# ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ
# ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

**ΕΠΛ 231: Δομές Δεδομένων και Αλγόριθμοι**

# ΘΕΩΡΗΤΙΚΗ ΑΣΚΗΣΗ 1 - ΛΥΣΕΙΣ
# Ανάλυση Πολυπλοκότητας

## ΠΕΡΙΓΡΑΦΗ

Σε αυτή την άσκηση καλείστε να αναλύσετε και να υπολογίσετε το χρόνο εκτέλεσης διαφόρων λειτουργιών χρησιμοποιώντας τους ορισμούς που διδαχτήκατε, την μέθοδο της μαθηματικής επαγωγής και την μέθοδο της αντικατάστασης.

## Άσκηση 1 (15 μονάδες)

Χρησιμοποιείστε τον ορισμό «f(x) είναι O( g(x) )» για να δείξετε ότι:

- Διαβάζετε προσεκτικά την εκφώνηση
- Μην θεωρείτε ότι κάποια βήματα εννοούνται

α) $x^4 + 9x^4 + 4x + 7$ είναι $\mathrm{O}(x^4)$

Τότε, εξ' ορισμού, πρέπει να υπάρχουν σταθερές c και $x_0$ τέτοιες ώστε $x^4 + 9x^4 + 4x + 7 \leq c\cdot x^4$ για κάθε $x \geq x_0$

$x^4 + 9x^4 + 4x + 7 \leq x^4 + 9x^4 + 4x^4 + 7x^4 = 21x^4$

Συνεπώς ισχύει αφού υπάρχουν σταθερές, δηλ. c=21, $x_0=0$ ώστε $x^4 + 9x^4 + 4x + 7 \leq c\cdot x^4$ για κάθε $x \geq x_0$

β) $\frac{(x^2+1)}{(x+1)}$ είναι $\mathrm{O}(x)$

Τότε, εξ' ορισμού, πρέπει να υπάρχουν σταθερές c και $x_0$ τέτοιες ώστε

$\frac{(x^2+1)}{(x+1)} \leq c\cdot x$ για κάθε $x \geq x_0$

$\frac{(x^2+1)}{(x+1)} \leq \frac{(x^2+2x+1)}{(x+1)} = \frac{(x+1)^2}{(x+1)} = (x+1) \leq 2x$ για κάθε $x \geq 1$

Συνεπώς ισχύει αφού υπάρχουν σταθερές, δηλ. c=2, $x_0=1$ ώστε

$\frac{(x^2+1)}{(x+1)} \leq c\cdot x$ για κάθε $x \geq x_0$

- Η απόδειξη πρέπει να είναι βάση του ορισμού
- Ασυμπτωματική ανάλυση δεν επιτρέπεται
- Η παραγοντοποίηση $\frac{(x^2+1)}{(x+1)} = \frac{(x+(-1))(x-(-1))}{(x+1)}$ είναι λάθος.

γ) $x^2 + 4\mathrm{x} + 17$ είναι $\mathrm{O}(x^3)$,
αλλά επίσης, ότι το $x^3$δεν είναι $\mathrm{O}(x^2 + 4\mathrm{x} + 17)$

i) Είναι προφανές ότι $x^2 + 4\mathrm{x} + 17$ είναι $\mathrm{O}(x^3)$.

ii) Ας υποθέσουμε ότι ισχύει $x^3$ είναι $\mathrm{O}(x^2 + 4\mathrm{x} + 17)$. Τότε, εξ' ορισμού, πρέπει να υπάρχουν σταθερές c και $x_0$ τέτοιες ώστε $x^3 \leq \mathrm{c}.(x^2 + 4\mathrm{x} + 17)$ για κάθε $x \geq x_0$.
$\mathrm{c}.(x^2 + 4\mathrm{x} + 17) \leq \mathrm{c}.(x^2 + 4x^2 + 17x^2) \leq \mathrm{c}.(21x^2)$

Συνεπώς θέλουμε να αποδείξουμε ότι $x^3 \leq \mathrm{c}.(21x^2)$
Το οποίο είναι ισοδύναμο με το να αποδείξουμε $x \leq 21c$ το οποίο προφανώς δεν ισχύει για κάθε $x \geq 21c$.

- Η απόδειξη πρέπει να είναι βάση του ορισμού
- Ασυμπτωματική ανάλυση δεν επιτρέπεται

**Άσκηση 2 (15 μονάδες)**

Χρησιμοποιείστε την μαθηματική επαγωγή για να αποδείξετε ότι το άθροισμα των πρώτων ***n*** **θετικών περιττών αριθμών είναι ίσο με** $\boldsymbol{n^2}$.

- Προσπαθήστε να γράφετε τι θέλετε να αποδείξετε (π.χ., Έστω ότι ισχύει για n=k, να αποδείξω ότι ισχύει για n=k+1

Βασικό βήμα: Π(1) ισχύει αφού $1 = 1^2$.
Επαγωγική Υπόθεση: Έστω ότι ισχύει η Π(k), δηλ. $1+3+\ldots+(2k-1) = k^2$.
Επαγωγικό Βήμα: Να αποδείξουμε ότι ισχύει για k+1
$1+3+\ldots+(2k-1)+(2(k+1)-1) = (k+1)^2$

| | |
|---|---|
| $1+3+\ldots+(2k-1)+(2(k+1)-1)$ | |
| $= k^2 + (2(k+1)-1)$ | (επαγωγική υπόθεση) |
| $= k^2 + 2k+1$ | (πράξεις) |
| $= (k+1)^2$ | (παραγοντοποίηση) |

,που είναι του ζητούμενο

**Άσκηση 3 (15 μονάδες)**

Υποθέστε ότι κάποιος έχει αναλύσει τον χρόνο εκτέλεσης μίας σειράς αλγορίθμων και δημιούργησε μία λίστα με τις πολυπλοκότητες των

συναρτήσεων τους. Καλείστε να κατατάξατε τις συναρτήσεις αυτές σε κατηγορίες έτσι ώστε δύο συναρτήσεις *f(n)* και *g(n)* να ανήκουν στην ίδια κατηγορία αν και μόνο αν ανήκουν στην ίδια τάξη (δηλαδή αν $f(n) \in \Theta(g(n))$ ή $g(n) \in \Theta(f(n))$ ).
Στην συνέχεια καλείστε να ταξινομήσετε τις κατηγορίες σε αύξουσα σειρά ως προς την τάξη τους.

| | | | |
|---|---|---|---|
| $\sqrt{n}$ | $n$ | $3^n$ | $n^2 + log(n)$ |
| $6$ | $log(n)$ | $n - n^3 + 7n^5$ | $nlog(n)$ |
| $n!$ | $n^2 + 5n^3$ | $n^2$ | $n^3$ |

- Κάποιοι, δεν βρήκατε σε ποιες κατηγορίες (τάξεις) ανήκουν οι συναρτήσεις
- Πρέπει να αποδείξετε ότι μία συνάρτηση Θ(f(n)) είναι πιο μικρή από μία άλλη Θ(g(n))

Πιο κάτω παρουσιάζεται η σωστή σειρά των συναρτήσεων ξεκινώντας από τη μικρότερη τάξη προς τη μεγαλύτερη. Συναρτήσεις που ανήκουν στην ίδια τάξη βρίσκονται γραμμένες στην ίδια γραμμή.

6
lg n
$\sqrt{n}$
n
n lg n
$n^2$, $n^2 + \log n$
$n^3$, $n^2 + 5n^3$
$n - n^3 + 7n^5$
$3^n$
n!

Ακολουθεί αιτιολόγηση κάποιων σχέσεων:

(i) Σύγκριση $3^n$ και n!

Αποδείχθηκε στις διαλέξεις ότι 3n ∈ O(n!)
Επιπλέον, μπορούμε να δείξουμε ότι οι δύο συναρτήσεις δεν ανήκουν στην ίδια τάξη, δηλαδή, 3n ∉ Ω(n!). Ας υποθέσουμε, για να φτάσουμε σε αντίφαση, ότι 3n ∈ Ω(n!). Τότε, εξ' ορισμού, πρέπει να υπάρχουν σταθερές c και $n_0$ τέτοιες ώστε 3n ≥ c· n! για κάθε n ≥ $n_0$

Τότε έχουμε
3n ≥ c· n! για κάθε n ≥ $n_0$
= c·n·(n-1)! για κάθε n ≥ n0
> (Αποδείχθηκε στις διαλέξεις ότι 3n ∈ O(n!))
c·n·3n-1 για κάθε n ≥ max(n0,7)

Επομένως

$3n \in \Omega(n!)$ αν και μόνο αν $\exists$ c τέτοιο ώστε $3n > c \cdot n \cdot 3n\text{-}1$ για κάθε $n \geq \max(n_0,7)$,
$\Leftrightarrow$ $3n \in \Omega(n!)$ αν και μόνο αν $\exists$ c τέτοιο ώστε $3 > c \cdot n$ για κάθε $n \geq \max(n0,7)$,
$\Leftrightarrow$ $3n \in \Omega(n!)$ αν και μόνο αν $\exists$ c τέτοιο ώστε $c < 3/n$ για κάθε $n \geq \max(n0,7)$,
Προφανώς όμως δεν υπάρχει θετική σταθερά c τέτοια ώστε $c < 3/n$ καθώς το n μεγαλώνει απεριόριστα. (Ο λόγος 3/n τείνει στο 0 καθώς το n τείνει στο άπειρο.)
Συμπέρασμα: Δεν υπάρχουν ζεύγος σταθερών c και n0 τέτοιες ώστε $3n \geq c \cdot n!$ Για κάθε $n \geq n_0$, και $3n \notin \Omega(n!)$.

(ii) Σύγκριση $n^2 + 5n^3$ και $n^3$
Είναι προφανές ότι $n^3 \in O(n^2 + 5n^3)$. Επίσης ισχύει $n^3 \in \Omega(n^2 + 5n^3)$ αφού για c=1 και k=1, $n^2 + 5n^3 \geq c \cdot n^3$, για κάθε $n \geq k$.

(iii) Σύγκριση lg n και n lg n.
Είναι προφανές ότι $n \in O(n \lg n)$, αφού για c = 1, και k = 1
$\lg n \leq n \lg n$, για κάθε $n \geq k$
($\Leftrightarrow$ $1 \leq n$, για κάθε $n \geq 1$)

Από την άλλη δεν ισχύει ότι $n \in \Omega(n \lg n)$.
Ας υποθέσουμε ότι ότι $n \in \Omega(n \lg n)$. Τότε υπάρχουν c και κάθε n0 τέτοια ώστε
$\lg n \geq c \cdot n \lg n$ για κάθε $n \geq n0$.

Αυτό συνεπάγεται ότι υπάρχουν c και κάθε n0 τέτοια ώστε
$1 \geq c \cdot n$ για κάθε $n \geq n0$
$\Leftrightarrow$ $c \leq 1/n$ για κάθε $n \geq n0$
Καθώς το n τείνει στο άπειρο ο λόγος 1/n τείνει στο 0. Επομένως δεν είναι δυνατόν να υπάρχει θετική σταθερά c που να ικανοποιεί την πιο πάνω σχέση. Συμπέρασμα:
$n \notin \Omega(n \lg n)$.

(iv) Σύγκριση lg n και $\sqrt{n}$
Ας υποθέσουμε ότι $\lg n \notin O(\sqrt{n})$ . Τότε για κάθε c και κάθε n0 υπάρχει k > n0 τέτοιο ώστε $\lg k > c\, k$ και κατά συνέπεια $\lg^2 k > (c \sqrt{k})^2 = c^2 \sqrt{k}$.
Αυτό συνεπάγεται ότι lg2 $n \notin O(n)$ που γνωρίζουμε από τις διαλέξεις ότι δεν ισχύει. Επομένως $\lg n \in O(\ n)$ και το ζητούμενο έπεται .
Μπορεί επίσης να δειχθεί ότι $\lg n \notin \Omega(\sqrt{n})$ .

**Άσκηση 4 (40 μονάδες)**

Να αναλύσετε τον χρόνο εκτέλεσης χείριστης περίπτωσης των πιο κάτω σαν συνάρτηση του *n*. Υποθέστε, χωρίς βλάβη της γενικότητας, ότι το *n* είναι δύναμη του 2.

A)

```
    sum=0;
    for ( i=1; i<=n; i*=2 )
         for ( j=1; j<=i; j++ )
              sum++;
```

Έστω ότι $i=2^k$

Συνεπώς μπορούμε να μετασχηματίσουμε το A ως εξής:

```
    sum=0;
    for ( k=0; k<=lg₂n; k++ )
         for ( j=1; j<=2^k; j++ )
              sum++;
```

A=Εσωτερικός Βρόγχος (for ( j=1; j<=2k; j++ )) = $2^k$

Εξωτερικός Βρόγχος (for ( k=0; k<=lg2n; k++ )) =$\sum_{k=0}^{lg_2 n} 2^{\kappa} = 2^{(lg_2 n)+1} - 1$

$= 2.\,2^{(lg_2 n)} - 1 = 2n - 1 \in O(n)$

B)

```
    r=0;
    for ( i=1; i<=√n; i++ )
         for ( j=1; j<=n; j*=2 )
              if ( n % 2 == 0 )    //το n είναι άρτιος
                   for ( k=1; k<=n; k++ )
                        r++;
              else                 //το n είναι περιττός
                   r--;
```

Επειδή είναι όλοι ανεξάρτητοι, η ανάλυση γίνεται πιο έυκολα

A=Εσωτερικός Βρόγχος (for ( k=1; k<=n; k++ ) r++;) = n

B=Εσωτερικός Βρόγχος (for ( j=1; j<=n; j*=2 )) = lgn * A(=n) = nlgn

Εξωτερικός Βρόγχος (for ( i=1; i<=$\sqrt{n}$; i++ )) = $\sqrt{n}$ * B= n$\sqrt{n}$ lgn

$\in$O(n$\sqrt{n}$ lgn)

**Άσκηση 5 (15 μονάδες)**

Να υπολογίσετε το χρόνο εκτέλεσης του παρακάτω αναδρομικού προγράμματος λύνοντας οποιεσδήποτε αναδρομικές εξισώσεις συναντήσετε με την μέθοδο της αντικατάστασης. Υποθέστε χωρίς βλάβη της γενικότητας ότι το *n* είναι δύναμη του 2.

```
SplitSum( int n ) {
     int sum = 0;
     for ( j=1; j<=n; j++ ) sum++;
     if ( n > 1 ) return SplitSum( n/2 );
     else return 1;
}
```

- Μία αναδρομική εξίσωση αποτελείται από δύο μέρη, το Βήμα Τερματισμού και το Αναδρομικό Βήμα.
- Μην ξεχνάτε να αναφέρετε την γενική μορφή.
- Από την γενική μορφή, σκοπός μας είναι να φθάσουμε στο Βήμα Τερματισμού και να απαλείψουμε τους όρους που έχουν σχέση με την συνάρτηση T.
- Επιβεβαιώνετε με το master theorem για να είστε πιο σίγουροι για τις απαντήσεις σας.

$T(1) = 1$

$T(n) = T(n/2) + n$ // Εκτέλεση 1

$= T(n/4) + n/2 + n$ // Εκτέλεση 2

$= T(n/8) + n/4 + n/2 + n$ // Εκτέλεση 3

$= T(n/2^k) + n/2^{(k-1)} + \dots n/2^1 + n/2^0$ //Γενίκευση

$= T(1) + (n/2^{(k-1)} + \dots + n/2 + n)$ //Έστω ότι $2^k = n \rightarrow k = \log_2 n$

$= 1 + (2n-1)$ //Γεωμετρική πρόοδος

$\in O(n)$

Επιβεβαίωση με Master Theorem

$a=1, b=2, c=1, d=1 \rightarrow a=1<b^d=2^1=2$

Συνεπώς $T(n)$ is $O(n^d)=O(n)$