

Συνέδρια της Ελληνικής Επιστημονικής Ένωσης Τεχνολογιών Πληροφορίας & Επικοινωνιών στην Εκπαίδευση

Τόμ. 1 (2012)

6ο Συνέδριο Διδακτική της Πληροφορικής



Πρόταση διδακτικής προσέγγισης του
αλγόριθμου της ταξινόμησης φυσαλίδας

Σ. Δουκάκης, Π. Γιαννοπούλου, Α. Πέρδος

Βιβλιογραφική αναφορά:

Δουκάκης Σ., Γιαννοπούλου Π., & Πέρδος Α. (2022). Πρόταση διδακτικής προσέγγισης του αλγόριθμου της ταξινόμησης φυσαλίδας. *Συνέδρια της Ελληνικής Επιστημονικής Ένωσης Τεχνολογιών Πληροφορίας & Επικοινωνιών στην Εκπαίδευση*, 1, 157-166. ανακτήθηκε από <https://eproceedings.epublishing.ekt.gr/index.php/cetpe/article/view/4435>

Πρόταση διδακτικής προσέγγισης του αλγόριθμου της ταξινόμησης φυσαλίδας

Σ. Δουκάκης¹, Π. Γιαννοπούλου², Α. Πέρδος³

¹Αμερικανικό Κολλέγιο Ελλάδος - PIERCE
sdoukakis@acg.edu

²Λεόντειο Λύκειο Πατησίων
gianagia@gmail.com

³Ελληνογαλλική Σχολή Καλαμαρί
perdos@kalamari.gr

Περίληψη

Ένα από τα βασικότερα ζητήματα που οι μαθητές/τριες και οι φοιτητές/τριες διαπραγματεύονται στα μαθήματα αλγοριθμικής και προγραμματισμού είναι η ταξινόμηση των στοιχείων ενός πίνακα. Ο αλγόριθμος της ταξινόμησης ευθείας ανταλλαγής ή ταξινόμησης φυσαλίδας αποτελεί έναν από τους πλέον διαδεδομένους αλγόριθμους που διδάσκονται στη δευτεροβάθμια και την τριτοβάθμια εκπαίδευση τόσο στη χώρα μας όσο και διεθνώς. Ωστόσο, φαίνεται ότι οι εκπαιδευόμενοι έχουν δυσκολίες στην περιγραφή και χρήση του αλγόριθμου, ενώ ορισμένοι εκπαιδευτικοί και ερευνητές θεωρούν ότι δεν είναι κατάλληλος ως εισαγωγικός αλγόριθμος για τη διδασκαλία της ταξινόμησης. Στην παρούσα εργασία παρουσιάζεται ένα εκπαιδευτικό-διδακτικό σενάριο για τη διδασκαλία του αλγόριθμου, η οποία αξιοποιεί προϋπάρχουσες γνώσεις των εκπαιδευόμενων, αναπτύσσοντας σκαλωσιά μάθησης που τους βοηθά να εξηγούν και να εφαρμόζουν τον αλγόριθμο.

Λέξεις κλειδιά: ταξινόμηση φυσαλίδας, ανάπτυξη εφαρμογών σε προγραμματιστικό περιβάλλον, αλγοριθμική.

Abstract

One of the fundamental issues in computer science is ordering a list of items. Bubble sort algorithm is one of the most commonly used sorting algorithms in programming courses around the world. However, it appears that students have difficulties to describe and use the algorithm. Moreover, some teachers and researchers believe that this algorithm is not the right algorithm as the very first example of a sorting algorithm in programming courses. This paper presents a teaching approach, which is based on previous student's knowledge and helps them to overcome learning difficulties, so as to be able to explain and use this algorithm.

Keywords: bubble sort, algorithm, applications development in a programming environment.

1. Εισαγωγή

Ο αλγόριθμος της ταξινόμησης ευθείας ανταλλαγής ή ταξινόμησης φυσαλίδας εμφανίζεται από τα μέσα του προηγούμενου αιώνα στην βιβλιογραφία (Astrachan, 2003). Ο αλγόριθμος έχει προκαλέσει διάφορες δυσκολίες στη μάθηση μαθητών και

πρωτοετών φοιτητών, αλλά παρόλα αυτά εισάγεται απευθείας στη σχολική γνώση (Kordaki et al., 2008) χωρίς να αποτελεί κατασκευή των ίδιων των μαθητών. Οι δυσκολίες αυτές, έχουν οδηγήσει σε επαναπροσδιορισμό του τρόπου διδασκαλίας του. Έτσι, έχει προταθεί ως τεχνική διδασκαλίας το παιχνίδι ρόλων (Μπούμπουκα κ.α., 2008), αλλά και η οπτικοποίηση του αλγόριθμου σε κατάλληλο περιβάλλον (Λαζαρίδης, 2005; Μπαλής, 2010). Στις επόμενες παραγράφους παρουσιάζεται μία πρόταση διδασκαλίας του αλγόριθμου της ταξινόμησης φυσαλίδας ώστε να τοποθετηθούν τα στοιχεία του πίνακα σε αύξουσα τάξη. Η πρόταση λαμβάνει υπόψη τις προϋπάρχουσες γνώσεις των μαθητών, τις οποίες αξιοποιεί ώστε να τους διευκολύνει να κατασκευάσουν τον σχετικό αλγόριθμο.

2. Πρόταση διδασκαλίας

Η μέθοδος της ταξινόμησης ευθείας ανταλλαγής ή ταξινόμησης φυσαλίδας βασίζεται στην αρχή της σύγκρισης και ανταλλαγής ζευγών γειτονικών στοιχείων, μέχρις ότου διαταχθούν όλα τα στοιχεία. Στη βιβλιογραφία εντοπίζονται διαφορετικές περιγραφές του αλγόριθμου ταξινόμησης φυσαλίδας. Έτσι, αναφέρεται ότι με τον αλγόριθμο: «τοποθετούνται τα μεγαλύτερα στοιχεία στο τέλος του πίνακα» (Knuth, 1998; Σιδερίδης κ.α., 2000; Sahni, 2004), ή ότι «τα μικρότερα στοιχεία ανεβαίνουν σαν φυσαλίδες στην αρχή του πίνακα» (Chang, 2003; Φωτάκης, 2006; Βογιατζής κ.α. 2009). Η δεύτερη προσέγγιση συνάδει με την περιγραφή του αλγόριθμου στο μάθημα Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον (ΑΕΠΠ) (Βακάλη κ.α., 2010) και αξιοποιείται στην παρούσα εργασία.

2.1 Δραστηριότητες στη δομή ακολουθίας, επιλογής και επανάληψης

Η διδακτική προσέγγιση της ταξινόμησης στοιχείων μπορεί να στηριχθεί σε μία σκαλωσιά μάθησης με τη διαπραγμάτευση εννοιών και αξιοποίηση στοχευμένων δραστηριοτήτων στις τρεις αλγοριθμικές συνιστώσες. Έτσι, οι μαθητές εργαζόμενοι στην ενότητα της δομής ακολουθίας μπορούν να εμπλακούν στη σύνταξη ενός αλγόριθμου ο οποίος αντιμετωπίζει το περιεχόμενο δύο μεταβλητών (Σχήμα 1.).

Να αναπτύξετε τμήμα αλγόριθμου το οποίο θα διαβάζει τις τιμές δύο μεταβλητών ίδιου τύπου, θα αντιμετωπίζει το περιεχόμενό τους και θα εκτυπώνει τις τιμές τους.	Διάβασε α, β Αντιμετάθεσε α, β Εκτύπωσε α, β	Βοηθητική $\leftarrow \alpha$ $\alpha \leftarrow \beta$ $\beta \leftarrow$ Βοηθητική
---	---	--

Σχήμα 1: Εντολή Αντιμετάθεση

Η εντολή Αντιμετάθεση α, β ανταλλάσσει το περιεχόμενο δύο θέσεων με τη βοήθεια μίας βοηθητικής θέσης (Βακάλη κ.α., 2010). Εναλλακτικά αυτό μπορεί να γίνει με τις τρεις εντολές που εμφανίζονται παραπάνω. Στην παρούσα εργασία, σε όλους τους αλγορίθμους αξιοποιείται για οικονομία χώρου η εντολή Αντιμετάθεση, παρότι θα μπορούσε να είχαν χρησιμοποιηθεί οι παραπάνω τρεις εντολές.

Στην ενότητα της δομής επιλογής οι μαθητές μπορούν να διαπραγματευτούν το θέμα της σύγκρισης των τιμών δύο μεταβλητών ίδιου τύπου (Σχήμα 2).

Να αναπτύξετε αλγόριθμο ο οποίος με δεδομένες τις τιμές δύο μεταβλητών α , β ίδιου τύπου, θα πραγματοποιεί τις απαραίτητες ενέργειες ώστε η μεταβλητή α να έχει την μικρότερη τιμή και η μεταβλητή β την άλλη.	Αλγόριθμος Α2 Δεδομένα // α , β // Αν $\alpha > \beta$ τότε Αντιμετάθεσε α , β Αποτελέσματα // α , β // Τέλος Α2
--	---

Σχήμα 2: Σύγκριση και αντιμετάθεση του περιεχομένου δύο μεταβλητών

Τέλος, στην ενότητα της δομής επανάληψης οι μαθητές μπορούν να διαπραγματευτούν κατάλληλες δραστηριότητες ώστε να αποδεχτούν ότι κατά τη χρήση των εντολών επανάληψης για γνωστό αριθμό επαναλήψεων δεν είναι απαραίτητο η τιμή του μετρητή να ξεκινά από 1 (Σχήμα 3).

Να αναπτύξετε τμήμα αλγόριθμου το οποίο θα διαβάζει τις τιμές 50 ζευγών μεταβλητών ίδιου τύπου, θα αντιμεταθέτει το περιεχόμενό τους και θα εκτυπώνει τις τιμές τους.	Για i από 1 μέχρι 50 Διάβασε α , β Αντιμετάθεσε α , β Εκτύπωσε α , β Τέλος επανάληψης	Για i από 2 μέχρι 51 Διάβασε α , β Αντιμετάθεσε α , β Εκτύπωσε α , β Τέλος επανάληψης
---	--	--

Σχήμα 3: Εντολή επανάληψης

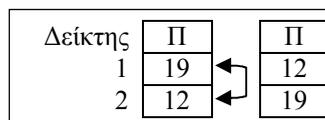
2.2 Δραστηριότητες με χρήση πίνακα

Ο αλγόριθμος Α2 που αποτέλεσε μία στοχευμένη δραστηριότητα στη δομή επιλογής (Σχήμα 2) μπορεί να εφαρμοστεί σε πίνακα και να προσεγγιστεί για πρώτη φορά η έννοια της ταξινόμησης στοιχείων πίνακα (Σχήμα 4).

Να αναπτύξετε αλγόριθμο ο οποίος με δεδομένο έναν πίνακα με 2 στοιχεία, θα πραγματοποιεί τις απαραίτητες ενέργειες έτσι ώστε να τοποθετεί στην πρώτη θέση του πίνακα το μικρότερο στοιχείο.	Αλγόριθμος ΤΣ1 Δεδομένα // Π// Αν $\Pi[1] > \Pi[2]$ τότε Αντιμετάθεσε $\Pi[1]$, $\Pi[2]$ Αποτελέσματα // Π// Τέλος ΤΣ1
---	---

Σχήμα 4: Τοποθέτηση μικρότερου στοιχείου στην κορυφή πίνακα 2 στοιχείων

Μετά την ολοκλήρωση του παραπάνω αλγόριθμου έχει τοποθετηθεί στην πρώτη θέση του πίνακα το μικρότερο στοιχείο. Με τον σχεδιασμό ενός πίνακα στο χαρτί με τιμές 19 και 12 και την εικονική εκτέλεση του αλγόριθμου, οι μαθητές παρακολουθούν την πορεία του αλγόριθμου και εξηγούν την λειτουργία του (Σχήμα 5).



Σχήμα 5: Εικονική τοποθέτηση μικρότερου στοιχείου στην κορυφή πίνακα 2 στοιχείων

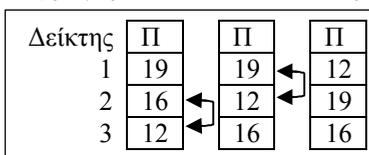
Η ίδια δραστηριότητα επαναλαμβάνεται με τρία στοιχεία. Όμως τώρα ζητείται η σύγκριση των στοιχείων ανά δυο και η τοποθέτηση του μικρότερου στην κορυφή. Οι μαθητές με το σχεδιασμό του πίνακα στο χαρτί και διερευνώντας τον τρόπο με τον οποίο θα αντιμετωπίσουν το πρόβλημα, θα διαπιστώσουν ότι χρειάζεται οι συγκρίσεις να ξεκινήσουν από το τελευταίο στοιχείο του πίνακα, ώστε να τοποθετηθεί στην πρώτη θέση του πίνακα το μικρότερο στοιχείο (Σχήμα 6).

Δίνεται ο πίνακας με στοιχεία 19, 16, 12. Να αναπτύξετε αλγόριθμο ο οποίος, θα πραγματοποιεί τις απαραίτητες ενέργειες έτσι ώστε να τοποθετεί στην πρώτη θέση του πίνακα το μικρότερο στοιχείο. Τα στοιχεία να συγκρίνονται ανά δυο.	Αλγόριθμος ΤΣ2 $\Pi[1] \leftarrow 19$ $\Pi[2] \leftarrow 16$ $\Pi[3] \leftarrow 12$ Αν $\Pi[3] < \Pi[2]$ τότε Αντιμετάθεσε $\Pi[2], \Pi[3]$ Αν $\Pi[2] < \Pi[1]$ τότε Αντιμετάθεσε $\Pi[1], \Pi[2]$ Αποτελέσματα // $\Pi //$ Τέλος ΤΣ2
---	---

Σχήμα 6: Τοποθέτηση μικρότερου στοιχείου στην κορυφή πίνακα 3 στοιχείων

Στον αλγόριθμο του σχήματος 6, προστέθηκε μία απλή εντολή επιλογής σε σχέση με τον αλγόριθμο ΤΣ1. Η νέα σύγκριση, η πρώτη δηλαδή εντολή Αν του αλγόριθμου ΤΣ2, συγκρίνει το τελευταίο στοιχείο με το προτελευταίο στοιχείο του πίνακα. Μετά την ολοκλήρωση του παραπάνω αλγόριθμου έχει τοποθετηθεί στην πρώτη θέση του πίνακα το μικρότερο στοιχείο.

Με τον σχεδιασμό του πίνακα στο χαρτί, οι μαθητές παρακολουθούν την πορεία του αλγόριθμου και εξηγούν την λειτουργία του. Επιπλέον, τους δίνεται η ευκαιρία να παρατηρήσουν ότι δεν έχουν τοποθετηθεί όλα τα στοιχεία του πίνακα σε σειρά διάταξης, αλλά η εικονική εκτέλεση του αλγόριθμου τοποθετεί το μικρότερο στοιχείο στην πρώτη θέση του πίνακα (η τιμή 19 τοποθετείται στην δεύτερη θέση) (Σχήμα 7).



Σχήμα 7: Εικονική τοποθέτηση μικρότερου στοιχείου στην κορυφή πίνακα 3 στοιχείων

Η δραστηριότητα αυτή μπορεί να επαναληφθεί για έναν πίνακα με 4 και 5 στοιχεία αντίστοιχα. Ο αλγόριθμος στις δύο αυτές περιπτώσεις, θα μπορούσε να αναπτυχθεί με παρόμοιο τρόπο. Αυτό που μπορούν να εντοπίσουν οι μαθητές από την προηγούμενη δραστηριότητα είναι ότι πρώτα συγκρίνονται τα δύο τελευταία στοιχεία του πίνακα και στη συνέχεια γίνεται η σύγκριση των υπόλοιπων στοιχείων και πάντα ανά ζεύγη. Έτσι, οι μαθητές θα μπορούσαν να προσδιορίσουν ότι απαιτείται η προσθήκη μίας ακόμα εντολής Αν, σε σχέση με τον προηγούμενο αλγόριθμο και να καταλήξουν μετά από διερεύνηση στον κατάλληλο αλγόριθμο για 4 στοιχεία (Αλγόριθμος ΤΣ3) και για 5 στοιχεία αντίστοιχα (Σχήμα 8). Η δραστηριότητα δίνει

την ευκαιρία στους μαθητές να γενικεύσουν τον αλγόριθμο. Οι μαθητές πιθανώς να προσδιορίσουν ότι η εντολή Αν επαναλαμβάνεται συγκεκριμένο αριθμό φορών ξεκινώντας από το τελευταίο στοιχείο προς το πρώτο με συνέπεια να αξιοποιήσουν την εντολή επανάληψης για να αναπτύξουν τον αλγόριθμο (Αλγόριθμος ΤΣ4) για πέντε στοιχεία (Σχήμα 8).

Αλγόριθμος ΤΣ3 Δεδομένα // Π // Αν Π[4] < Π[3] τότε Αντιμετάθεσε Π[3], Π[4] Αν Π[3] < Π[2] τότε Αντιμετάθεσε Π[2], Π[3] Αν Π[2] < Π[1] τότε Αντιμετάθεσε Π[1], Π[2] Αποτελέσματα // Π // Τέλος ΤΣ3	Αλγόριθμος ΤΣ4 Δεδομένα // Π // Για j από 5 μέχρι 2 με_βήμα -1 Αν Π[j] < Π[j - 1] τότε Αντιμετάθεσε Π[j - 1], Π[j] Τέλος_αν Τέλος_επανάληψης Αποτελέσματα // Π // Τέλος ΤΣ4
--	---

Σχήμα 8: Τοποθέτηση μικρότερου στοιχείου στην κορυφή πίνακα 4 και 5 στοιχείων

Στη συνέχεια ζητείται από τους μαθητές να εκτελέσουν εικονικά τον αλγόριθμο ΤΣ4, και να παρουσιάσουν τα αποτελέσματα με δεδομένο ότι ο πίνακας Π περιέχει τα ακόλουθα 5 στοιχεία 19, 12, 16, 14, 13. Με την εικονική εκτέλεση οι μαθητές μπορούν να παρατηρήσουν αυτά που έχουν ήδη διαπραγματευτεί στις προηγούμενες δραστηριότητες.

Η επόμενη δραστηριότητα μπορεί να υλοποιηθεί με διαφορετικό βαθμό καθοδήγησης των μαθητών. Έτσι, μπορεί να ζητηθεί από τους μαθητές να επαναλάβουν τον αλγόριθμο ΤΣ4 δύο φορές και στη συνέχεια τέσσερις φορές (Σχήμα 9).

Δίνεται πίνακας Π που περιέχει τα ακόλουθα 5 στοιχεία 19, 12, 16, 14, 13. Αν ο αλγόριθμος ΤΣ4, επαναληφθεί δύο φορές, δηλαδή αν έχετε το τμήμα αλγόριθμου που παρουσιάζεται στην διπλανή εικόνα τι πιστεύετε ότι θα συμβεί; Αν ο αλγόριθμος ΤΣ4 επαναληφθεί τέσσερις φορές, τι πιστεύετε ότι θα συμβεί;	Δεδομένα // Π // Για i από 1 μέχρι 2 Για j από 5 μέχρι 2 με_βήμα -1 Αν Π[j] < Π[j - 1] τότε Αντιμετάθεσε Π[j - 1], Π[j] Τέλος_αν Τέλος_επανάληψης Τέλος_επανάληψης
---	---

Σχήμα 9: Τοποθέτηση δύο μικρότερων στοιχείων στην κορυφή πίνακα 5 στοιχείων

Η δραστηριότητα μπορεί να γίνει περισσότερο διερευνητική, αν δεν προσδιοριστεί η επανάληψη του αλγόριθμου ΤΣ4 για τέσσερις φορές, αλλά ζητηθεί από τους μαθητές, αφού έχουν επαναλάβει τον παραπάνω αλγόριθμο δύο φορές, να προσδιορίσουν το πλήθος των επαναλήψεων του αλγόριθμου ΤΣ4 ώστε να τοποθετηθούν τα στοιχεία του πίνακα σε μία σειρά διάταξης από το μικρότερο στο μεγαλύτερο. Παρότι οι μαθητές με τις παραπάνω δραστηριότητες έχουν προσδιορίσει έναν αλγόριθμο για την ταξινόμηση κατά αύξουσα διάταξη των στοιχείων ενός πίνακα, είναι χρήσιμο να συζητηθούν ορισμένα επιπλέον ζητήματα.

2.3 Ο αλγόριθμος ταξινόμησης φουσαλίδας

Με την προαναφερθείσα προσέγγιση οι μαθητές είναι πιθανό να διερωτηθούν για ποιο λόγο/ους η εξωτερική εντολή επανάληψης εκτελείται 4 φορές στην περίπτωση που χρειάζεται να ταξινομηθούν τα 5 στοιχεία ενός πίνακα ή ακόμα σε ποιες περιπτώσεις μπορεί η εξωτερική επανάληψη να εκτελεστεί λιγότερες από 4 φορές. Από τις δραστηριότητες που έχουν ήδη υλοποιηθεί έχει φανεί ότι σε κάθε εκτέλεση της εξωτερικής επανάληψης ένα στοιχείο του πίνακα διατασσόταν σε σχέση με τα υπόλοιπα. Έτσι, για τα 5 στοιχεία του πίνακα χρειάστηκαν 4 περάσματα για να ταξινομηθούν όλα τα στοιχεία του πίνακα. Αφού, λοιπόν, η εξωτερική εντολή επανάληψης εκτελείται 4 φορές και με σκοπό να διευκολυνθούν οι μαθητές να ανακαλούν ευκολότερα τον αλγόριθμο της ταξινόμησης φουσαλίδας καθοδηγούνται να χρησιμοποιήσουν ως αρχική και τελική τιμή της εξωτερικής επανάληψης την τελική και αρχική τιμή της εσωτερικής επανάληψης αντίστοιχα. Στο σημείο αυτό ο σχεδιασμός της συνολικής δραστηριότητας είναι καθοδηγητικός, αλλά είναι χρήσιμη αυτή η προσέγγιση, αφού οι μαθητές έχουν ήδη επαληθεύσει τα όρια του εσωτερικού βρόχου. Έτσι, ο αλγόριθμος ΤΣ5 μπορεί να βελτιωθεί (Αλγόριθμος ΤΣ6), όπου η αρχική και η τελική τιμή της εξωτερικής εντολής επανάληψης έχουν τις ίδιες τιμές με την τελική και αρχική τιμή της εσωτερικής επανάληψης αντίστοιχα (Σχήμα 10).

<p>Αλγόριθμος ΤΣ5 Δεδομένα // Π // Για i από 1 μέχρι 4 Για j από 5 μέχρι 2 με_βήμα -1 Αν Π[j] < Π[j - 1] τότε Αντιμετάθεσε Π[j - 1], Π[j] Τέλος_αν Τέλος_επανάληψης Τέλος_επανάληψης Αποτελέσματα // Π //</p>	<p>Αλγόριθμος ΤΣ6 Δεδομένα // Π // Για i από 2 μέχρι 5 Για j από 5 μέχρι 2 με_βήμα -1 Αν Π[j] < Π[j - 1] τότε Αντιμετάθεσε Π[j - 1], Π[j] Τέλος_αν Τέλος_επανάληψης Τέλος_επανάληψης Αποτελέσματα // Π //</p>
Τέλος ΤΣ5	Τέλος ΤΣ6

Σχήμα 10: Αλγόριθμος ταξινόμησης φουσαλίδας για πίνακα 5 στοιχείων

Μία πρόσθετη «βελτίωση» του αλγόριθμου μπορεί να επιτευχθεί αν οι μαθητές είναι σε θέση να αναγνωρίσουν ότι μετά την ολοκλήρωση της εσωτερικής εντολής επανάληψης (δηλαδή από το πρώτο πέρασμα) το μικρότερο στοιχείο τοποθετήθηκε στην κορυφή και στη συνέχεια να υποστηρίξουν ότι είναι περιττός ο έλεγχος του πρώτου στοιχείου με τα υπόλοιπα στα επόμενα περάσματα. Ομοίως να προσδιορίσουν ότι στο δεύτερο πέρασμα, το 2ο μικρότερο στοιχείο τοποθετήθηκε στη δεύτερη θέση και είναι περιττός ο έλεγχος του πρώτου και δεύτερου στοιχείου με τα υπόλοιπα στα επόμενα περάσματα. Άρα, ο αλγόριθμος μπορεί να βελτιωθεί, ώστε να μην πραγματοποιεί περιττούς ελέγχους. Για να επιτευχθεί αυτό, χρειάζεται να καθοριστεί το πλήθος των επαναλήψεων της εσωτερικής επανάληψης από το πλήθος των στοιχείων που έχουν ήδη ταξινομηθεί. Με την χρήση του δείκτη της εξωτερικής

επανάληψης (i) μπορεί να επιτευχθεί η εξάλειψη των περιττών περασμάτων (ΤΣ7), όπου η μεταβλητή i αποτελεί την τελική τιμή της εσωτερικής επανάληψης. Οι μαθητές παροτρύνονται να εκτελέσουν εικονικά τον αλγόριθμο και να επαληθεύσουν αυτό που έχουν ήδη περιγράψει, δηλαδή ότι κάθε φορά που συγκρίνουν το κάτω στοιχείο με το πάνω, αν το j πάρει τιμή μικρότερη του i, τότε θα βρεθεί σε τμήμα του πίνακα το οποίο έχει ήδη ταξινομηθεί από τις προηγούμενες επαναλήψεις. Επίσης, είναι σημαντικό, οι μαθητές να γενικεύσουν τον αλγόριθμο (ΤΣ8) για N στοιχεία (Σχήμα 11).

<p>Αλγόριθμος ΤΣ7 Δεδομένα // Π // Για i από 2 μέχρι 5 Για j από 5 μέχρι i με_βήμα -1 Αν $\Pi[j] < \Pi[j - 1]$ τότε Αντιμετάθεσε $\Pi[j - 1], \Pi[j]$ Τέλος_αν Τέλος_επανάληψης Τέλος_επανάληψης Αποτελέσματα // Π // Τέλος ΤΣ7</p>	<p>Αλγόριθμος ΤΣ8 Δεδομένα // Π, N // Για i από 2 μέχρι N Για j από N μέχρι i με_βήμα -1 Αν $\Pi[j] < \Pi[j - 1]$ τότε Αντιμετάθεσε $\Pi[j - 1], \Pi[j]$ Τέλος_αν Τέλος_επανάληψης Τέλος_επανάληψης Αποτελέσματα // Π // Τέλος ΤΣ8</p>
--	---

Σχήμα 11: Ο αλγόριθμος ταξινόμησης φυσαλίδας για πίνακα N στοιχείων

Επιπλέον είναι χρήσιμο να προσδιορίσουν οι μαθητές τον τρόπο υλοποίησης του αλγόριθμου της ταξινόμησης των στοιχείων του πίνακα κατά φθίνουσα διάταξη (ΤΣ9), αλλά και να συσχετίσουν τον αλγόριθμο της ταξινόμησης φυσαλίδας με τον αλγόριθμο ταξινόμησης που τοποθετεί τη σωστή τιμή (μικρότερη ή μεγαλύτερη) στο τελευταίο στοιχείο του πίνακα (ΤΣ10), που φαίνονται στο σχήμα 12.

<p>Αλγόριθμος ΤΣ9 Δεδομένα // Π, N // Για i από 2 μέχρι N Για j από N μέχρι i με_βήμα -1 Αν $\Pi[j] > \Pi[j - 1]$ τότε Αντιμετάθεσε $\Pi[j - 1], \Pi[j]$ Τέλος_αν Τέλος_επανάληψης Τέλος_επανάληψης Αποτελέσματα // Π // Τέλος ΤΣ9</p>	<p>Αλγόριθμος ΤΣ10 Δεδομένα // Π, N // Για i από 1 μέχρι N - 1 Για j από 1 μέχρι N - i Αν $\Pi[j + 1] > \Pi[j]$ τότε Αντιμετάθεσε $\Pi[j], \Pi[j + 1]$ Τέλος_αν Τέλος_επανάληψης Τέλος_επανάληψης Αποτελέσματα // Π // Τέλος ΤΣ10</p>
---	--

Σχήμα 12: Αλγόριθμος ταξινόμησης φυσαλίδας σε φθίνουσα διάταξη

2.4 Οι βελτιώσεις του αλγόριθμου ταξινόμησης φουσαλίδας

Ένα από τα πλεονεκτήματα του συγκεκριμένου αλγόριθμου είναι οι δυνατότητες βελτίωσής του και παραλλαγής, με αποτέλεσμα να κρίνεται ως αλγόριθμος που προσφέρει σημαντικά μαθησιακά οφέλη (Liu et al., 2009; Fuguo et al., 2011).

Δύο τέτοιες παραλλαγές περιέχονται στο διδακτικό πακέτο (Βακάλη κ.α., 2010). Η πρώτη αποτελεί μία δραστηριότητα η οποία αναφέρει ότι: «Ο αλγόριθμος της φουσαλίδας όπως διατυπώθηκε στην προηγούμενη παράγραφο έχει το μειονέκτημα ότι δεν είναι αρκετά «έξυπνος» ώστε να διαπιστώνει στην αρχή ή στο μέσο της διαδικασίας αν ο πίνακας είναι ταξινομημένος» (Τετράδιο Μαθητή, Βακάλη κ.α. σ. 33, 2010). Με την δραστηριότητα αυτή, οι μαθητές μπορούν είτε μέσω κατάλληλων διερευνητικών ερωτήσεων του τύπου: «πώς μπορώ να βρω αν ο πίνακας είναι ταξινομημένος» ή/και «πώς μπορώ να σταματήσω τον αλγόριθμο ώστε να μην κάνει περιττές επαναλήψεις;» είτε με κατάλληλη υπόδειξη να τροποποιήσουν τον αλγόριθμο, ώστε να σταματά η διαδικασία σύγκρισης, όταν διαπιστωθεί ότι τα στοιχεία του πίνακα είναι ήδη ταξινομημένα (Σχήμα 13).

<p>Αλγόριθμος ΠΤΣ1 Δεδομένα // Π, N // $i \leftarrow 2$ Αρχή_επανάληψης EA ← Ψευδής ! EA: έγινε αντιμετάθεση Για j από N μέχρι i με_βήμα -1 Αν $\Pi[j] < \Pi[j - 1]$ τότε Αντιμετάθεσε $\Pi[j - 1], \Pi[j]$ EA ← Αληθής Τέλος_αν Τέλος_επανάληψης $i \leftarrow i + 1$ Μέχρις_ότου EA = Ψευδής ή $i > N$ Αποτελέσματα // Π // Τέλος ΠΤΣ1</p>	<p>Αλγόριθμος ΠΤΣ2 Δεδομένα // Π, N // $i \leftarrow 2$ EA ← Αληθής Όσο EA = Αληθής και $i \leq N$ επανάλαβε EA ← Ψευδής Για j από N μέχρι i με_βήμα -1 Αν $\Pi[j] < \Pi[j - 1]$ τότε Αντιμετάθεσε $\Pi[j - 1], \Pi[j]$ EA ← Αληθής Τέλος_αν Τέλος_επανάληψης $i \leftarrow i + 1$ Τέλος_επανάληψης Αποτελέσματα // Π // Τέλος ΠΤΣ2</p>
--	--

Σχήμα 13: Βελτιωμένος αλγόριθμος ταξινόμησης φουσαλίδας

Με τον παραπάνω αλγόριθμο, το ζητούμενο είναι να ελεγχθεί αν κατά την εκτέλεση της εσωτερικής επανάληψης, εκτελέστηκε η εντολή Αντιμετάθεση. Για το λόγο αυτό χρησιμοποιείται μία λογική μεταβλητή, η οποία ελέγχει αν θα εκτελεστεί η εντολή Αντιμετάθεση. Η τιμή της μεταβλητής υποδεικνύει αν ο πίνακας είναι ή δεν είναι ταξινομημένος. Η ανάπτυξη του σχετικού αλγόριθμου θα μπορούσε να υλοποιηθεί και με τον αλγόριθμο ΠΤΣ3 ή με τον αλγόριθμο ΠΤΣ4 (Σχήμα 14)

<p>Αλγόριθμος ΠΤΣ3 Δεδομένα // Π, N // Αρχή_επανάληψης EA ← Ψευδής Για i από 1 μέχρι N - 1 Αν Π[i+1] < Π[i] τότε Αντιμετάθεσε Π[i + 1], Π[i] EA ← Αληθής Τέλος_αν Τέλος_επανάληψης Μέχρις_ότου EA = Ψευδής Αποτελέσματα // Π // Τέλος ΠΤΣ3</p>	<p>Αλγόριθμος ΠΤΣ4 Δεδομένα // Π, N // N1 ← N Αρχή_επανάληψης Θ ← 0 Για i από 1 μέχρι N1 - 1 Αν Π[i+1] < Π[i] τότε Αντιμετάθεσε Π[i + 1], Π[i] Θ ← i Τέλος_αν Τέλος_επανάληψης N1 ← Θ Μέχρις_ότου Θ = 0 Αποτελέσματα // Π // Τέλος ΠΤΣ4</p>
---	--

Σχήμα 14: Βελτιώσεις του αλγόριθμου ταξινόμησης φυσαλίδας

Οι παραλλαγές του αλγόριθμου μπορούν να επεκταθούν με διάφορες βελτιώσεις που καταγράφονται στην βιβλιογραφία (Fuguo et al., 2011), αλλά και σε θέματα όπως των εξετάσεων του ημερησίου λυκείου για το σχολικό έτος 2009-2010.

3. Συμπεράσματα

Ο αλγόριθμος ταξινόμησης φυσαλίδας, αποτελεί έναν αλγόριθμο που διδάσκεται διαχρονικά στην εκπαίδευση. Παρότι έχουν δημοσιευτεί εργασίες που θεωρούν τον αλγόριθμο απαρχαιωμένο (Astrachan, 2003; Nieminen, 2005), ο αλγόριθμος επιδέχεται παραλλαγές και βελτιώσεις, με αποτέλεσμα η διδασκαλία του να προσφέρει μαθησιακά οφέλη. Επιπλέον, η σχεδίαση ενός διδακτικού-εκπαιδευτικού σεναρίου (Γρηγοριάδου κ.α., 2010) και η αξιοποίηση ψηφιακών εργαλείων στο πλαίσιο διδασκαλίας του αλγόριθμου (Στέργου, 2010), φαίνεται να έχει πρόσθετη διδακτική αξία στα ζητήματα της εικονικής εκτέλεσής του (Kordaki et al., 2008), της υλοποίησης παραλλαγών και τροποποιήσεων ώστε να ταξινομείται μέρος του πίνακα.

Η διδακτική προσέγγιση που προτάθηκε επιχειρεί να αμβλύνει τις δυσκολίες διδασκαλίας και αξιοποίησης του αλγόριθμου, οικοδομώντας την διδασκαλία του αλγόριθμου σε προϋπάρχουσες γνώσεις και προηγούμενες έννοιες, ώστε οι μαθητές/τριες να είναι σε θέση να τον περιγράψουν, να τον εφαρμόζουν και να τον επαληθεύουν.

Βιβλιογραφία

Astrachan, O. (2003). Bubble sort: an archaeological algorithmic analysis. In *Proceedings of the 34th SIGCSE technical symposium on Computer science education (SIGCSE '03)*. ACM, New York, NY, USA, 1-5.

- Chang, S.K. (2003). Data structures and algorithms, World Scientific.
- Fuguo, D., Lan, W., & Da, Y. (2011). Overview of the study on bubble sort algorithm, *International Journal of Advancements in Computing Technology*, 3(5), 1-7.
- Knuth, D. (1998). The Art of Computer Programming: Sorting and Searching, 2 ed., vol. 3, Addison-Wesley.
- Kordaki, M., Miatidis, M., & Kapsampelis, G. (2008). A computer environment for beginners' learning of sorting algorithms: Design and pilot evaluation. *Computers & Education*, 51(2), 708-723.
- Liu, C.H., Jiu, Y.W., & Jason, J-Y. (2009). Using Design Sketch to Teach Bubble Sort in High School, *The Journal of Computing*, 1(1), 20-25.
- Nieminen, J. (2005). *Bubble sort as the first sorting algorithm*, Ανακτήθηκε από <http://warp.povusers.org>.
- Sahni, S. (2004). *Δομές Δεδομένων, Αλγόριθμοι, Εφαρμογές στην C++*, Εκδόσεις Τζιόλα.
- Βακάλη, Α., Γιαννόπουλος, Η., Ιωαννίδης, Χ., Κοΐλιας, Χ., Μάλαμας, Κ., Μανωλόπουλος, Ι., & Πολίτης, Π. (2010). *Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον*, Αθήνα, ΥΠΔΒΜΘ.
- Βογιατζής, Ι., Ιωαννίδης, Ν., Κοΐλιας, Χ., Μελετίου, Γ., & Μόρμορης, Μ., (2009). *Εισαγωγή στην Αλγοριθμική*, Εκδόσεις Νέων Τεχνολογιών.
- Γρηγοριάδου, Μ., Βεργίνης, Η., & Γόγουλου, Α. (2010). Αναπτύσσοντας και Αξιολογώντας Εκπαιδευτικό Υλικό που βασίζεται στο ECLiP για Εισαγωγικά Μαθήματα Προγραμματισμού, *Πρακτικά 5ου Πανελληνίου Συνεδρίου «Διδακτικής της Πληροφορικής»*, Αθήνα, σελ. 51-60.
- Δουκάκης, Σ., Ψαλτίδου, Α., Γιαννοπούλου, Π. (2012). *Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον*, Τόμος Β, Εκδόσεις Πατάκη.
- Λαζαρίδης, Β. (2005). *Ασύγχρονη εξ αποστάσεως εκπαίδευση αλγορίθμων γραμμικής βελτιστοποίησης*, Δημοσίευτη διδακτορική διατριβή, Τμήμα εφαρμοσμένης πληροφορικής, Πανεπιστήμιο Μακεδονίας.
- Μπαλής, Π. (2010). Αλγόριθμος ταξινόμησης φουσαλίδας, Ανακτήθηκε από <http://www.youtube.com/watch?v=zSg3kxBkVVc>.
- Μπούμπουκα, Μ., Κανίδης, Ε., Γρηγοριάδου, Μ., & Σουλιώτη, Σ. (2008). Επανασχεδιάζοντας μια Δραστηριότητα Παιχνιδιού Ρόλων για τη Διδασκαλία του Αλγορίθμου Ταξινόμησης στη Δευτεροβάθμια Εκπαίδευση, *4ο Πανελλήνιο Συνέδριο «Διδακτική της Πληροφορικής»*.
- Σιδερίδης, Α., Γιαλούρης, Κ., Μπακογιάννης, Σ., & Σταθόπουλος, Κ. (2000). Προγραμματισμός υπολογιστών, Αθήνα, ΥΠΔΒΜΘ.
- Στέργου, Σ. (2010). pseudoglossa.gr - Online διερμηνευτής για την Ψευδογλώσσα του μαθήματος ΑΕΠΠ. Στο Δουκάκης Σ. (Επιμ.) *Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον, Παρελθόν, Παρόν και Μέλλον*, ΕΠΥ, Αθήνα, Εκδόσεις Νέων Τεχνολογιών, 101-108.
- Φωτάκης, Δ. (2006). Ταξινόμηση-Αναζήτηση-Επιλογή, Ανακτήθηκε από <http://www.softlab.ntua.gr/~fotakis>.