

Προϋπάρχουσες γνώσεις των μαθητών στους αλγόριθμους ταξινόμησης. Ποιον αλγόριθμο επινοούν;

Ευριπίδης Βραχνός, Αθανάσιος Τζιμογιάννης

evrachnos@gmail.com, ajimoyia@uop.gr

Τμήμα Κοινωνικής και Εκπαιδευτικής Πολιτικής, Πανεπιστήμιο Πελοποννήσου

Περίληψη

Η ταξινόμηση αποτελεί ένα από τα θεμελιώδη προβλήματα της επιστήμης της Πληροφορικής. Οι αλγόριθμοι ταξινόμησης χρησιμοποιούνται ως εισαγωγικά παραδείγματα σε μαθήματα ανάλυσης και σχεδίασης αλγορίθμων. Σύμφωνα με τη θεωρία του εποικοδομισμού η νέα γνώση χτίζεται πάνω σε προϋπάρχουσα γνώση ή στις εμπειρίες που έχουν οι μαθητές, οπότε η διερεύνηση των προϋπαρχουσών γνώσεων και αναπαραστάσεων των μαθητών είναι πολύ χρήσιμη για το σχεδιασμό διδακτικών στρατηγικών. Στην εργασία αυτή παρουσιάζουμε τα αποτελέσματα μιας έρευνας που έγινε σε μαθητές λυκείου στους οποίους δόθηκε ένα πρόβλημα ταξινόμησης και τους ζητήθηκε να καταγράψουν στο χαρτί τον αλγόριθμο με τον οποίο θα τοποθετήσουν ένα σύνολο αριθμών στη σειρά. Ο αλγόριθμος ευθείας ανταλλαγής που διδάσκεται στη Δευτεροβάθμια Εκπαίδευση χρησιμοποιήθηκε μόλις από το 4% των μαθητών σε αντίθεση με τον αλγόριθμο εισαγωγής που «επινόησε» η πλειοψηφία των μαθητών.

Λέξεις κλειδιά: αλγόριθμοι ταξινόμησης, προϋπάρχουσα γνώση, Διδακτική της Πληροφορικής

Εισαγωγή

Η διδασκαλία του προγραμματισμού υπολογιστών και της αλγοριθμικής σχεδίασης συνιστούν ένα έργο με ιδιαίτερες δυσκολίες. Πολλοί μαθητές, ακόμη και φοιτητές, δεν μπορούν να σχεδιάσουν ολοκληρωμένα και λογικά ορθά προγράμματα ακόμη και μετά από πολλά μαθήματα στον προγραμματισμό (De Raadt, 2007; Robins et al., 2003; Κόμης & Τζιμογιάννης, 2006; Τζιμογιάννης, 2005). Ένας από τους λόγους που οι αρχάριοι προγραμματιστές συναντούν τόσες δυσκολίες, είναι οι αφηρημένες αλγοριθμικές έννοιες που πρέπει να διαχειριστούν, οι οποίες έχουν μικρή σχέση με τα αντικείμενα της καθημερινότητας.

Σύμφωνα με την εποικοδομιστική προσέγγιση (Ben-Ari, 2001), οι μαθητές πριν ακόμα διδαχθούν κάποιο αντικείμενο διαθέτουν ήδη κάποιες γνώσεις ή εμπειρίες σχετικά με αυτό. Η μάθηση συνίσταται στην τροποποίηση ή επέκταση των γνώσεων και συνεπώς εξαρτάται άμεσα από τις προϋπάρχουσες γνώσεις. Ο σχεδιασμός των διδακτικών στρατηγικών που θα πρέπει να ακολουθηθούν έτσι ώστε να διευκολυνθούν οι μαθητές στην πρόσκτηση νέων γνώσεων θα πρέπει να λαμβάνει υπόψη του τις προϋπάρχουσες γνώσεις των μαθητών στο γνωστικό αντικείμενο της διδασκαλίας. Για να γίνει όμως αυτό θα πρέπει πρώτα να εντοπιστούν οι γνώσεις αυτές.

Στη βιβλιογραφία δεν υπάρχουν πολλές έρευνες σχετικά με τις προϋπάρχουσες γνώσεις των μαθητών και των φοιτητών σε έννοιες της Πληροφορικής. Οι περισσότερες εργασίες είναι σχετικά πρόσφατες και διερευνούν την προϋπάρχουσα γνώση που έχει κάποιος σε θέματα όπως ο καταναμημένος προγραμματισμός (Kolikant, 2001; Lewandowski, 2007) και η ορθότητα (Kolikant, 2005) και η πολυπλοκότητα (McCartney, 2009) των αλγορίθμων. Η ομάδα του “Common sense computing” διενήργησε μια έρευνα (Simon et al., 2006) για τη διερεύνηση των πρότερων γνώσεων των μαθητών σχετικά με τους αλγόριθμους ταξινόμησης. Η εργασία αυτή είναι πολύ σχετική με τη δική μας δουλειά, η οποία αφορά

την αναγνώριση και διερεύνηση των προϋπαρχουσών γνώσεων των μαθητών σχετικά με τους αλγόριθμους ταξινόμησης. Δηλαδή αν ζητήσουμε από τους μαθητές να βάλουν στη σειρά κάποια αντικείμενα ποια στρατηγική θα ακολουθήσουν; Ποιόν αλγόριθμο θα «επινοήσουν» για να λύσουν αυτό το πρόβλημα;

Η έρευνα έγινε σε ένα δείγμα 108 μαθητών λυκείου και τα αποτελέσματα παρουσιάζουν αρκετό ενδιαφέρον. Οι μαθητές επινόησαν πολλούς γνωστούς αλγορίθμους για να λύσουν το πρόβλημα της ταξινόμησης χωρίς να τους έχουν διδαχθεί. Η συντριπτική πλειοψηφία των μαθητών χρησιμοποίησε τους αλγορίθμους εισαγωγής και επιλογής για να ταξινομήσει μια σειρά αντικειμένων. Έλαχιστοι μαθητές χρησιμοποίησαν τον αλγόριθμο ευθείας ανταλλαγής που διδάσκεται στη Δευτεροβάθμια Εκπαίδευση, κάτι που δείχνει ότι ο αλγόριθμος αυτός δεν προκύπτει από τις καθημερινές εμπειρίες των μαθητών. Αυτό το αποτέλεσμα αποτελεί μια εξήγηση για τη δυσκολία που έχουν οι μαθητές στην κατανόηση της λειτουργίας του συγκεκριμένου αλγορίθμου ταξινόμησης (Astrachan, 2003; Βραχνός & Τζιμογιάννης 2014).

Τα αποτελέσματα της έρευνας θεωρούμε ότι είναι χρήσιμα για τη διάρθρωση του προγράμματος σπουδών εισαγωγικών μαθημάτων αλγορίθμων και προγραμματισμού στην ενότητα της ταξινόμησης, αφού μπορούν να χρησιμοποιηθούν για την επιλογή των κατάλληλων αλγορίθμων ταξινόμησης που μπορούν να διδαχθούν οι μαθητές.

Θεωρητικό πλαίσιο

Στη βιβλιογραφία υπάρχουν πολύ λίγες έρευνες σχετικά με τις προϋπαρχουσες γνώσεις των μαθητών σε αλγόριθμους ταξινόμησης. Οι Simon et. al. (2006) εκπόνησαν μια έρευνα αντίστοιχη με τη δική μας. Στη συγκεκριμένη εργασία οι ερευνητές ζήτησαν από τους φοιτητές να περιγράψουν τον τρόπο με τον οποίο θα ταξινομούσαν μια σειρά από 10 αριθμούς σε αύξουσα σειρά. Τα αποτελέσματα έδειξαν ότι οι περισσότεροι φοιτητές αντιμετώπισαν τους αριθμούς σαν συμβολοσειρές από ψηφία και έκαναν συγκρίσεις ψηφίο-ψηφίο, ομαδοποιώντας τους αριθμούς σε δεκάδες, εκατοντάδες κλπ., σαν οργάνωση ευρετηρίου. Η δεύτερη πιο δημοφιλής μέθοδος ήταν ο αλγόριθμος επιλογής (selection sort). Άλλοι αλγόριθμοι που χρησιμοποιήθηκαν από μαθητές ήταν ο αλγόριθμος εισαγωγής (insertion sort) και σε λίγες περιπτώσεις ο αλγόριθμος ευθείας ανταλλαγής (bubble sort).

Τα αποτελέσματα που θα παρουσιάσουμε στη συνέχεια είναι κάπως διαφορετικά από τα αποτελέσματα της έρευνας των Simon et al. (2006), στην οποία οι συγγραφείς δεν όρισαν το σύνολο των επιτρεπτών κινήσεων-εντολών που μπορούσαν να χρησιμοποιήσουν οι φοιτητές, δίνοντας τους έτσι απόλυτη ελευθερία κινήσεων.

Αποτέλεσμα αυτής της ελευθερίας που δόθηκε στους φοιτητές ήταν, πολλές από τις απαντήσεις να μη «δείχνουν» κάποιον γνωστό αλγόριθμο ταξινόμησης, ή κάποιον αλγόριθμο ταξινόμησης που να είναι αποδεκτός από το υπολογιστικό μοντέλο που χρησιμοποιούμε. Για παράδειγμα, ένας φοιτητής μπορεί να απαντήσει ότι με μια ματιά βρίσκει αμέσως το μικρότερο στοιχείο. Δεν μας λέει όμως με ποια διαδικασία έγινε αυτό, δηλαδή με ποια συγκεκριμένα βήματα, άρα δεν μπορούμε να ταυτοποιήσουμε με βεβαιότητα τον αλγόριθμο ταξινόμησης που χρησιμοποίησε.

Το σύνολο των επιτρεπτών εντολών που μπορούν να χρησιμοποιηθούν ορίζει τη νοητή μηχανή που προγραμματίζουμε για την επίλυση του προβλήματος. Η πρώτη αναφορά στην έννοια της νοητής μηχανής (notional machine) έγινε από τον Du Boulay (1986). Σύμφωνα με τον Du Boulay, η νοητή μηχανή περικλείει το σύνολο των γενικών χαρακτηριστικών που έχει η μηχανή στην οποία προγραμματίζουμε τη λύση ενός προβλήματος. Στη δική μας έρευνα η νοητή μηχανή την οποία προγραμματίζουν οι μαθητές προκύπτει έμμεσα από την εκφώνηση του προβλήματος που τους δόθηκε:

Σας δίνουν 20 κλειστούς φακέλους. Σε κάθε έναν από αυτούς υπάρχει ένα χαρτί με έναν αριθμό γραμμένο πάνω σε αυτό. Σας ζητείται να βάλετε τους φακέλους σε αύξουσα σειρά (δηλαδή από το μικρότερο στο μεγαλύτερο) με βάση τα νούμερα αυτά. Μπορείτε να ανοίξετε κάθε φάκελο όσες φορές θέλετε αλλά μπορείτε να έχετε ανοικτούς την ίδια στιγμή το πολύ δύο φακέλους και να βλέπετε τι έχουν μέσα. Μετά τους κλείνετε πάλι και τους τοποθετείτε όπου εσείς κρίνετε. Να περιγράψετε όσο πιο αναλυτικά μπορείτε τον αλγόριθμο με τον οποίο θα βάλετε τους φακέλους στη σειρά.

Στη δική μας νοητή μηχανή οι φάκελοι παίζουν το ρόλο των μεταβλητών και οι αριθμοί το ρόλο του περιεχομένου των μεταβλητών. Το άνοιγμα ενός φακέλου μοντελοποιεί την προσπέλαση στη μνήμη της αντίστοιχης μεταβλητής. Επειδή κάθε φορά μπορεί να συγκριθούν μόνο δύο στοιχεία, θέσαμε τον περιορισμό των δύο ανοικτών φακέλων για κάθε βήμα του αλγορίθμου, ώστε να υποχρεώσουμε τους μαθητές να συγκρίνουν μόνο δύο αντικείμενα κάθε φορά.

Οι βασικοί κανόνες-περιορισμοί που θέσαμε έμμεσα στους μαθητές είναι οι εξής:

- Την ίδια στιγμή μπορούν να έχουν ανοικτούς το πολύ δύο φακέλους
- Αφού δουν το περιεχόμενο ενός φακέλου τον ξανακλείνουν και τον τοποθετούν πάλι στη θέση του, εκτός αν θέλουν να τον μετακινήσουν σε άλλο σημείο.

Επίσης, επιτρέψαμε στους μαθητές να τοποθετούν τους φακέλους όπου ήθελαν στο τραπέζι. Αυτή η απόφαση αποδείχθηκε κρίσιμη διότι όπως θα δούμε στην ανάλυση των αποτελεσμάτων, πολλοί μαθητές επινόησαν αναδρομικούς αλγόριθμους όπως η Merge Sort και η Quicksort οι οποίοι χρειάζονται πολλούς πίνακες. Η υλοποίηση πάνω στον ίδιο πίνακα (on-the-fly) είναι αρκετά δύσκολη για αρχάριους προγραμματιστές και δεν προσφέρει κάτι σημαντικό στην κατανόηση της λειτουργίας του αλγορίθμου. Είναι θέμα βέλτιστης χρήσης της μνήμης. Επίσης οι on-the-fly αλγόριθμοι δεν είναι αλγόριθμοι που χρησιμοποιούνται στην καθημερινή ζωή συχνά, διότι στο φυσικό κόσμο δεν έχουμε τους περιορισμούς στη μνήμη που μας επιβάλλει η αρχιτεκτονική των υπολογιστών. Είναι συνηθισμένο όταν βάζουμε στη σειρά μια ομάδα από αντικείμενα να τοποθετούμε τα ταξινομημένα ξεχωριστά από τα υπόλοιπα και όχι δίπλα τους. Αυτό το είδαμε και στην παρούσα έρευνα. Η τοποθέτηση των ταξινομημένων αντικειμένων σε ξεχωριστή περιοχή υποδηλώνει τη χρήση δεύτερου πίνακα, αν προσπαθήσουμε να μεταφράσουμε το πρόγραμμα της νοητής μηχανής σε μια συμβατική γλώσσα προγραμματισμού.

Μεθοδολογία της έρευνας

Η έρευνα έλαβε χώρα τον Ιανουάριο του 2014 και συμμετείχαν 106 μαθητές του Ζάννειου Πειραματικού Λυκείου Πειραιά που είχαν επιλέξει το μάθημα της Πληροφορικής.

Πίνακας 1. Κατανομή των μαθητών του δείγματος

Τάξη	Μαθητές	Κατεύθυνση	Μαθητές
Α' Λυκείου	39		
Β' Λυκείου	35		
Γ' Λυκείου	32	{ Θεωρητική Θετική Τεχνολογική 1 Τεχνολογική 2	15
			10
			4
			3
Σύνολο	106		

Οι μαθητές της Α' και της Β' Λυκείου είχαν διδαχθεί κάποιες βασικές έννοιες προγραμματισμού όπως αυτές της μεταβλητής και της δομής επιλογής, αλλά δεν είχαν

ακόμα φτάσει στη δομή επανάληψης και στους πίνακες. Επίσης δεν είχαν έρθει σε επαφή με κανέναν αλγόριθμο ταξινόμησης. Το ίδιο ίσχυε και για όλους τους μαθητές της Γ' τάξης εκτός από τέσσερις (4) μαθητές του κύκλου Πληροφορικής και Υπηρεσιών της τεχνολογικής κατεύθυνσης οι οποίοι είχαν ήδη διδαχθεί τον αλγόριθμο ευθείας ανταλλαγής. Η κατανομή των μαθητών ανά τάξη και κατεύθυνση φαίνεται στον Πίνακα 1.

Στους μαθητές δόθηκε το πρόβλημα που αναφέρθηκε παραπάνω και τους ζητήθηκε να περιγράψουν τα βήματά του όσο πιο αναλυτικά γίνεται και όχι να δώσουν μόνο την ιδέα του αλγορίθμου.

Τα βασικά ερευνητικά ερωτήματα αυτής της έρευνας είναι τα παρακάτω:

- Πως αντιλαμβάνονται οι μαθητές τη διαδικασία της ταξινόμησης;
- Με ποια μέθοδο / αλγόριθμο ταξινομούν οι μαθητές μια σειρά από αντικείμενα;
- Πως περιγράφουν οι μαθητές τα βήματα ενός αλγορίθμου;
- Πως περιγράφουν/κωδικοποιούν μια επαναληπτική διαδικασία;

Ανάλυση των αποτελεσμάτων

Στον Πίνακα 2 παρουσιάζονται τα συγκεντρωτικά αποτελέσματα της έρευνας.

Πίνακας 2. Απαντήσεις των μαθητών ανά τάξη

α/α	Απάντηση	Κατανομή Απαντήσεων ανά τάξη			
		Α' Τάξη	Β' Τάξη	Γ' Τάξη	Σύνολο
1	Δεν απάντησαν	3	6		9
2	Ταξινόμηση Εισαγωγής	23	12	20	55
3	Ταξινόμηση Επιλογής	4	9	5	18
4	Ταξινόμηση Συγχώνευσης	3	6	3	12
5	Ταξινόμηση Φυσαλίδας			4	4
6	Γρήγορη Ταξινόμηση	3	2		5
7	Ταξινόμηση με χρήση Ευρετηρίου	3			3

Το 55% των μαθητών επινόησε τον αλγόριθμο ταξινόμησης εισαγωγής, το 18% τον αλγόριθμο επιλογής και το 12% τον αλγόριθμο συγχώνευσης. Τον αλγόριθμο της φυσαλίδας τον χρησιμοποίησαν μόλις 4 μαθητές από τους οποίους όμως οι 3 ήταν μαθητές της τεχνολογικής κατεύθυνσης οπότε τον είχαν διδαχθεί και αυτό φάνηκε από το γεγονός ότι έδωσαν αλγόριθμο με τη μορφή ψευδογλώσσας.

Εδώ πρέπει να σημειώσουμε ότι οι περισσότερες απαντήσεις των μαθητών εξετάστηκαν με αρκετή επεικεία όσον αφορά τη σαφήνεια και την αυστηρότητα διατύπωσης των αλγορίθμων που πρότειναν. Αυτό έγινε επειδή μας ενδιέφερε κυρίως να διακρίνουμε το είδος του αλγορίθμου και όχι να σταθούμε σε συντακτικά ή εκφραστικά λάθη. Με άλλα λόγια ήμασταν αρκετά ελαστικοί όσον αφορά την παραβίαση των κριτηρίων της αποτελεσματικότητας και της καθοριστικότητας των αλγορίθμων των μαθητών.

Ταξινόμηση εισαγωγής (Insertion sort)

Περισσότεροι από τους μισούς μαθητές χρησιμοποίησαν τον αλγόριθμο ταξινόμησης εισαγωγής. Παραθέτουμε και σχολιάζουμε κάποιες χαρακτηριστικές απαντήσεις μαθητών:

Παίρνω τους δύο πρώτους και τους βάζω στη σειρά. Στη συνέχεια παίρνω τον 3^ο και τον τοποθετώ στη σωστή θέση (πρώτο, τελευταίο ή ανάμεσα στους 2). Κάνω το ίδιο για τους υπόλοιπους αριθμούς τοποθετώντας κάθε φορά τον επόμενο στη σωστή θέση.

Η δομή αυτής της απάντησης είναι και η πιο συνηθισμένη που είδαμε. Δηλαδή οι μαθητές περιγράφουν τα 2-3 πρώτα βήματα και τελειώνουν με μια φράση όπως «κ.ο.κ», «ομοίως» κλπ. Δεν δίνουν δηλαδή τον αλγόριθμο στη μορφή που γνωρίζουμε χρησιμοποιώντας τις γνωστές δομές ακολουθίας, επιλογής και επανάληψης, αλλά με ελεύθερο κείμενο στο οποίο ωστόσο είναι διακριτή η βασική ιδέα του αλγορίθμου.

Επίσης είναι φανερό ότι η μαθήτρια που έδωσε την παραπάνω απάντηση έχει χωρίσει τους φακέλους σε δύο ομάδες, τους ταξινομημένους και τους μη ταξινομημένους. Κάθε φορά παίρνει έναν φάκελο από το αρχικό σύνολο και το τοποθετεί στη σωστή θέση μεταξύ των ταξινομημένων φακέλων, μέχρι να τους τοποθετήσει όλους. Δεν το αναφέρει ρητά γιατί μάλλον το θεωρεί αυτονόητο. Αυτό είναι πιο ξεκάθαρο στην παρακάτω απάντηση μαθητή:

Χωρίζω τους φακέλους σε δύο ομάδες, αυτούς που είναι ταξινομημένοι και αυτούς που δεν είναι. Κάθε φορά παίρνω έναν από το πάνω μέρος της στοιβας των μη ταξινομημένων και τον βάζω στη σωστή θέση στους ταξινομημένους.

Στην παραπάνω απάντηση φαίνεται ξεκάθαρα η τάση που έχουν οι μαθητές να ομαδοποιούν τα αντικείμενα κατά την ταξινόμηση σε δύο σύνολα, τα ταξινομημένα και τα μη ταξινομημένα. Βαθμιαία αφαιρούν ένα στοιχείο από το αρχικό σύνολο και το εισάγουν στο ήδη ταξινομημένο σύνολο αφού πρώτα βρουν τη θέση του.

Βάζω τον πρώτο φάκελο στο κέντρο. Αν ο επόμενος είναι μεγαλύτερος τον βάζω δεξιά και αν είναι μικρότερος αριστερά. Αφήνω αρκετή απόσταση ώστε να χωρέσουν και άλλοι φάκελοι αν υπάρχουν μεταξύ αυτών που έχω ήδη βάλει. Επαναλαμβάνω τη διαδικασία μέχρι να τελειώσουν οι φάκελοι.

Η παραπάνω προσέγγιση αποτελεί ένα ακόμα παράδειγμα του περιορισμού της εφευρετικότητας και της φαντασίας των μαθητών που επιτελείται από τη διδασκαλία του προγραμματισμού μέσα σε συγκεκριμένο πλαίσιο με αυστηρούς κανόνες. Οι μαθητές που δεν είχαν διδαχθεί προγραμματισμό παρουσίασαν πρωτότυπες λύσεις οι οποίες δεν περιορίζονται από τους κανόνες που θέτει μια εκπαιδευτική γλώσσα προγραμματισμού. Η ικανότητα των μαθητών να επανεφεύρουν τον τρόπο ήταν εντυπωσιακή.

Ο παραπάνω αλγόριθμος ενώ παραπέμπει στη βασική ιδέα του αλγορίθμου εισαγωγής δεν ταυτίζεται με αυτόν. Θα μπορούσαμε να πούμε ότι είναι μια παραλλαγή του η οποία είναι σχεδιασμένη για ένα φυσικό νοητό μοντέλο υπολογισμού, το οποίο περιλαμβάνει τις κινήσεις που μπορεί να κάνει κάποιος σε ένα τραπέζι με απεριόριστη επιφάνεια. Η μεταφορά αυτού του αλγορίθμου σε μια γλώσσα προγραμματισμού δεν είναι τόσο απλή υπόθεση, ειδικά αν δεν έχουμε αρκετή μνήμη.

Ταξινόμηση επιλογής (Selection sort)

Ο δεύτερος πιο δημοφιλής αλγόριθμος που επινοήθηκε από τους μαθητές για την επίλυση του προβλήματος της ταξινόμησης ήταν ο αλγόριθμος επιλογής (selection sort).

Εδώ θα πρέπει να σημειώσουμε ότι ενώ αναμέναμε ότι αυτός θα ήταν ο δημοφιλέστερος αλγόριθμος, οι μαθητές μας διέψευσαν και προτίμησαν τον αλγόριθμο εισαγωγής. Η υπόθεσή μας βασιζόταν στο γεγονός ότι ο αλγόριθμος επιλογής είναι ο απλούστερος αλγόριθμος αφού αφορά μια επαναλαμβανόμενη εύρεση μεγίστου κάτι που καθιστά την υλοποίησή του αρκετά απλή και κατανοητή. Αυτό όμως ισχύει για το υπολογιστικό μοντέλο του υπολογιστή το οποίο διαφέρει από τη νοητή μηχανή που περιγράφει τις φυσικές κινήσεις που μπορούν να κάνουν οι μαθητές με τα χέρια τους. Ακολουθεί υποδειγματική απάντηση μαθητή:

Ανοίγουμε 2 φακέλους και κρατάμε το φάκελο με τον μεγαλύτερο αριθμό, ενώ τον άλλο τον βάζουμε πίσω στη στοιβα με τους άλλους που έχουμε ανοίξει. Το κάνουμε αυτό μέχρι να ανοιχτούν όλοι οι φάκελοι και να βρεθούν στη στοιβα. Αυτός που μένει στο χέρι μας είναι ο μεγαλύτερος. Επαναλαμβάνουμε την ίδια διαδικασία και ο φάκελος που μένει στο χέρι μας είναι ο δεύτερος

μεγαλύτερος. Τον τοποθετούμε δίπλα στον πρώτο φάκελο. Επαναλαμβάνουμε άλλες 18 φορές. Στο τέλος όλοι οι φάκελοι θα είναι στη σειρά.

Η παραπάνω απάντηση περιγράφει αρκετά καλά τον αλγόριθμο επιλογής. Ωστόσο παρατηρούμε ότι και εδώ οι μαθητές αποθηκεύουν τους ταξινομημένους αριθμούς σε ξεχωριστή δομή. Αν για παράδειγμα υλοποιήσουμε τον παραπάνω αλγόριθμο με πίνακες θα χρειαστούμε δύο πίνακες. Έναν για τα αρχικά δεδομένα και έναν για τους ταξινομημένους αριθμούς.

Οι μαθητές έδωσαν και μερικές παραλλαγές της ταξινόμησης επιλογής. Σε μια από αυτές υπολογίζουν σε κάθε βήμα και το μέγιστο και το ελάχιστο στοιχείο του πίνακα και τοποθετούν το ένα στην αρχή και το άλλο στο τέλος του νέου πίνακα.

Αναδρομικοί αλγόριθμοι ταξινόμησης

Όταν σχεδιάζαμε αυτή την έρευνα δεν φανταζόμασταν ότι κάποιοι μαθητές θα πρότειναν αναδρομικούς αλγορίθμους όπως ο αλγόριθμος της συγχώνευσης και η γρήγορη ταξινόμηση. Η επιλογή που δόθηκε στους μαθητές να μπορούν να τοποθετούν τους φακέλους σε οποιοδήποτε σημείο του χώρου θέλουν, βοήθησε στην ανάπτυξη αναδρομικών αλγορίθμων ταξινόμησης οι οποίοι χρειάζονται πολλούς βοηθητικούς πίνακες.

Ταξινόμηση συγχώνευσης (Merge sort)

Οι περιγραφές των μαθητών που πρότειναν αυτόν τον αλγόριθμο δεν είναι ιδιαίτερα ακριβείς κυρίως λόγω της δυσκολίας περιγραφής του αναδρομικού βήματος και της συγχώνευσης των ταξινομημένων λιστών. Μια ενδεικτική απάντηση μαθητή είναι η παρακάτω:

Φτιάχνω 20 θέσεις στο χώρο τις οποίες χωρίζω στα δύο, δέκα δεξιά και δέκα αριστερά. Στη συνέχεια χωρίζω τους 10 πάλι σε ομάδες των 5 και τις πεντάδες σε δυάδες και τριάδες. Βάζω τις δυάδες και τις τριάδες στη σειρά και στη συνέχεια ενώνω αντίστροφα μέχρι να φτιάσω στις δύο δεκάδες οπότε ενώνω και αυτές.

Η παραπάνω περιγραφή παραπέμπει σε αλγόριθμο διαίρει και βασίλευε, αφού το πρόβλημα διαιρείται διαδοχικά σε απλούστερα προβλήματα. Η ιδέα μοιάζει αρκετά με αυτή της συγχώνευσης. Το μόνο πρόβλημα είναι η έλλειψη περιγραφής του βήματος της ένωσης των δύο ταξινομημένων λιστών. Αυτό που έχει σημασία είναι ότι ο μαθητής δείχνει να έχει καταλάβει πολύ καλά ότι η διαίρεση του προβλήματος οδηγεί σε απλούστερα προβλήματα.

Γρήγορη ταξινόμηση (QuickSort)

Η επινόηση της γρήγορης ταξινόμησης από κάποιους μαθητές ήταν αποτέλεσμα της σχετικής ελευθερίας κινήσεων που τους δόθηκε. Πέντε μαθητές (τρεις από την Α΄ Λυκείου) έδωσαν απαντήσεις οι οποίες έχουν αρκετά κοινά στοιχεία με το δημοφιλή αλγόριθμο της γρήγορης ταξινόμησης (QuickSort), με πιο χαρακτηριστικό το διαχωρισμό των αριθμών σε δύο λίστες με βάση την τιμή ενός στοιχείου του πίνακα. Οι αναδρομικές κλήσεις είναι πολύ δύσκολο να περιγραφούν σωστά από τους μαθητές και για αυτό είδαμε απαντήσεις όπως «κάνω το ίδιο και για τα δύο σύνολα», «επιλέγω πάλι έναν αριθμό κ.ο.κ.», «επαναλαμβάνω την ίδια διαδικασία» κλπ. Δηλαδή οι μαθητές προσπαθούν να περιγράψουν τα αναδρομικά βήματα με παρόμοιο τρόπο με τις επαναληπτικές διαδικασίες. Αυτό αποδεικνύει ότι οι μαθητές αντιλαμβάνονται ότι η αναδρομή συνδέεται με την έννοια της επαναληπτικής διαδικασίας (Rinderknecht, 2014). Ακολουθεί μια περιγραφή της Quicksort από μαθητή της Α΄ Λυκείου.

Παίρνω στην τύχη ένα φάκελο και το βάζω στη μέση. Μετά ανοίγω όλους τους φακέλους έναν-έναν και αν είναι μικρότεροι τους βάζω αριστερά ενώ αν είναι μεγαλύτεροι δεξιά. Στη συνέχεια από τους

μικρότερους ανοίγω πάλι ένα στην τύχη και επαναλαμβάνω την ίδια διαδικασία. Το ίδιο και για τους μεγαλύτερους.

Ταξινόμηση ευθείας ανταλλαγής (Φυσαλίδας)

Ο αλγόριθμος ταξινόμησης φυσαλίδας προτάθηκε μόνο από 4 μαθητές, από τους οποίους οι 3 ήταν μαθητές της τεχνολογικής κατεύθυνσης που το είχαν ήδη διδαχθεί. Ουσιαστικά μόνο 1 στους 106 μαθητές σκέφτηκε να κάνει ταξινόμηση με αυτόν τον τρόπο. Οι αλγόριθμοι των τριών μαθητών ήταν γραμμένοι στην ψευδογλώσσα του σχολικού βιβλίου. Παρακάτω δίνουμε την απάντηση της μαθήτριας της θετικής κατεύθυνσης όπως δόθηκε σε ελεύθερο κείμενο:

1. Τοποθετώ όλους τους φακέλους στη σειρά
2. Ανοίγω τον πρώτο και το δεύτερο φάκελο και τους συγκρίνω
3. Διαδικασία 1
*Αν ο πρώτος είναι μικρότερος ή ίσος από το δεύτερο τους αφήνω όπως ήταν
 Αν ο δεύτερος είναι μικρότερος τους αλλάζω θέση*
4. Ανοίγω το δεύτερο και τον τρίτο φάκελο
5. Επαναλαμβάνω τη Διαδικασία 1 μέχρι τον τελευταίο φάκελο μέχρι οι αριθμοί να είναι στη σειρά

Η περιγραφή της μαθήτριας μοιάζει αρκετά με φυσική γλώσσα κατά βήματα, αφού η παραπομπή στη Διαδικασία 1, δείχνει ουσιαστικά άλμα σε προηγούμενη εντολή.

Είναι φανερό λοιπόν ότι η ταξινόμηση φυσαλίδας δε συνάδει με τις αρχές του εποικοδομισμού, αφού είναι πολύ δύσκολο όχι μόνο να χτιστεί σε προϋπάρχουσα γνώση των μαθητών αλλά και να συνδεθεί με εμπειρίες τους από την καθημερινή ζωή, αφού σχεδόν κανείς δεν πρόκειται να ταξινομήσει ένα σύνολο αντικειμένων με τα χέρια του χρησιμοποιώντας αυτόν τον αλγόριθμο. Αυτός ίσως να είναι ο βασικότερος λόγος που οι μαθητές δυσκολεύονται τόσο πολύ στην κατανόηση της λειτουργίας του.

Συμπεράσματα

Σύμφωνα με τη θεωρία του εποικοδομισμού η νέα γνώση χτίζεται πάνω σε προϋπάρχουσα γνώση και στις εμπειρίες που έχουν οι μαθητές από την καθημερινότητά τους. Για αυτό η μελέτη της προϋπάρχουσας γνώσης των μαθητών για ένα αντικείμενο όπως οι αλγόριθμοι ταξινόμησης έχει πολύ μεγάλη σημασία για τις διδακτικές στρατηγικές που αναπτύσσουν οι διδάσκοντες.

Στους μαθητές δόθηκε ένα πρόβλημα το οποίο ήταν σχεδιασμένο με τέτοιο τρόπο ώστε οι επιτρεπτές κινήσεις που μπορούσαν να χρησιμοποιηθούν να είναι όσο το δυνατόν πιο κοντά στο μοντέλο υπολογισμού των περισσότερων γλωσσών προγραμματισμού χωρίς όμως να περιορίζεται η φαντασία των μαθητών.

Τα αποτελέσματα είχαν αρκετό ενδιαφέρον και εν μέρει επαλήθευσαν την αρχική μας υπόθεση ότι ο αλγόριθμος ευθείας ανταλλαγής, γνωστός και ως αλγόριθμος ταξινόμησης της φυσαλίδας, είναι μια από τις χειρότερες επιλογές για την εισαγωγή των μαθητών στους αλγορίθμους. Ο αλγόριθμος αυτός παρότι διδάσκεται στη Δευτεροβάθμια Εκπαίδευση χρησιμοποιήθηκε μόλις από το 4% των μαθητών σε αντίθεση με τους αλγορίθμους εισαγωγής και επιλογής που επινόησε η πλειοψηφία των μαθητών. Επίσης, πολλοί μαθητές χρησιμοποίησαν αναδρομικούς αλγορίθμους όπως η ταξινόμηση με συγχώνευση και η γρήγορη ταξινόμηση ως απόρροια της δυνατότητας να χρησιμοποιήσουν επιπλέον μνήμη.

Τα αποτελέσματα αυτά επαλήθευσαν την υπόθεση ότι ο αλγόριθμος της ευθείας ανταλλαγής δεν προκύπτει φυσικά από τις εμπειρίες των μαθητών οπότε δεν μπορεί να χτιστεί πάνω σε προϋπάρχουσα γνώση. Για αυτό κατά τη διδασκαλία του παρουσιάζονται πολλά

προβλήματα (Geller & Dios, 1998; Nieminen, 2006; 2008; Simon et al., 2006; Βραχνός & Τζιμογιάννης 2014). Επίσης, επιβεβαιώνεται η τάση των μαθητών να καταχωρούν τα ταξινομημένα στοιχεία σε μια νέα βοηθητική δομή ξεχωρίζοντας τα από τα μη ταξινομημένα, κάτι που ήταν πολύ εύκολο στο υπολογιστικό μοντέλο της νοητής μηχανής που ορίσαμε. Το μοντέλο αυτό επέτρεψε στους μαθητές να επινοήσουν αλγορίθμους όπως η γρήγορη ταξινόμηση και η ταξινόμηση συγχώνευσης, κάτι ομολογουμένως εντυπωσιακό.

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.
- Ben-Ari, M. (2001). Constructivism in computer science education. *Journal of Computers in Mathematics and Science Teaching*, 20(1), 45-73.
- de Raadt, M. (2007). A review of Australian investigations into problem solving and the novice programmer. *Computer Science Education*, 17(3), 201-213.
- 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.
- Kolikant, Y. (2001). Gardeners and cinema tickets: High schools' preconceptions of concurrency. *Computer Science Education*, 11(3), 221-245.
- Kolikant, Y. (2005). Students' alternative standards for correctness. In *Proceedings of the 1st International Computing Education Research Workshop* (pp. 37-43). New York: ACM.
- McCartney, R., Bouvier, D. J., Chen, T.-Y., Lewandowski, G., Sanders, K., Simon, B., & VanDeGrift, T. (2009). Commonsense computing (episode 5): algorithm efficiency and balloon testing. In *Proceedings of the fifth international workshop on Computing education research workshop (ICER '09)* (pp. 51-62), New York: ACM.
- Lewandowski, G., Bouvier, D. J., McCartney, R., Sanders, K., & Simon, B. (2007). Commonsense computing (episode 3): concurrency and concert tickets. In *Proceedings of the third international workshop on Computing education research workshop (ICER '07)* (pp. 51-62), New York: ACM.
- Rinderknecht, C. (2014). A Survey on Teaching and Learning Recursive Programming. *Informatics in Education*, 13(1), 87-119.
- 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.
- Βραχνός, Ε., & Τζιμογιάννης, Α. (2014). Αναπαραστάσεις μαθητών και φοιτητών για τον αλγόριθμο ταξινόμησης ευθείας ανταλλαγής: Μια ανάλυση βασισμένη στην ταξινομία SOLO. Στο Π. Αναστασιάδης, Ν. Ζαράνης, Β. Οικονομίδης & Μ. Καλογιαννάκης (επιμ.), *Πρακτικά 7ου Πανελληνίου Συνεδρίου «Διδακτική της Πληροφορικής»* (σ. 72-81), Ρέθυμνο: ΕΤΠΕ.
- 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 Πανελληνίου Συνεδρίου «Διδακτική της Πληροφορικής»* (σ. 99-111), Κόρινθος: ΕΤΠΕ.