ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΜ&ΜΥ
Αλγόριθμοι και Πολυπλοκότητα
$1^{η}$ Σειρά Γραπτών Ασκήσεων
Ακ. έτος 2010-2011

Λύρας Γρηγόρης
Α.Μ.: 03109687

8 Δεκεμβρίου 2011

1 Ασυμπτωτικός συμβολισμός, Αναδρομικές Σχέσεις

α′ *Να ταξινομήσετε τις παρακάτω συναρτήσεις σε αύξουσα σειρά τάξης μεγέθους, να βρείτε δηλαδή μια διάταξη* $g_1, g_2, g_3, ...$ *τέτοια ώστε* $g_1 = O(g_2), g_2 = O(g_3)$, *κοκ. Σε αυτή τη διάταξη, να επισημάνετε τις συναρτήσεις που έχουν ίδια τάξη μεγέθους.*

(i) $\log n^3$

(ii) $\sqrt{n} * \log^{50} n$

(iii) $\frac{n}{\log \log n}$

(iv) $\log n!$

(v) $n * \log^{10} n$

(vi) $n^{1.01}$

(vii) $5^{\log_2 n}$

(viii) $\sum_{k=1}^{n} k^5$

(ix) $\log^{\log n} n = n^{\log \log n}$

(x) $2^{\log_2^4 n}$

(xi) $\log^{\sqrt{n}} n$

(xii) $e^{\frac{n}{\ln n}}$

(xiii) $n * 3^n$

(xiv) 2^{2*n}

(xv) $\sqrt{n!}$

β′ *Να υπολογίσετε την τάξη μεγέθους* Θ *των λύσεων των παρακάτω αναδρομικών σχέσεων. Για όλες τις σχέσεις, να θεωρήσετε ότι* $T(1) = \Theta(1)$

(i) $T(n) = 5 * T(n/7) + n * \log(n) \Rightarrow n^{\log_7 5} = n^{0.827} \Rightarrow n^{0.827} < n * \log n$
$\Rightarrow T(n) \in \Theta(n * \log n)$

(ii) $T(n) = 4 * T(n/5) + n/\log^2 n \Rightarrow n^{\log_5 4} = n^{0.861} \Rightarrow n^{0.861} > n/\log^2 n$
$\Rightarrow T(n) \in \Theta(n/\log^2 n)$

(iii) $T(n) = T(n/3) + 3 * T(n/7) + n$
$\Rightarrow T(n) \in O(n)$

(iv) $T(n) = 6 * T(n/6) + n \Rightarrow n^{\log_6 6} = n$
$\Rightarrow T(n) \in \Theta(n * \log n)$

(v) $T(n) = T(n/3) + T(2n/3) + n$
$\Rightarrow T(n) \in \Theta(n * \log n)$

(vi) $T(n) = 16 * T(n/4) + n^3 * \log^2 n \Rightarrow n^{\log_4 16} = n^2 \Rightarrow n^2 < n^3 * \log^2 n$
$\Rightarrow T(n) \in \Theta(n^3 * \log^2 n)$

(vii) $T(n) = T(\sqrt(n)) + \Theta(\log \log n)$ $(k = \log n, f(k) = T(n))$
$\Rightarrow T(\sqrt{n}) = f(\log \sqrt{n}) = f(\frac{1}{2} * \log n) = f(\frac{k}{2})$
$\Rightarrow f(k) = f(\frac{k}{2}) + \Theta(\log k)$
$\Rightarrow \Theta(\frac{1}{2} * \log^2 k * (\log k + 1))$

(viii) $T(n) = T(n-3) + \log n$
$\Rightarrow T(n) \in \Theta(n * logn)$

2 Ταξινόμηση σε Πίνακα με Πολλά Ίδια Στοιχεία

Έστω πίνακας ακεραίων $A[1...n]$ που χαρακτηρίζεται από πολλές πολλαπλές εμφανίσεις των στοιχείων του. Συγκεκριμένα, θεωρούμε ότι το πλήθος των διαφορετικών στοιχείων του A είναι μόλις πολυλογαριθμικό (δηλ. $O(\log^d n)$, για κάποια σταθερά $d \geq 1$). Να διατυπώσετε έναν συγκριτικό αλγόριθμο που ταξινομεί τον πίνακα A σε χρόνο $O(n * \log\log n)$. Γιατί δεν ισχύει το κάτω φράγμα του $\Omega(n * \log n)$ σε αυτή την περίπτωση;

Για να κάνουμε ταξινόμηση σε ένα τέτοιο πίνακα μπορούμε να κάνουμε το εξής. Χρησιμοποιούμε ένα πίνακα $2 * \log^d n$ όπου θα κρατάμε ταξινομημένα τα στοιχεία και τη συχνότητα εμφάνισής τους. Για κάθε στοιχείο του αρχικού πίνακα κάνουμε binary search στον δεύτερο πίνακα. Αν το βρούμε αυξάνουμε τη συχνότητά του. Αν όχι κάνουμε insertion sort για το στοιχείο στον πίνακα στον δεύτερο πίνακα. Συνολικά θα κάνουμε χρόνο $\Theta(n * \log\log^d n + (\log^d n)^2)$

Αφού ολοκληρωθεί η διαδικασία επεκτείνουμε τα στοιχεία όσο ορίζει η συχνότητα του κάθε ενός με τη σειρά (αφού ο πίνακας είναι ήδη ταξινομημένος) σε χρόνο n. Συνολικά η πολυπλοκότητα του αλγορίθμου είναι $O(n * \log\log n)$.

3 Δυαδική Αναζήτηση

α′ *Μας δίνεται πίνακας A με n ακέραιους ταξινομημένους σε αύξουσα σειρά, αλλά* δεν γνωρίζουμε την τιμή του n. *Ο τύπος δεδομένων του A έχει υλοποιηθεί ώστε να επιστρέφει το μήνυμα σφάλματος* inf *κάθε φορά που επιχειρείται προσπέλαση στοιχείου $A[i]$ με $i > n$. Να διατυπώσετε αλγόριθμο με χρόνο εκτέλεσης $O(\log n)$, που δέχεται ως είσοδο έναν ακέραιο x, και βρίσκει μια θέση του πίνακα A που περιέχει τον x, αν υπάρχει τέτοια θέση.*

Ξεκινάμε με ένα στοιχείο. Και συγκρίνουμε με τον x. Όσο ο x είναι μεγαλύτερος από αυτόν διπλασιάζουμε τον αριθμό των στοιχείων και ελέγχουμε πάλι με τον τελευταίο. Μόλις φτάσουμε σε μεγαλύτερο αριθμό από τον x (δηλαδή ακέραιο ή ∞) έστω στη θέση k και εφαρμόζουμε κλασσική δυαδική αναζήτηση στο τμήμα $A[m...k]$ όπου m το k της προηγούμενης επανάληψης. Μέγιστος αριθμός επαναλήψεων είναι $O(\log k)$.

β′ *Έστω δύο ταξινομημένοι πίνακες ακεραίων $A[1...n]$ και $B[1...n]$. Να διατυπώσετε αλγόριθμο με χρόνο εκτέλεσης $O(\log k)$ που υπολογίζει το k-οστό μικρότερο στοιχείο της ένωσης των πινάκων A και B. Για ευκολία, μπορείτε να υποθέσετε ότι όλα τα στοιχεία των A και B είναι διαφορετικά.*

Παίρνω τα k πρώτα στοιχεία από κάθε πίνακα, αν ο κάθε πίνακας είναι πιο μικρός τότε επεκτείνω προς τα πίσω χρησιμοποιώντας το $maxA[i], B[m]$, όπου $A[i], B[m]$ τα τελευταία στοιχεία των πινάκων. Στη συνέχεια τα χωρίζω στη μέση και συγκρίνω τα $A[k/2]$ με $B[k/2+1]$ και $A[k/2+1]$ με $B[k/2]$. Αν $A[k/2] < B[k/2+1]$ και $B[k/2] < A[k/2+1]$ τότε το k-οστό στοιχείο είναι το $max\{A[k/2], B[k/2]\}$. Αν $A[k/2] < B[k/2+1]$ και $B[k/2] > A[k/2+1]$ τότε επαναλαμβάνω χρησιμοποιώντας τα στοιχεία $[A[k/2+1]..A[k]]$ και $[B[1]..B[k/2]]$ για νέο $k = \frac{k}{2}$.

4 Συλλογή Comics

Ένας φανατικός συλλέκτης comics θυμάται ότι από τη συλλογή του αγαπημένου του geek comic έχει χάσει ένα τεύχος, αλλά δεν θυμάται ποιο. Χρειάζεται να εντοπίσει το τεύχος αυτό, γιατί όλα τα τεύχη της σειράς πωλούνται στο Internet σε τιμή ευκαιρίας, σε μια δημοπρασία που λήγει άμεσα.

Ένας από τους λόγους που η συγκεκριμένη σειρά comics είναι η αγαπημένη του έχει να κάνει με τον ιδιόρρυθμο τρόπο αρίθμησης των τευχών: η αρίθμηση των τευχών είναι δυαδική, ξεκινά από το $0...0$ *και ολοκληρώνεται στο* $1...1$*. Κάθε τεύχος έχει* k *σελίδες, το πλήθος των τευχών είναι* $n = 2^k$*, και τα bits του αριθμού κάθε τεύχους είναι γραμμένα από ένα σε κάθε σελίδα. Έτσι το μοναδικό που μπορεί να κάνει ο συλλέκτης για να εντοπίσει το τεύχος που λείπει είναι να εξετάζει κάθε φορά το* i*-οστό bit της αρίθμησης ενός τεύχους* j*.*

Ο συλλέκτης χρειάζεται λοιπόν έναν αλγόριθμο που εντοπίζει το τεύχος που λείπει με όσο το δυνατόν λιγότερες ερωτήσεις της μορφής "ποιο είναι το i*-οστό bit αρίθμηση του τεύχους* j*;". Μπορείτε να βοηθήσετε τον συλλέκτη διατυπώνοντας έναν αλγόριθμο που χρειάζεται το πολύ* $2 * n$ *τέτοιες ερωτήσεις;*

Ζητάμε πρώτα το MSB για όλα τα τεύχη. Έτσι τα χωρίζουμε σε $\frac{n}{2}$ και $\frac{n}{2} - 1$. Προφανώς αυτό που λείπει είναι στα λιγότερα. Εφαρμόζουμε πάλι κάνοντας $\frac{n}{2}$ ερωτήσεις και κρατάμε πάλι το υποσύνολο με το μικρότερο πλήθος. Αναδρομικά θα καταλήξουμε στο τεύχος που λείπει από τη συλλογή έχοντας ρωτήσει $n + \frac{n}{2} + \frac{n}{4} + .. + 1 = 2 * n * (1 - \frac{1}{2^n}) < 2 * n$ φορές.

5 Πολυκατοικίες χωρίς Θέα

Κατά μήκος ενός πυκνοκατοικημένου δρόμου, που είναι χαραγμένος στη διεύθυνση ανατολή - δύση, υπάρχουν n *πολυκατοικίες με διάφορα ύψη. Κατά μια περίεργη σύμπτωση, όλοι οι κάτοικοι του δρόμου είναι αμετανόητα ρομαντικοί, και επιθυμούν να βλέπουν το ηλιοβασίλεμα από την ταράτσα της πολυκατοικίας τους. Καθώς όμως έχουν χτιστεί όλες αυτές οι πολυκατοικίες, κάτι τέτοιο δεν είναι εφικτό για τους περισσότερους. Θέλοντας λοιπόν να ρυθμίσουν την κατάσταση, αποφασίζουν την οργάνωση ενός είδους μονομαχίας (ως ρομαντικοί), ώστε να λυθεί το πρόβλημα. Εφοδιάζονται λοιπόν με όπλα, και αποφασίζουν το επόμενο πρωί να ανέβουν στις ταράτσες τους και να προσπαθήσουν να διαλύσουν την πολυκατοικία που έχει μεγαλύτερο ύψος από τη δική τους και εμποδίζει τη θέα της δύσης (ή την πρώτη που βλέπουν από αυτές, αν είναι πολλές). Τα όπλα τους ρίχνουν μόνο στην ευθεία παράλληλα με το έδαφος, και οι αποστάσεις μεταξύ των πολυκατοικιών είναι τέτοιες ώστε η βαρύτητα δεν παίζει κανένα ρόλο.*

Ο Δήμαρχος πληροφορείται την κατάσταση, και αντιλαμβάνεται πως πρέπει να ενεργήσει γρήγορα, αν θέλει να προλάβει τα χειρότερα. Για αυτό, προσπαθεί να υπολογίσει σε ποια πολυκατοικία θα σημαδέψουν εκείνο το πρωί οι κάτοικοι κάθε πολυκατοικίας. Αριθμεί τις πολυκατοικίες από τα δυτικά προς τα ανατολικά, και συμπληρώνει έναν πίνακα ακεραίων $A[1...n]$ *με τα ύψη τους (το ύψος των ανθρώπων θεωρείται αμελητέο). Δεδομένου του πίνακα* A*, ο Δήμαρχος χρειάζεται να συμπληρώσει έναν πίνακα* $B[1...n]$*,που κάθε στοιχείο του* $B[i]$ *δίνει την πολυκατοικία που θα δεχθεί τα πυρά των κατοίκων της πολυκατοικίας* i *(αν όλες οι πολυκατοικίες δυτικά της* i *έχουν μικρότερο η ίσο του* $A[i]$*, τότε* $B[i] = 0$*). Μπορείτε να βοηθήσετε το Δήμαρχο διατυπώνοντας έναν αλγόριθμο γραμμικού χρόνου για αυτό το πρόβλημα; Να επιμείνετε στην αναλυτική αιτιολόγηση του γραμμικού χρόνου εκτέλεσης του αλγορίθμου σας.* Ξεκινάμε

με τον από τον $A[1]$ θέτοντας το $B[1] = 0$ εφόσον δεν υπάρχει κάποιος δυτικότερα από αυτόν. Προχωράμε στον επόμενο $A[2]$ και συγκρίνουμε το ύψος του με τον $A[1]$. Αν $A[2] < A[1]$ τότε θέτουμε $B[2] = 1$ αλλιώς συγκρίνουμε το ύψος $A[2]$ με το ύψος του στόχου του $A[1]$. Αν πάλι δεν βρούμε ψηλότερο τρέχουμε πάλι για το στόχο του στόχου αναδρομικά μέχρι να φτάσουμε σε 0 ή σε κάποιον ψηλότερο του $A[2]$ και τον θέτουμε ως στόχο στο πεδίο $B[2]$. Επαναλαμβάνουμε μέχρι n. Μέγιστος αριθμός συγκρίσεων είναι $2n$.