Ασκήσεις Φυλλου εργασιας 1

Βασίλειος Ροδανίδης 12164

1 Άσκηση 1

$(i)\sqrt{25n^2+3n}-2n \in \Theta(n)$

Θετουμε $t(n)=\sqrt{25n^2+3n}-2n$ και $g(n)=n$

Παιρνουμε το οριο $\lim_{n\to\infty}\dfrac{t(n)}{g(n)}$

$\lim_{n\to\infty}\dfrac{t(n)}{g(n)}=\lim_{n\to\infty}\dfrac{\sqrt{25n^2+3n}-2n}{n}=\lim_{n\to\infty}\dfrac{\sqrt{n^2(25+3/n)}-2n}{n}=$
$\lim_{n\to\infty}\dfrac{n\sqrt{(25+3/n)}-2n}{n}=\lim_{n\to\infty}\dfrac{1\sqrt{(25+3/n)}-2\cdot 1}{1}=$

$=\lim_{n\to\infty}\sqrt{(25+3/n)}-2=\lim_{n\to\infty}\sqrt{25}-2=\lim_{n\to\infty}5-2=\lim_{n\to\infty}3=$

$\implies \lim_{n\to\infty}\dfrac{t(n)}{g(n)}=3=c>0$

Αρα $t(n)\in\Theta(g(n))$

Αρα $\sqrt{25n^2+3n}-2n \in \Theta(n)$

(ii) $\log n \in O(2^{\sqrt{logn}})$

Θετουμε $t(n) = \log n$ και $g(n) = 2^{\sqrt{logn}}$

Παιρνουμε το οριο $\lim_{n\to\infty} \dfrac{t(n)}{g(n)}$

$\lim_{n\to\infty} \dfrac{t(n)}{g(n)} = \lim_{n\to\infty} \dfrac{\log n}{2^{\sqrt{\log n}}}$

ϑετω $w = \log n$,$n \to \infty \implies \log n \to \infty \implies w \to \infty$

$$\implies \lim_{n\to\infty} \frac{\log n}{2^{\sqrt{\log n}}} = \lim_{w\to\infty} \frac{w}{2^{\sqrt{w}}} \overset{\frac{\infty}{\infty}}{\implies} \lim_{w\to\infty} \frac{1}{(2^{\sqrt{w}})'} = ... = \lim_{w\to\infty} \frac{c}{\infty} =$$

$= 0$

$$\implies \lim_{n\to\infty} \frac{t(n)}{g(n)} = 0 = c$$

Αρα $t(n) \in O(g(n))$

Αρα $logn \in O(2^{\sqrt{logn}})$

(iii) $log(n!) \in \Theta(nlogn)$

Θετουμε $t(n) = \log(n!)$ και $g(n) = nlogn$

Παιρνουμε το οριο $\lim_{n\to\infty} \dfrac{t(n)}{g(n)}$

$\lim_{n\to\infty} \dfrac{t(n)}{g(n)} = \lim_{n\to\infty} \dfrac{\log(n!)}{nlogn} = \lim_{n\to\infty} \dfrac{\log(n!)}{log(n^n)}$

Ισχυει οτι: $n^n = \underbrace{\cdot n \cdot n \cdot ... \cdot n}_{n \text{ φορες}} > n \cdot (n-1) \cdot (n-2) \cdot ... \cdot 1 = n!$

$\implies n^n > n! \overset{logx \text{ αυξουσα}}{\implies} log(n^n) > log(n!)$ Αρα στο κλασμα μας ο παρονομαστης μηδενιζεται με γρηγοροτερο ρυθμο απ οτι ο αριθμητης επομενως :

$\implies \lim_{n\to\infty} \dfrac{\log(n!)}{log(n^n)} = \lim_{n\to\infty} \dfrac{1}{\infty} = 0$

Αρα $t(n) \in \Theta(g(n)) \implies t(n) \in O(g(n))$

Αρα $log(n!) \in O(nlogn)$

2 Ασκηση 2

(i) $SUM_X SYGKRISI(A[1,...,n])$

$akeraios_x = x$

```
for (i=0,...,n; i++) do
    for(j=(i+1),...,n-1; j++) do
        if(A[i]+A[j]==x)
            return true;
             else
             return false;
```

Το συνολικο πληθος προσθεσης ενος ζευγους αριθμων και συγκρισης του αθροισματος με τον ακεραιο αριθμο x ειναι:

$1 + 2 + 3 + ... + (n - 1) = \frac{1}{2}n(n - 1) = O(n^2)$

(ii)Για το δευτερο προβλημα θα χωρισουμε τον πινακα A[1,...,n] σε δυο υποπινακες A[1,...,$\frac{n}{2}$] και A[$\frac{n}{2}+1$,...n].Ετσι θα κανουμε merge 2 ταξινομημενων πινακων μεγεθους $\frac{n}{2}$ και $\frac{n}{2}$ πραγμα το οποιο γινεται με 2$\frac{n}{2}$ - 1=n-1 συγκρισεις.

π.χ. A[1,3,7,8,2,4,5,11] Με την Merge σπαει ο πινακας σε A[1,3,7,8] και A[2,4,5,11]

και στην συνεχεια ακολουθειται η εξης διαδικασια:

Προσθετω 1 με 2 και συγκρινω με τον x,

Προσθετω 1 με 4 και συγκρινω με τον x,

Προσθετω 1 με 5 και συγκρινω με τον x,

Προσθετω 1 με 11 και συγκρινω με τον x,

Προσθετω 3 με 2 και συγκρινω με τον x,

Προσθετω 3 με 4 και συγκρινω με τον x,

Προσθετω 3 με 5 και συγκρινω με τον x,

Προσθετω 3 με 11 και συγκρινω με τον x,

Προσθετω 7 με 2 και συγκρινω με τον x,

Προσθετω 7 με 4 και συγκρινω με τον x,

Προσθετω 7 με 5 και συγκρινω με τον x,

Προσθετω 7 με 11 και συγκρινω με τον x,

Προσθετω 8 με 2 και συγκρινω με τον x,

Προσθετω 8 με 4 και συγκρινω με τον x,

Προσθετω 8 με 5 και συγκρινω με τον x,

Προσθετω 8 με 11 και συγκρινω με τον x.

Οπου για την Mergesort γνωριζουμε → O(nlogn).

3 Άσκηση 3

ΜΕΘΟΔΟΣ 1

Ορίζω εναν πινακα (πλαγια στοιβα) που υποστηριζει τις παρακατω πραξεις

1. Push(A[i],x):Ενθετει το στοιχειο x στην αριστεροτερη ελευθερη θεση του πινακα A[i] με κοστος: O(i)σταθερο

2. Pop(A[i]):Επιστρεφει και διαγραφει το στοιχειο που μπηκε τελευταιο στον πινακα A[i] με κοστος : O(i) σταθερο

3. Multipop(A[i],m):Επιστρεφει και διαγραφει απ την A[i] τα τελευταια $min\{m, |A[i]|\}$στοιχεια με γραμμικο κοστος ως προς το $min\{m, |A[i]|\}$: $o(min\{m, |A[i]|\}$

Ο πινακας A[i] θα εχει την εξης μορφη βλ. Πινακα 1

Πινακας 1: Αυτος ειναι ο Πινακας A[i]

*	*	*	...	

Ο πινακας ειναι συμπληρωμενος με k στοιχεια (οπου * τυχαιο στοιχειο). Αν εφαρμοσω την push τοτε:

1. Εαν k=i δηλαδη ο πινακας ειναι πληρης τοτε παιρνω εναν καινουργιο πινακα $A[2i]$ με k+1 στοιχεια της μορφης βλ. Πινακα2

Πινακας 2: Πινακας $A[2i]$

*	*	*	...			

2. Εαν k <i τοτε παιρνω τον ιδιο πινακα $A[i]$ με k+1 στοιχεια ο οποιος θα ειναι της μορφης βλ. Πινακα 3

Πινακας 3: Πινακας $A[i]$

$*$	$*$	$*$	$*$	...	

Αν εφαρμοσω την pop τοτε:

Παιρνω τον ιδιο πινακα $A[i]$ με k-1 στοιχεια ο οποιος θα ειναι της μορφης βλ. Πινακα 4

Πινακας 4: Πινακας $A[i]$

$*$	$*$	...			

Αν εφαρμοσω την Multipop τοτε:

Παιρνω τον ιδιο πινακα $A[i]$ με max{0,k-$min\{m, |A[i]|\}$ στοιχεια ο οποιος θα ειναι της μορφης βλ. Πινακα 5

Πινακας 5: Πινακας $A[i]$

$*$	...				

(i)Αναλυση χειροτερης περιπτωσης

n πραξεις, ολες ειναι ακριβες $\implies$ ολες εχουν γραμμικο κοστος

Συνολικο κοστος O(n^2) τετραγωνικο (ii) Επιμερισμενο κοστος

ΜΕΘΟΔΟΣ 2

Ορίζω εναν πινακα (πλαγια στοιβα) που υποστηριζει τις παρακατω πραξεις

1. Push(A[i],x):Ενθετει το στοιχειο x στην αριστεροτερη ελευθερη θεση του πινακα A[i] με κοστος: 1

2. Pop(A[i]):Επιστρεφει και διαγραφει το στοιχειο που μπηκε τελευταιο στον πινακα A[i] με κοστος : 1

3. Multipop(A[i],m):Επιστρεφει και διαγραφει απ την A[i] τα τελευταια $min\{m, |A[i]|\}$στοιχεια με γραμμικο κοστος ως προς το $min\{m, |A[i]|\}$: $(min\{m, |A[i]|\}$

Ο πινακας A[i] θα εχει την εξης μορφη βλ. Πινακα 6

Πινακας 6: Αυτος ειναι ο Πινακας A[i]

$*$	$*$	$*$	...	

Ο πινακας ειναι συμπληρωμενος με k στοιχεια (οπου * τυχαιο στοιχειο). Αν εφαρμοσω την push τοτε:

1. Εαν k=i τοτε παιρνω εναν καινουργιο πινακα $A[2i]$ με k+1 στοιχεια της μορφης βλ. Πινακα7

Πινακας 7: Πινακας $A[2i]$

$*$	$*$	$*$	...			

2. Εαν k <i τοτε παιρνω τον ιδιο πινακα $A[i]$ με k+1 στοιχεια ο οποιος θα ειναι της μορφης βλ. Πινακα 8

Πινακας 8: Πινακας $A[i]$

$*$	$*$	$*$	$*$	...	

Αν εφαρμοσω την pop τοτε:

Παιρνω τον ιδιο πινακα $A[i]$ με k-1 στοιχεια ο οποιος θα ειναι της μορφης βλ. Πινακα 9

Πινακας 9: Πινακας $A[i]$

$*$	$*$	...			

Αν εφαρμοσω την Multipop τοτε:

Παιρνω τον ιδιο πινακα $A[i]$ με max{0,k-$min\{m, |A[i]|\}$ στοιχεια ο οποιος θα ειναι της μορφης βλ. Πινακα 10

Πινακας 10: Πινακας $A[i]$

*	...				

(i)Αναλυση χειροτερης περιπτωσης

Εχω n πραξεις, το πολυ εχω n push αρα το κοστος χειριστης περιπτωσης ειναι 2n. Για Pop/Multipop δεν εχω καποιο κοστος. Συνολικο κοστος n πραξεων $\leq 2n \implies$ Επιμερισμενο κοστος $\leq \frac{2n}{n} = 2$

(ii) Για τον δευτερο πινακα εχουμε την διαφορα οτι ο πινακας $A[i]$ στην εκτελεση της pop εχει την παρακατω ιδιαιτεροτητα:

Για k-1> $i/2$ παιρνω τον ιδιο πινακα $A[i]$ με k-1 στοιχεια ο οποιος θα ειναι της μορφης βλ. Πινακα 11

Πινακας 11: Πινακας $A[i]$

*	*	...			

Για k-1$\leq$ $i/2$ παιρνω εναν καινουργιο πινακα $A[i/2]$ με k-1 στοιχεια ο οποιος θα ειναι της μορφης βλ. Πινακα 12

Πινακας 12: Πινακας $A[i]$

$*$	...	

4 Ασκηση 4

(i)Θα αναπαραστησουμε καθε αλγοριθμο ταξινομησης με συγκρισεις στο δυαδικο δεντρο:

A[1] $\leq A[2]$

Αν ισχυει $\downarrow$ Αν δεν ισχυει $\downarrow$

A[2] $\leq$ A[3] A[1] $\leq$ A[3]

Αν ισχυει $\downarrow$ Αν δεν ισχυει $\downarrow$ Αν ισχυει $\downarrow$ Αν δεν ισχυει $\downarrow$

A[1]$\leq$A[2]$\leq$A[3] A[1]$\leq$A[3] A[2]$\leq$A[1]$\leq$A[3] A[2]$\leq$A[3]

—— $\Downarrow$ —— $\Downarrow$

Αν ισχυει $\downarrow$ Αν δεν ισχυει $\downarrow$ Αν ισχυει $\downarrow$ Αν δεν ισχυει $\downarrow$

A[1]≤A[3]≤A[2] A[3]≤A[1]≤A[2] A[2]≤A[3]≤A[1] A[3]≤A[2]≤A[1]

1. Τα φυλλα του δεντρου T ειναι τουλαχιστον n!

2. Ο αλγοριθμος εκτελει h(T] συγκρισεις στην χειροτερη περιπτωση, οπου h(T) ειναι το υψος του T.

3. To T εχει n! φυλλα $\Longrightarrow$ h(T)$\geq log_2(n!) = log(1 \cdot 2 \cdot ... \cdot \frac{n}{2} \cdot ... \cdot n) \geq log(\frac{n}{2}...n) \geq log(\frac{n}{2}...\frac{n}{2}) = log(\frac{n}{2})^{\frac{n}{2}} = \frac{n}{2} \cdot log\frac{n}{2} = O(nlogn)$

Αρα απο 1,2,3 $\Longrightarrow$ το πληθος συγκρισεων καθε αλγοριθμου ταξινομησης ειναι $\Omega(nlogn)$

(ii) 1. Τα φυλλα του δεντρου T2 ειναι τουλαχιστον 2n

2. Ο αλγοριθμος εκτελει h(T2] συγκρισεις στην χειροτερη περιπτωση, οπου h(T2) ειναι το υψος του T2.

3. To T2 εχει 2n φυλλα $\Longrightarrow$ h(T2)$\geq log_2(2n) = log_2 2 + log_2 n = 1 + log_2 n \geq log_2 n = O(logn)$

Αρα απο 1,2,3 $\Longrightarrow$ το προβλημα της αποφασης ειναι της ταξης του $\Omega(logn)$

Το δενδρο θα ειναι της μορφης:

πχ x ενας ακεραιος και A[i] i=1,..,4 ενα ταξινομημενο διανυσμα με 4 στοιχεια

x=A[1]

Αν ισχυει ↓ Αν δεν ισχυει ↓

——— x=A[2]

Αν ισχυει ↓ Αν δεν ισχυει ↓

——— x=A[3]

Αν ισχυει ↓ Αν δεν ισχυει ↓

——— x=A[4]