n + 1 > m$ teljesül.

Összegfoglalva azt kapjuk, hogy mindkét tárolási módszerrel jobban járunk n^2-hez képest, és a naiv tárolási módszert, akkor éri meg jobban választanunk, ha m nagyon "pici".

3. Gyors mátrixszorzás

3.1. A természetes mátrixszorzás

Legyen A és B két $n \times n$-es mátrix és szorzatuk C. Ekkor a természetes szorzással C elemei a következőképpen állnak elő:

$$c_{ik} = \sum_{k=1}^{n} a_{ij} b_{jk} \text{ minden } 1 \leq i, j \leq n \text{ esetén.}$$

Például: $c_{11} = a_{11}b_{11} + a_{12}b_{21} + a_{13}b_{31} + ... + a_{1n}b_{n1}$:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \dots & a_{1n} \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix} \cdot \begin{bmatrix} b_{11} & 0 & \dots\dots & 0 \\ b_{21} & 0 & \dots\dots & 0 \\ b_{31} & 0 & \dots\dots & 0 \\ \vdots & \vdots & & \vdots \\ b_{n1} & 0 & \dots\dots & 0 \end{bmatrix}$$

Látható, hogy minden elem kiszámolásakor összeszorzunk egy sort és egy oszlopot, itt látszik hogy a két vektor elemei számának meg kell egyeznie, tehát A oszlopainak és B sorainak ugyanannyinak kell lennie, hogy össze tudjuk szorozni őket. Továbbá pedig ilyenkor n szorzást és $n-1$ összeadást illetve kivonást végzünk.

Tehát, ha A és B két $n \times n$-es méretű mátrix, akkor C is $n \times n$-es méretű lesz, azaz n^2 eleme lesz.Összesen:

$$n^2 \cdot n = n^3 \text{ szorzást kell elvégeznünk és}$$

$$n^2 \cdot (n-1) = n^3 - n^2 \text{ összeadást.}$$

3.2. Strassen algoritmusa

Korábban láttuk, hogy n^3 szorzás és $n^3 - n^2$ összeadás, illetve kivonás kell két $n \times n$ mátrix összeszorzása esetén, és sokáig a számítógép nélküli világban, nem is volt igény rá, hogy ezeket a lépésszámokat csökkentsék.

Némileg váratlan volt 1969-ben Volker Strassen bejelentése, ő egy olyan módszert talált, melynél a mátrixszorzás $n^{\log_2 7} \approx n^{2.81}$ szorzással volt végrehajtható: Lásd[S69]. Ennek szemléltetésére legyenek elöször: A és B 2×2-es mátrixok és jelöljük a szorzatukat C-vel.

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, B = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$$

$$AB = C = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} = \begin{bmatrix} a_{11}b_{11} + a_{12}b_{21} & a_{11}b_{12} + a_{12}b_{22} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{12} + a_{22}b_{22} \end{bmatrix},$$

A fönt is látható mátrixszorzással a 2×2-es mátrixok szorzása 8 számszorzást és 4 összeadást igényel, az alábbi új ötlettel tudta a szükséges szorzások számát csökkenteni Strassen.
Nézzük a következő mennyiségeket:

$$\begin{aligned}
\alpha_1 &= (a_{11} + a_{22}) \cdot (b_{11} + b_{22}) \\
\alpha_2 &= (a_{21} + a_{22}) \cdot b_{11} \\
\alpha_3 &= a_{11} \cdot (b_{12} - b_{22}) \\
\alpha_4 &= a_{22} \cdot (b_{21} - b_{11}) \\
\alpha_5 &= (a_{11} + a_{12}) \cdot b_{22} \\
\alpha_6 &= (a_{21} - a_{11}) \cdot (b_{11} + b_{12}) \\
\alpha_7 &= (a_{12} - a_{22}) \cdot (b_{12} + b_{22})
\end{aligned}$$

Ezekből már szorzás nélkül is ki tudjuk fejezni C elemeit, ugyanis:

$$\begin{aligned}
\alpha_1 + \alpha_4 - \alpha_5 + \alpha_7 &= a_{11}b_{11} + a_{11}b_{22} + a_{22}b_{11} + a_{22}b_{22} + a_{22}b_{21} - a_{22}b_{11} \\
&\quad - a_{11}b_{22} - a_{12}b_{22} + a_{12}b_{12} + \\
&\quad + a_{12}b_{22} - a_{22}b_{12} - a_{22}b_{22} \\
&= a_{11}b_{11} + a_{12}b_{21} \\
&= c_{11}.
\end{aligned}$$

$$\begin{aligned}\alpha_3 + \alpha_5 &= a_{11}b_{12} - a_{11}b_{22} + a_{11}b_{22} + a_{11}b_{22} + a_{12}b_{22} \\ &= a_{11}b_{12} + a_{12}b_{22} \\ &= c_{12}.\end{aligned}$$

$$\begin{aligned}\alpha_2 + \alpha_4 &= a_{21}b_{11} + a_{22}b_{11} + a_{22}b_{21} - a_{22}b_{11} \\ &= a_{21}b_{11} + a_{22}b_{21} \\ &= c_{21}.\end{aligned}$$

$$\begin{aligned}\alpha_1 + \alpha_3 - \alpha_2 + \alpha_6 &= a_{11}b_{11} + a_{11}b_{22} + a_{22}b_{11} + a_{22}b_{22} + \\ &= +a_{11}b_{12} - a_{11}b_{22} - a_{21}b_{11} - a_{22}b_{11} + \\ &+ a_{21}b_{11} + a_{21}b_{12} - a_{11}b_{11} - a_{11}b_{12} \\ &= a_{21}b_{12} + a_{22}b_{22} \\ &= c_{22}.\end{aligned}$$

Mindegyik α_i kiszámolásánál 1 számszorzást végzünk, utána pedig már csak összeadjuk, illetve kivonjuk egymásból ezeket. A számszorzások száma tehát 7 lesz, az összeadások és kivonások száma pedig összesen 18.
Azért azt meg kell jegyeznünk, hogy természetes mátrixszorzással 2×2-es mátrixok esetén 4 összeadást, illetve kivonást végzünk, szemben a mostani 18-cal.

Mi van akkor, ha a mátrixaink 4×4-es méretűek?
A következőt tudjuk elmondani: ha $A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, B = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}$, ahol A_{ij}-k és B_{ij}-k 2×2-es blokkok, akkor

$$AB = \begin{bmatrix} A_{11}B_{11} + A_{12}B_{21} & A_{11}B_{12} + A_{12}B_{22} \\ A_{21}B_{11} + A_{22}B_{21} & A_{21}B_{12} + A_{22}B_{22} \end{bmatrix},$$

Tehát ugyanúgy szorozzuk össze őket, mint a 2×2-es mátrixokat, csak a mátrixok elemeit 2×2-es blokkokkal hellyetesítjük. Vegyük észre, hogy a

2×2-es mátrixok eseténél nem használjuk ki a szorzás kommutativitását, így az előbbi képleteket most 2×2-es blokkok szorzására is használhatjuk. Itt minden blokkszorzás kiszámolásánál 7 számszorzást végzünk, mivel 2×2-es mátrixokat szorzunk össze. Összesen 7 blokkszorzásunk van, tehát $7 \cdot 7 = 7^2$=49 számszorzást fogunk elvégezni.

Ugyanígy tudjuk ezt rekurzívan folytatni minden 2-hatványra: legyen $n = 2^k$ és minden k-ra $2^k \times 2^k$ méretű mátrixok szorzásakor az α_i-knél a $2^{k-1} \times 2^{k-1}$-es blokkokat szorozzuk, 7-szer a $2^{k-1} \times 2^{k-1}$ méretű mátrixok szorzásakor lévő számszorzások számát kapjuk. Tehát legyen

$M(k)$:= a $2^k \times 2^k$ méretű mátrixok szorzásánál lévő számszorzások száma

Ekkor $M(0) = 1$, és $M(k) = 7 \cdot M(k-1)$. Ekkor a rekurziót megoldva azt kapjuk, hogy

$$M(k) = 7^k = 7^{\log_2 n} = (2^{\log_2 7})^{\log_2 n} = 2^{\log_2 n \cdot \log_2 7} = (2^{\log_2 n})^{\log_2 7} =$$

$$= n^{\log_2 7} = n^{2.807\ldots} \approx n^{2.81}.$$

Láthattuk, hogy a 2×2-es blokkoknál 8 helyett 7 számszorzást kaptunk, ami csak egy picivel jobb eredmény, de elég nagy n-re már sokkal jobb becslést kaptunk ezzel a módszerrel, hiszen $n^{2.81} < n^3$, és nagy n-ekre ez már sokat számít.

Mi a helyzet az összeadásokkal és kivonásokkal? Legyen

A(k):= a $2^k \times 2^k$ méretű mátrixok szorzásánál lévő számösszeadások és kivonások száma.

Tudjuk, hogy a $2^{k-1} \times 2^{k-1}$-es blokkok között 18 összeadást illetve kivonást végzünk, így ezeknél a számösszeadások száma: $18 \cdot (2^{k-1})^2 = 18 \cdot (2^{2k-2}) = 9 \cdot 2^{2k-1}$ lesz.

De számösszeadásra szükség van a blokkok szorzásánál is: az α_i-k kiszámításánál, a 7 blokkszorzásnál a számösszeadások illetve számkivonások

száma $7A(k-1)$. Ezeket összerakva: $A(k) = 9 \cdot 2^{2k-1} + 7A(k-1)$.
Vezessük be a $B(k) = 7^{-k}A(k)$ jelölést. Ekkor:

$$B(k) = 7^{-k}\left(\frac{9}{2}\right)2^{2k} + 7^{-k+1} \cdot A(k-1) = \left(\frac{9}{2}\right)\left(\frac{2^{2k}}{7^k}\right) + 7^{-(k-1)}A(k-1) =$$

$$= \frac{9}{2}\left(\frac{4}{7}\right)^k + B(k-1).$$

Azt is tujduk, hogy $B(0) = 0$. Ezeket k szerint összeadva

$$\sum_{i=0}^{k} B(i) = \sum_{i=0}^{k} \frac{9}{2}\left(\frac{4}{7}\right)^i + \sum_{i=1}^{k} B(i-1).$$

Ha kivonjuk mindkét oldalból $\sum_{i=1}^{k} B(i-1)$-t akkor a következőt kapjuk:

$$B(k) = \sum_{i=0}^{k} \frac{9}{2}\left(\frac{4}{7}\right)^i = \frac{9}{2}\sum_{i=0}^{k}\left(\frac{4}{7}\right)^i \leq \left(\frac{9}{2}\right)\left(\frac{4}{3}\right) = 6.$$

A $\sum_{i=0}^{k}\left(\frac{4}{7}\right)^i \leq \frac{4}{3}$ egyenlőtlenséget a mértani sorozat öszegképletéből kapjuk, ugyanis

$$\sum_{i=0}^{k}\left(\frac{4}{7}\right)^i = \frac{1\left(\left(\frac{4}{7}\right)^{k+1} - 1\right)}{\frac{4}{7} - 1} = \frac{\left(\frac{4}{7}\right)^{k+1} - 1}{\frac{-3}{7}} = \frac{1 - \left(\frac{4}{7}\right)^{k+1}}{\frac{3}{7}} = \frac{7 - \frac{4^{k+1}}{7^k}}{3} \leq \frac{4 - \frac{4^{k+1}}{7^k}}{3} \leq \frac{4}{3}$$

Ezekután már tudunk felső becslést adni $A(k)$-ra:

$$A(k) = B(k)7^k \leq 6 \cdot 7^k = 6 \cdot n^{log_2 7}.$$

Az alábbi szorzást még áltáthatóbbá tehetjük, mert $A(k)$-ra egy könnyen belátható explicit képletet is tudunk adni.

$$A(k) = 6 \cdot 7^k - 6 \cdot 4^k.$$

A fönti képletet k szerinti indukcióval fogunk bizonyítani
Nézzük $k = 0$-ra:
$A(k)$=két 1×1 méretű mátrix összeszorzásánál lévő összeadások és kivonások

száma=0, másrészt

$6 \cdot 7^k - 6 \cdot 4^k = 6 \cdot 1 - 6 \cdot 1 = 0$

Tehát k=0-ra teljesül az egyenlőség.

Nézzük meg még $k = 1$-re:

$A(k)$=két 2×2 méretű mátrix összeszorzásánál lévő összeadások és kivonások száma=18.

$6 \cdot 7^k - 6 \cdot 4^k = 18.$

Teljesül az egyenlőség.

Tegyük föl, hogy $k-1$-re teljesül az állítás, azaz $A(k-1) = 6 \cdot 7^{k-1} - 6 \cdot 4^{k-1}$ és lássuk be, hogy k-ra is teljesül:

$$A(k) = 9 \cdot 2^{2k-1} + 7A(k-1) = 9 \cdot 2^{2k-1} + 7 \cdot 6 \cdot 7^{k-1} - 7 \cdot 6 \cdot 4^{k-1} = \frac{9}{2}4^k + 6 \cdot 7^k - 42 \cdot 4^{k-1} =$$

$$= 6 \cdot 7^k + \frac{9}{2}4^k - \frac{42}{4}4^k = 6 \cdot 7^k + 4^k \left(\frac{9}{2} - \frac{42}{4}\right) = 6 \cdot 7^k + 4^k \left(\frac{18}{4} - \frac{42}{4}\right) =$$

$$= 6 \cdot 7^k + 4^k \left(\frac{-24}{4}\right) = 6 \cdot 7^k + 4^k (-6) = 6 \cdot 7^k - 6 \cdot 4^k.$$

És ezzel be is láttuk a képletet.

Mivel $4^k = 2^{2^k} = 2^{2^{\log_2 n}} = (2^{\log_2 n})^2 = n^2$.

Innen már látszik, hogy $A(k) \approx 6n^{2.81} - 6n^2 < n^3 - n^2$, ha n elég nagy. Elég nagy n-re ez is kisebb, mint az eredeti műveletigény. Ilyenkor ugyanis még az összeadások száma is kevesebb a Strassen-algoritmusnál, mint a szokásos szorzásnál. Érdemes megjegyezzünk, hogy a szorzások általában bonyolultabb műveletnek tekinthetők az összeadásnál, így a Strassen-algoritmus "jó"-ságának mércéje is elsősorban az $M(k)$-ra adott becslés.

Egy dologgal még adósok vagyunk: mi van, ha a mátrix mérete nem 2-hatvány? Egy egyszerű módszerrel visszavezethetjük az általános esetet az előbb tárgyaltakra: A, B helyett

$$\tilde{A} = \begin{bmatrix} A & 0 \\ 0 & I \end{bmatrix} \text{ és } \tilde{B} = \begin{bmatrix} B & 0 \\ 0 & I \end{bmatrix}$$

mátrixokkal számolunk, ahol $\tilde{A}$ és $\tilde{B}$ már $2^k \times 2^k$-os mátrixok. Nyilván:

$$\tilde{A}\tilde{B} = \begin{bmatrix} AB & 0 \\ 0 & I \end{bmatrix}.$$

Ilyenkor a kibővített méretre vonatkozó képletek fogják megadni a műveletigényeket, de nagy n-ekre még így is kedvezőbb értéket kapunk, mint a természetes mátrixszorzásnál. A mátrix méretét ugyanis kevesebb, mint a kétszeresére fogjuk megnövelni, s a művelet igénye a Strassen algoritmust használva legfeljebb $(2n)^{2.81} = 2^{2.81} \cdot n^{2.81} \approx 7 \cdot n^{2.81}$, a szokásos mátrixszorzással pedig n^3 szorzásunk van. Nagy n-ekre viszont $7 \cdot n^{2.81} < n^3$.

Végezetül itt jegyezzük meg, hogy 1990 Winograd talált egy olyan módszert a 2×2-es mátrixok szorzására, ahol a számszorzások száma szintén 7 lesz, viszont az összeadások illetve kivonások száma 16 helyett csak 15: Nevetesen vegyük a következő kifejezéseket:

$$\begin{aligned}
\beta_1 &= a_{21} + a_{22} & \gamma_1 &= \beta_2\beta_6 & \beta_9 &= \gamma_2 + \gamma_3 \\
\beta_2 &= \beta_1 - a_{11} & \gamma_2 &= a_{11}b_{11} & \beta_{10} &= \gamma_1 + \gamma_2 \\
\beta_3 &= a_{11} - a_{21} & \gamma_3 &= a_{12}b_{21} & \beta_{11} &= \beta_{10} + \gamma_4 \\
\beta_4 &= a_{12} - \beta_2 & \gamma_4 &= \beta_3\beta_7 & \beta_{12} &= \beta_{10} + \gamma_5 \\
\beta_5 &= b_{12} - b_{11} & \gamma_5 &= \beta_1\beta_5 & \beta_{13} &= \beta_{12} + \gamma_6 \\
\beta_6 &= b_{22} - \beta_5 & \gamma_6 &= \beta_4 b_{22} & \beta_{14} &= \beta_{11} - \gamma_7 \\
\beta_7 &= b_{22} - b_{12} & \gamma_7 &= a_{22}\beta_8 & \beta_{15} &= \beta_{11} + \gamma_{15} \\
\beta_8 &= \beta_6 - b_{21}
\end{aligned}$$

Ekkor:

$$AB = C = \begin{bmatrix} \beta_9 & \beta_{13} \\ \beta_{14} & \beta_{15} \end{bmatrix}$$

Mindegyik β_i együtthatóban egy számösszeadást, illetve kivonást végzünk, és mindegyik γ_i kifejezésnél egy számszorzást hajtunk végre. A β_i-k száma 15, a γ_i-ik száma pedig 7, innen is látszik, hogy összesen 15 számösszeadásra, illetve kivonásra és 7 számszorzásra van szükségünk.

Ezzel az eljárással a szükséges összeadások száma kicsit csökken, de aszimptotikusan nem változik a becslés a Strassen-algoritmushoz képest.

3.3. Egyéb algoritmusok

Strassen módszerét újabb eredmények követték. A 70-es években Pan, valamint Bini et al., majd a 80-as években Schönhage, Romani, Coppersmith és Winograd, és Strassen hoztak jobb eredményt. 1990-ben Coppersmith és Winograd immáron egy $O(n^{2.376})$ lépésigényű algoritmust talált. Jelenleg William[VW11] eredménye a legjobb: $O(n^{2.373})$.

Évszám	Szerző	ω lépésszám
Kezdetekben	Hagyományos	$O(n^3)$
1969	Volker Strassen	$\approx O(n^{2.81})$
1978	Pan	$\approx O(n^{2.796})$
1978	Bini et al.	$\approx O(n^{2.78})$
1981	Scönhage	$\approx O(n^{2.548})$
1981	Pan, Schönhage	$\approx O(n^{2.522})$
1982	Romani	$\approx O(n^{2.517})$
1985	Coppersmith, Winograd	$\approx O(n^{2.496})$
1986	Strassen	$\approx O(n^{2.479})$
1990	Coppersmith, Winograd	$\approx O(n^{2.376})$
2014	Williams	$\approx O(n^{2.373})$

3.4. Egy naiv becslés ritka mátrixok szorzására

Nézzük meg elöször, mi a helyzet akkor, ha ritka mátrixokat szorzunk.

A szokásos eljárásnál és a Strassen-algoritmusnál nem tudjuk kihaszálni, ha 0 elem van valamelyik tényezőbe, legalábbis a lépésszám becslésénél a mátrix "ritkasága" nem hoz javulást. Ezzel szemben a következő, a szokásostól eltérő, de viszonylag egyszerű naiv mátrixszorzási eljárás lehetővé teszi, hogy a lépésszám becslésénél figyelembe vegyük, ha kevés a nem nulla elemünk.

Legyen A és B két $n \times n$-es mátrix és szorzatuk C. Jelöljük A_{*k}-val az A mátrix k-adik oszlopát, és az ebben szereplő nem nulla elemek számát jelölje a_k, B_{k*} B mátrix $k-$adik sorát, és a benne lévő nem nulla elemek száma legyen b_k. Ekkor az AB mátrixszozatot az alábbi, diadikus szorzatok

összegeként is felírhatjuk:

$$AB = \sum_{k=1}^{n} A_{*k} B_{k*}.$$

Mivel A_{*k} egy oszlopvektor, tehát $n \times 1$-es méretű, B_{k*} pedig egy sorvektor azaz $1 \times n$-es méretű, így a szorzatuk $n \times n$-es méretű lesz minden k-ra. Összeadva ezeket minden $k \in \{1, 2, ..., n\}$-ra, pont AB-t fogjuk kapni:
$\sum_k A_{*k} B_{k*} = A_{*1} B_{1*} + A_{*2} B_{2*} + ... + A_{*n} B_{n*} = AB.$

$$\begin{bmatrix} a_{11} & 0 & \dots & 0 \\ a_{21} & 0 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ a_{n1} & 0 & \dots & 0 \end{bmatrix} \cdot \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1n} \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix} + \begin{bmatrix} 0 & a_{12} & \dots & 0 \\ 0 & a_{22} & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & a_{n2} & \dots & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 & 0 & \dots & 0 \\ b_{21} & b_{22} & \dots & b_{2n} \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix} +$$

$$+ \dots + \begin{bmatrix} 0 & 0 & \dots & a_{1n} \\ 0 & 0 & \dots & a_{2n} \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & 0_{nn} \end{bmatrix} \cdot \begin{bmatrix} 0 & 0 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & 0 \\ b_{n1} & b_{n2} & \dots & b_{nn} \end{bmatrix}$$

Amikor összeszorzunk egy oszlopot és egy sort, az oszlopban lévő összes nem nulla elemet szorozzuk össze a sorban lévő összes nem nulla elemmel. Így ha a_k jelöli A mátrix k-adik oszlopában lévő nem 0 elemek számát, b_k pedig a B-nek k-adik sorában lévő nem 0 elemek számát, akkor az $A_{*k} B_{k*}$ szorzat kiszámitásakor pontosan $a_k b_k$ darab számszorzást kell elvégeznünk. Így az AB kiszámításának műveletigénye:

$$\sum_{k=1}^{n} a_k b_k \text{ lesz.}$$

Hogy lehetne felülről becsülni ezt a szorzatot? Ha most A ritka mátrix és m darab nem nulla elemet tartalmaz, akkor ezt a fönti szorzatot felülről tudjuk becsülni mn-nel, mivel:

$$\begin{aligned} \sum_{k=1}^{n} a_k b_k &= a_1 b_1 + a_2 b_2 + a_3 b_3 + ... + a_n b_n \\ &\leq a_1 n + a_2 n + a_3 n + ... + a_n n = \\ &= (\sum_{k=1}^{n} a_k) n = mn. \end{aligned}$$

Természetesen ezt akkor is meg tudjuk tenni, ha B tartalmaz m darab nem nulla elemet. Ez azt mutatja, hogy ha m lényegesen kisebb, mint n^2, pl. $m \approx n^{\frac{3}{2}}$, akkor a szorzások száma lényegesen kisebb, mint n^3. (Az alábbi példa esetén $n^{2.5}$.) Nézzünk, mi van akkor, ha mindkét mátrix ritka, és legfeljebb m darab nem nulla elemet tartalmaz: kaphatunk-e még jobb becslést.: Sajnos nem, amint azt a következő példa is mutatja:
Legyen $m \geq n$ és nézzük azt az esetet, amikor A és B mátrix első m/n oszlopában illetve sorában vannak a nem nulla elemek, a többi oszlopba illetve sorba pedig csupa 0, vagyis

$$\begin{cases} a_i = b_i = n & i \leq \frac{m}{n} \\ a_i = b_i = 0 & \text{különben} \end{cases}$$

Ekkor

$$\sum_{k=1}^{n} a_k b_k = \frac{m}{n} n^2 = mn.$$

Ez tehát azt mutatja, hogy nem kaptunk jobb felső korlátot, azaz nem tudjuk tovább javítani a lépésszámot. Viszont egy kicsivel jobb becslést kapunk, ha a nem 0 elemek egyenletesen oszlanak el A-ban és B-ben. Legyen pl. $m \approx n^{\frac{3}{2}}$; akkor egyenletes eloszlás mellett az alábbit kapjuk:.

$$a_k = b_k = m/n \text{ minden } 1 \leq k \leq n\text{-re.},$$

és így

$$\sum_{k=1}^{n} a_k b_k = n \left(\frac{m}{n}\right)^2 = \frac{m^2}{n} = n^2 < m \cdot n.$$

Ez általában kisebb, mint mn, mivel $m \leq n^2$, ezért $\frac{m}{n} \leq n$, tehát

$$\frac{m^2}{n} = m \left(\frac{m}{n}\right) \leq mn.$$

3.5. Sűrű mátrixok szorzása Coppersmith és Winograd módszerével

Mielőtt a Yuster és Zwick által ritka mátrixok szorzására adott eljárást ismertetünk, térjük vissza egy kis időre a "közönséges" mátrixok szorzására. Míg

Strassen algoritmusa elsősorban négyzetes mátrixokra müködött, a későbbi eljárásoknál már általában tetszőleges téglalap alakú mátrixok szorzására új módszert adtak.

Jelölje $M(a,b,c)$ az $a \times b$ és $b\times$-es mátrixok összeszorzásánál szükséges számszorzások számát. Kicsit áttekinthetőbbé válik az irásmód, ha áttérünk a logaritmikus jelölésre: $a = n^r$, $b = n^s$, $c = n^t$ esetén jelölje $k(r,s,t)$ a logaritmikus műveletigényt, azaz legyen $k(r,s,t)$ a legkisebb olyan ω szám, melyre $M(n^r, n^s, n^t) = O(n^{\omega+o(1)})$.

Jelölje ω a $k(1,1,1)$ kitevőt. Ez épp azt jelenti, hogy $n \times n$-es mátrixok szorzásánál legalább n^ω nagyságrendű számszorzásra van szükségünk. Általános mátrixokra tudható, hogy legalább n^2 szorzás biztosan szükséges, így $\omega \geq 2$, másrészt a Strassen algoritmus azt mutatja, hogy $\omega \leq 2.81$. Tehát ω értékét nem ismerjük pontosan, de becslést adhatunk rá.
1990-be Coppersmith és Winograd egy jobb felső becslést adott ω-ra:

3.5.1. Tétel: *(Coppersmith és Winograd[CW90]):*
ω<2.376.

Tehát, bármely n-re $M(n,n,n)$, azaz két $n \times n$-es mátrix szorzásának minimális költsége legfeljebb $O(n^{2.376+o(1)})$ lehet.
Szükségünk lesz még két konstansra a lépésszám becsléséhez: Legyen

$$\alpha = sup\left\{0 \leq r \leq 1 | k(1,r,1) = 2\right\},$$

Tehát α az a maximális kitevő, amire $M(n, n^\alpha, n) = O(n^{2+o(1)})$,
ahol $\alpha \in (0,1)$. Legyen továbbá

$$\beta = \frac{\omega - 2}{1 - \alpha}.$$

7 évvel később Coppersmith α-ra is adott egy becslést:

3.5.2. Tétel: *(Coppersmith[C97]):*
α>0.294.

Az 1. és 2. tételből levezethető az alábbi becslés:

3.5.3. Tétel: $k(1,r,1) \leq \begin{cases} 2 & 0 \leq r \leq \alpha \\ 2+\beta(r-\alpha) & \textit{különben} \end{cases}$

3.5.4. Következmény: $M(n,l,n) = n^{2-\alpha\beta+o(1)}l^{\beta} + n^{2+o(1)}$.

A bizonyításoktól itt most eltekintünk, ezek megtalálhatók pl. Huang és Pan cikkében. Láthatjuk, hogy α és β ismeretében, bármely n-re és l-re $M(n,l,n)$-t ki tudjuk számítani. Illetve, ha tudjuk ω-t és α-t, akkor meg tudjuk mondani a β értékét is.
ω=2.376-tal és α=0.294-gyel, azt kapjuk hogy $\beta \simeq$0.533. Ha ω=2 lenne igaz, akkor az azt jeletené, hogy két $n \times n$-es mátrix szorzását n^2 lépéssel is megtudnánk csinálni, azaz α=1. Ebben az esetben β-t nem tudjuk definiálni, de nincs is rá szükségünk.

3.6. Az új, ritka mátrixokat szorzó algoritmus

Ebben a fejezetben ismeretettem azt az eljárást, melyet Yuster és Zwick adott a [YZ05] cikkben ritka mátrixok szorzására.

Legyen A és B a két $n \times n$-es mátrixunk, egy tetszőleges R gyűrű fölött, és legyen a szorzatuk C, amit ki szeretnénk számítani. Jelölje A_{*k} A k-adik oszlopát, és B_{k*} pedig B k-adik sorát.
Korábban már láttuk, hogy AB=$\sum_k A_{*k}B_{k*}$.

Legyen a_k az A mátrix k-adik oszlopában lévő nem nulla elemeinek a száma, és b_k pedig B mátrix k-adik sorában lévő nem nulla elemienk a száma. Bármely $I \subseteq \{1,2,\ldots,n\}$-re legyen A_{*I}, A-nak az a részmátrixa, ami azokból a oszlopokból áll, amik benne vannak I-ben, és B_{I*}, B-nek az a részmátrixa ami azokból a sorokból áll, amik benne vannak I-ben. Ekkor az is látszik, hogy ha $J = \{1,2,\ldots,n\} - I$, akkor $A_{*I}B_{I*}+A_{*J}B_{J*} = AB$. Az új ritka mátrix szorzó algoritmusunk fő ötlete abban rejlik, hogy két féle mátrixokat szorzó algoritmust használ, ketté szedi a mátrixokat ennek alapján, hogy hol van kevesebb nem nulla elem van: itt tudunk spórolni a naiv mátrix szorzó algoritmussal, a sűrű részére pedig valamelyik gyors, sűrű téglalap ala-

kú mátrixok szorzására alkalmazható algoritmust használjuk. Ahhoz, hogy megállapítsuk, hogyan is szedjük szét a mátrixokat, először definiálunk egy π permutációt, melyre $a_{\pi(1)}b_{\pi(1)} \geq a_{\pi(2)}b_{\pi(2)} \geq a_{\pi(3)}b_{\pi(3)} \geq ..... \geq a_{\pi(n)}b_{\pi(n)}$. Ez a rendezés azt mutatja, hogy hol a legsűrűbb a két mátrix, hol kell a legtöbb(vagy a legkevesebb) szorzást végeznünk a naiv algoritmussal. Így a következőféleképpen fog kinézni az algoritmusunk:

$$Algoritmus : SMP(A, B)$$

$$Input :\ \text{A és B n} \times \text{n mátrixok}$$

$$Output : C = AB.$$

1. Legyen a_k az A k-adik oszlopában, azaz A_{*k}-ban lévő nem nulla elemek száma, minden $1 \leq k \leq n$-re.

2. Legyen b_k a B k-adik sorában, azaz B_{k*}-ban lévő nem nulla elemek száma, minden $1 \leq k \leq n$-re.

3. Legyen π permutáció olyan, melyre $a_{\pi(1)}b_{\pi(1)} \geq a_{\pi(2)}b_{\pi(2)} \geq ..... \geq a_{\pi(n)}b_{\pi(n)}$.

4. Megnézzük, mely $0 \leq l \leq$ n-re minimális $M(n, l, n) + \sum_{k>l}\ a_{\pi(k)}b_{\pi(k)}$.

5. Legyen $I = \{\pi(1), \pi(2), ..., \pi(l)\}$ és $J = \{\pi(l+1), ..., \pi(n)\}$.

6. Számítsuk ki $C_1 = A_{*I}B_{I*}$-t pl. a Coppersmith-Winograd féle mátrixokat szorzó általános algoritmussal.

7. Számítsuk ki $C_2 = A_{*J}B_{J*}$-t a ritka mátrixokat szorzó „naiv" algoritmussal.

8. Output $C_1 + C_2$.

3.6.1. Tétel: *(Yuster és Zwick[YZ05])*
Az SMP(A,B) algoritmus két $n \times n$-es mátrix szorzatát számítja ki az R gyűrű felett. Ha A-ban m_1, és B-ben pedig m_2 nem nulla elem van, akkor az

algoritmus legfeljebb

$$O\left(min\left\{(m_1m_2)^{\frac{\beta}{\beta+1}}n^{\frac{2-\alpha\beta}{\beta+1}+o(1)}+n^{2+o(1)},m_1n,m_2n,n^{\omega+o(1)}\right\}\right)$$

gyűrűbeli műveletet használ.

Bizonyítás:

Először megmutatjuk, ahhoz hogy az algoritmus kiválassza azt az l-et, amivel minimalizálja az $M(n,l,n)+\sum_{k>l}a_{\pi(k)}b_{\pi(k)}$ kifejezést, illetve így a további lépésekben a műveletek számát is,

$$O\left(min\left\{m_1n,m_2n,n^{\omega+o(1)}\right\}\right)$$

műveletre van szükségünk, azaz l kiválasztása önmagában nem túl "költséges".

Ezt úgy fogjuk belátni, hogy mégnézzük az algoritmusnak azokat a lépéseit, amiktől függ az $M(n,l,n)+\sum_{k>l}a_{\pi(k)}b_{\pi(k)}$ érték minimalizálása és belátjuk, hogy ezek műveletigénye minden esetben kisebb vagy egyenlő m_1n,m_2n és $n^{\omega+o(1)}$. Nézzük a lépéseket:
Tegyük föl,hogy $m_1\leq m_2$, és egyik sem 0.

1. lépésben:

Végig megyünk A elemein és meghatározzuk az a_i értékeket, azaz megszámoljuk, hogy oszloponként hány nem nulla elemet tartalmaz. A tételből tudjuk, hogy összesen m_1 darab nem nulla elem van, és mivel azt nem számoljuk lépésnek, amikor 0 elemet találunk, így ez m_1 lépést jelent. Az is könnyen látszik, hogy ezt felülről tudjuk becsülni mindhárom taggal:
mivel $1\leq n$, ezért $m_1\leq m_1n$
mivel feltettük, hogy:$m_1\leq m_2$, ezért $m_1\leq m_2\leq m_2n$
mivel $m_1\leq n^2$, ugyanis A legfeljebb n^2 elemet tartalmaz és korábban már láttuk, hogy $2\leq\omega$, ezért $m_1\leq n^{\omega+o(1)}$

2. lépésben:

Ugyanígy tudjuk, hogy B m_2 darab nem nulla elemet tartalmaz, így ha végig

megyünk B elemein és így tároljuk el őket, akkor ez m_2 lépést fog jelent. Ugyanakkor ezt megtehetjük úgy is hogy csak az A nem nulla oszlopaihoz tartozó sorokat nézzük meg B-ben, ezekben mind n elem van, így ez $m_1 n$ lépés lesz.

Tehát ennek a lépésnek a műveletigénye legfeljebb $k \leq min\{m_2, m_1 n\}$.

Így a k lépésszám:

$k \leq m_1 n$; $k \leq m_2 \leq m_2 n$

az $m_2 \leq n^2$ egyenlőtlenségből itt $k \leq n^{\omega+o(1)}$ (hiszen $\omega \leq 2$).

3. lépésben: Sorba rendezzük a szorzatokat. Mivel legfeljebb n szorzatunk lehet, ennek a sorba rendezése legfeljebb $n \cdot \log n$ lépés lesz.

Ha $n \leq m_1 \leq m_2$

Tudjuk, hogy: $\log n \leq n$

akkor $n \log n \leq m_1 n \leq m_2 n$

és $n \log n \leq n^2 \leq n^{\omega+o(1)}$

Ha $m_1 \leq m_2 \leq n$ vagy $m_1 \leq n \leq m_2$

$\Rightarrow$ A-nak a skatulya elv miatt nem minden oszlopába lesz nem nulla elem, azaz lesz olyan $k \in \{1, 2, ..., n\}$, amire a_k=0. Mivel A m_1 nem nulla elemet tartalmaz, ezért legfeljebb m_1 darab $a_k b_k$ nem nulla szorzat lesz és ezeket kell sorba rendeznünk. Ennek a lépésszáma $m_1 * \log m_1$ lesz.

$\Rightarrow \log m_1 \leq \log n \leq n$

Ekkor $m_1 \log m_1 \leq m_1 n$ és

$m_1 \log m_1 \leq m_1 n \leq n^2 \leq n^{\omega+o(1)}$

és mivel $m_1 \leq n$, ezért $m_1 n \leq m_2 n$.

4. lépésben pedig:

Kiválasztjuk azt az l-et, amire minimális lesz $M(n, l, n) + \sum_{k>l} a_{\pi(k)} b_{\pi(k)}$.

Az ötlet az az, hogy a $\sum_{k>l} a_{\pi(k)} b_{\pi(k)}$ értéket, visszafele számítjuk ki minden $k \in \{1, 2, ..., n\}$-ra, tehát $l = n$-től kezdjük, és mindig csak az eggyel kisebb $a_k b_k$ szorzatot adjuk hozzá.

Az $M(n, l, n)$ l-ben csökkenni fog, ha l csökken, mert l^β monoton növő.

Ha $m_1 \leq m_2 \leq n$ vagy $m_1 \leq n \leq m_2$

$\Rightarrow$ Biztos, hogy lesz A-nak olyan oszlopa ami csupa 0 lesz. Tehát az $a_{\pi(k)}b_{\pi(k)}$ szorzatok közül m_1 darab lesz nem nulla. Mivel a π permutáció csökkenő sorrendbe rendezi a szorzatokat, ezért $\sum_{k>m_1} a_{\pi(k)}b_{\pi(k)}$=0, ezeket nem is kell kiszámolnunk.

És azt is tudjuk, hogy $M(n,l,n)$ érték csökken, ha l csökken, így $l > m_1$-re $M(n,l,n)+\sum_{k>l} a_{\pi(k)}b_{\pi(k)} \geq M(n,m_1,n)+\sum_{k>m_1} a_{\pi(k)}b_{\pi(k)}$. Tehát $M(n,l,n)$-t sem érdemes kiszámolni az m_1-nél nagyobb l-ekre. Így összesen: $O(m_1)$ lépésből választjuk ki l-et.

Mivel $m_1 \leq m_2, n$ ezért $O(m_1) \leq O\left(min\left\{m_1 n, m_2 n, n^{\omega+o(1)}\right\}\right)$.

Ha $n \leq m_1 \leq m_2$

$\Rightarrow O(n)$ lépésünk lesz.

Ekkor $O(n) \leq O\left(min\left\{m_1 n, m_2 n, n^{\omega+o(1)}\right\}\right)$.

Miután pedig kiszámítottuk az $M(n,l,n) + \sum_{k>l} a_{\pi(k)}b_{\pi(k)}$ értékeket, megkeressük a minimumot, és mivel a szóbajövő l értékek száma $\leq m_1 \leq m_2$, ezért ez szintén $\leq O\left(min\left\{m_1 n, m_2 n, n^{\omega+o(1)}\right\}\right)$ lépést jelent.

(Természetesen, ha $m_2 \leq m_1$, ugyanígy szimmetrikusan be tudjuk ezeket látni.)

Ha ezeket összeadom, az még mindig csak konstansszorosa lesz a lépéseknek itt megkapott fölső korlátjának, tehát, az egész eljárás lépésszám igénye $O\left(min\left\{m_1 n, m_2 n, n^{\omega+o(1)}\right\}\right)$-nagyságrendű.

Ezekután már csak azt kellene belátnunk, hogy a további lépésekben,az algoritmus

$$O((m_1 m_2)^{\frac{\beta}{\beta+1}} n^{\frac{2-\alpha\beta}{\beta+1}+o(1)} + n^{2+o(1)})$$

műveletet használ. □

Ehhez egy segéd lemmára lesz szükségünk, ami a naiv mátrix szorzó algoritmus lépésszámával kapcsolatos:

3.6.2. Lemma: *Bármely* $1 \leq l < n$*-re* $\sum_{k>l} a_{\pi(k)}b_{\pi(k)} \leq \frac{m_1 m_2}{l}$.

Lemma bizonyítás: Feltehetjük, hogy $a_1 \geq a_2 \geq \ldots\ldots \geq a_n$ és legyen $1 \leq l < n$.

Ekkor $la_{l+1} = a_{l+1} + a_{l+1} + ... + a_{l+1} \leq a_1 + a_2 + ... + a_{l-1} + a_l = \sum_{k \leq l} a_k \leq \sum_{k=1}^{n} a_k = m_1$. Ebből következik, hogy $a_{l+1} \leq \frac{m_1}{l}$. Az is látszik, hogy $\sum_{k>l} a_{\pi(k)} b_{\pi(k)} \leq \sum_{k>l} a_k b_k$, π definiciója miatt. Ezeket összerakva

$$\sum_{k>l} a_{\pi(k)} b_{\pi(k)} \leq \sum_{k>l} a_k b_k \leq a_{l+1} \sum_{k>l} b_k \leq \frac{m_1}{l} m_2 = \frac{m_1 m_2}{l}.$$

□

Visszatérve a lépésekhez:

Bizonyítás:

Az 5.lépésben nem végzünk műveletet.

Miután l-et megkaptuk, elvégezzük a mátrixszorzást, és ezek után végzünk gyűrűműveleteket.

Mivel l-et úgy választottuk, hogy a naiv mátrixszorzás és a hagyományos gyors mátrixszorzás optimális kombinációját adja, a gyűrűműveletek száma= $O(min\left\{m_1 n, m_2 n, n^{\omega+o(1)}\right\})$. Azt kell tehát megmutatnunk, hogy a 6.-8. lépések műveletigénye=$O(\left\{(m_1 m_2)^{\frac{\beta}{\beta+1}} n^{\frac{2-\alpha\beta}{\beta+1}+o(1)} + n^{2+o(1)}\right\})$.

6. és 7.lépésben, a két algoritmus műveletigényeinek összege: a két különböző algoritmus lépéseinek száma:$M(n,l,n) + \sum_{k>l} a_{\pi(k)} b_{\pi(k)}$.
Két esetet különböztetünk meg:
1) ha $m_1 m_2 \leq n^{2+\alpha}$;
2) ha $m_1 m_2 \geq n^{2+\alpha}$.

1) Legyen $l = \frac{m_1 m_2}{n^2}$, mivel $l \leq \frac{n^{2+\alpha}}{n^2} = n^\alpha$, így α definiciójából azt kapjuk, hogy $M(n,l,n) = O(n^{2+o(1)})$.

   Felhasználva a segéd lemmát és l kiválasztott értékét:

$$\begin{aligned} M(n,l,n) + \sum_{k>l} a_{\pi(k)} b_{\pi(k)} &\leq n^{2+o(1)} + \frac{m_1 m_2}{l} = \\ &= n^{2+o(1)} + \frac{(m_1 m_2) n^2}{m_1 m_2} = \\ &= n^{2+o(1)} + n^2 \\ &= O(n^{2+o(1)}). \end{aligned}$$

2) Legyen ebben az esetben $l = (m_1 m_2)^{\frac{1}{\beta+1}} n^{\frac{\alpha\beta-2}{\beta+1}}$.

Mivel

$$\begin{aligned} l \geq (n^{2+\alpha})^{\frac{1}{\beta+1}} n^{\frac{\alpha\beta-2}{\beta+1}} &= n^{\frac{2+\alpha}{\beta+1}} n^{\frac{\alpha\beta-2}{\beta+1}} = \\ &= n^{\frac{2+\alpha+\alpha\beta-2}{\beta+1}} = \\ &= n^{\frac{\alpha(1+\beta)}{\beta+1}} = \\ &= n^{\alpha}, \end{aligned}$$

így

$n^{2-\alpha\beta+o(1)} l^{\beta} \geq n^{2-\alpha\beta+o(1)} n^{\alpha\beta} = n^{2+o(1)}$, ezért

$$\begin{aligned} M(n,l,n) = n^{2-\alpha\beta+o(1)} l^{\beta} &= \\ &= n^{2-\alpha\beta+o(1)} (m_1 m_2)^{\frac{\beta}{\beta+1}} n^{\frac{\alpha\beta^2-2\beta}{\beta+1}} = \\ &= (m_1 m_2)^{\frac{\beta}{\beta+1}} n^{\frac{2\beta+2-\alpha\beta^2-\alpha\beta+o(1)+\alpha\beta^2-2\beta}{\beta+1}} = \\ &= (m_1 m_2)^{\frac{\beta}{\beta+1}} n^{\frac{2-\alpha\beta}{\beta+1}+o(1)}. \end{aligned}$$

Itt is felhasználjuk a segédlemmát:

$$\sum_{k>l} a_{\pi(k)} b_{\pi(k)} \leq \frac{m_1 m_2}{l} = (m_1 m_2)^{1-\frac{1}{\beta+1}} n^{\frac{2-\alpha\beta}{\beta+1}} = (m_1 m_2)^{\frac{\beta}{\beta+1}} n^{\frac{2-\alpha\beta}{\beta+1}}.$$

Ezeket összerakva $M(n,l,n) + \sum_{k>l} a_{\pi(k)} b_{\pi(k)} = O((m_1 m_2)^{\frac{\beta}{\beta+1}} n^{\frac{2-\alpha\beta}{\beta+1}+o(1)})$.

Tehát a 6. és a 7.-ben

$$\begin{aligned} O((m_1 m_2)^{\frac{\beta}{\beta+1}} n^{\frac{2-\alpha\beta}{\beta+1}+o(1)}) + O(n^{2+o(1)}) = \\ = O((m_1 m_2)^{\frac{\beta}{\beta+1}} n^{\frac{2-\alpha\beta}{\beta+1}+o(1)} + n^{2+o(1)}) \end{aligned}$$

műveletet végzünk.

Végül pedig a 8.lépésben egy összeadást végzünk, összeadunk két $n \times n$-es mátrixot, tehát a műveletigény általában n^2, azaz $O(n^{2+o(1)})$.

Viszont, ha $m_1 \leq m_2$, és $m_1 < n$, akkor C_1-ben és C_2-ban legfeljebb $m_1 n$

nem nulla elemünk van, így $C_1 + C_2$ kiszámolásánál$\leq m_1 n \leq m_2 n$ összeadást végzünk. □

Érdemes megemlítenünk egyébként l választására két speciális esetet:
Abban az esetben, amikor l=0, I üres halmaz lesz, csak a naiv mátrix szorzó algoritmust használjuk, ekkor a mátrixaink valószínűleg elég kevés nem nulla elemet tartalmaznak.
A 6. és a 8.lépésben egyáltalán nem fogunk műveletet végezni. A 7.lépésben a művelet igényünk $\sum_{k=1}^{n} a_{\pi(k)} b_{\pi(k)} = \sum_{k=1}^{n} a_k b_k$. Korábban már láttuk, hogy ez felülről becsülhető $min\{m_1 n, m_2 n\}$-nel.
Amikor pedig l=n, ilyenkor J lesz üres halmaz, és ekkor csak a gyors sűrű téglalap alakú mátrix szorzó algoritmust használjuk, ennek a művelet igénye: $M(n,l,n) = M(n,n,n) = n^{\omega+o(1)}$ lesz.

Ha $m_1 = m_2 = m$, akkor az algoritmus legfeljebb $O\left(m^{\frac{2\beta}{\beta+1}} n^{\frac{2-\alpha\beta}{\beta+1}+o(1)} + n^{2+o(1)}\right)$ műveletet használ.

A legjobb eredményeket használva ω-ra, α-ra és β-ra ez az érték legfeljebb $O(m^{0.7} n^{1.2} + n^{2+o(1)})$.

Ha $m = O(n^{\frac{\omega+1}{2}-\epsilon})$, bármely ϵ-ra akkor $O(n^{\omega})$ műveletet használunk.
Ha $m = O(n^{1+\frac{\alpha}{2}})$, akkor $n^{2+o(1)}$ műveletet használunk.

Ha pedig, m_1 vagy $m_2 = 0$, akkor $C \equiv 0$, ez az eset nem túl érdekes, az algoritmus minden lépésben 0 lépést tesz meg, ezért nem is foglalkoztunk vele a bizonyításban sem.

4. Irodalomjegyzék

Hivatkozások

[S69] Strassen, Volker: Gaussian elimination is not optimal. *Numer. Math.* **13** (1969), 354–356.

[YZ05] Yuster, Raphael; Zwick, Uri: Fast Sparse matrix multiplication. *ACM Transactions on Algorithms* **1** (2005), 2–13.

[FH13] Faragó, István;Hórváth, Róbert: Numerikus módszerek. (2013)

[F07] DA Fonesca, C.M.: On the eigenvalues of some tridiagonal matrices. *Journal of Computational and Applied Mathematics*, **200** (2007), 283–286

[GA16] Aurél, Galantai: Algoritmuselmélet(jegyzet) (2016), URL: uni-obuda.hu/users/galantai/AlgElm2016tav.pdf

[VW11] Virginia, Vassilevska Williams: Multiplying matrices in O(n2.373) time. (2014), URL: http://theory.stanford.edu/˜virgi/matrixmult-f.pdf

[VP84] Pan, Victor: How to multiply matrices faster. *Springer LNCS,* **179,** (1984).

[LP08] Lax, Peter D.: Lineáris algebra és alkalmazásai. *Akadémiai kiadó,* (2008),

[RM] https://en.wikipedia.org/wiki/Sparse_matrix