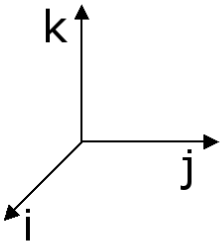
Κεφάλαιο 2.4
Matrix Algorithms

Παρουσίαση στα πλαίσια του μαθήματος
«Παράλληλοι Αλγόριθμοι»

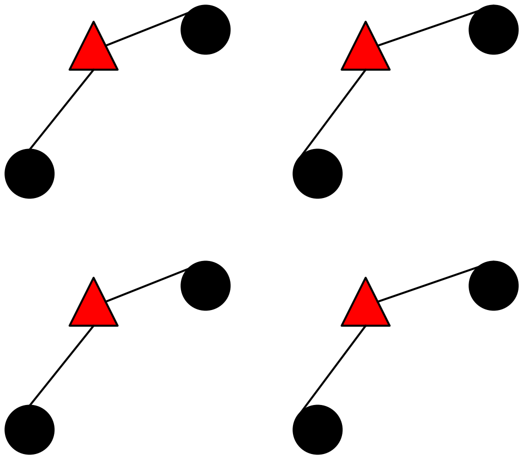
Καούρη Γεωργία
Μήτσου Βασιλική

Κατασκευή NxNxN Mesh of trees (1/3)

Στον NxNxN κύβο προσθέτω τους εξής κόμβους:



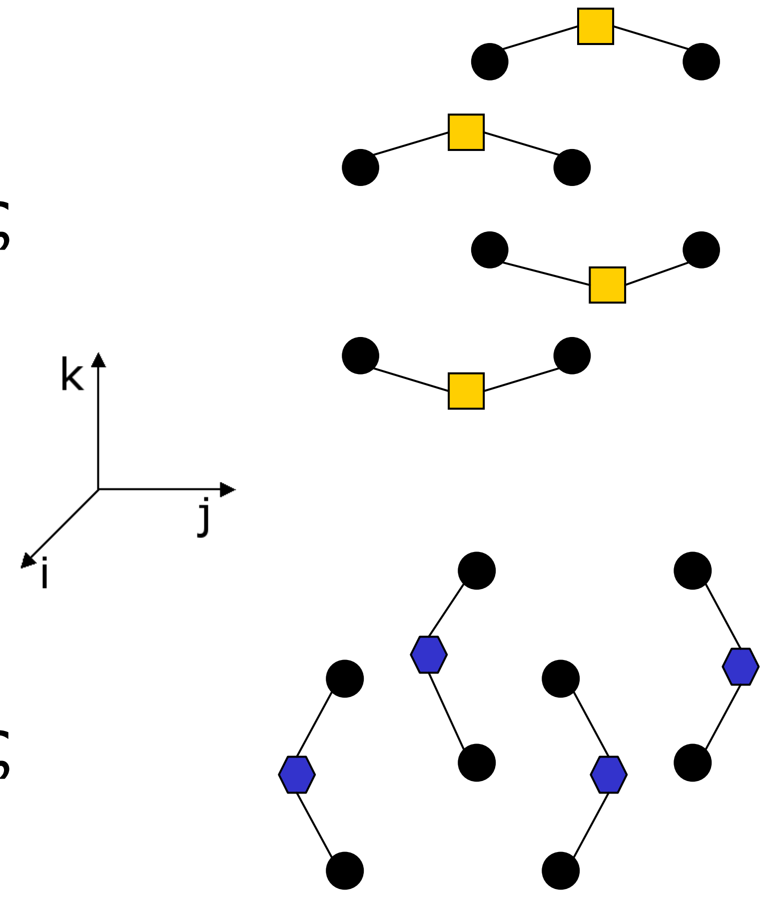
- Για κάθε j,k ($1 \le j,k \le N$) προσθέτω N-1 κόμβους ώστε να φτιάξω ένα πλήρες δέντρο (αυτό καλείται dimension 1 tree).



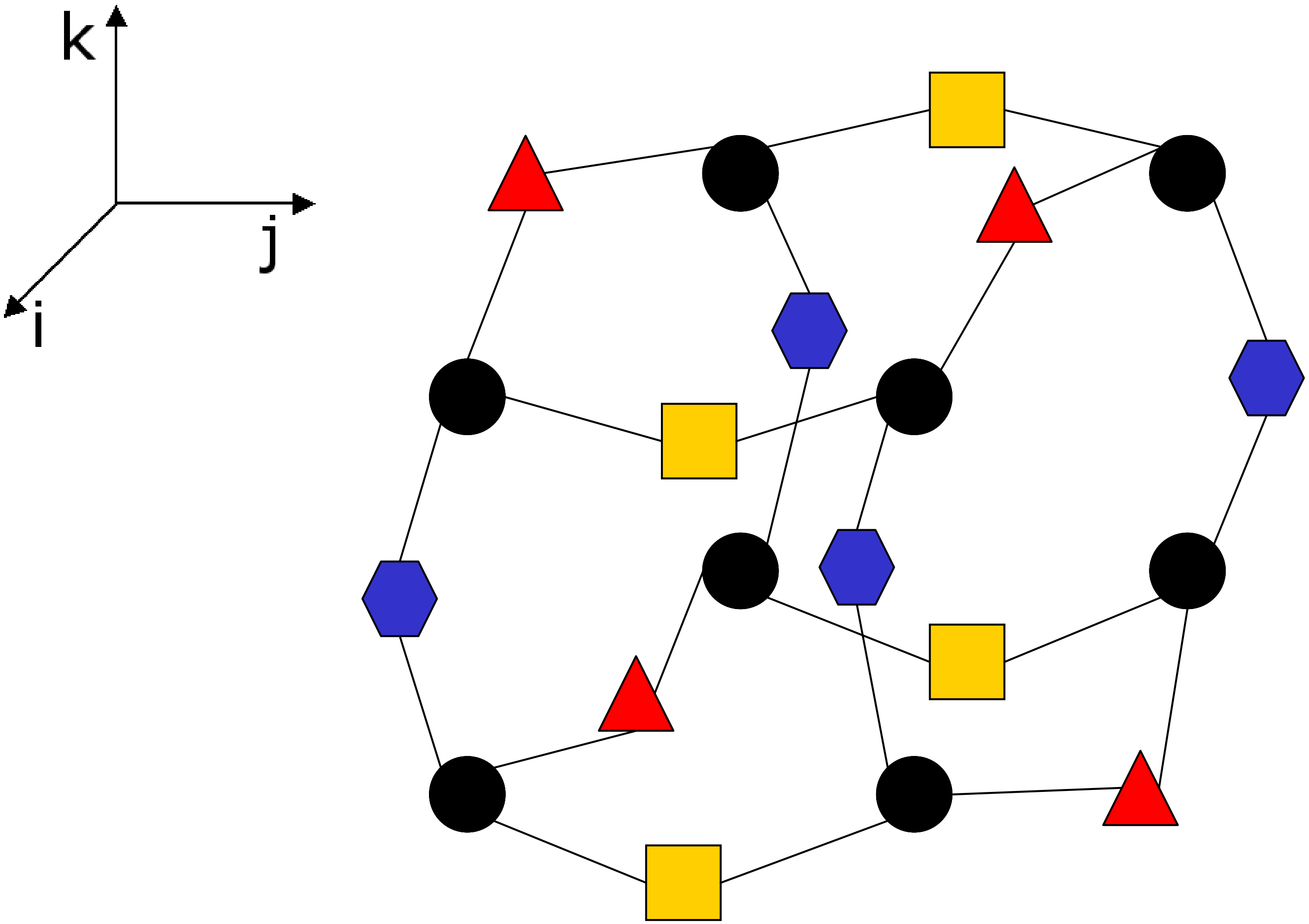
Κατασκευή NxNxN Mesh of trees (2/3)

- Για κάθε i,j ($1 \le i, k \le N$) προσθέτω N-1 κόμβους ώστε να φτιάξω ένα πλήρες δέντρο (αυτό καλείται dimension 2 tree).

- Για κάθε i,k ($1 \le i, j \le N$) προσθέτω N-1 κόμβους ώστε να φτιάξω ένα πλήρες δέντρο (αυτό καλείται dimension 3 tree).



Κατασκευή NxNxN Mesh of trees (3/3)



NxNxN Mesh of trees (1/2)

- Σύνολο κόμβων: $N^3+3N^2(N-1) =$
  $= 4N^3-3N^2$
- Σύνολο ακμών: $3N^2(2N-2) = 6N^3-6N^2$
- Όλοι οι κόμβοι έχουν βαθμό 3 εκτός από τις ρίζες που έχουν βαθμό 2
- Διάμετρος: 6logN
- Bisection width: N^2

NxNxN Mesh of trees (2/2)

- Κάθε NxNxN Mesh of trees αποτελείται από N NxN Meshes of trees των οποίων τα φύλλα είναι συνδεδεμένα με N^2 δυαδικά δέντρα.
- Κάθε NxNxN Mesh of trees περιέχει 2^{3i} $N/2^i \times N/2^i \times N/2^i$ meshes of trees.

Matrix Multiplication(1/3)

Για να πολλαπλασιάσω τους NxN πίνακες $A=(a_{ij})$ και $B=(b_{ij})$ σε ένα NxNxN mesh of trees κάνω τα εξής βήματα:

- Εισάγω την τιμή του a_{ij} στη ρίζα του i,j dimension 3 δέντρου και την τιμή του b_{jk} στη ρίζα του j,k dimension 1 δέντρου.

Matrix Multiplication(2/3)

- Στα επόμενα logN βήματα οι τιμές των A και B κατεβαίνουν και στο βήμα logN+1 έχουν φτάσει στα φύλλα και πολλαπλασιάζονται μεταξύ τους. Τα αποτελέσματα στέλνεται στους γονιούς τους στο δέντρο dimension 2.
- Στα επόμενα logN βήματα τα γινόμενα αυτά προστίθενται και τελικά φτάνει στη ρίζα του 2 dimension δέντρου η τιμή $c_{ik} = \sum_{j=1}^{N} a_{ij} b_{jk}$, το οποίο είναι το i,k στοιχείο του πίνακα C=AxB.

Matrix Multiplication(3/3)

- Ο Αλγόριθμος αυτός εκτελείται σε 2logN+1, δηλαδή σε O(logN) βήματα κι αυτό δε μπορεί να βελτιωθεί περαιτέρω. Μπορούμε όμως να μειώσουμε τον αριθμό των επεξεργαστών σε M(N), όπου M(N) είναι ο αριθμός των πράξεων που εκτελεί ο σειριακός αλγόριθμος (Μέχρι σήμερα $M(N)= N^{2.38}$).
- Με pipelining μπορούμε να πολλαπλασιάσουμε k ζευγάρια NxN πινάκων σε 2logN+k βήματα.

Αντιστροφή Κάτω Τριγωνικού Πίνακα(1/6)

Κάτω τριγωνικός πίνακας NxN είναι ένας πίνακας της μορφής:

$$\begin{bmatrix} x & 0 & 0 & \cdots & 0 \\ x & x & & 0 & 0 \\ x & & \ddots & & \vdots \\ \vdots & x & & x & 0 \\ x & x & x & \cdots & x \end{bmatrix}$$

Στα επόμενα θεωρώ ότι το N είναι δύναμη του 2.

Αντιστροφή Κάτω Τριγωνικού Πίνακα(2/6)

Χωρίζω τον πίνακα σε τέσσερις ισομεγέθεις μικρότερους ως εξής:

$$A = \begin{bmatrix} A_1 & 0 \\ A_3 & A_2 \end{bmatrix}$$

όπου οι A_1 και A_2 είναι κάτω τριγωνικοί

Τότε ο A^{-1} είναι αντιστρέψιμος και μάλιστα

$$A^{-1} = \begin{bmatrix} A_1{}^{-1} & 0 \\ X & A_2{}^{-1} \end{bmatrix}$$

όπου $X=-A_2{}^{-1}A_3\,A_1{}^{-1}$

Αντιστροφή Κάτω Τριγωνικού Πίνακα(3/6)

- Στην πρώτη φάση υπολογίζουμε τα διαγώνια στοιχεία $a_{ii}^{-1} = 1/a_{ii}$
- Στη δεύτερη φάση υπολογίζουμε τα διαγώνια 2x2 blocks του A^{-1} υπολογίζοντας τα στοιχεία $a_{2i,\ 2i-1}^{-1} = -a_{2i,\ 2i}^{-1} \cdot a_{2i,\ 2i-1} \cdot a_{2i,\ 2i}^{-1}$
- Στην φάση k υπολογίζουμε τα διαγώνια $2^{k-1} \times 2^{k-1}$ blocks του A^{-1} αφού έχουμε υπολογίσει τα $2^{k-2} \times 2^{k-2}$ blocks κατά τη διάρκεια της προηγούμενης φάσης.

Αντιστροφή Κάτω Τριγωνικού Πίνακα(4/6)

$$\begin{matrix} x & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ x & x & 0 & 0 & 0 & 0 & 0 & 0 \\ x & x & x & 0 & 0 & 0 & 0 & 0 \\ x & x & x & x & 0 & 0 & 0 & 0 \\ x & x & x & x & x & 0 & 0 & 0 \\ x & x & x & x & x & x & 0 & 0 \\ x & x & x & x & x & x & x & 0 \\ x & x & x & x & x & x & x & x \end{matrix}$$

Αρχική Κατάσταση

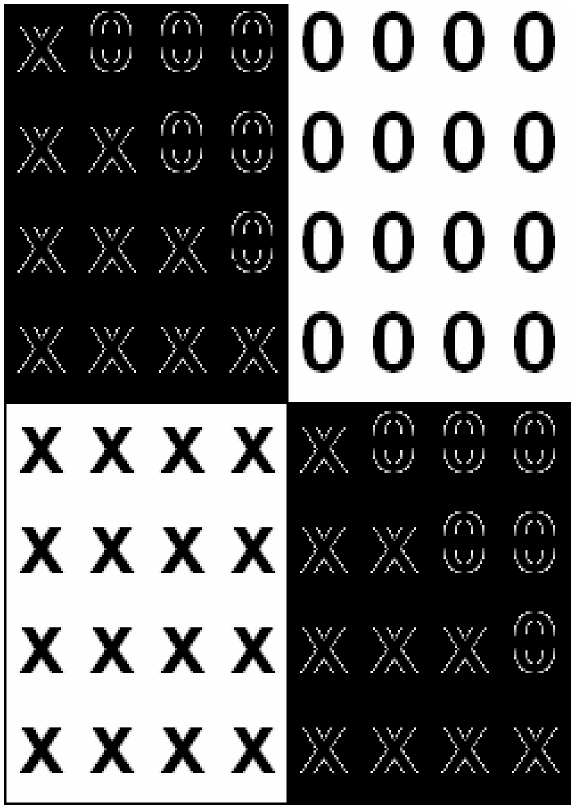
$$\begin{matrix} x & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ x & x & 0 & 0 & 0 & 0 & 0 & 0 \\ x & x & x & 0 & 0 & 0 & 0 & 0 \\ x & x & x & x & 0 & 0 & 0 & 0 \\ x & x & x & x & x & 0 & 0 & 0 \\ x & x & x & x & x & x & 0 & 0 \\ x & x & x & x & x & x & x & 0 \\ x & x & x & x & x & x & x & x \end{matrix}$$

1η Φάση

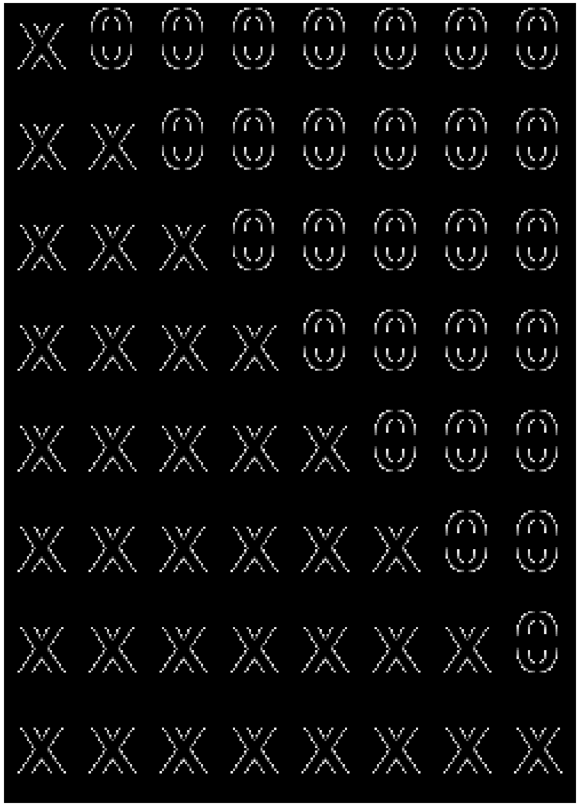
$$\begin{matrix} x & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ x & x & 0 & 0 & 0 & 0 & 0 & 0 \\ x & x & x & 0 & 0 & 0 & 0 & 0 \\ x & x & x & x & 0 & 0 & 0 & 0 \\ x & x & x & x & x & 0 & 0 & 0 \\ x & x & x & x & x & x & 0 & 0 \\ x & x & x & x & x & x & x & 0 \\ x & x & x & x & x & x & x & x \end{matrix}$$

2η Φάση

Αντιστροφή Κάτω Τριγωνικού Πίνακα(5/6)



3^{η} Φάση



4^{η} Φάση

Αντιστροφή Κάτω Τριγωνικού Πίνακα(6/6)

Ο αλγόριθμος εκτελείται σε 2 $\log^2 N$ βήματα σε ένα NxNxN mesh of trees ως εξής:

Στη φάση k>1 υπολογίζουμε τον εκάστοτε A_3 εκτελώντας παράλληλα $N / 2^{k-1}$ ζευγάρια πολλαπλασιασμών πινάκων $2^{k-2} \times 2^{k-2}$ σε κάποιο $2^{k-2} \times 2^{k-2} \times 2^{k-2}$ mesh of trees που περιέχεται στο NxNxN mesh of trees. Ο καθένας πραγματοποιείται σε $2(k-2)+1 = 2k-3$ βήματα. Άρα θα εκτελεστούν $2 \cdot (2k-3)$, δηλαδή το πολύ 4k βήματα.

Έτσι συνολικά θα εκτελεστούν το πολύ 2 $\log^2 N$ βήματα.

Αντιστροφή Τυχαίου Πίνακα σε $O(\log^2 N)$ βήματα.

Αλγόριθμος του Csanky	***Newton iteration***
▪ Ακριβής λύση ▪ Μεγάλος $\{\Theta(N^4)\}$ αριθμός επεξεργαστών ▪ Ασταθής	▪ Προσεγγιστική λύση ▪ Μικρός $\{O(N^3)\}$ αριθμός επεξεργαστών ▪ Ευσταθής

Αντιστροφή Τυχαίου Πίνακα: Αλγόριθμος του Csanky(1/5)

Χαρακτηριστικό Πολυώνυμο του Α:

$C_A(x) = \det(xI - A) =$

$= x^N + c_1 x^{N-1} + c_2 x^{N-2} + \dots + c_N$

Αν τα c_i είναι γνωστά τότε από θ. Cayley - Hamilton :

$A^{-1} = -1/c_N \left(A^{N-1} + c_1 A^{N-2} + \dots + c_{N-2} A + \dots + c_{N-1}\right)$

Αντιστροφή Τυχαίου Πίνακα: Αλγόριθμος του Csanky(2/5)

Λήμμα Leverier

Οι συντελεστές c_i του χαρακτηριστικού πολυωνύμου ικανοποιούν την εξίσωση:

$$\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ s_1 & 2 & 0 & \cdots & 0 \\ s_2 & s_1 & & & 0 \\ \vdots & & \ddots & \ddots & \vdots \\ s_{N-1} & \cdots & s_2 & s_1 & N \end{bmatrix} \cdot \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_N \end{bmatrix} = -\begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ \vdots \\ s_N \end{bmatrix} \qquad (1)$$

όπου το s_k είναι το ίχνος του πίνακα A^k.

Αντιστροφή Τυχαίου Πίνακα: Αλγόριθμος του Csanky(3/5)

Απόδειξη:

Το ίχνος ενός πίνακα ισούται με το άθροισμα των ιδιοτιμών του.

Οι ιδιοτιμές του A^k είναι οι k-οστές δυνάμεις των ιδιοτιμών του A.

Άρα $s_k = \lambda_1^k + \ldots + \lambda_N^k$, όπου $\lambda_1, \ldots, \lambda_N$ είναι οι ιδιοτιμές του A.

Αντιστροφή Τυχαίου Πίνακα: Αλγόριθμος του Csanky(4/5)

Οι ιδιοτιμές του A είναι οι ρίζες του χαρακτηριστικού πολυωνύμου, άρα

$$C_A(x) = \prod_{i=1}^{N} (x - \lambda_i) \qquad (2)$$

Επίσης ισχύει ότι

$$C_A(x) = x^N + c_1 x^{N-1} + c_2 x^{N-2} + \ldots + c_N \qquad (3)$$

Παραγωγίζοντας τις (2) και (3) και εξισώνοντας τα δεύτερα μέλη προκύπτει το ζητούμενο.

Αντιστροφή Τυχαίου Πίνακα: Αλγόριθμος του Csanky(5/5)

- Υπολογίζουμε κάθε A^i σε ένα NxNxN mesh of trees με διαδοχικό τετραγωνισμό σε $O(\log^2 N)$ βήματα.
- Υπολογίζουμε σε $O(\log N)$ βήματα το s_k αθροίζοντας τα διαγώνια στοιχεία του A^k.
- Λύνουμε την εξίσωση (1) ως προς τα c_i αντιστρέφοντας τον κάτω τριγωνικό πίνακα σε $O(\log^2 N)$ βήματα.
- Υπολογίζουμε τον A^{-1}, υπολογίζοντας το άθροισμα $-1/c_N\,(A^{N-1} + c_1 A^{N-2} + \ldots + c_{N-2} A + \ldots + c_{N-1})$ σε επιπλέον $\log N$ βήματα.

Αντιστροφή με την επαναληπτική μέθοδο Newton(1/3)

Υπολογίζουμε την t+1 προσέγγιση του A^{-1} σύμφωνα με τον κανόνα

$$X_{t+1} = 2X_t - X_t A X_t \qquad (4)$$

όπου X_t είναι η t-οστή προσέγγιση του πίνακα A^{-1}.

Αντιστροφή με την επαναληπτική μέθοδο Newton(2/3)

Για να αποδείξουμε ότι η μέθοδος συγκλίνει θα εξετάσουμε τον πίνακα $R_t = I - A X_t$, που υπολογίζει πόσο απέχει ο X_t από τον A^{-1}.

$R_{t+1} = I - AX_{t+1}$

$= I - A (2X_t - X_t A X_t)$

$= I - 2AX_t + (Axt)(AXt)$

$= (I - 2AX_t)^2$

$= R_t^2$

Άρα $R_t = R_0^{2^t}$ κι αυτό σημαίνει ότι το R_t τείνει πολύ γρήγορα στο μηδέν αν το X_0 είναι μια καλή αρχική προσέγγιση του A^{-1}.

Αντιστροφή με την επαναληπτική μέθοδο Newton(3/3)

Χρόνος

Αποδεικνύεται ότι αν το $K = ||A||_2 \, ||A^{-1}||_2$ είναι πολυωνυμικό ως προς το N τότε τα πρώτα N bits κάθε εισόδου του A^{-1} υπολογίζονται σε O(logN) επαναλήψεις της εξίσωσης (4).

Αφού η εξίσωση (4) απαιτεί μόνο πολ/σμους πινάκων και προσθέσεις, ο αλγόριθμος μπορεί να τρέξει σε $O(\log^2 N)$ βήματα σε ένα NxNxN mesh of trees.

Related Problems(1/9)

Θεώρημα

Με πολυωνυμικές (ως προς το N) αλλαγές του μεγέθους των πινάκων και αλλαγές στους σταθερούς παράγοντες του χρόνου τρεξίματος των αλγορίθμων, ο παράλληλος χρόνος που χρειάζονται οι παρακάτω υπολογισμοί για να εκτελεστούν είναι ίδιος:

1. Αντιστροφή Πίνακα
2. Υπολογισμός Ορίζουσας
3. Υπολογισμός Χαρακτηριστικού Πολυωνύμου
4. Αντιστροφή Κάτω Τριγωνικού Πίνακα και
5. N-οστη Δύναμη Πίνακα

Related Problems(2/9)

<u>Απόδειξη:</u>

(1→2) Επειδή $A^{-1} = \dfrac{(-1)^{i+j}\det(A_{ij})}{\det(A)}$,

όπου A_{ij} είναι ο πίνακας χωρίς την i-οστή σειρά και τη j-οστή στήλη, άρα για να υπολογίσουμε τον A^{-1} αρκεί να υπολογίσουμε τις ορίζουσες det(A) και $\det(A_{ij})$ παράλληλα.

Related Problems(3/9)

(2→3) Ο υπολογισμός της ορίζουσας δεν είναι πιο δύσκολος από την εύρεση του χαρακτηριστικού πολυωνύμου, αφού η σταθερά του χαρακτηριστικού πολυωνύμου είναι η ορίζουσα.

Related Problems(4/9)

(3→4) Ο υπολογισμός χαρακτηριστικού πολυωνύμου ενός πίνακα NxN απαιτεί:

- O(logN) χρόνο +
- χρόνο αντιστροφής ενός κάτω τριγωνικού πίνακα +
- χρόνο εύρεσης δύναμης.

Related Problems(5/9)

- Τα δύο τελευταία απαιτούν χρόνο Ω(logN) άρα το πρώτο είναι μη σημαντικό.
- Επίσης ο υπολογισμός δύναμης ενός NxN πίνακα δεν είναι πιο δύσκολος από τον υπολογισμό αντιστρόφου ενός N^2 x N^2 πίνακα.
- Άρα ο υπολογισμός χαρακτηριστικού πολυωνύμου είναι το πολύ σταθερό αριθμό πιο αργός από τον υπολογισμό αντιστρόφου ενός κάτω τριγωνικού πίνακα.

Related Problems(6/9)

$$\begin{pmatrix} I & & & & & 0 \\ A & I & & & & \\ & A & I & & & \\ & & A & I & & \\ & & & \ddots & \ddots & \\ 0 & & & & A & I \end{pmatrix}^{-1} = \begin{pmatrix} I & & & & & 0 \\ -A & I & & & & \\ A^2 & -A & I & & & \\ -A^3 & A^2 & & I & & \\ \vdots & \ddots & \ddots & \ddots & \ddots & \\ (-A)^{N-1} & & -A^3 & A^2 & -A & I \end{pmatrix}$$

Υπολογισμός Δύναμης Πίνακα μέσω Αντιστροφής ενός Κάτω Τριγωνικού Πίνακα

Related Problems(7/9)

(4→5) Ένας κάτω τριγωνικός πίνακας L L εκφράζεται ως:

$L = (I - L_0)\,D$, όπου ο D είναι διαγώνιος και ο L_0 κάτω τριγωνικός με 0 στη διαγώνιο.

Related Problems(8/9)

Οι L_0, D, D^{-1} υπολογίζονται σε O(logN).

Ο $L^{-1} = D^{-1} (I - L_0)^{-1}$ υπολογίζεται σε O(logN) αν ξέρω τον $(I - L_0)^{-1}$.

Όμως $L_0^N = 0$ άρα

$$(I - L_0)^{-1} = I + L_0 + L_0^2 + \ldots + L_0^{N-1}$$

Άρα η αντιστροφή του L δεν είναι πιο δύσκολη από τον υπολογισμό της j-οστής δύναμης του L_0.

Related Problems(9/9)

(5→1) Ο υπολογισμός δύναμης ενός NxN πίνακα δεν είναι πιο δύσκολος από τον υπολογισμό αντιστρόφου ενός N^2 x N^2 πίνακα.