ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ
ΕΡΕΥΝΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ

ΠΕΡΙΦΕΡΕΙΑΚΗ ΔΙΕΥΘΥΝΣΗ Π.Ε. & Δ.Ε. ΑΤΤΙΚΗΣ
ΓΡΑΦΕΙΟ ΣΧΟΛΙΚΩΝ ΣΥΜΒΟΥΛΩΝ Δ.Ε. Β΄ ΑΘΗΝΑΣ

Δρ. Ευάγγελος Κανίδης
Σχολικός Σύμβουλος Πληροφορικής

Χαλάνδρι, 15-03-2016
Αρ. Πρωτ.:

ΠΡΟΣ: ΓΕΛ Β' Αθήνας
(Δημόσια και Ιδιωτικά)

Ταχ. Δ/νση:	Έλλης 3 15232 Χαλάνδρι
Τηλ:	210 6080654, 6843551
Fax:	210 6850754
e-mail:	symbath@sch.gr
website:	http://users.sch.gr/symbath/

ΘΕΜΑ Οδηγίες – διευκρινήσεις για την διδασκαλία του 5ου κεφ. ΑΕΠΠ

Αγαπητοί συνάδελφοι

Μετά από πολλά ερωτήματα που έχω δεχθεί για τη διδασκαλία του κεφαλαίου 5 (Πολυπλοκότητα) στο μάθημα Ανάπτυξη Εφαρμογών σε Προγραμματιστικό περιβάλλον (ΑΕΠΠ), σας στέλνω τις προτάσεις μου σχετικά με την προσέγγιση της διδασκαλίας αυτού του κεφαλαίου.

Υπενθυμίζω ότι οδηγίες για το μάθημα ΑΕΠΠ καθορίζονται από το Υπουργείο Παιδείας καθώς και από το Ινστιτούτο Εκπαιδευτικής Πολιτικής (ΙΕΠ), ως εκ τούτου είναι προφανές ότι οι προτάσεις αυτές δεν είναι επίσημες και δεν δεσμεύουν ούτε το Υπουργείο ούτε φυσικά την Κεντρική Επιτροπή Γενικών Εξετάσεων.

Προτεινόμενη διδακτική προσέγγιση - επεξηγήσεις:

Στην παράγραφο 5.1.1 αναφέρεται η χειρότερη περίπτωση ενός αλγορίθμου. Η έννοια αυτή συνδέεται με τον αριθμό των πράξεων που πρέπει να εκτελέσει ο αλγόριθμος για τη λύση του προβλήματος και για πρώτη φορά αναφέρονται οι "βασικές πράξεις" με την πρόταση

" Για παράδειγμα μια βασική πράξη μπορεί να είναι

- ανάθεση τιμής
- σύγκριση 2 μεταβλητών
- αριθμητική πράξη μεταξύ 2 μεταβλητών"

Παρατηρούμε ότι οι συγγραφείς δεν ορίζουν με αυστηρότητα ποιες πράξεις θεωρούνται "βασικές" Για παράδειγμα δεν αναφέρεται ως βασική η πράξη της εκχώρησης. Αυτό όμως δεν είναι λάθος διότι οι βασικές πράξεις σχετίζονται άμεσα με την αρχιτεκτονική που χρησιμοποιεί ο ΗΥ, με τον compiler καθώς και με τη γλώσσα με την οποία θα υλοποιηθεί ο αλγόριθμος. Αφού τονιστεί το ανωτέρω, ο εκπαιδευτικός μπορεί να συμπληρώσει ότι για το βιβλίο βασικές πράξεις είναι και οι :

- Εκχώρηση τιμής σε μια μεταβλητή
- Πρόσβαση στην τιμή ενός συγκεκριμένου κελιού ενός πίνακα
- Αύξηση (ή μείωση) της τιμής μιας μεταβλητής

Η τελευταία αναφερόμενη πράξη συνάγεται από το παράδειγμα μέτρησης των πράξεων που αναφέρεται στην παράγραφο 5.1.3, όπου η θεωρητική έκφραση i ←i +1 υπολογίζεται ως μια πράξη.

Συνεπώς η αύξηση (ή μείωση) της τιμής οποιασδήποτε μεταβλητής (π.χ x ← x+1) υπολογίζεται ως μια πράξη ενώ αν εμπλέκονται και άλλες μεταβλητές (x ← y+1), υπολογίζεται ως δύο.

Ως αιτιολόγηση των ανωτέρω μπορούν να χρησιμοποιηθούν τα παρακάτω**:** Στην αρχή του βιβλίου (παράγραφος 1.6) αναφέρεται σε επίπεδο αρχιτεκτονικής " Όσο και αν τυχόν ξαφνιάζει, ο υπολογιστής δεν μπορεί να εκτελεί παρά μόνο τρεις λειτουργίες: πρόσθεση, ….. σύγκριση,…. μεταφορά δεδομένων, …..". Συνεπώς η x ← y+1 εκτελεί δύο λειτουργίες μια πρόσθεση και μια μεταφορά δεδομένων σε άλλη θέση μνήμης. Στην x ← x+1 δεν υπάρχει η μεταφορά δεδομένων σε άλλη θέση μνήμης. Σε επίπεδο γλώσσας , για παράδειγμα στη γλώσσα assembly, υπάρχει η εντολή ADD x, 1 (μια πράξη), ενώ η μεταφορά είναι διαφορετική πράξη. Θεωρώ αυτονόητο ότι τουλάχιστον η τελευταία αιτιολόγηση απευθύνεται στους εκπαιδευτικούς πληροφορικής και όχι για χρήση σε μαθητές.

Σε κάθε περίπτωση να τονιστεί ότι δεν έχει νόημα ο ακριβής αριθμός των πράξεων και αν γίνει λάθος σε μια εντολή εκχώρησης ως προς το πόσες πράξεις είναι (αν είναι μία, δύο ή πέντε!) δεν αλλάζει την πολυπλοκότητα (τάξη) του αλγόριθμου. Λόγω των ανωτέρω θεωρώ απίθανο να ζητηθεί σε εξετάσεις ο αριθμός των πράξεων ενός αλγορίθμου.

Οι υπολογισμοί του πίνακα 5.2 για τον αλγόριθμο της παραγράφου 5.1.3 ισχύουν αν υποτεθεί ότι η εντολή "Για" εκτελείται Ν-φορές και κάθε βασική πράξη θέλει 1 μs.

Για τις εντολές που εκτελούνται μόνο μια φορά x ←123, y ← 234, εκτύπωσε x, εκτύπωσε y. Εκτύπωσε z : 5 πράξεις (5 μs)

Οι εντολές μέσα στο βρόχο Για εκτελούνται:
Αρχική τιμή i : 1 πράξη
Έλεγχος i : N+1 πράξεις
Αύξηση : Ν πράξεις
Εκτύπωση : Ν πράξεις
Υπολογισμός (z ← x*y) : 2Ν πράξεις
Συνολικές πράξεις

5+1+N+1+N+N+2N**= 5N+7**

Ο ανωτέρω τύπος για Ν=5 δίνει 32μs, για Ν=10 δίνει 57 μs και Ν=1.000.000 βγάζει (περίπου 5.000.000 πράξεις δηλαδή 5.000.000 μs = 5 sec (1 sec = 1.000.000 μs).

Η τελευταία μέτρηση δείχνει ότι δεν έχει νόημα να προσθέσεις στα 5 εκ πράξεις ακόμα 7 πράξεις. Στους μαθητές πρέπει να περάσουμε την αξία αυτής της διαδικασίας και το ρόλο που παίζουν οι επαναληπτικές δομές στην αύξηση των πράξεων και όχι την ακριβή μέτρηση μιας εντολής που δεν έχει νόημα εφόσον δεν αλλάζει την πολυπλοκότητα του αλγορίθμου.

Για το σκοπό αυτό να αξιοποιηθεί το παράδειγμα 2 της παραγράφου 5.2 από το τετράδιο του μαθητή το οποίο υπολογίζει σε Ν πράξεις κάθε εντολή επανάληψης και απλά υπολογίζει ότι ο αλγόριθμος είναι τετραγωνικός.

Υπενθυμίζω την διδακτική προσέγγιση του κεφαλαίου 5 που είχα προτείνει τον Νοέμβριο:

"Από την παράγραφο 5.3 διδάσκεται το τμήμα μέχρι τον ορισμό της πολυπλοκότητας. Ο Ορισμός της πολυπλοκότητας διδάσκεται από το βιβλίο της Β΄ ΓΕΛ. *"Η πολυπλοκότητα ενός αλγορίθμου δίνει ένα μέτρο της χρονικής καθυστέρησης του αλγορίθμου για την επίλυση ενός προβλήματος " 'η* *ισοδύναμα "Η πολυπλοκότητα ενός αλγορίθμου δίνει ένα μέτρο της ταχύτητας εκτέλεσης του αλγορίθμου".*

Οι μαθητές να συγκρίνουν ως προς την αποδοτικότητα τον αλγόριθμο σειριακής και δυαδικής αναζήτησης. Για τη σύγκριση αυτή, αφού βρουν το μέσο αριθμό πράξεων που απαιτεί ο αλγόριθμος σειριακής αναζήτησης Ν στοιχείων, να τον συγκρίνουν με τον πίνακα που δείχνει τον αριθμό των συγκρίσεων στη δυαδική αναζήτησης για διάφορα πλήθη στοιχείων.

Αριθμός συγκρίσεων στη δυαδική αναζήτηση

Στοιχεία Ν	**Συγκρίσεις**
10	4
100	7
1.000	10
10.000	14
100.000	17
1.000.000	20
10.000.000	24
100.000.000	27
1.000.000.000	30

Για τον συμβολισμό Ο της πολυπλοκότητας δεν πρέπει να αναλυθεί τι ακριβώς εκφράζει και πως υπολογίζεται σε ένα αλγόριθμο. Προτείνεται ο εκπαιδευτικός να δείξει τον πίνακα 2.2 και την εικόνα 2.10 από τη σελίδα 24 του βιβλίου της Β' ΓΕΛ, καθώς και τον πίνακα 5.4 του βιβλίου της Γ΄ τάξης και να συζητήσει με τους μαθητές, για την αύξηση του χρόνου ολοκλήρωσης που απαιτεί ένας αλγόριθμος, καθώς αυξάνεται η πολυπλοκότητά του.

Τέλος μπορεί να αναφερθεί ότι πρακτικά τα απλά προγράμματα μπορούν να αναλυθούν μετρώντας τους φωλιασμένους βρόχους που υπάρχουν στο πρόγραμμα. Ένας απλός βρόχος, που διασχίζει Ν στοιχεία, δίνει πολυπλοκότητα Ν, ένας βρόχος μέσα σ᾽ ένα βρόχο δίνει n^2, ένας βρόχος μέσα μέσα σ᾽ ένα βροχο δίνει n^3 κ.λπ.

Με εκτίμηση

Δρ. Ευάγγελος Κανίδης
Σχ. Σύμβουλος Πληροφορικής