

Υπολογισμός Παραγοντικού (1/3)

Μαθηματικός τύπος:

$$n! = 1 \cdot \dots \cdot (n - 1) \cdot n = (n - 1)! \cdot n \quad \text{for } n \geq 1$$

Βασική λειτουργία: πολ/μος

Το πλήθος των πολ/μων έχει ως εξής:

$$M(n) = \underbrace{M(n - 1)}_{\text{to compute } F(n-1)} + \underbrace{1}_{\text{to multiply } F(n-1) \text{ by } n} \quad \text{for } n > 0$$

Κάθε φορά το πλήθος εξαρτάται από το πλήθος στο $n-1$

ALGORITHM $F(n)$

//Computes $n!$ recursively

//Input: A nonnegative integer n

//Output: The value of $n!$

if $n = 0$ **return** 1

else return $F(n - 1) * n$

Υπολογισμός Παραγοντικού (2/3)

Η αρχική συνθήκη βρίσκεται στο if

Όταν $n=0$ ο αλγόριθμος δεν κάνει πολ/μους

$M(0) = 0.$

the calls stop when $n = 0$  no multiplications when $n = 0$

Εφαρμόζουμε τη μέθοδο των ***backward substitutions***

$$\begin{aligned} M(n) &= M(n-1) + 1 && \text{substitute } M(n-1) = M(n-2) + 1 \\ &= [M(n-2) + 1] + 1 = M(n-2) + 2 && \text{substitute } M(n-2) = M(n-3) + 1 \\ &= [M(n-3) + 1] + 2 = M(n-3) + 3. \end{aligned}$$

Υπολογισμός Παραγοντικού (3/3)

Γενικός τύπος

$$M(n) = M(n - i) + i$$

Αφού η αρχική συνθήκη είναι $n=0$, αντικαθιστούμε με $i=n$ και έχουμε:

$$M(n) = M(n - 1) + 1 = \dots = M(n - i) + i = \dots = M(n - n) + n = n$$

Complexity $\rightarrow O(n)$

Μεθοδολογία

Η γενική μεθοδολογία για την ανάλυση αναδρομικών αλγορίθμων έχει ως εξής:

- **Αποφασίζουμε** την παράμετρο του μεγέθους της εισόδου
- **Αναγνωρίζουμε** τη βασική λειτουργία του αλγορίθμου (π.χ. σύγκριση, ανάθεση τιμής)
- **Ελέγχουμε** αν το πλήθος εκτέλεσης της βασικής λειτουργίας εξαρτάται μόνο από το μέγεθος εισόδου. Αν εξαρτάται από κάποια επιπλέον παράμετρο τότε μελετούμε ξεχωριστά την καλύτερη, τη χειρότερη και τη μέση περίπτωση.
- **Δημιουργούμε** μια αναδρομική σχέση με μια κατάλληλη αρχική συνθήκη που απεικονίζει το πλήθος των εκτελέσεων της βασικής λειτουργίας.
- **Επιλύουμε** την αναδρομική σχέση.

Backward Substitution

Δύο βασικά βήματα:

- Περιγράφουμε τη μορφή της λύσης
- Χρησιμοποιούμε επαγωγή για να βρούμε σταθερές που αποδεικνύουν ότι η λύση ισχύει

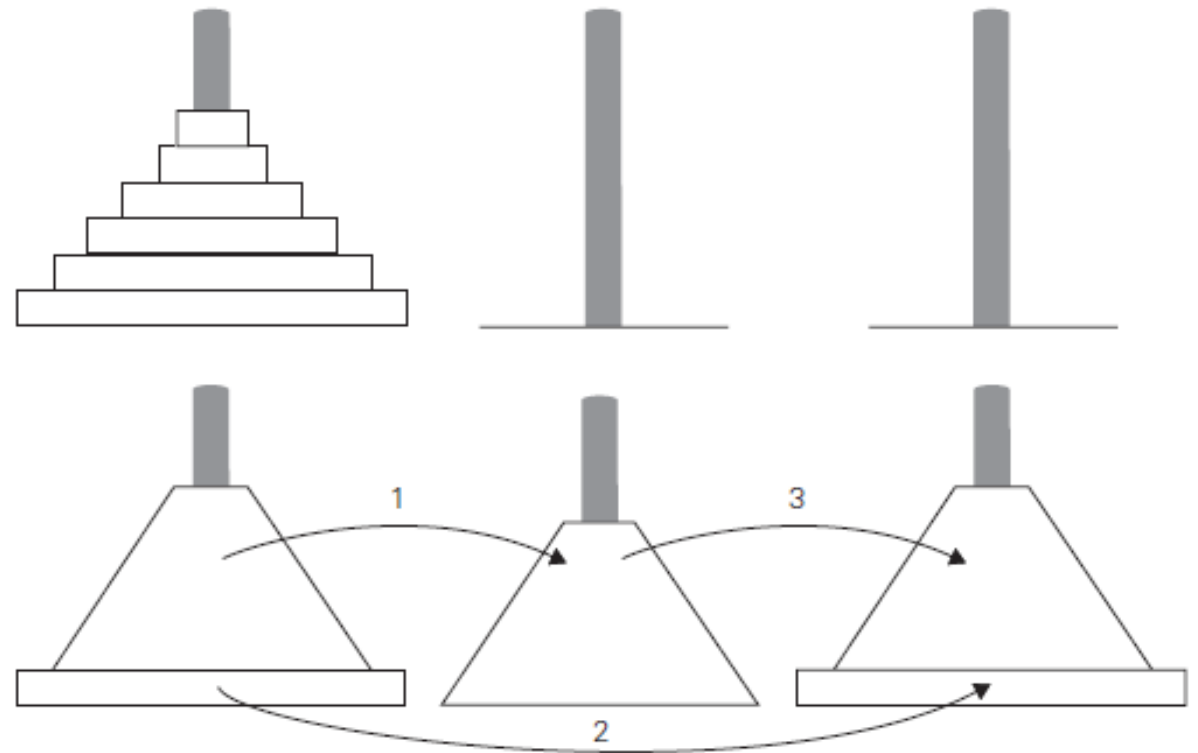
Η λύση που προτείνουμε είναι μια συνάρτηση όπου εφαρμόζουμε την επαγωγή

Οι Πύργοι του Ανόϊ (1/5)

Έχουμε n δίσκους διαφορετικού μεγέθους

Οι δίσκοι πρέπει να μεταφερθούν από τον 1^ο στύλο στον 3^ο αλλά:

- Ένας δίσκος κάθε φορά μπορεί να μεταφερθεί
- Μεγαλύτερος δίσκος δεν πρέπει να τοποθετηθεί πάνω από μικρότερο



Οι Πύργοι του Ανόϊ (2/5)

Για την μετακίνηση n δίσκων πρώτα μεταφέρουμε αναδρομικά $n-1$ δίσκους από τον 1^ο στύλο στον 2^ο (με τον 3^ο στύλο βοηθητικό) και έπειτα μεταφέρουμε το μεγαλύτερο δίσκο στον 3^ο στύλο

Στη συνέχεια μεταφέρουμε αναδρομικά $n-1$ δίσκους από τον 2^ο δίσκο στον 3^ο (με τον 1^ο δίσκο βοηθητικό)

Όταν $n=1$, απλά μετακινούμε το δίσκο από τον στύλο 'πηγή' στο στύλο 'προορισμό'

Το μέγεθος εισόδου είναι n

Βασική λειτουργία: μετακίνηση / μεταφορά δίσκου

Οι Πύργοι του Ανόϊ (3/5)

Η αναδρομική σχέση είναι

$$M(n) = M(n - 1) + 1 + M(n - 1) \quad \text{for } n > 1.$$

Έχουμε:

$$M(n) = 2M(n - 1) + 1 \quad \text{for } n > 1$$

$$M(1) = 1$$

Εφαρμόζοντας backward substitution παίρνουμε:

$$M(n) = 2M(n - 1) + 1 \qquad \text{sub. } M(n - 1) = 2M(n - 2) + 1$$

$$= 2[2M(n - 2) + 1] + 1 = 2^2M(n - 2) + 2 + 1 \quad \text{sub. } M(n - 2) = 2M(n - 3) + 1$$

$$= 2^2[2M(n - 3) + 1] + 2 + 1 = 2^3M(n - 3) + 2^2 + 2 + 1$$

Οι Πύργοι του Ανόϊ (4/5)

Η σχέση που φαίνεται μέσα από την αναδρομική σχέση είναι:

$$M(n) = 2^i M(n - i) + 2^{i-1} + 2^{i-2} + \dots + 2 + 1 = 2^i M(n - i) + 2^i - 1$$

$$n - i = 1 \rightarrow i = n - 1$$

Η αρχική συνθήκη ισχύει για $n=1$ και την έχουμε για $i=n-1$ έχουμε:

$$\begin{aligned} M(n) &= 2^{n-1} M(n - (n - 1)) + 2^{n-1} - 1 \\ &= 2^{n-1} M(1) + 2^{n-1} - 1 = 2^{n-1} + 2^{n-1} - 1 = 2^n - 1 \end{aligned}$$

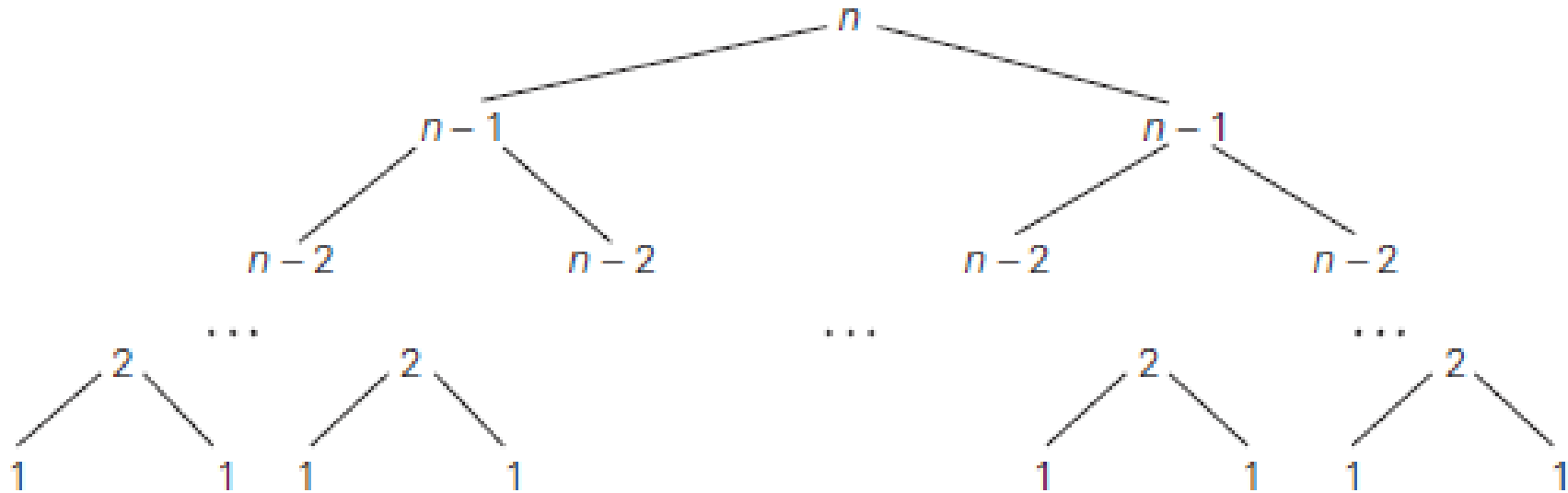
Ένα δένδρο που δείχνει τα βήματα της αναδρομής μπορεί να βοηθήσει στην επίλυση

Complexity $O(2^n)$

Οι Πύργοι του Ανόϊ (5/5)

Πλήθος κόμβων στο δένδρο: $C(n) = \sum_{l=0}^{n-1} 2^l$

Όπου l είναι τα επίπεδα του δένδρου



Εύρεση Ψηφίων Αναδρομικά (1/3)

Στόχος: εύρεση αναδρομικής σχέσης

Πλήθος προσθέσεων: $A(\lfloor n/2 \rfloor) + 1$

Αναδρομική σχέση:

$$A(n) = A(\lfloor n/2 \rfloor) + 1 \quad \text{for } n > 1.$$

Πρόβλημα:

- Η μέθοδος backward substitution είναι δύσκολο να εφαρμοστεί για άλλους αριθμούς εκτός από τα πολλαπλάσια του 2

ALGORITHM *BinRec*(n)

//Input: A positive decimal integer n

//Output: The number of binary digits in n 's

if $n = 1$ **return** 1

else return *BinRec*($\lfloor n/2 \rfloor$) + 1

Εύρεση Ψηφίων Αναδρομικά (2/3)

Επιλύουμε το πρόβλημα για τα
πολλαπλάσια του 2 και εφαρμόζουμε
τον **κανόνα εξομάλυνσης (smoothness
rule)**

Έχουμε:

$$A(2^k) = A(2^{k-1}) + 1 \quad \text{for } k > 0$$

$$A(2^0) = 0.$$

και με backward substitution
παίρνουμε:

$$\begin{aligned} A(2^k) &= A(2^{k-1}) + 1 && \text{substitute } A(2^{k-1}) = A(2^{k-2}) + 1 \\ &= [A(2^{k-2}) + 1] + 1 = A(2^{k-2}) + 2 && \text{substitute } A(2^{k-2}) = A(2^{k-3}) + 1 \\ &= [A(2^{k-3}) + 1] + 2 = A(2^{k-3}) + 3 && \dots \\ &\dots && \\ &= A(2^{k-i}) + i && \\ &\dots && \\ &= A(2^{k-k}) + k \end{aligned}$$

Εύρεση Ψηφίων Αναδρομικά (3/3)

Τελικά καταλήγουμε στο:

$$A(2^k) = A(1) + k = k$$

και επιστρέφοντας στη συλλογιστική μας για οποιοδήποτε n έχουμε:

$$A(n) = \log_2 n \in \Theta(\log n)$$

αφού $2^k = n$

Ουσιαστικά

$$A(n) = \lfloor \log_2 n \rfloor$$

The Master Theorem (1/7)

Μας προσφέρει μια μεθοδολογία για να επιλύουμε αναδρομικές εξισώσεις της μορφής:

$$T(n) = aT(n/b) + f(n)$$

με $a \geq 1$ και $b > 1$ να είναι σταθερές και $f(n)$ είναι μια ασυμπτωτικά θετική συνάρτηση

Η παραπάνω συνάρτηση χωρίζει το πρόβλημα μεγέθους n σε a υπο-προβλήματα μεγέθους n/b

Τα a υπο-προβλήματα λύνονται πάλι αναδρομικά σε χρόνο n/b

Η $f(n)$ απεικονίζει το κόστος της διαίρεσης του προβλήματος και της συγχώνευσης των λύσεων

The Master Theorem (2/7)

Το n/b μπορεί να μην είναι ακέραιος

Παίρνουμε το floor ή το ceiling του n/b χωρίς να επηρεάζεται η ασυμπτωτική συμπεριφορά

Θα πρέπει να θυμόμαστε τρεις περιπτώσεις

Υπάρχουν κενά στη μέθοδο αφού η $f(n)$ πρέπει να είναι πολυωνυμικά μικρότερη / μεγαλύτερη από το $n^{\log_b a}$

Αν είναι τότε δεν μπορούμε να χρησιμοποιήσουμε το master theorem για την επίλυση αναδρομικών σχέσεων

The Master Theorem (3/7)

Let $a \geq 1$ and $b > 1$ be constants, let $f(n)$ be a function, and let $T(n)$ be defined on the nonnegative integers by the recurrence

$$T(n) = aT(n/b) + f(n) ,$$

where we interpret n/b to mean either $\lfloor n/b \rfloor$ or $\lceil n/b \rceil$. Then $T(n)$ has the following asymptotic bounds:

1. If $f(n) = O(n^{\log_b a - \epsilon})$ for some constant $\epsilon > 0$, then $T(n) = \Theta(n^{\log_b a})$.
2. If $f(n) = \Theta(n^{\log_b a})$, then $T(n) = \Theta(n^{\log_b a} \lg n)$.
3. If $f(n) = \Omega(n^{\log_b a + \epsilon})$ for some constant $\epsilon > 0$, and if $af(n/b) \leq cf(n)$ for some constant $c < 1$ and all sufficiently large n , then $T(n) = \Theta(f(n))$.

The Master Theorem (4/7)

Ας δούμε ένα παράδειγμα:

Για την $T(n) = 9T(n/3) + n$

έχουμε $a = 9, b = 3, f(n) = n,$

και έτσι $n^{\log_b a} = n^{\log_3 9} = \Theta(n^2)$

Αφού $f(n) = O(n^{\log_3 9 - \epsilon})$

με $\epsilon = 1$

Εφαρμόζουμε την 1^η περίπτωση και καταλήγουμε στο

$$T(n) = \Theta(n^2)$$

The Master Theorem (5/7)

Παράδειγμα

Για την $T(n) = T(2n/3) + 1$

έχουμε $a = 1, b = 3/2, f(n) = 1$

και έτσι $n^{\log_b a} = n^{\log_{3/2} 1} = n^0 = 1$

Αφού $f(n) = \Theta(n^{\log_b a}) = \Theta(1)$

Εφαρμόζουμε τη 2^η περίπτωση και καταλήγουμε στο

$$T(n) = \Theta(\lg n)$$

The Master Theorem (6/7)

Παράδειγμα

Για την $T(n) = 3T(n/4) + n \lg n$

έχουμε $a = 3, b = 4, f(n) = n \lg n$

και έτσι $n^{\log_b a} = n^{\log_4 3} = O(n^{0.793})$

Αφού $f(n) = \Omega(n^{\log_4 3 + \epsilon})$

με $\epsilon \approx 0.2$

για μεγάλο n έχουμε $af(n/b) = 3(n/4) \lg(n/4) \leq (3/4)n \lg n = cf(n)$ for $c = 3/4$

Εφαρμόζουμε τη 3^η περίπτωση και καταλήγουμε στο $T(n) = \Theta(n \lg n)$

The Master Theorem (7/7)

Παράδειγμα

Για την $T(n) = 2T(n/2) + n \lg n$

η master μέθοδος δεν μπορεί να εφαρμοστεί αφού για

$$a = 2, b = 2, f(n) = n \lg n, \quad \text{και} \quad n^{\log_b a} = n$$

Δεν μπορεί να εφαρμοστεί η 3^η περίπτωση αφού η $f(n)$ είναι ασυμπτωτικά μεγαλύτερη από την $n^{\log_b a} = n$ αλλά όχι πολυωνυμικά μεγαλύτερη. Ο λόγος $f(n)/n^{\log_b a} = (n \lg n)/n = \lg n$

Είναι ασυμπτωτικά μικρότερος από το n^ϵ

για οποιοδήποτε ϵ

Ασκήσεις (1/2)

1. $T(n) = 3T(n/2) + n^2$

2. $T(n) = 4T(n/2) + n^2$

3. $T(n) = T(n/2) + 2^n$

4. $T(n) = 2^n T(n/2) + n^n$

5. $T(n) = 16T(n/4) + n$

6. $T(n) = 2T(n/2) + n \log n$

1. $T(n) = 3T(n/2) + n^2 \implies T(n) = \Theta(n^2)$ (Case 3)

2. $T(n) = 4T(n/2) + n^2 \implies T(n) = \Theta(n^2 \log n)$ (Case 2)

3. $T(n) = T(n/2) + 2^n \implies \Theta(2^n)$ (Case 3)

4. $T(n) = 2^n T(n/2) + n^n \implies$ Does not apply (a is not constant)

5. $T(n) = 16T(n/4) + n \implies T(n) = \Theta(n^2)$ (Case 1)

6. $T(n) = 2T(n/2) + n \log n \implies T(n) = n \log^2 n$ (Case 2)

Ασκήσεις (2/2)

7. $T(n) = 2T(n/2) + n/\log n$
8. $T(n) = 2T(n/4) + n^{0.51}$
9. $T(n) = 0.5T(n/2) + 1/n$
10. $T(n) = 16T(n/4) + n!$
11. $T(n) = \sqrt{2}T(n/2) + \log n$
12. $T(n) = 3T(n/2) + n$
7. $T(n) = 2T(n/2) + n/\log n \implies$ Does not apply (non-polynomial difference between $f(n)$ and $n^{\log_b a}$)
8. $T(n) = 2T(n/4) + n^{0.51} \implies T(n) = \Theta(n^{0.51})$ (Case 3)
9. $T(n) = 0.5T(n/2) + 1/n \implies$ Does not apply ($a < 1$)
10. $T(n) = 16T(n/4) + n! \implies T(n) = \Theta(n!)$ (Case 3)
11. $T(n) = \sqrt{2}T(n/2) + \log n \implies T(n) = \Theta(\sqrt{n})$ (Case 1)
12. $T(n) = 3T(n/2) + n \implies T(n) = \Theta(n^{\lg 3})$ (Case 1)

The Master Theorem

Εναλλακτικά:

If $f(n) \in \Theta(n^d)$ $d \geq 0$

$$T(n) \in \begin{cases} \Theta(n^d) & \text{if } a < b^d \\ \Theta(n^d \log n) & \text{if } a = b^d \\ \Theta(n^{\log_b a}) & \text{if } a > b^d \end{cases}$$

Ασκήσεις (1/12)

Να λύσετε την επόμενη αναδρομική σχέση:

$$T(n)=3T(n/2)+n^2$$

Ασκήσεις (2/12)

Να λύσετε την επόμενη αναδρομική σχέση: $T(n)=3T(n/2)+n^2$

Λύση

$$n^{\log_2 3} = n^{1.58}$$

$$\text{Άρα } f(n) = \Omega(n^{1.58})$$

Κοιτάζουμε την ανίσωση: $af(n/b) \leq cf(n)$

Έχουμε: $3/4 n^2 \leq c n^2$, ισχύει άρα 3^η περίπτωση του Master, συνεπώς $T(n)=\Theta(n^2)$

Ασκήσεις (3/12)

Να λύσετε την επόμενη αναδρομική σχέση:

$$T(n)=4T(n/2)+n^2$$

Ασκήσεις (4/12)

Να λύσετε την επόμενη αναδρομική σχέση: $T(n)=4T(n/2)+n^2$

Λύση

$$n^{\log_2 4} = n^2$$

$$\text{Άρα } f(n) = \Theta(n^2)$$

2^η περίπτωση του Master, συνεπώς $T(n)=\Theta(n^2 \log n)$

Ασκήσεις (5/12)

Να λύσετε την επόμενη αναδρομική σχέση:

$$T(n)=T(n/2)+2^n$$

Ασκήσεις (6/12)

Να λύσετε την επόμενη αναδρομική σχέση: $T(n)=T(n/2)+2^n$

Λύση

$$n^{\log_2 1} = n^0 = 1$$

$$\text{Άρα } f(n) = \Omega(n^0)$$

Κοιτάζουμε την ανίσωση: $af(n/b) \leq cf(n)$

Έχουμε: $2^{n/2} \leq c 2^n$, ισχύει άρα 3^η περίπτωση του Master, συνεπώς
 $T(n) = \Theta(2^n)$

Ασκήσεις (7/12)

Να λύσετε την επόμενη αναδρομική σχέση:

$$T(n)=16T(n/4)+n$$

Ασκήσεις (8/12)

Να λύσετε την επόμενη αναδρομική σχέση: $T(n)=16T(n/4)+n$

Λύση

$$n^{\log_4 16} = n^2$$

$$\text{Άρα } f(n) = O(n^2)$$

1^η περίπτωση του Master, συνεπώς $T(n)=\Theta(n^2)$

Ασκήσεις (9/12)

Να λύσετε την επόμενη αναδρομική σχέση:

$$T(n) = \sqrt{2}T(n/2) + \log n$$

Ασκήσεις (10/12)

Να λύσετε την επόμενη αναδρομική σχέση: $T(n) = \sqrt{2}T(n/2) + \log n$

Λύση

$$n^{\log_2 \sqrt{2}} = n^{1/2}$$

$$\text{Άρα } f(n) = O(n^{1/2})$$

1^η περίπτωση του Master, συνεπώς $T(n) = \Theta(n^{1/2})$

Ασκήσεις (11/12)

Να λύσετε την επόμενη αναδρομική σχέση:

$$T(n)=3T(n/2)+n$$

Ασκήσεις (12/12)

Να λύσετε την επόμενη αναδρομική σχέση: $T(n)=3T(n/2)+n$

Λύση

$$n^{\log_2 3} = n^{1.58}$$

$$\text{Άρα } f(n) = O(n^{1.58})$$

1^η περίπτωση του Master, συνεπώς $T(n)=\Theta(n^{1.58})$

Διαίρει και βασίλευε

(συνέχεια)

Πολ/μός Μεγάλων Αριθμών (1/9)

Κάποιες εφαρμογές (π.χ. κρυπτογραφία) απαιτούν το χειρισμό ακεραίων ακόμη και με 100 ψηφία

Αυτού του είδους οι αριθμοί δεν μπορούν να χωρέσουν σε μια λέξη ενός υπολογιστή

Απαιτούν ειδική μεταχείριση

Αν προσπαθήσουμε να πολλαπλασιάσουμε αριθμούς n ψηφίων χρειαζόμαστε n^2 πολλαπλασιασμούς (κάθε ένα ψηφίο του πρώτου αριθμού πολλαπλασιάζεται με όλα τα ψηφία του δεύτερου)

Πολ/μός Μεγάλων Αριθμών (2/9)

Βασική ιδέα αλγορίθμου

- Παράδειγμα: πολλαπλασιασμός του 23 και 14
- Οι αριθμοί γράφονται ως εξής

$$23 = 2 \cdot 10^1 + 3 \cdot 10^0 \quad \text{and} \quad 14 = 1 \cdot 10^1 + 4 \cdot 10^0$$

- Οπότε

$$\begin{aligned} 23 * 14 &= (2 \cdot 10^1 + 3 \cdot 10^0) * (1 \cdot 10^1 + 4 \cdot 10^0) \\ &= (2 * 1)10^2 + (2 * 4 + 3 * 1)10^1 + (3 * 4)10^0 \end{aligned}$$

Πολ/μός Μεγάλων Αριθμών (3/9)

Ο τελευταίος τύπος εξάγει το σωστό αποτέλεσμα αλλά απαιτεί 4 πολλαπλασιασμούς

Μπορεί να γίνει μια βελτίωση ως εξής:

$$2 * 4 + 3 * 1 = (2 + 3) * (1 + 4) - 2 * 1 - 3 * 4$$

Για δύο αριθμούς $a = a_1a_0$ και $b = b_1b_0$

Το γινόμενο τους μπορεί να δοθεί ως εξής:

$$c = a * b = c_210^2 + c_110^1 + c_0; \quad \text{με} \quad \begin{aligned} c_2 &= a_1 * b_1 \\ c_0 &= a_0 * b_0 \\ c_1 &= (a_1 + a_0) * (b_1 + b_0) - (c_2 + c_0) \end{aligned}$$

Πολ/μός Μεγάλων Αριθμών (4/9)

Εφαρμόζουμε τον τύπο για τον πολλαπλασιασμό αριθμών με n ψηφία

Διαιρούμε τους αριθμούς στη μέση

a_1 είναι τα πρώτα ψηφία του a και a_0 είναι τα υπόλοιπα

b_1 είναι τα πρώτα ψηφία του b και b_0 είναι τα υπόλοιπα

Συνεπώς:

$$a = a_1 10^{n/2} + a_0$$

$$b = b_1 10^{n/2} + b_0$$

Πολ/μός Μεγάλων Αριθμών (5/9)

Οπότε

$$\begin{aligned}c &= a * b = (a_1 10^{n/2} + a_0) * (b_1 10^{n/2} + b_0) \\&= (a_1 * b_1) 10^n + (a_1 * b_0 + a_0 * b_1) 10^{n/2} + (a_0 * b_0) \\&= c_2 10^n + c_1 10^{n/2} + c_0\end{aligned}$$

με

$$c_2 = a_1 * b_1$$

$$c_0 = a_0 * b_0$$

$$c_1 = (a_1 + a_0) * (b_1 + b_0) - (c_2 + c_0)$$

Πολ/μός Μεγάλων Αριθμών (6/9)

Αν το $n/2$ είναι άρτιος μπορούμε να συνεχίσουμε με την ίδια μέθοδο για τα c_2, c_1, c_0

Συνεπώς αν το n είναι δύναμη του 2 μπορούμε εύκολα να καταλήξουμε στην αναδρομική σχέση

Η αναδρομική σχέση καταλήγει όταν $n = 1$

Μπορεί να καταλήξει όταν το n είναι πολύ μικρό

Απαιτούνται 3 πολλαπλασιασμοί αριθμών με $n/2$ ψηφία

Πολ/μός Μεγάλων Αριθμών (7/9)

Υπολογισμοί

$$M(n) = 3M(n/2) \quad \text{for } n > 1, \quad M(1) = 1.$$

$$\begin{aligned} M(2^k) &= 3M(2^{k-1}) = 3[3M(2^{k-2})] = 3^2 M(2^{k-2}) \\ &= \dots = 3^i M(2^{k-i}) = \dots = 3^k M(2^{k-k}) = 3^k. \end{aligned}$$

$$k = \log_2 n$$

$$a^{\log_b c} = c^{\log_b a}$$

$$M(n) = 3^{\log_2 n} = n^{\log_2 3} \approx n^{1.585}$$

Πολ/μός Μεγάλων Αριθμών (8/9)

Προσθέσεις και αφαιρέσεις

- Απαιτούνται πέντε προσθέσεις και μια αφαίρεση
- Παίρνουμε την ακόλουθη αναδρομική σχέση

$$A(n) = 3A(n/2) + cn \quad \text{for } n > 1, \quad A(1) = 1.$$

- Και καταλήγουμε ότι

$$A(n) \in \Theta(n^{\log_2 3}).$$

Πολ/μός Μεγάλων Αριθμών (9/9)

Η διαίρει και βασίλευε είναι γρηγορότερη ακόμα και για αριθμούς 8 ψηφίων σε σχέση με τις συμβατικές μεθόδους

Για αριθμούς με 300 ψηφία είναι δύο φορές πιο γρήγορη

Στις γλώσσες Java, C++ υπάρχουν ειδικές κλάσεις για χειρισμό μεγάλων αριθμών

Ερωτήματα:

- Τι γίνεται αν οι δύο αριθμοί δεν είναι του ιδίου μεγέθους?
- Τι γίνεται αν το πλήθος των ψηφίων n δεν είναι πολλαπλάσιο του 2?

Matrix Multiplication (1/6)

Αν $A = (a_{ij})$ και $B = (b_{ij})$ είναι $n \times n$ πίνακες τότε το γινόμενο τους είναι

$$c_{ij} = \sum_{k=1}^n a_{ik} \cdot b_{kj}$$

Αλγόριθμος

SQUARE-MATRIX-MULTIPLY(A, B)

```
1   $n = A.rows$ 
2  let  $C$  be a new  $n \times n$  matrix
3  for  $i = 1$  to  $n$ 
4      for  $j = 1$  to  $n$ 
5           $c_{ij} = 0$ 
6          for  $k = 1$  to  $n$ 
7               $c_{ij} = c_{ij} + a_{ik} \cdot b_{kj}$ 
8  return  $C$ 
```

Matrix Multiplication (2/6)

Προσέγγιση με διαίρει και βασίλευε

- Υποθέτουμε ότι σπάμε τους πίνακες σε 4 $n/2 \times n/2$ πίνακες

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}, \quad B = \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix}, \quad C = \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix}$$

$$\begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \cdot \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix}$$

$$C_{11} = A_{11} \cdot B_{11} + A_{12} \cdot B_{21}$$

$$C_{12} = A_{11} \cdot B_{12} + A_{12} \cdot B_{22}$$

$$C_{21} = A_{21} \cdot B_{11} + A_{22} \cdot B_{21}$$

$$C_{22} = A_{21} \cdot B_{12} + A_{22} \cdot B_{22}$$

Matrix Multiplication (3/6)

Για κάθε μια από τις προηγούμενες 4 εξισώσεις απαιτούνται 2 πολ/μοι και η πρόσθεση των $n/2 \times n/2$ γινομένων

Νέος αλγόριθμος

SQUARE-MATRIX-MULTIPLY-RECURSIVE(A, B)

```
1   $n = A.rows$ 
2  let  $C$  be a new  $n \times n$  matrix
3  if  $n == 1$ 
4       $c_{11} = a_{11} \cdot b_{11}$ 
5  else partition  $A, B$ , and  $C$  as in equations (4.9)
6       $C_{11} = \text{SQUARE-MATRIX-MULTIPLY-RECURSIVE}(A_{11}, B_{11})$ 
           +  $\text{SQUARE-MATRIX-MULTIPLY-RECURSIVE}(A_{12}, B_{21})$ 
7       $C_{12} = \text{SQUARE-MATRIX-MULTIPLY-RECURSIVE}(A_{11}, B_{12})$ 
           +  $\text{SQUARE-MATRIX-MULTIPLY-RECURSIVE}(A_{12}, B_{22})$ 
8       $C_{21} = \text{SQUARE-MATRIX-MULTIPLY-RECURSIVE}(A_{21}, B_{11})$ 
           +  $\text{SQUARE-MATRIX-MULTIPLY-RECURSIVE}(A_{22}, B_{21})$ 
9       $C_{22} = \text{SQUARE-MATRIX-MULTIPLY-RECURSIVE}(A_{21}, B_{12})$ 
           +  $\text{SQUARE-MATRIX-MULTIPLY-RECURSIVE}(A_{22}, B_{22})$ 
10 return  $C$ 
```


Matrix Multiplication (4/6)

Χωρίζουμε τους πίνακες χωρίς να αντιγράφουμε στοιχεία

Χρησιμοποιούμε index calculations

Αναγνωρίζουμε ένα υπο-πίνακα από ένα εύρος γραμμών και στηλών στον αρχικό πίνακα

Έστω $T(n)$ είναι ο χρόνος για τον πολλαπλασιασμό των πινάκων

Όταν $n=1$ εκτελούμε ένα πολλαπλασιασμό, οπότε $T(1)=\Theta(1)$

Η αναδρομική σχέση ισχύει για $n>1$

Matrix Multiplication (5/6)

Το κόστος της γραμμής 5 του αλγορίθμου είναι $\Theta(1)$ SQUARE-MATRIX-MULTIPLY-RECURSIVE(A, B)

Στις γραμμές 6-9 εκτελούμε 8 κλήσεις της αναδρομής

Κάθε κλήση συνεισφέρει με $T(n/2)$ στο συνολικό χρόνο

Άρα το συνολικό κόστος των κλήσεων είναι $8T(n/2)$

Στις γραμμές 6-9 εκτελούμε 4 προσθέσεις πινάκων

Κάθε υπο-πίνακας περιέχει $n^2/4$ στοιχεία

Οι 4 προσθέσεις έχουν πολυπλοκότητα $\Theta(n^2)$

```
1   $n = A.rows$ 
2  let  $C$  be a new  $n \times n$  matrix
3  if  $n == 1$ 
4       $c_{11} = a_{11} \cdot b_{11}$ 
5  else partition  $A, B$ , and  $C$  as in equations (4.9)
6       $C_{11} = \text{SQUARE-MATRIX-MULTIPLY-RECURSIVE}(A_{11}, B_{11})$ 
           +  $\text{SQUARE-MATRIX-MULTIPLY-RECURSIVE}(A_{12}, B_{21})$ 
7       $C_{12} = \text{SQUARE-MATRIX-MULTIPLY-RECURSIVE}(A_{11}, B_{12})$ 
           +  $\text{SQUARE-MATRIX-MULTIPLY-RECURSIVE}(A_{12}, B_{22})$ 
8       $C_{21} = \text{SQUARE-MATRIX-MULTIPLY-RECURSIVE}(A_{21}, B_{11})$ 
           +  $\text{SQUARE-MATRIX-MULTIPLY-RECURSIVE}(A_{22}, B_{21})$ 
9       $C_{22} = \text{SQUARE-MATRIX-MULTIPLY-RECURSIVE}(A_{21}, B_{12})$ 
           +  $\text{SQUARE-MATRIX-MULTIPLY-RECURSIVE}(A_{22}, B_{22})$ 
10 return  $C$ 
```

Matrix Multiplication (6/6)

Ο συνολικός χρόνος για τις αναδρομικές κλήσεις και τους πολλαπλασιασμούς – προσθέσεις είναι:

$$\begin{aligned}T(n) &= \Theta(1) + 8T(n/2) + \Theta(n^2) \\ &= 8T(n/2) + \Theta(n^2)\end{aligned}$$

Αν στη διάσπαση των πινάκων επιλέξουμε την αντιγραφή (κόστος $\Theta(n^2)$) η αναδρομική σχέση δεν θα αλλάξει

Συνεπώς

$$T(n) = \begin{cases} \Theta(1) & \text{if } n = 1 \\ 8T(n/2) + \Theta(n^2) & \text{if } n > 1 \end{cases}$$

Πολυπλοκότητα: $O(n^3)$ από την 1^η περίπτωση του Θ.Κ.

Strassen's Method(1/7)

1. Divide the input matrices A and B and output matrix C into $n/2 \times n/2$ submatrices. This step takes $\Theta(1)$ time by index calculation, just as in SQUARE-MATRIX-MULTIPLY-RECURSIVE.
2. Create 10 matrices $S_1, S_2, \dots, S_{10}$, each of which is $n/2 \times n/2$ and is the sum or difference of two matrices created in step 1. We can create all 10 matrices in $\Theta(n^2)$ time.
3. Using the submatrices created in step 1 and the 10 matrices created in step 2, recursively compute seven matrix products $P_1, P_2, \dots, P_7$. Each matrix P_i is $n/2 \times n/2$.
4. Compute the desired submatrices $C_{11}, C_{12}, C_{21}, C_{22}$ of the result matrix C by adding and subtracting various combinations of the P_i matrices. We can compute all four submatrices in $\Theta(n^2)$ time.

Strassen's Method (2/7)

Αναδρομική σχέση

$$T(n) = \begin{cases} \Theta(1) & \text{if } n = 1 \\ 7T(n/2) + \Theta(n^2) & \text{if } n > 1 \end{cases}$$

Με τη χρήση του master θεωρήματος παίρνουμε ότι $T(n) = \Theta(n^{\lg 7})$.

Strassen's Method (3/7)

Δημιουργία 10 υπο-πινάκων στο 2^ο βήμα

$$S_1 = B_{12} - B_{22}$$

$$S_2 = A_{11} + A_{12}$$

$$S_3 = A_{21} + A_{22}$$

$$S_4 = B_{21} - B_{11}$$

$$S_5 = A_{11} + A_{22}$$

$$S_6 = B_{11} + B_{22}$$

$$S_7 = A_{12} - A_{22}$$

$$S_8 = B_{21} + B_{22}$$

$$S_9 = A_{11} - A_{21}$$

$$S_{10} = B_{11} + B_{12}$$

Κόστος $\Theta(n^2)$

Strassen's Method (4/7)

Πολλαπλασιασμός στο 3^ο βήμα

$$P_1 = A_{11} \cdot S_1 = A_{11} \cdot B_{12} - A_{11} \cdot B_{22}$$

$$P_2 = S_2 \cdot B_{22} = A_{11} \cdot B_{22} + A_{12} \cdot B_{22}$$

$$P_3 = S_3 \cdot B_{11} = A_{21} \cdot B_{11} + A_{22} \cdot B_{11}$$

$$P_4 = A_{22} \cdot S_4 = A_{22} \cdot B_{21} - A_{22} \cdot B_{11}$$

$$P_5 = S_5 \cdot S_6 = A_{11} \cdot B_{11} + A_{11} \cdot B_{22} + A_{22} \cdot B_{11} + A_{22} \cdot B_{22}$$

$$P_6 = S_7 \cdot S_8 = A_{12} \cdot B_{21} + A_{12} \cdot B_{22} - A_{22} \cdot B_{21} - A_{22} \cdot B_{22}$$

$$P_7 = S_9 \cdot S_{10} = A_{11} \cdot B_{11} + A_{11} \cdot B_{12} - A_{21} \cdot B_{11} - A_{21} \cdot B_{12}$$

Strassen's Method (5/7)

Προσθέσεις στο 4^ο βήμα

$$C_{11} = P_5 + P_4 - P_2 + P_6$$

$$\begin{array}{r}
 A_{11} \cdot B_{11} + A_{11} \cdot B_{22} + A_{22} \cdot B_{11} + A_{22} \cdot B_{22} \\
 \phantom{A_{11} \cdot B_{11} + A_{11} \cdot B_{22} + A_{22} \cdot B_{11} + A_{22} \cdot B_{22}} - A_{22} \cdot B_{11} \phantom{+ A_{22} \cdot B_{21}} + A_{22} \cdot B_{21} \\
 \phantom{A_{11} \cdot B_{11} + A_{11} \cdot B_{22} + A_{22} \cdot B_{11} + A_{22} \cdot B_{22}} - A_{11} \cdot B_{22} \phantom{+ A_{22} \cdot B_{21}} - A_{12} \cdot B_{22} \\
 \phantom{A_{11} \cdot B_{11} + A_{11} \cdot B_{22} + A_{22} \cdot B_{11} + A_{22} \cdot B_{22}} - A_{22} \cdot B_{22} - A_{22} \cdot B_{21} + A_{12} \cdot B_{22} + A_{12} \cdot B_{21} \\
 \hline
 A_{11} \cdot B_{11} \phantom{+ A_{11} \cdot B_{22} + A_{22} \cdot B_{11} + A_{22} \cdot B_{22}} + A_{12} \cdot B_{21}
 \end{array}$$

Strassen's Method (6/7)

$$C_{12} = P_1 + P_2$$

$$\begin{array}{r} A_{11} \cdot B_{12} - A_{11} \cdot B_{22} \\ + A_{11} \cdot B_{22} + A_{12} \cdot B_{22} \end{array}$$

$$A_{11} \cdot B_{12} \qquad + A_{12} \cdot B_{22}$$

$$C_{21} = P_3 + P_4$$

$$\begin{array}{r} A_{21} \cdot B_{11} + A_{22} \cdot B_{11} \\ - A_{22} \cdot B_{11} + A_{22} \cdot B_{21} \end{array}$$

$$A_{21} \cdot B_{11} \qquad + A_{22} \cdot B_{21}$$

Strassen's Method (7/7)

$$C_{22} = P_5 + P_1 - P_3 - P_7$$

$$\begin{array}{r}
 A_{11} \cdot B_{11} + A_{11} \cdot B_{22} + A_{22} \cdot B_{11} + A_{22} \cdot B_{22} \\
 \quad - A_{11} \cdot B_{22} \qquad \qquad \qquad + A_{11} \cdot B_{12} \\
 \qquad \qquad \qquad - A_{22} \cdot B_{11} \qquad \qquad \qquad - A_{21} \cdot B_{11} \\
 - A_{11} \cdot B_{11} \qquad \qquad \qquad - A_{11} \cdot B_{12} + A_{21} \cdot B_{11} + A_{21} \cdot B_{12} \\
 \hline
 \qquad \qquad \qquad A_{22} \cdot B_{22} \qquad \qquad \qquad + A_{21} \cdot B_{12}
 \end{array}$$