ePublishing
ΕΘΝΙΚΟ ΚΕΝΤΡΟ ΤΕΚΜΗΡΙΩΣΗΣ NATIONAL DOCUMENTATION CENTRE

Συνέδρια της Ελληνικής Επιστημονικής Ένωσης Τεχνολογιών Πληροφορίας & Επικοινωνιών στην Εκπαίδευση

Τόμ. 1 (2014)

7ο Πανελλήνιο Συνέδριο Διδακτικής της Πληροφορικής

Μελέτη της συμβολής του περιβάλλοντος οπτικοποίησης αλγορίθμων DAVE στην οικοδόμηση αλγορίθμων ταξινόμησης από μαθητές Γ' Λυκείου

Ευριπίδης Βραχνός, Αθανάσιος Τζιμογιάννης

Βιβλιογραφική αναφορά:

Βραχνός Ε., & Τζιμογιάννης Α. (2022). Μελέτη της συμβολής του περιβάλλοντος οπτικοποίησης αλγορίθμων DAVE στην οικοδόμηση αλγορίθμων ταξινόμησης από μαθητές Γ' Λυκείου. *Συνέδρια της Ελληνικής Επιστημονικής Ένωσης Τεχνολογιών Πληροφορίας & Επικοινωνιών στην Εκπαίδευση*, *1*, 082–091. ανακτήθηκε από https://eproceedings.epublishing.ekt.gr/index.php/cetpe/article/view/4376

https://epublishing.ekt.gr | **e-Εκδότης:** EKT | **Πρόσβαση:** 21/01/2026 00:48:37

Μελέτη της συμβολής του περιβάλλοντος οπτικοποίησης αλγορίθμων DAVE στην οικοδόμηση αλγορίθμων ταξινόμησης από μαθητές Γ' Λυκείου

Ευριπίδης Βραχνός, Αθανάσιος Τζιμογιάννης
evrachnos@gmail.com, ajimoyia@uop.gr
Τμήμα Κοινωνικής και Εκπαιδευτικής Πολιτικής, Πανεπιστήμιο Πελοποννήσου

Περίληψη

Η διδασκαλία των βασικών αλγοριθμικών δομών σε μαθητές και φοιτητές αποτελεί μια ιδιαίτερα δύσκολη και ενδιαφέρουσα διαδικασία. Τα προγραμματιστικά αντικείμενα και οι δομές που χρησιμοποιούνται για τον σχεδιασμό αλγορίθμων είναι δύσκολο να παρουσιαστούν με συμβατικά μέσα λόγω της αφηρημένης και δυναμικής φύσης τους. Τα τελευταία χρόνια έχουν αναπτυχθεί διάφορα εκπαιδευτικά περιβάλλοντα οπτικοποίησης αλγορίθμων, τα οποία έχουν ως στόχο να βοηθήσουν τους μαθητές να οικοδομήσουν επαρκείς αναπαραστάσεις για τις προγραμματιστικές έννοιες μέσα από την ανάδειξη σημαντικών χαρακτηριστικών των αλγορίθμων. Στην εργασία αυτή παρουσιάζονται τα αποτελέσματα της μελέτης μιας πειραματικής διδακτικής παρέμβασης σε μαθητές Λυκείου, στην ενότητα της ταξινόμησης ευθείας ανταλλαγής, η οποία βασίστηκε στη χρήση του εκπαιδευτικού περιβάλλοντος οπτικοποίησης αλγορίθμων DAVE. Τα αποτελέσματα ανέδειξαν σημαντικές πληροφορίες για τη συμβολή του λογισμικού στην εξέλιξη της αλγοριθμικής σκέψης των μαθητών και στην οικοδόμηση αλγορίθμων με στόχο την επίλυση προβλημάτων ταξινόμησης πινάκων.

Λέξεις κλειδιά: Οπτικοποίηση αλγορίθμων, περιβάλλον DAVE, πίνακες, αλγόριθμοι ταξινόμησης

Εισαγωγή

Ο προγραμματισμός υπολογιστών και η αλγοριθμική σχεδίαση συνιστούν το σημαντικότερο και συνάμα το πιο απαιτητικό αντικείμενο της εκπαίδευσης στην πληροφορική. Βασικές έννοιες όπως είναι η μεταβλητή, η δομή επιλογής και η δομή επανάληψης, δυσκολεύουν αρκετά τους μαθητές όχι μόνο ως προς την κατανόηση αλλά και ως προς την εφαρμογή τους για την επίλυση προβλημάτων (Soloway & Spohrer, 1989; Jimoyiannis, 2011; Robins et al., 2003). Πιο σύνθετες έννοιες, όπως είναι οι δομές δεδομένων και οι αλγόριθμοι επεξεργασίας τους, είναι αναμενόμενο να δυσκολεύουν ακόμη περισσότερο τους αρχάριους προγραμματιστές (Danielsiek, 2012). Σύμφωνα με μια μεγάλη έρευνα που έγινε από καθηγητές Πληροφορικής (Dale, 2006), δυο από τις έννοιες που δυσκολεύουν περισσότερο τους φοιτητές στον προγραμματισμό είναι οι δομές επανάληψης και οι πίνακες, οι οποίοι αποτελούν την πρώτη δομή δεδομένων με την οποία έρχονται σε επαφή οι φοιτητές. Πάνω στις δυο αυτές βασικές έννοιες χτίζονται οι αλγόριθμοι αναζήτησης και ταξινόμησης, οι οποίοι βασίζονται σε συνεχείς αντιμεταθέσεις, μετακινήσεις και συγκρίσεις των στοιχείων ενός πίνακα. Η λειτουργικότητα των αλγορίθμων αυτών δεν είναι εύκολο να παρουσιαστεί με συμβατικά μέσα (πίνακας, χαρτί και μολύβι).

Τα τελευταία χρόνια, διαπιστώνεται αυξημένο ενδιαφέρον για το σχεδιασμό μαθησιακών δραστηριοτήτων που βασίζονται στις αρχές του εποικοδομισμού (Hadas et al., 2013; Κόμης & Τζιμογιάννης, 2006; Τζιμογιάννης, 2005). Κύριοι άξονες των εποικοδομιστικών διδακτικών προσεγγίσεων είναι η εκτίμηση των προϋπαρχουσών γνώσεων και αντιλήψεων των μαθητών και η οργάνωση διδακτικών-μαθησιακών δραστηριοτήτων που να ευνοούν τη

Π. Αναστασιάδης, Ν. Ζαράνης, Β. Οικονομίδης & Μ. Καλογιαννάκης, (Επιμ.), *Πρακτικά 7ου Πανελλήνιου Συνεδρίου «Διδακτική της Πληροφορικής»*. Πανεπιστήμιο Κρήτης, Ρέθυμνο, 3-5 Οκτωβρίου 2014.

διερευνητική, ανακαλυπτική και συνεργατική μάθηση. Οι προσεγγίσεις αυτές δίνουν έμφαση στον παιδαγωγικό σχεδιασμό της διδασκαλίας του προγραμματισμού και στη μετατόπιση από το συντακτικό στην καλλιέργεια δεξιοτήτων επίλυσης προβλημάτων (αναλυτική-συνθετική σκέψη, αφαιρετική ικανότητα, μοντελοποίηση λύσεων).

Στο πλαίσιο αυτό, έχουν προταθεί κατάλληλα σχεδιασμένες μαθησιακές δραστηριότητες με χρήση εκπαιδευτικών περιβαλλόντων προσομοίωσης-οπτικοποίησης δομών δεδομένων και αλγορίθμων (Sorva et al., 2013; Urquiza–Fuentes et al., 2009; Βραχνός & Τζιμογιάννης, 2009; 2014). Τα περιβάλλοντα αυτά παρέχουν νέες δυνατότητες για την οικοδόμηση γνώσεων και την ανάπτυξη δεξιοτήτων στον προγραμματισμό. Το βασικό χαρακτηριστικό τους είναι η επεξηγηματική αναπαράσταση της δυναμικής φύσης των αφηρημένων δομών και αλγορίθμων που είναι δύσκολο να αναπαρασταθούν με συμβατικά μέσα.

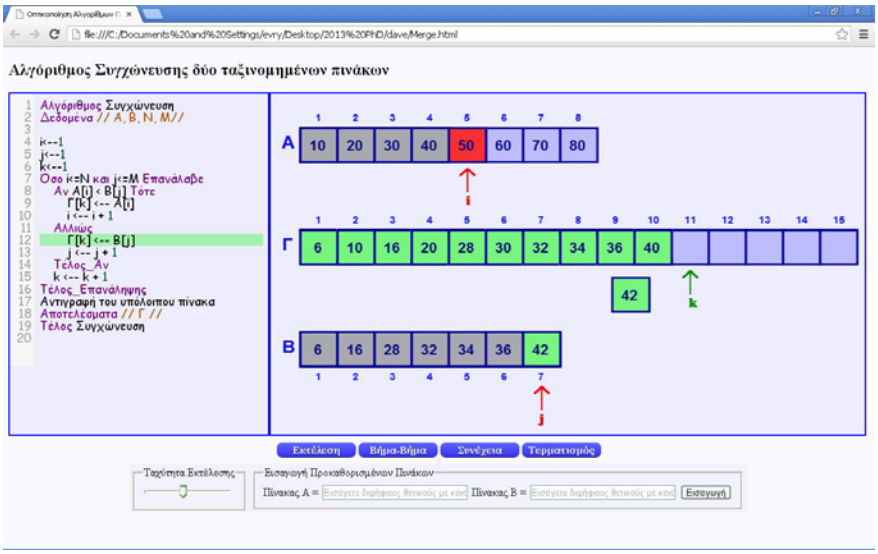
Το DAVE (Dynamic Algorithm Visualization Environment) αποτελεί ένα εκπαιδευτικό περιβάλλον που επιτρέπει τη δυναμική οπτικοποίηση αλγορίθμων ταξινόμησης και αναζήτησης σε πίνακες, αναδεικνύοντας τα ιδιαίτερα χαρακτηριστικά (κρίσιμα στοιχεία) της λογικής κάθε αλγορίθμου (Vrachnos & Jimoyiannis, 2014). Το λογισμικό προωθεί τον πειραματισμό με την οπτικοποίηση του αλγορίθμου, με στόχο την υποστήριξη των μαθητών ώστε να οικοδομήσουν αποτελεσματικές αναπαραστάσεις για δομές και αλγορίθμους.

Στην εργασία αυτή παρουσιάζονται τα αποτελέσματα μιας πειραματικής παρέμβασης με χρήση του λογισμικού DAVE που έγινε σε 45 μαθητές Γ΄ Λυκείου. Η δραστηριότητα επικεντρώθηκε στη μελέτη του αλγόριθμου ταξινόμησης ευθείας ανταλλαγής, ευρύτερα γνωστού και ως αλγόριθμου φυσαλίδας. Τα αποτελέσματα έδειξαν ότι οι δυνατότητες πειραματισμού με την οπτικοποίηση της εκτέλεσης του αλγορίθμου, ενθάρρυναν τη συμμετοχή και την ενεργοποίηση των μαθητών κατά την επίλυση προβλημάτων ταξινόμησης πινάκων αυξημένης δυσκολίας. Έδωσαν δε, σημαντικές πληροφορίες για τη συμβολή του λογισμικού στην εξέλιξη της αλγοριθμικής σκέψης των μαθητών με στόχο την οικοδόμηση αλγόριθμων ταξινόμησης.

Ο αλγόριθμος ταξινόμησης ευθείας ανταλλαγής

Η ταξινόμηση αποτελεί την πρώτη ενότητα αλγορίθμων με την οποία έρχονται σε επαφή οι αρχάριοι προγραμματιστές. Αποτελεί ένα από τα θεμελιώδη αλγοριθμικά προβλήματα που συχνά χρησιμοποιείται για την εισαγωγή μαθητών ή φοιτητών σε τεχνικές σχεδίασης και ανάλυσης αλγορίθμων. Παρότι υπάρχει μεγάλη ποικιλία αλγόριθμων ταξινόμησης, ο αλγόριθμος ευθείας ανταλλαγής (straight exchange sort), που είναι ευρύτερα γνωστός και ως αλγόριθμος ταξινόμησης φυσαλίδας (bubble sort), είναι από τους πιο δημοφιλείς αλγόριθμους ταξινόμησης στην εκπαίδευση. Βασίζεται στην σύγκριση και αντιμετάθεση γειτονικών στοιχείων ενός πίνακα μέχρις ότου τα στοιχεία του πίνακα να διαταχθούν στην επιθυμητή σειρά. Διδάσκεται δε, στα πλαίσια του μαθήματος Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον της τεχνολογικής κατεύθυνσης Γ΄ Λυκείου.

Η έρευνα έχει δείξει ότι οι μαθητές αντιμετωπίζουν αρκετές δυσκολίες, όχι τόσο στην χρήση του αλγόριθμου, αφού τον χρησιμοποιούν σαν μαύρο κουτί, όσο στη βαθύτερη κατανόησή του (Nieminen, 2006; 2008; Geller & Dios, 1998; Astrachan et al., 2003). Οι δυσκολίες αυτές οφείλονται σε δυο λόγους: α) στο γεγονός ότι ο αλγόριθμος φυσαλίδας δεν προκύπτει άμεσα από την εμπειρία, β) σε αντίθεση με τους αλγόριθμους επιλογής (selection sort) και εισαγωγής (insertion sort), δεν μπορεί να χτιστεί εύκολα σε προϋπάρχουσα γνώση (Nieminen, 2006; 2008; Geller & Dios, 1998; Simon et al., 2006; Hadas, 2013). Για παράδειγμα, ο αλγόριθμος επιλογής αποτελεί μια επαναλαμβανόμενη εύρεση του ελάχιστου στοιχείου του πίνακα, έναν αλγόριθμο ο οποίος θεωρείται απλούστερος για τους μαθητές και μπορεί να αναπαρασταθεί ακόμη και με φυσικά αντικείμενα.



Σχήμα 1. Οθόνη δυναμικής οπτικοποίησης του αλγορίθμου συγχώνευσης

Το λογισμικό DAVE

Το περιβάλλον οπτικοποίησης αλγορίθμων DAVE υποστηρίζει όλους τους γνωστούς αλγορίθμους ταξινόμησης και αναζήτησης σε πίνακες (Vrachnos & Jimoyiannis, 2014). Στο Σχήμα 1 φαίνεται μια οθόνη του λογισμικού με την οπτικοποίηση του αλγόριθμου συγχώνευσης πινάκων. Το παράθυρο εργασίας του DAVE χωρίζεται σε τρία μέρη: α) τον συντάκτη πηγαίου κώδικα του αλγόριθμου (αριστερά) σε ψευδογλώσσα, β) την περιοχή οπτικοποίησης (δεξιά) και γ) τον πίνακα ελέγχου (κάτω). Το σύστημα είναι σχεδιασμένο ώστε να υποστηρίζει τον πειραματισμό των μαθητών και τη διερευνητική μάθηση-οικοδόμηση αλγορίθμων. Ο χρήστης μπορεί να εισάγει τα δικά του δεδομένα, να ελέγξει την ταχύτητα της οπτικοποίησης ή να την σταματήσει σε συγκεκριμένο σημείο. Το σύστημα παρέχει τη δυνατότητα στον μαθητή να τροποποιήσει την κωδικοποίηση του αλγορίθμου, να πειραματιστεί με την οπτικοποίηση της εκτέλεσης του δικού του αλγορίθμου και να εντοπίσει πιθανά λάθη. Οι οπτικοποιήσεις που παράγονται είναι διαφορετικές για αλγορίθμους με διαφορετική λογική και εστιασμένες στα βασικά χαρακτηριστικά που αναδεικνύουν τη φιλοσοφία κάθε αλγορίθμου.

Οι μαθητές γίνονται ενεργά υποκείμενα της μάθησης και υποστηρίζονται με στόχο την κατανόηση της δυναμικής συμπεριφοράς του αλγορίθμου, την ανάδειξη των δικών τους αναπαραστάσεων για τον αλγόριθμο ή μέρη αυτού και την ανίχνευση των λογικών σφαλμάτων κατά την ανάπτυξη του αλγορίθμου. Σε τελική ανάλυση, οι μαθητές μαθαίνουν μέσα από διαδικασίες δοκιμής και άμεσης παρατήρησης του αποτελέσματος στην οθόνη του υπολογιστή. Η κατανόηση της λειτουργίας του αλγορίθμου βασίζεται στη διδακτική χρησιμότητα του λάθους, κατά τη διαδικασία της ανάπτυξης του αλγορίθμου από τους ίδιους τους μαθητές, και στην αξιοποίηση των διαφορών ανάμεσα στα αναμενόμενα και στα παρατηρούμενα αποτελέσματα, με στόχο την οικοδόμηση αποτελεσματικών αναπαραστάσεων για τις αλγοριθμικές δομές και τα προγραμματιστικά αντικείμενα.

Το λογισμικό είναι ανεξάρτητο υπολογιστικής πλατφόρμας αφού είναι εξολοκλήρου γραμμένο σε HTML5 και Javascript. Αρκεί λοιπόν ένας φυλλομετρητής (browser) για να το

χρησιμοποιήσουμε. Δεν χρειάζεται εγκατάσταση, ούτε μεταφόρτωση κάποιας βιβλιοθήκης όπως συμβαίνει για παράδειγμα για τα java applets, και μπορεί να εκτελεστεί και σε άλλες συσκευές όπως ταμπλέτες και smartphones.

Το πλαίσιο της έρευνας

Στόχοι της έρευνας

Ο βασικός στόχος της έρευνας ήταν να μελετήσουμε κατά πόσο το περιβάλλον δυναμικής οπτικοποίησης αλγορίθμων DAVE και τα χαρακτηριστικά που ενσωματώνει συμβάλλουν στη διερεύνηση της λειτουργίας των αλγορίθμων ταξινόμησης και στην οικοδόμηση της λογικής τους από τους μαθητές. Τα ερευνητικά ερωτήματα που τέθηκαν ήταν τα εξής:

- σε ποιο βαθμό το λογισμικό οπτικοποίησης DAVE ενισχύει την εμπλοκή των μαθητών κατά την επίλυση αλγοριθμικών προβλημάτων ταξινόμησης;
- πώς συμβάλλει το λογισμικό οπτικοποίησης ώστε οι μαθητές να οικοδομήσουν επαρκείς αναπαραστάσεις για τη λειτουργία των αλγορίθμων ταξινόμησης;
- ποιες δυνατότητες του περιβάλλοντος αξιοποιούν οι μαθητές και ποιες πρακτικές αναπτύσσουν κατά την ενασχόλησή τους με την οπτικοποίηση αλγορίθμων;

Δείγμα

Η έρευνα διεξήχθη, κατά το σχολικό έτος 2012-2013, σε δύο γενικά λύκεια (7ο και 10ο) του δήμου Περιστερίου Αττικής. Το δείγμα περιελάβανε 45 μαθητές της Γ΄ τάξης της τεχνολογικής κατεύθυνσης που παρακολουθούσαν το μάθημα Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον (ΑΕΠΠ).

Μεθοδολογία

Η έρευνα έλαβε χώρα τον Μάρτιο του 2013, ένα μήνα μετά τη διδασκαλία της σχετικής ενότητας του μαθήματος που αναφέρεται στην ταξινόμηση της ευθείας ανταλλαγής ευρύτερα γνωστής και ως ταξινόμηση της φυσαλίδας (bubble sort). Διεξήχθη στο εργαστήριο Πληροφορικής, σε ομάδες των 11-12 μαθητών κάθε φορά, έτσι ώστε να αντιστοιχεί ένας Η/Υ ανά μαθητή. Διατέθηκαν δύο συνεχόμενες διδακτικές ώρες, κατά τις οποίες ήταν παρών και ο καθηγητής πληροφορικής κάθε τμήματος.

Φύλλο εργασίας

Οι μαθητές κλήθηκαν να απαντήσουν σε μια σειρά από ερωτήσεις και να επιλύσουν προβλήματα ταξινόμησης πινάκων, έχοντας τη δυνατότητα να χρησιμοποιήσουν παράλληλα το εκπαιδευτικό περιβάλλον προγραμματισμού DAVE. Για την υλοποίηση των δραστηριοτήτων χρησιμοποιήθηκε ειδικό φύλλο εργασίας, το οποίο σχεδιάστηκε στο πλαίσιο της παρούσας μελέτης. Το φύλλο εργασίας ήταν επώνυμο, ώστε να έχουμε τη δυνατότητα στη συνέχεια να διερευνήσουμε περαιτέρω τις απόψεις των μαθητών που θα είχαν ερευνητικό ενδιαφέρον (π.χ. μέσω συνεντεύξεων). Από τους μαθητές ζητήθηκε να παραθέσουν τις σκέψεις και τις προσεγγίσεις τους σχετικά με τη λειτουργία μικρών τμημάτων αλγορίθμων που δόθηκαν στις δραστηριότητες του φύλλου εργασίας.

Ανάλυση των αποτελεσμάτων

Στην παράγραφο αυτή παρουσιάζεται η ανάλυση των απαντήσεων που δόθηκαν από τους μαθητές στις δραστηριότητες της έρευνας. Λόγω έλλειψης χώρου, παρουσιάζουμε τρεις

ενδεικτικές δραστηριότητες-προβλήματα, με τα οποία ασχολήθηκαν οι μαθητές του δείγματος, καθώς και τα σημαντικότερα αποτελέσματα.

Δραστηριότητα 1

Η πρώτη δραστηριότητα αποτελείται από 6 βήματα (υπο-δραστηριότητες) με διττό σκοπό. Αφενός, να εξοικειωθούν οι μαθητές με το λογισμικό και, αφετέρου, να ελεγχθούν κάποιες βασικές γνώσεις των μαθητών πάνω στον αλγόριθμο ταξινόμησης, χρησιμοποιώντας απλές ερωτήσεις ώστε να τονωθεί η αυτοπεποίθησή τους και να ενθαρρυνθούν να συμμετέχουν. Στη συνέχεια αναλύουμε τα τρία τελευταία ερωτήματα της δραστηριότητας, με τα οποία ζητήθηκε από τους μαθητές να τροποποιήσουν τον αλγόριθμο που τους δόθηκε.

Βήμα 1

Αλγόριθμος	**Ερωτήσεις**
Για i από 2 μέχρι 2 Για j από N μέχρι i με βήμα –1 Αν A[j] < A[j – 1] Τότε Αντιμετάθεσε A[j], A[j – 1] Τέλος_Αν Τέλος_Επανάληψης Τέλος_Επανάληψης	Να τροποποιήσετε τον αλγόριθμο έτσι ώστε στην πρώτη θέση του πίνακα να μεταφερθεί ο μεγαλύτερος αριθμός. Να αιτιολογήσετε την απάντησή σας.

40 μαθητές έδωσαν σωστή απάντηση τροποποιώντας το συγκριτικό τελεστή από '<' σε '>'. Μόνο 6 από αυτούς αιτιολόγησαν την απάντησή τους γράφοντας *"πρέπει ο επόμενος να είναι μεγαλύτερος από τον προηγούμενο"*. Ενδιαφέρον παρουσιάζει το γεγονός ότι 2 μαθητές δεν άλλαξαν τη φορά της ανίσωσης αλλά τους δείκτες των στοιχείων του πίνακα αλλάζοντας τη συνθήκη από A[j] < A[j–1] σε A[j] < A[j+1]. Αυτό, στη γενική περίπτωση είναι σωστό, αλλά στις οριακές τιμές έχουμε προσπέλαση πέρα από τα όρια του πίνακα. Οι μαθητές εντόπισαν το λάθος, αφού το DAVE εμφανίζει μήνυμα για αυτή την περίπτωση, και διόρθωσαν κατάλληλα τα όρια των επαναληπτικών δομών, όπως φαίνεται παρακάτω :

Προσπέλαση πέρα από τα όρια του πίνακα	**Διορθωμένος αλγόριθμος**
Για i από 2 μέχρι 2 Για j από N μέχρι i με βήμα –1 Αν A[j] < A[j + 1] Τότε Αντιμετάθεσε A[j], A[j + 1] Τέλος_Αν Τέλος_Επανάληψης Τέλος_Επανάληψης	Για i από 2 μέχρι 2 Για j από **N–1** μέχρι **i–1** με βήμα –1 Αν A[j] < A[j + 1] Τότε Αντιμετάθεσε A[j], A[j + 1] Τέλος_Αν Τέλος_Επανάληψης Τέλος_Επανάληψης

Βήμα 2

Να τροποποιήσετε τον αλγόριθμο έτσι ώστε στις 3 πρώτες θέσεις του πίνακα να βρεθούν οι 3 μεγαλύτεροι αριθμοί. Να αιτιολογήσετε την απάντησή σας.

Στο ερώτημα αυτό 14 μαθητές έδωσαν σωστή απάντηση τροποποιώντας την εξωτερική επανάληψη (*Για i από 2 μέχρι 4*). 10 μαθητές έδωσαν πλήρη αλγόριθμο ταξινόμησης, ο οποίος

ταξινομεί όλα τα στοιχεία και όχι μόνο τα 3 πρώτα (*Για i από 2 μέχρι N*) ενώ 6 μαθητές απάντησαν ότι αρκούν 3 επαναλήψεις χωρίς να δώσουν τμήμα κώδικα.

Άλλοι 4 μαθητές απάντησαν αφού πρώτα, πειραματίστηκαν με το λογισμικό. Δηλαδή έκαναν δοκιμές, τροποποιώντας τα όρια των επαναλήψεων μέχρι να καταλήξουν στη σωστή απάντηση. Άλλοι 4 μαθητές δήλωσαν ότι βοηθήθηκαν από το λογισμικό στον εντοπισμό ενός λάθους και έδωσαν τη λύση *Για i από 1 μέχρι 3*. Στην περίπτωση αυτή έχουμε προσπέλαση πέρα από τα όρια του πίνακα αφού για j = i = 1 έχουμε A[j–1] = A[i–1] = A[1–1] = A[0]. Το λάθος αυτό αναδεικνύεται γραφικά με την κίνηση του δείκτη j εκτός των ορίων του πίνακα, αλλά και με διαγνωστικό μήνυμα και διακοπή της εκτέλεση στο σημείο αυτό.

Βήμα 3

Τι πρέπει να αλλάξετε στον αλγόριθμο έτσι ώστε να ταξινομηθούν όλα τα στοιχεία του πίνακα σε φθίνουσα σειρά;

Το ερώτημα αυτό είναι συνέχεια του προηγούμενου. Οι μαθητές πρέπει να αλλάξουν και το πλήθος των επαναλήψεων, ώστε να ταξινομούνται όλα τα στοιχεία του πίνακα και όχι μόνο τα τρία πρώτα. Την απάντηση (*Για i από 2 μέχρι 8*) έδωσαν 28 μαθητές ενώ 7 απάντησαν ότι "*πρέπει να εκτελεστεί 7 φορές*" χωρίς να δώσουν κωδικοποίηση. Τρεις μαθητές έδωσαν την απάντηση "*ο αλγόριθμος πρέπει να εκτελεστεί 9 φορές*". Όταν ζητήθηκε να διευκρινίσουν αυτό που εννοούσαν, έδωσαν το τμήμα κώδικα που εκτελεί την ταξινόμηση αλλά κάνει μια επανάληψη παραπάνω (*Για i από 2 μέχρι 9*). Εδώ αναδείχθηκε η παρανόηση ότι η συγκεκριμένη επανάληψη εκτελείται 9 φορές, επιβεβαιώνοντας τα ευρήματα άλλων ερευνών (Du Boulay, 1986; Robins et al., 2003; Dale 2006) ότι οι παρανοήσεις στις δομές επανάληψης συνιστούν παράγοντα δυσκολιών στην κατανόηση πιο σύνθετων δομών. Όταν οι μαθητές αυτοί πειραματιστήκαν με το λογισμικό, εντόπισαν ότι υπάρχει προσπέλαση πέρα από τα όρια του πίνακα αφού έλαβαν σχετικό διαγνωστικό μήνυμα.

Δραστηριότητα 2

Εισάγετε πίνακα με τα εξής πέντε στοιχεία: 1 4 3 2 5. Πόσα περάσματα θα χρειαστούν ώστε να ταξινομηθεί ο πίνακας σε αύξουσα σειρά; Πόσες αντιμεταθέσεις θα γίνουν; Μπορείτε να προτείνετε έναν τρόπο ταξινόμησης του πίνακα με πολύ λιγότερες αντιμεταθέσεις; Να αιτιολογήσετε την απάντησή σας.

Από τους 45 μαθητές, 25 απάντησαν ότι αρκούν 2 περάσματα με 3 αντιμεταθέσεις, κάτι το οποίο επιβεβαίωσαν εκτελώντας τον αλγόριθμο με τη βοήθεια του λογισμικού. Ενδιαφέρον παρουσιάζει η προσέγγιση εννέα μαθητών, οι οποίοι παρατήρησαν ότι αρκεί να συγκριθούν τα στοιχεία που βρίσκονται στις ζυγές θέσεις και έδωσαν τον παρακάτω αλγόριθμο:

```
Για i από 2 μέχρι 6 με βήμα 2
   Για j από 6 μέχρι i με βήμα –2
        Αν A[ j ] < A[ j – 2 ] Τότε Αντιμετάθεσε A[ j ], A[ j – 2 ]
   Τέλος_Επανάληψης
Τέλος_Επανάληψης
```

Όταν όμως εκτέλεσαν τον αλγόριθμο, το λογισμικό εντόπισε ότι για i=j=2 υπήρχε προσπέλαση πέρα από τα όρια του πίνακα στο στοιχείο A[j–2]=A[0]. Το λάθος αυτό εντοπίστηκε εύκολα από τους μαθητές, καθώς α) ο αντίστοιχος δείκτης (j) μετακινήθηκε πέρα από τα όρια του πίνακα και β) το σύστημα εμφάνισε σχετικό διαγνωστικό μήνυμα. Στη συνέχεια οι μαθητές τροποποίησαν τον αλγόριθμό τους όπως φαίνεται στην 1η προσέγγιση.

1η προσέγγιση	**2η προσέγγιση**

Για i από 2 μέχρι 6 με βήμα 2
  Για j από 6 μέχρι i+1 με βήμα –2
  Αν A[j]<A[j–2] Τότε Αντιμετάθεσε A[j], A[j–2]
  Τέλος_Επανάληψης
Τέλος_Επανάληψης
	Για i από 2 μέχρι 2
	  Για j από 3 μέχρι 3 με βήμα –1
	  Αν A[j]<A[j–2] Τότε Αντιμετάθεσε A[j], A[j–2]
	  Τέλος_Επανάληψης
	Τέλος_Επανάληψης

Τρεις (3) μαθητές έδωσαν απαντήσεις όπως στη 2η προσέγγιση. Το λογισμικό ανίχνευσε το λογικό λάθος και εμφάνισε μήνυμα ότι τα στοιχεία που αντιμετατίθενται πρέπει να είναι αυτά που εμφανίζονται και στη συνθήκη σύγκρισης. Αυτό σημειώθηκε από τους μαθητές οι οποίοι έκαναν αλλαγές στον αλγόριθμο. Ωστόσο, 24 μαθητές δεν απάντησαν στο τελευταίο ερώτημα που αφορούσε την βελτιστοποίηση του αλγορίθμου.

Δραστηριότητα 3

Εισάγετε τον πίνακα 7 στοιχείων 32 38 98 54 60 90 20
Να σχεδιάσετε τον αλγόριθμο ταξινόμησης της φυσαλίδας για την περίπτωση που η φυσαλίδα κατευθύνεται προς τα κάτω και όχι προς τα πάνω σε έναν πίνακα Α, Ν θέσεων. Δηλαδή ξεκινάει από τη θέση 1 και κινείται προς το τέλος του πίνακα όπως φαίνεται παρακάτω για έναν πίνακα 7 θέσεων.
Σημειώστε τον τρόπο σκέψης σας και τις δοκιμές ή τα λάθη που έγιναν μέχρι να καταλήξετε στον ζητούμενο αλγόριθμο.

1ο πέρασμα

1	32	38	38	38	38	38
2	38	32	98	98	98	98
3	98	98	32	54	54	54
4	54	54	54	32	60	60
5	60	60	60	60	32	90
6	90	90	90	90	90	32
7	20	20	20	20	20	20

2ο πέρασμα

38	98	98	98	98	98
98	38	54	54	54	54
54	54	38	60	60	60
60	60	60	38	90	90
90	90	90	90	38	38
32	32	32	32	32	32
20	20	20	20	20	20

Η δραστηριότητα αυτή είναι αυξημένης δυσκολίας, καθώς απαιτεί βαθιά κατανόηση του αλγόριθμου ταξινόμησης ευθείας ανταλλαγής. Μόνο 20 μαθητές από τους 45 του δείγματος ασχολήθηκαν με το πρόβλημα αυτό. Όλοι οι μαθητές που ασχολήθηκαν με το θέμα τροποποίησαν από την αρχή την εσωτερική επανάληψη της ταξινόμησης ευθείας ανταλλαγής, έτσι ώστε τα μικρότερα στοιχεία να κατευθύνονται προς τα κάτω, ως αντίστροφη φυσαλίδα (sinking sort). Από αυτούς, 16 έφτασαν στη σωστή απάντηση μετά από διαδοχικούς πειραματισμούς με το περιβάλλον οπτικοποίησης. Αρκετοί από αυτούς σημείωσαν στο φύλλο εργασίας ότι τους βοήθησε πολύ το γεγονός ότι, κατά την εκτέλεση του αλγορίθμου, το σύστημα εντοπίζει την προσπέλαση πέρα από τα όρια του πίνακα.

Τέσσερις μαθητές έφτασαν σε μια ημιτελή λύση από την οποία όμως φαινόταν ότι είχαν κατανοήσει τη λογική του αλγορίθμου. Τα λάθη τους εντοπίζονταν κυρίως σε λεπτομέρειες παρά στη βασική φιλοσοφία του αλγορίθμου που ήταν σωστή. Παρακάτω παρουσιάζουμε έναν χαρακτηριστικό τρόπο σκέψης, όπως τον αποτύπωσε μια μαθήτρια στο φύλλο εργασίας της. Στην αρχή έγραψε τον αλγόριθμο 1α. Αφού τον εκτέλεσε με το λογισμικό, διαπίστωσε ότι στην τελευταία επανάληψη ο πίνακας είναι ήδη ταξινομημένος και τροποποίησε τη λύση της με τον αλγόριθμο 2α. Είναι ενδιαφέρον το γεγονός ότι η τελική απάντηση είναι πιο απλή από αυτές που προηγήθηκαν.

Η μαθήτρια, όταν ρωτήθηκε για τον τρόπο εργασίας της, απάντησε ότι ξεκίνησε αρχικά με τον αλγόριθμο της ευθείας ανταλλαγής που γνώριζε, τον οποίο προσπάθησε να τροποποιήσει. Ανάλογη προσέγγιση ακολούθησαν και οι περισσότεροι μαθητές, οι οποίοι προσπάθησαν να χτίσουν πάνω σε προϋπάρχουσα γνώση, δηλαδή στον αλγόριθμο ευθείας ανταλλαγής. Στη συνέχεια όμως κατέληξαν να σχεδιάσουν τον αλγόριθμο από την αρχή. Όπως έγραψαν στο φύλλο εργασίας, τους βοήθησε πολύ η δυνατότητα εντοπισμού λογικών λαθών που παρείχε το λογισμικό είτε μέσω παρατήρησης της οπτικοποίησης είτε μέσω διαγνωστικών μηνυμάτων, ώστε να καταλήξουν στην τελική απάντησή τους.

Αλγόριθμος 1α	**Αλγόριθμος 2α**	**Αλγόριθμος 3α**
Για i από 2 μέχρι 3	Για i από 2 μέχρι 3	Για i από 1 μέχρι 6
Για j από i μέχρι N=7	Για j από i μέχρι N–1 = 6	Για j από 1 μέχρι 6
Αν A[j] > A[j – 1] Τότε	Αν A[j] > A[j – 1] Τότε	Αν A[j+1] > A[j] Τότε
Αντιμετάθεσε A[j], A[j–1]	Αντιμετάθεσε A[j], A[j–1]	Αντιμετάθεσε A[j], A[j+1]
Τέλος_αν	Τέλος_αν	Τέλος_αν
Τέλος_Επανάληψης	Τέλος_Επανάληψης	Τέλος_Επανάληψης
Τέλος_Επανάληψης	Τέλος_Επανάληψης	Τέλος_Επανάληψης

Στη συνέχεια παραθέτουμε τον τρόπο σκέψης και ενός άλλου μαθητή. Ο πρώτος αλγόριθμος (1β) που έδωσε ο μαθητής εκτελεί φθίνουσα ταξινόμηση των στοιχείων του πίνακα. Κατά την εκτέλεση του αλγορίθμου από το λογισμικό ο μαθητής παρατήρησε ότι τα μεγαλύτερα στοιχεία κινούνται προς τις πρώτες θέσεις του πίνακα και όχι τα μικρότερα προς τις τελευταίες. Αυτό τον οδήγησε στην δεύτερη έκδοση του αλγόριθμου (2β).

Αλγόριθμος 1β	**Αλγόριθμος 2β**
Για i από 2 μέχρι 7	Για i από 2 μέχρι 7
Για j από 7 μέχρι i με βήμα –1	Για j από i μέχρι 7
Αν A[j]>A[j–1] Τότε Αντιμετάθεσε A[j], A[j–1]	Αν A[j]>A[j–1] Τότε Αντιμετάθεσε A[j], A[j–1]
Τέλος_Επανάληψης	Τέλος_Επανάληψης
Τέλος_Επανάληψης	Τέλος_Επανάληψης

Εκτελώντας πάλι όμως τον αλγόριθμο, ο μαθητής παρατήρησε ότι στο τέλος τα στοιχεία δεν είναι ταξινομημένα (32 98 60 90 54 38 20). Αυτό τον οδήγησε στην τροποποίηση των οριακών τιμών της εσωτερικής επανάληψης όπως φαίνεται στον αλγόριθμο 3β. Κατά την εκτέλεση της οπτικοποίησης του νέου αλγορίθμου ο μαθητής παρατήρησε ότι το αμέσως μικρότερο στοιχείο, που κινείται προς τις τελευταίες θέσεις του πίνακα, σταματάει μόλις φτάσει στο ταξινομημένο τμήμα του. Όμως, οι συγκρίσεις συνεχίζονται και για τα τελευταία στοιχεία που είναι ήδη στις σωστές θέσεις. Τελικά ο μαθητής, αφού πειραματίστηκε με το λογισμικό οπτικοποίησης τροποποιώντας τις αρχικές και τελικές τιμές της επανάληψης, κατέληξε στην τελική του απάντηση, στον αλγόριθμο 4β.

Αλγόριθμος 3β	**Αλγόριθμος 4β**
Για i από 2 μέχρι 7	Για i από 2 μέχρι 7
Για j από 2 μέχρι 7	Για j από 2 μέχρι 9–i
Αν A[j]>A[j–1] Τότε Αντιμετάθεσε A[j], A[j–1]	Αν A[j]>A[j–1] Τότε Αντιμετάθεσε A[j], A[j–1]
Τέλος_Επανάληψης	Τέλος_Επανάληψης
Τέλος_Επανάληψης	Τέλος_Επανάληψης

Παρατηρήσεις - Σχόλια μαθητών

Το περιβάλλον οπτικοποίησης DAVE ενεργοποίησε σε μεγάλο βαθμό τους μαθητές, ενίσχυσε τη γνωστική εμπλοκή τους για την επίλυση αλγοριθμικών προβλημάτων μέσα από

τον πειραματισμό και τη διερεύνηση εναλλακτικών εκδοχών του αλγορίθμου, την αναθεώρηση και, τελικά, τη βελτίωση των δικών τους αλγορίθμων. Οι περισσότεροι μαθητές ασχολήθηκαν με όλες δραστηριότητες του φύλλου εργασίας, ακόμη και με τα πιο απαιτητικά ερωτήματα. Είναι χαρακτηριστικό ότι σε παρόμοια έρευνα για τη μελέτη των τρόπων επίλυσης αλγοριθμικών προβλημάτων, που έγινε με χαρτί και μολύβι, ελάχιστοι μαθητές ασχολήθηκαν με τα αντίστοιχα ερωτήματα (Βραχνός & Τζιμογιάννης 2014).

Στους μαθητές δόθηκε επίσης ειδικό φύλλο αξιολόγησης του περιβάλλοντος DAVE. Οι μαθητές, στην συντριπτική τους πλειονότητα έγραψαν ότι η ενασχόληση με το λογισμικό τους βοήθησε σημαντικά στο να κατανοήσουν τη λειτουργία του αλγορίθμου της ευθείας ανταλλαγής, και θεωρούν απαραίτητη τη χρήση του για το συγκεκριμένο μάθημα.

Ένα από τα βασικά σχόλια των μαθητών ήταν ότι το λογισμικό τους επέτρεψε να πειραματιστούν με τον αλγόριθμο που είχαν σχεδιάσει και με συνεχείς αλλαγές, να καταλήξουν στη σωστή απάντηση. Αρκετοί σημείωσαν ότι τους βοήθησε πολύ στην κατανόηση του αλγορίθμου επειδή έβλεπαν την εκτελούμενη εντολή και ακριβώς δίπλα το αποτέλεσμα της οπτικοποίησής της. Ένα άλλο σημείο που αξιολογήθηκε θετικά από τους μαθητές ήταν η κίνηση των μεταβλητών i, j και η επισήμανση που έκαναν ως δείκτες στα αντίστοιχα στοιχεία του πίνακα.

Από τις σημειώσεις παρατήρησης του ερευνητή προέκυψε ότι ελάχιστοι μαθητές έκαναν χρήση της βήμα-βήμα εκτέλεσης με στόχο να επικεντρωθούν σε συγκεκριμένα σημεία του αλγορίθμου. Συνήθως, οι μαθητές επέλεγαν να εκτελούν τον αλγόριθμο πολλές φορές και να μειώνουν την ταχύτητα της εκτέλεσης στα σημεία ενδιαφέροντος. Επίσης, κάποιοι μαθητές θα ήθελαν να υπάρχει δυνατότητα αντίστροφης (προς τα πίσω) εκτέλεσης, ώστε να εστιάζουν σε δύσκολα σημεία του αλγορίθμου χωρίς να εκτελούν κάθε φορά τον αλγόριθμο από την αρχή. Η αντίστροφη εκτέλεση του αλγορίθμου είναι μια δυνατότητα που αναμένεται να προστεθεί σε μελλοντική έκδοση του DAVE.

Συμπεράσματα

Τα αποτελέσματα της εργασίας αυτής επιβεβαιώνουν τα ευρήματα προηγούμενων ερευνών για τη συμβολή των περιβαλλόντων δυναμικής οπτικοποίησης αλγορίθμων στην ανάπτυξη προγραμματιστικών ικανοτήτων (Sorva et al., 2013; Urquiza–Fuentes et al., 2009). Η χρήση του λογισμικού DAVE ενίσχυσε την προσπάθεια των μαθητών του δείγματος για επίλυση αλγοριθμικών προβλημάτων με πίνακες, προωθώντας την ενεργό συμμετοχή και τον πειραματισμό τους με τις οπτικοποιήσεις αλγορίθμων. Οι μαθητές αξιοποίησαν σχεδόν όλες τις δυνατότητες του λογισμικού και, κυρίως, την τροποποίηση του κώδικα του αλγορίθμου και την αυξομείωση της ταχύτητας της εκτέλεσης της οπτικοποίησης. Τα δυναμικά χαρακτηριστικά του λογισμικού και οι δυνατότητες εντοπισμού λογικών λαθών που υποστηρίζει το λογισμικό, όπως η προσπέλαση πέρα από τα όρια του πίνακα, συνέβαλαν στην ανάδειξη λογικών σφαλμάτων και στην οικοδόμηση επαρκών αναπαραστάσεων των μαθητών για τις έννοιες του πίνακα, του δείκτη, της αντιμετάθεσης και της σύγκρισης στοιχείων.

Ακολουθώντας στρατηγικές διερευνητικής μάθησης, και αξιοποιώντας την προσομοίωση του αλγορίθμου μέσω του λογισμικού DAVE, οι μαθητές είχαν την ευκαιρία να δοκιμάσουν την δική τους εκδοχή του αλγορίθμου επίλυσης του προβλήματος και να πειραματιστούν με αυτήν, εντοπίζοντας τυχόν λογικά λάθη μέσα από την οπτικοποίηση. Η πειραματική καταγραφή των διαδοχικών βημάτων προσέγγισης ανέδειξε δυσκολίες, παρανοήσεις και μοντέλα σκέψης των μαθητών που έχουν μεγάλη διδακτική αξία. Η έρευνά μας συνεχίζεται σχετικά με αλγόριθμους αναζήτησης και συγχώνευσης πινάκων και αναμένεται να δώσει σημαντικά αποτελέσματα.

Αναφορές

Astrachan, O. (2003). Bubble sort: an archaeological algorithmic analysis. *Proceedings of the 34th SIGCSE Technical Symposium on Computer Science Education (SIGCSE '03)* (pp. 1-5). NY: ACM.

Dale, N. B. (2006). Most difficult topics in CS1: results of an online survey of educators. *SIGCSE Bulletin*, 38(2), 49-53.

Danielsiek, H. (2012). Detecting and understanding student's misconceptions related to algorithms and data structures. *Proceedings of the SIGCSE 2012 Technical Symposium on Computer Science Education* (pp. 21-26). Raleigh, North Carolina: ACM Press.

Du Boulay, B. (1986). Some Difficulties of Learning to Program. In E. Soloway & J. C. Spohrer (eds.), *Studying the Novice Programmer* (pp. 283-299). Hillsdale, NJ: Lawrence Erlbaum Associates.

Geller, J., & Dios, R. (1998). A low-tech, hands-on approach to teaching sorting algorithms to working students. *Computers & Education,* 31(1), 89-103.

Green, T. R. G. (1990). *Psychology of Programming*, London: Academic Press.

Hadas, L., R. (2013). A derivation-first approach to teaching algorithms. *Proceeding of the 44th ACM technical symposium on Computer science education (SIGCSE '13)* (pp. 573-578). NY: ACM.

Jimoyiannis, A. (2011). Using SOLO taxonomy to explore students' mental models of the programming variable and the assignment statement. *Themes in Science and Technology Education*, 4(2), 53-74.

Nieminen, J. (2006). *Bubble sort as the first sorting algorithm*. Retrieved 20 January 2014, from http://warp.povusers.org/grrr/bubblesort_eng.html

Nieminen, J. (2008). *Bubble sort misconceptions*. Retrieved 20 January 2014, from http://warp.povusers.org/grrr/bubblesort_misconceptions.html

Robins, A., Rountree, J., & Rountree, N. (2003). Learning and teaching programming: A review and discussion. *Computer Science Education*, 13(2), 137–172.

Simon, B., Chen, T.-Y., Lewandowski, G., McCartney, R., & Sanders, K. (2006). Commonsense computing: What students know before we teach (Episode 1): Sorting. *Proceedings of the 2nd International Workshop on Computing Education Research* (pp. 29-40). NY: ACM.

Soloway, E., & Spohrer, J. C. (1989) (eds.). *Studying the Novice Programmer*. Hillsdale, NJ: Lawrence Erlbaum.

Sorva, J., Karavirta, V., & Malmi, L. (2013). A review of generic program visualization systems of introductory programming education. *ACM Transactions on Computing Education*, 13(4). 15:1-15:64.

Urquiza–Fuentes, J., & Velazquez–Iturbide, J.A. (2009). A survey of successful evaluations of program visualization and algorithm animation systems. *ACM Transactions of Computing Education*, 9(2), 1-21.

Vrachnos, E., & Jimoyiannis, A. (2014). Design and evaluation of a web-based dynamic algorithm visualization environment for novices. *Procedia Computer Science*, 27, 229-239.

Βραχνός, Ε., & Τζιμογιάννης, Α. (2009). Εκπαιδευτικά περιβάλλοντα οπτικοποίησης αλγορίθμων: Μια επισκόπηση των τεχνικών και παιδαγωγικών χαρακτηριστικών. *Θέματα Επιστημών και Τεχνολογίας στην Εκπαίδευση*, 2(3), 215-245.

Βραχνός, Ε., & Τζιμογιάννης, Α. (2010). Μελέτη των αναπαραστάσεων μαθητών της Γ' Λυκείου για την έννοια του πίνακα χρησιμοποιώντας την ταξινομία SOLO. Στο Μ. Γρηγοριάδου (επιμ.), *Πρακτικά 5ου Πανελλήνιου Συνεδρίου ''Διδακτική της Πληροφορικής''* (σ. 81-90). Αθήνα

Βραχνός, Ε., & Τζιμογιάννης, Α. (2014). Αναπαραστάσεις μαθητών και φοιτητών για τον αλγόριθμο ταξινόμησης ευθείας ανταλλαγής: Μια ανάλυση βασισμένη στην ταξινομία SOLO. *Πρακτικά 7ου Πανελλήνιου Συνεδρίου ''Διδακτική της Πληροφορικής''*. Ρέθυμνο (υπό δημοσίευση)

Κόμης, Β., & Τζιμογιάννης, Α. (2006). Ο Προγραμματισμός ως μαθησιακή δραστηριότητα: από τις εμπειρικές προσεγγίσεις στη γνώση παιδαγωγικού περιεχομένου. *Θέματα στην Εκπαίδευση*, 7(3), 229-255.

Τζιμογιάννης, Α. (2005). Προς ένα παιδαγωγικό πλαίσιο διδασκαλίας του προγραμματισμού στη δευτεροβάθμια εκπαίδευση. Στο Α. Τζιμογιάννης (επιμ.), *Πρακτικά 3ου Πανελλήνιου Συνεδρίου ''Διδακτική της Πληροφορικής''* (σ. 99-111). Κόρινθος.