Κατασκευή δυνάμεων

Παράδειγμα 1.10.7

Πρόβλημα κατασκευής δυνάμεων

- Πρόβλημα 1: Υπολογισμός n-οστής δύναμης πίνακα Α μεγέθους $m \times m$, όπου $n \geq 1$ και συνήθως n «πολύ μεγάλος».

$$\mathrm{A}^n = ?$$

- Πρόβλημα 2: Υπολογισμός γινομένου πίνακα με διάνυσμα ή πίνακα, $y = \mathrm{A}^n x$, όπου x, y κατάλληλα διανύσματα ή πίνακες.

$$y = \mathrm{A}^n x = ?$$

Υπολογιστικό κόστος γινομένου πινάκων

- Πολλαπλασιασμός πινάκων μεγέθους $m \times m$:

  Από Γραμμική Άλγεβρα

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{pmatrix} \quad B = \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{pmatrix}$$

$$A\,B = \begin{pmatrix} a_{11}b_{11} + a_{12}b_{21} + a_{13}b_{31} & a_{11}b_{12} + a_{12}b_{22} + a_{13}b_{32} & a_{11}b_{13} + a_{12}b_{23} + a_{13}b_{33} \\ a_{21}b_{11} + a_{22}b_{21} + a_{23}b_{31} & a_{21}b_{12} + a_{22}b_{22} + a_{23}b_{32} & a_{21}b_{13} + a_{22}b_{23} + a_{23}b_{33} \end{pmatrix}$$

- Πόσες πράξεις χρειάζονται για τον τελικό υπολογισμό του γινομένου AB όπου A και B πίνακες μεγέθους $m \times m$;

  #πολλαπλασιασμοί=$(m \cdot m) \cdot m$

  #προσθέσεις=$(m \cdot m) \cdot (m - 1)$

- ***Παραδοχή: θεωρούμε την πράξη $A \cdot B$ ή $A \cdot A$ σαν 1 βήμα***

Υπολογιστικό κόστος γινομένου πινάκων

Στόχος η εύρεση αλγορίθμου κατασκευής δυνάμεων με τα λιγότερα δυνατά βήματα, τους λιγότερους δυνατούς πολλαπλασιασμούς πινάκων

- ***Παραδοχή: θεωρούμε την πράξη $A \cdot B$ ή $A \cdot A$ σαν 1 βήμα***

Κατασκευή δυνάμεων (1/2)

- Απλοϊκός τρόπος υπολογισμού δύναμης πίνακα:

$$\mathrm{A}^n = A \cdot A \cdot \cdots \cdot A$$

#βημάτων=$n-1$

- Έστω $n = 2^r$

Για $r = 1$: $A^n = A^2 = A \cdot A \rightarrow$ #βημάτων=1=r

Για $r = 2$: $A^n = A^4 = A^2 \cdot A^2 \rightarrow$ #βημάτων=2=r

Για $r = 3$: $A^n = A^8 = A^4 \cdot A^4 \rightarrow$ #βημάτων=3=r

Γενικά: αν $n = 2^r$ χρειαζόμαστε r βήματα για την κατασκευή της δύναμης A^n

Κατασκευή δυνάμεων (1/2)

- Απλοϊκός τρόπος υπολογισμού δύναμης πίνακα:

$$A^n = A \cdot A \cdot \cdots \cdot A$$

#βημάτων=$n-1$

Τι ισχύει για οποιοδήποτε n ;
Πως μπορούμε να εκμεταλλευτούμε την προηγούμενη ιδιότητα;

- ~~Έστω $n = 2^r$~~

Για $r=1$: $A^n = A^2 = A \cdot A \to$ #βημάτων=1=r

Για $r=2$: $A^n = A^4 = A^2 \cdot A^2 \to$ #βημάτων=2=r

Για $r=3$: $A^n = A^8 = A^4 \cdot A^4 \to$ #βημάτων=3=r

Γενικά: αν $n = 2^r$ χρειαζόμαστε r βήματα για την κατασκευή της δύναμης A^n

Κατασκευή δυνάμεων (2/2)

- Οποιοσδήποτε αριθμός στο δεκαδικό σύστημα μπορεί να αναπαραστεί και στο δυαδικό σύστημα

$$(39)_{10} \rightarrow (100111)_2$$

- Οποιαδήποτε δύναμη μπορεί να «σπάσει» σε δυνάμεις του 2

$A^{39} = A^{32} \cdot A^{4} \cdot A^{2} \cdot A \rightarrow$ # βημάτων= ???

Κατασκευή δυνάμεων (2/2)

- Οποιοσδήποτε αριθμός στο δεκαδικό σύστημα μπορεί να αναπαρασταθεί και στο δυαδικό σύστημα

$$(39)_{10} \rightarrow (100111)_2$$

- Οποιαδήποτε δύναμη μπορεί να «σπάσει» σε δυνάμεις του 2

$$A^{39} = A^{32} \cdot A^4 \cdot A^2 \cdot A$$

\# βημάτων= **5**+**3**

Γιατί δεν προσθέτω και τα βήματα που χρειάζονται για να υπολογίσω τις δυνάμεις A^2 και A^4;

Γενικά: αν $(n)_{10} = (b_r \cdots b_1)_2$ με $b_r \neq 0$ απαιτούνται $r-1$ βήματα για τον σχηματισμό της τελικής δύναμης A^r και για τον πολλαπλασιασμό των δυνάμεων A^i με $b_i = 1$ απαιτούνται ακόμα $r-1$ βήματα στη χειρότερη περίπτωση. Με $r = \lceil logn \rceil$

$$\boldsymbol{\Pi \times \Pi(n) = 2(\lceil logn \rceil - 1) = O(logn)}$$

Αλγόριθμος Δυνάμεων

ΕΙΣΟΔΟΣ: $n \geq 2, m \geq 1$, ο $m \times m$ πίνακας A και το $m \times 1$ διάνυσμα x

ΕΞΟΔΟΣ: το αποτέλεσμα $A^n x$ αποθηκευμένο στο διάνυσμα x (και ο τελικός αριθμός των βημάτων r)

$r = 0$
$q = n/2$
$b = mod(n, 2)$
if $b \neq 0$ then $x = Ax$
while $q \neq 0$ do
    $A = A \times A$
    $r = r + 1$
    $b = mod(q, 2)$
    $q = q/2$
    if $b \neq 0$ then $x = Ax$
end-while-loop

Αλγόριθμος Winograd για πολλαπλασιασμό πινάκων

Παράδειγμα 1.10.7

Πολλαπλασιασμός διανυσμάτων

- Εσωτερικό γινόμενο 2 διανυσμάτων v, w μεγέθους $n \times 1$:

$$v^T w = w^T v = [v_1 \cdots v_n] \cdot \begin{bmatrix} w_1 \\ \vdots \\ w_n \end{bmatrix} = v_1 w_1 + \cdots + v_n w_n$$

- Πολυπλοκότητα:

#πολλαπλασιασμών = ?

#προσθέσεων = ?

- Βέλτιστος τρόπος υπολογισμού;

Πολλαπλασιασμός διανυσμάτων

- Εσωτερικό γινόμενο 2 διανυσμάτων v, w μεγέθους $n \times 1$:

$$v^T w = w^T v = [v_1 \cdots v_n] \cdot \begin{bmatrix} w_1 \\ \vdots \\ w_n \end{bmatrix} = v_1 w_1 + \cdots + v_n w_n$$

- Πολυπλοκότητα:

#πολλαπλασιασμών = $\boldsymbol{n}$

#προσθέσεων = $\boldsymbol{n - 1}$

- Βέλτιστος τρόπος υπολογισμού; ΝΑΙ

Πολλαπλασιασμός πίνακα με διάνυσμα

- Πολλαπλασιασμός πίνακα A μεγέθους $m \times n$ με διάνυσμα v μεγέθους $n \times 1 \rightarrow$ **αποτέλεσμα διάνυσμα $\boldsymbol{w}$ μεγέθους $\boldsymbol{m} \times \mathbf{1}$**

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{i1} & \cdots & a_{in} \\ \vdots & & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix}, \qquad v = \begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix}, \qquad w = \begin{bmatrix} w_1 \\ \vdots \\ w_m \end{bmatrix}$$

- Κάθε συντεταγμένη του w είναι το εσωτερικό γινόμενο της αντίστοιχης γραμμής του A με το v:

$$w_i = [a_{i1} \cdots a_{in}] \cdot \begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix} = a_{i1}v_1 + \cdots + a_{in}v_n, \qquad i = 1, \ldots, m$$

- Πολυπλοκότητα:

#πολλαπλασιασμών = ?

#προσθέσεων = ?

- Βέλτιστος τρόπος υπολογισμού;

Πολλαπλασιασμός πίνακα με διάνυσμα

- Πολλαπλασιασμός πίνακα A μεγέθους $m \times n$ με διάνυσμα v μεγέθους $n \times 1 \rightarrow$ **αποτέλεσμα διάνυσμα $\boldsymbol{w}$ μεγέθους $\boldsymbol{m \times 1}$**

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{i1} & \cdots & a_{in} \\ \vdots & & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix}, \quad v = \begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix}, \quad w = \begin{bmatrix} w_1 \\ \vdots \\ w_m \end{bmatrix}$$

- Κάθε συντεταγμένη του w είναι το εσωτερικό γινόμενο της αντίστοιχης γραμμής του A με το v:

$$w_i = [a_{i1} \cdots a_{in}] \cdot \begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix} = a_{i1}v_1 + \cdots + a_{in}v_n, \qquad i = 1, \dots, m$$

Όπου m το πλήθος των εσωτερικών γινομένων που χρειάζεται να υπολογιστούν

- Πολυπλοκότητα:

#πολλαπλασιασμών $= \boldsymbol{mn}$

#προσθέσεων $= \boldsymbol{m(n-1)}$

- Βέλτιστος τρόπος υπολογισμού; NAI

Πολλαπλασιασμός πινάκων

- Πολλαπλασιασμός πίνακα A μεγέθους $m \times n$ με πίνακα B μεγέθους $n \times q$ $\rightarrow$ **αποτέλεσμα πίνακας $\boldsymbol{C}$ μεγέθους $\boldsymbol{m \times q}$**

- Κάθε στοιχείο c_{ij} του πίνακα C ισούται με το εσωτερικό γινόμενο της i-οστής γραμμής του πίνακα A με την j-οστή στήλη του πίνακα B.

$$c_{ij} = [a_{i1} \cdots a_{in}] \cdot \begin{bmatrix} b_{1j} \\ \vdots \\ b_{nj} \end{bmatrix} = \sum_{k=1}^{n} a_{ik} b_{jk}$$

με $i = 1, \dots, m$ και $j = 1, \dots, q$

- Πολυπλοκότητα:

#πολλαπλασιασμών = ?

#προσθέσεων = ?

- Βέλτιστος τρόπος υπολογισμού;

Πολλαπλασιασμός πινάκων

- Πολλαπλασιασμός πίνακα Α μεγέθους $m \times n$ με πίνακα Β μεγέθους $n \times q$ → **αποτέλεσμα πίνακας $\boldsymbol{C}$ μεγέθους $\boldsymbol{m \times q}$**

- Κάθε στοιχείο c_{ij} του πίνακα C ισούται με το εσωτερικό γινόμενο της i-οστής γραμμής του πίνακα Α με την j-οστή στήλη του πίνακα Β.

$$c_{ij} = [a_{i1} \cdots a_{in}] \cdot \begin{bmatrix} b_{1j} \\ \vdots \\ b_{nj} \end{bmatrix} = \sum_{k=1}^{n} a_{ik} b_{jk}$$

με $i = 1, \dots, m$ και $j = 1, \dots, q$

- Πολυπλοκότητα:

#πολλαπλασιασμών =$\boldsymbol{mqn}$

#προσθέσεων =$\boldsymbol{mq(n-1)}$

Όπου mq το πλήθος των εσωτερικών γινομένων που χρειάζεται να υπολογιστούν

- Βέλτιστος τρόπος υπολογισμού; ΟΧΙ

Πολλαπλασιασμός πινάκων

- Πολλαπλασιασμός πίνακα A μεγέθους $m \times n$ με πίνακα B μεγέθους $n \times q$ → **αποτέλεσμα πίνακας $\boldsymbol{C}$ μεγέθους $\boldsymbol{m} \times \boldsymbol{q}$**

- Κάθε στοιχείο c_{ij} του πίνακα C ισούται με το εσωτερικό γινόμενο της i-οστής γραμμής του πίνακα A με την j-οστή στήλη του πίνακα B.

$$c_{ij} = [a_{i1} \cdots a_{in}] \cdot \begin{bmatrix} b_{1j} \\ \vdots \\ b_{nj} \end{bmatrix} = \sum_{k=1}^{n} a_{ik} b_{jk}$$

$$\text{με } i = 1, \dots, m \text{ και } j = 1, \dots, q$$

- Πολυπλοκότητα:

$$\#\text{πολλαπλασιασμών} = \boldsymbol{mqn}$$

$$\#\text{προσθέσεων} = \boldsymbol{mq(n-1)}$$

- Για τετραγωνικούς πίνακες A, B $(m = q = n)$

$$\#\text{πολλαπλασιασμών} = \boldsymbol{n^3}$$

$$\#\text{προσθέσεων} = \boldsymbol{n^3 - n^2}$$

Αλγόριθμος Winograd (1/3)

- Εσωτερικό γινόμενο διανυσμάτων v, w μεγέθους 4×1:

$$v^T w = v_1 w_1 + v_2 w_2 + v_3 w_3 + v_4 w_4$$
$$= (v_1 + w_2)(v_2 + w_1) + (v_3 + w_4)(v_4 + w_3) - \alpha - \beta$$

Όπου

$$\alpha = v_1 v_2 + v_3 v_4,\ \beta = w_1 w_2 + w_3 w_4$$

- Γενίκευση για διανύσματα μεγέθους $n \times 1$, με $n = 2p$

$$v^T w = v_1 w_1 + \cdots + v_{2p} w_{2p} = \sum_{i=1}^{p} (v_{2i-1} + w_{2i}) \cdot (v_{2i} + w_{2i-1}) - \alpha - \beta$$

Όπου

$$\alpha = \sum_{i-1}^{p} v_{2i-1} v_{2i}, \qquad \beta = \sum_{i-1}^{p} w_{2i-1} w_{2i}$$

Αλγόριθμος Winograd (1/3)

- Εσωτερικό γινόμενο διανυσμάτων v, w μεγέθους 4×1:

$$v^T w = v_1 w_1 + v_2 w_2 + v_3 w_3 + v_4 w_4$$
$$= (v_1 + w_2)(v_2 + w_1) + (v_3 + w_4)(v_4 + w_3) - \alpha - \beta$$

Όπου

$$\boldsymbol{\alpha = v_1 v_2 + v_3 v_4, \ \beta = w_1 w_2 + w_3 w_4}$$

Οι συντελεστές $\alpha . \beta$ περιέχουν όρους μόνο από το διάνυσμα v και w αντίστοιχα

- Γενίκευση για δια

$$v^T w = v_1 w_1 + \cdots + v_{2p} w_{2p} \ \ldots \sum_{i=1} \ldots_{2i-1} + w_{2i}) \ (v_{2i} + w_{2i-1}) \ \ldots \alpha - \beta$$

Όπου

$$\boldsymbol{\alpha = \sum_{i-1}^{p} v_{2i-1} v_{2i}, \qquad \beta = \sum_{i-1}^{p} w_{2i-1} w_{2i}}$$

Αλγόριθμος Winograd (1/3)

- Εσωτερικό γινόμενο διανυσμάτων v, w μεγέθους 4×1:

$v^T w = v_1 w_1 + v_2 w_2$

$= (v_1 + w_2)(v_2$

Όπου

$\alpha =$

#πολλαπλασιασμών = $3p = 3n/2$
#προσθέσεων =$2p + 2 + p + p = 2n + 2$
Επομένως είναι χειρότερος από τον απλοϊκό τρόπο

- Γενίκευση για διανύσματα μεγέθους $n \times 1$, με $n = 2p$

$$v^T w = v_1 w_1 + \cdots + v_{2p} w_{2p} = \sum_{i=1}^{p} (v_{2i-1} + w_{2i}) \cdot (v_{2i} + w_{2i-1}) - \alpha - \beta$$

Όπου

$$\alpha = \sum_{i-1}^{p} v_{2i-1} v_{2i}\,, \qquad \beta = \sum_{i-1}^{p} w_{2i-1} w_{2i}$$

Αλγόριθμος Winograd (2/3)

- Πολλαπλασιασμός πίνακα Α μεγέθους $m \times n$ με πίνακα Β μεγέθους $n \times q$, με n ζυγό ($n = 2p$)

$$v^T w = v_1 w_1 + \cdots + v_{2p} w_{2p} = \sum_{i=1}^{p} (v_{2i-1} + w_{2i}) \cdot (v_{2i} + w_{2i-1}) - \alpha - \beta$$

Όπου

$$\alpha = \sum_{i-1}^{p} v_{2i-1} v_{2i}, \qquad \beta = \sum_{i-1}^{p} w_{2i-1} w_{2i}$$

- Υπολογίζουμε το άθροισμα α, για κάθε γραμμή v του Α.

  #πολλαπλασιασμών: $\mu_{\mathrm{A}} = ?$

  #προσθέσεων: $\pi_{\mathrm{A}} = ?$
- Παρόμοια υπολογίζουμε το άθροισμα β, για όλες τις στήλες w του Β.

  #πολλαπλασιασμών: $\mu_{\mathrm{B}} = ?$

  #προσθέσεων: $\pi_{\mathrm{B}} = ?$
- Υπολογίζουμε το πρώτο άθροισμα του εσωτερικού γινομένου για κάθε γραμμή του Α και κάθε στήλη του Β.

  #πολλαπλασιασμών: $\mu_{\mathrm{AB}} = ?$

  #προσθέσεων: $\pi_{\mathrm{AB}} = ?$

Αλγόριθμος Winograd (2/3)

- Πολλαπλασιασμός πίνακα A μεγέθους $m \times n$ με πίνακα B μεγέθους $n \times q$, με n ζυγό ($n = 2p$)

- Υπολογίζουμε το άθροισμα α, για κάθε γραμμή v του A.

  #πολλαπλασιασμών: $\mu_{\mathrm{A}} = pm = \left(\frac{n}{2}\right)m$

  #προσθέσεων: $\pi_{\mathrm{A}} = pm = \left(\frac{n}{2}\right)m$
- Παρόμοια υπολογίζουμε το άθροισμα β, για όλες τις στήλες w του B.

  #πολλαπλασιασμών: $\mu_{\mathrm{B}} = pq = \left(\frac{n}{2}\right)q$

  #προσθέσεων: $\pi_{\mathrm{B}} = qm = \left(\frac{n}{2}\right)q$
- Υπολογίζουμε το πρώτο άθροισμα του εσωτερικού γινομένου για κάθε γραμμή του A και κάθε στήλη του B.

  #πολλαπλασιασμών: $\mu_{\mathrm{AB}} = pmq = \left(\frac{n}{2}\right)mq$

  #προσθέσεων: $\pi_{\mathrm{AB}} = 2pmq + 2mq = (n+2)mq$

Αλγόριθμος Winograd (3/3)

- Συνολική πολυπλοκότητα:

#πολλαπλασιασμών $= \mu_{\mathrm{A}} + \mu_{\mathrm{B}} + \mu_{\mathrm{AB}} = \left(\frac{n}{2}\right)(mq + m + q)$

#προσθέσεων $= \pi_{\mathrm{A}} + \pi_{\mathrm{B}} + \pi_{\mathrm{AB}} = \left(\frac{n}{2}\right)(m + q) + (n + 2)mq$

- Αν A, B τετραγωνικοί πίνακες → $m = q = n$ ζυγός

#πολλαπλασιασμών = $\boldsymbol{\frac{n^3}{2} + n^2}$

#προσθέσεων= $\boldsymbol{n^3 + 3n^2}$

- Απλοϊκός τρόπος:

#πολλαπλασιασμών = $\boldsymbol{n^3}$

#προσθέσεων = $\boldsymbol{n^3 - n^2}$

Γίνεται εξοικονόμηση πολλαπλασιασμών ενώ πληρώνουμε σε (φθηνότερες ίσως) προσθέσεις.

Αλγόριθμος Winograd (3/3)

- Συνολική πολυπλοκότητα:

#πολλαπλασιασμών $=\mu_{\mathrm{A}} + \mu_{\mathrm{B}} + \mu_{\mathrm{AB}} = \left(\frac{n}{2}\right)(mq + m + q)$

#προσθέσεων $=\pi_{\mathrm{A}} + \pi_{\mathrm{B}} + \pi_{\mathrm{AB}} = \left(\frac{n}{2}\right)(m + q) + (n + 2)mq$

- Αν A, B τετραγωνικοί πίνακες $\rightarrow m = q = n$ ζυγός

#πολλαπλασιασμών = $\boldsymbol{\frac{n^3}{2} + n^2}$

#προσθέσεων= $\boldsymbol{n^3 + 3n^2}$

- Απλοϊκός τρόπος:

#πολλαπλασιασμών = $\boldsymbol{n^3}$

#προσθέσεων = $\boldsymbol{n^3 - n^2}$

Ο κυρίαρχος όρος n^3 ελαττώθηκε αλλά δεν εξαφανίστηκε.