

Υλοποιώντας Αλγορίθμους και Δομές Δεδομένων στο μάθημα ΑΕΠΠ της Θετικής Κατεύθυνσης

Ελένη Ρόμπολα

Καθηγήτρια Πληροφορικής, 5ο ΓΕΛ Βύρωνα
eleni.rompola@gmail.com

Περίληψη

Η διδασκαλία του προγραμματισμού (ΑΕΠΠ) σε μαθητές της Γ' τάξης ΓΕΛ Θετικής Κατεύθυνσης παρουσιάζει το πλεονέκτημα της ανεξάρτητης από τις Πανελλαδικές Εξετάσεις. Αξιόλογο ποσοστό των μαθητών αυτών προσανατολίζεται σε πολυτεχνικές σχολές και σχολές Πληροφορικής, όπου μια βασική προϋπάρχουσα γνώση αλγορίθμων και δομών δεδομένων αποτελεί χρήσιμο εφόδιο. Η γνώση αυτή μπορεί να δομηθεί στο πλαίσιο του ΑΕΠΠ, μέσω προσεκτικά επιλεγμένων αλγορίθμων της Πληροφορικής και την υλοποίησή τους σε διάφορες γλώσσες προγραμματισμού. Σε αντίθεση με μια τυπική εξετασιοκεντρική διδασκαλία, η μελέτη και υλοποίηση πραγματικών αλγορίθμων, η ανακάλυψη της επιστημονικής έρευνας πίσω από αυτούς και η αξιολόγηση της αποτελεσματικότητάς τους, προσδίδει σφαιρικότητα και αυθεντικότητα στην αλγοριθμική σκέψη των μαθητών.

Λέξεις κλειδιά: Αλγόριθμοι, Δομές Δεδομένων, Προγραμματισμός, ΑΕΠΠ.

1. Εισαγωγή

Αλγόριθμοι και Δομές Δεδομένων αποτελούν δύο άρρηκτα συνδεδεμένες γνωστικές περιοχές της Πληροφορικής και θα μπορούσαμε να υποστηρίξουμε ότι αποτελούν την βάση για κάθε προϊόν σε επίπεδο λογισμικού αλλά και υλικού. Η μοντελοποίηση των χαρακτηριστικών και πληροφοριών που συνθέτουν το προς επίλυση πρόβλημα ή το μελλοντικό προϊόν και η αναπαράστασή τους με δομές δεδομένων, αποτελούν την μια όψη του νομίσματος. Ο αποδοτικός χειρισμός των δομών αυτών, δηλ. ο αλγόριθμος, με απώτερο στόχο την βέλτιστη δυνατή επίλυση του προβλήματος ή την βέλτιστη υλοποίηση του προϊόντος, είναι η άλλη όψη του ίδιου νομίσματος. Η σχεδίαση τόσο των δομών δεδομένων όσο και της λογικής του αλγορίθμου εξαρτάται από την εφευρετικότητα του προγραμματιστή και την νοητική ικανότητά του να σκέφτεται δημιουργικά.

Η καταλληλότητα των δομών δεδομένων αποδεικνύεται στην πράξη από το αν οδηγούν σε βέλτιστους αλγορίθμους ή όχι, γιαυτό και υποστηρίζουμε ότι οι δύο αυτές γνωστικές περιοχές της Πληροφορικής είναι άρρηκτα συνδεδεμένες. Για τον ίδιο λόγο, έστω κι αν η αλγοριθμική σκέψη υποστηρίζει τους μαθητές και σε άλλα μαθήματα, το κατεξοχήν περιβάλλον στο οποίο ανήκει και μέσω του οποίου μπορεί

να αναπτυχθεί στον μέγιστο βαθμό είναι η Πληροφορική, ακριβώς διότι δεν νοούνται αλγόριθμοι χωρίς δομές δεδομένων.

Από την άλλη πλευρά, οι γλώσσες προγραμματισμού εξελίσσονται, καταργούνται ή εμπλουτίζονται, δημιουργούνται νέες. Κάθε γλώσσα παρέχει στον προγραμματιστή διαφορετικά εργαλεία και δυνατότητες για την υλοποίηση των ίδιων αλγορίθμων και δομών δεδομένων. Ως προγραμματιστική σκέψη θα μπορούσε να οριστεί, πιο εξειδικευμένα, η ικανότητα μετατροπής ενός αλγορίθμου σε πρόγραμμα. Η ικανότητα χρήσης συγκεκριμένων γλωσσών προγραμματισμού και συγκεκριμένων προγραμματιστικών προτύπων.

Η ανάπτυξη αλγορίθμων και δομών δεδομένων αποτελεί συνεπώς διαφορετική δεξιότητα από την προγραμματιστική ικανότητα και προηγείται αυτής. Η αλγοριθμική σκέψη, γενικά, ενθαρρύνει τους μαθητές να κατασκευάσουν μια λύση, να αποδείξουν την ορθότητά της και να αναλύσουν την πολυπλοκότητά της. Σε αρκετούς μαθητές η αδυναμία δεξιοτήτων προγραμματισμού οφείλεται στην έλλειψη γενικών ικανοτήτων τους στην επίλυση προβλημάτων και στη συνεπαγόμενη δυσκολία τους κατά τη δημιουργία αλγορίθμων (Μαστρογιάννης, 2017).

Η διδασκαλία του προγραμματισμού έχει μελετηθεί από πολλές πλευρές, και πολλές διδακτικές μεθοδολογίες έχουν διατυπωθεί και δοκιμαστεί στην πράξη σε διάφορες εκπαιδευτικές βαθμίδες καθώς και για διαφορετικά προφίλ μαθητών. Η στοχοθεσία συμπεριλαμβάνει συχνά την καλλιέργεια της αλγοριθμικής σκέψης (Πολίτης & Κόμης, 1999), ωστόσο η διδακτική πράξη περιορίζεται πολλές φορές σε μικρά προβλήματα, ιδίως στο πλαίσιο του μαθήματος της Ανάπτυξης Εφαρμογών σε Προγραμματιστικό Περιβάλλον (στο εξής ΑΕΠΠ), όπου τα προς επίλυση προβλήματα δεν απαιτούν ούτε ιδιαίτερες δομές δεδομένων ούτε ιδιαίτερη εφευρετικότητα. Ο γενικός στόχος της επιτυχίας στις Πανελλαδικές Εξετάσεις σε συνδυασμό με την χρονική στενότητα των δύο διδακτικών ωρών ανά εβδομάδα, θέτει περιορισμούς: η επιλογή εκπαιδευτικών και μαθητών στρέφεται συχνά προς την ασφαλή πεπατημένη οδό των τυποποιημένων ασκήσεων προκειμένου να “βγει η ύλη μέχρι τα Χριστούγεννα”. Μια τέτοια στρεβλή πραγματικότητα, μέσω μιας συνεχούς, άκρατης ασκησιολογίας και μιας αυτοματοποιημένης ενεργοποίησης συμπεριφοριστικών τεχνικών, βιώνουν οι Έλληνες μαθητές κατά την προετοιμασία τους για τις Πανελλαδικές Εξετάσεις (Μαστρογιάννης, 2017β).

Η διδασκαλία του ΑΕΠΠ κατά τον τρόπο αυτό στερείται αυθεντικότητας, δεν μπορεί να μεταδώσει στους μαθητές σφαιρική εικόνα της αλγοριθμικής και κατ' επέκταση της Πληροφορικής και δυστυχώς πολλές φορές εκφυλίζεται σε αποστήθιση μεθοδολογιών επίλυσης των ασκήσεων. Η αλγοριθμική σκέψη ωστόσο είναι μια ευρύτερη και περισσότερο περίπλοκη διαδικασία από ότι ένας απλός αλγόριθμος και η προσπάθεια για οργάνωση της σκέψης μέσω αλγορίθμων, οδηγεί σε βαθύτερα και πληρέστερα μαθησιακά αποτελέσματα σε σχέση με άλλες μεθόδους (Μαστρογιάννης, 2017).

Η απεξάρτηση του ΑΕΠΠ της Θετικής Κατεύθυνσης από τις Πανελλαδικές Εξετάσεις πιστεύουμε ότι απελευθέρωσε το μάθημα από τα παραπάνω προβλήματα και πρόσθεσε σε αυτό νέες δυνατότητες. Είναι πλέον εφικτή η διδασκαλία πραγματικών αλγορίθμων της Πληροφορικής, οι οποίοι επιλύουν σημαντικά προβλήματα, στηρίζονται σε επιστημονική έρευνα και για τους οποίους έχει μελετηθεί η αποδοτικότητά τους. Καθώς μεγάλο ποσοστό των μαθητών Θετικής Κατεύθυνσης προσανατολίζεται σε σχολές πληροφορικής και πολυτεχνείων, πιστεύουμε ότι μια βασική προϋπάρχουσα γνώση αλγορίθμων και δομών δεδομένων θα αποτελούσε επιθυμητό εφόδιο. Εκτός αυτού, η δημιουργική σκέψη, η συστηματοποιημένη συλλογιστική, η εργασία σε ομάδες, είναι ικανότητες χρήσιμες για την μετέπειτα πορεία και εξέλιξη του μαθητή σε όλους τους τομείς και τα πεδία (Τζελέπη & Κοτίνη, 2012).

Στην παρούσα εργασία παρουσιάζουμε μια πρώτη προσπάθεια επιλογής αλγορίθμων και αντίστοιχων δομών δεδομένων, κατάλληλων για διδασκαλία στην τάξη: majority algorithm, broadcasting algorithm, shortest-path algorithm, eulerian path algorithms, online algorithms, cake-cutting algorithms, dynamic programming algorithms, κ.α. Οι αλγόριθμοι αυτοί μπορούν να διατυπωθούν σε ΓΛΩΣΣΑ - εάν αυτό είναι επιθυμητό - και να υλοποιηθούν με στατικές δομές πινάκων, ώστε να μην παραβιάζεται το πλαίσιο της διδακτέας ύλης του μαθήματος. Δεν υποστηρίζουμε πως το υπάρχον Αναλυτικό Πρόγραμμα Σπουδών του ΑΕΠΠ είναι επαρκές, πιστεύουμε το αντίθετο. Ωστόσο, έστω και με τον περιορισμό αυτού του ΑΠΣ, η διδασκαλία του μαθήματος μπορεί να εμπλουτιστεί.

Σημείο ιδιαίτερης βαρύτητας στην παρουσίαση κάθε αλγορίθμου αποτελεί η επιλογή του αρχικού παραδείγματος-προβλήματος, κι αυτό διότι η χρήση εννοιών από την καθημερινή ζωή έχει μια θετική επίπτωση στην ικανότητα των μαθητών να μαθαίνουν (Τζελέπη & Κοτίνη, 2013). Οι μαθητές κατασκευάζουν ενεργά νέες ιδέες ή έννοιες με βάση τις τρέχουσες και τις προηγούμενες γνώσεις και εμπειρίες τους αλληλεπιδρώντας με το κοινωνικό περιβάλλον (Hanley, 1994). Τα προβλήματα προσεγγίζονται μέσα από κατ' εξοχήν διαδικασίες επίλυσης προβλημάτων Πληροφορικής (ανάλυση, σχεδίαση, υπολογισμός πολυπλοκότητας), χωρίς ωστόσο η φάση της ανάλυσης να παρουσιάζει ομοιομορφία. Η φύση των επιλεγμένων παραδειγμάτων είναι διαφορετική, ως προβλήματα κατατάσσονται σε διαφορετικές κατηγορίες, άρα δεν υπάρχει προκαθορισμένος συστηματικός τρόπος σκέψης για την επίλυσή τους. Το ζητούμενο είναι η ενεργοποίηση της δημιουργικής και εφευρετικής σκέψης των μαθητών, η οποία - όπως τουλάχιστον είδαμε στη διάρκεια των μαθημάτων αυτών - λειτουργεί εξαιρετικά καλά. Ο εκπαιδευτικός προσπαθεί να δημιουργήσει ένα μαθησιακό περιβάλλον που να δίνει στους μαθητές τη δυνατότητα να αναλύσουν, να ερευνήσουν, να συνεργαστούν, να μοιραστούν και να δημιουργήσουν με βάση αυτά που ήδη γνωρίζουν (Hanley, 1994), εντάσσοντας έτσι την όλη διδασκαλία σε ένα πλαίσιο εποικοδομητισμού.

Τέλος, προτείνεται οι μαθητές να υλοποιήσουν κάθε αλγόριθμο σε τουλάχιστον δύο γλώσσες προγραμματισμού, π.χ. C και Python, είτε υποβοηθούμενοι από τον εκπαιδευτικό είτε εργαζόμενοι ομαδικά. Δεν αποτελεί αυτοσκοπό η εκμάθηση των συγκεκριμένων γλωσσών προγραμματισμού, αλλά η εξοικείωση με την διαδικασία μετατροπής ενός αλγορίθμου σε πρόγραμμα. Η επίδοση των μαθητών δεν αξιολογήθηκε γραπτώς, διότι οι γραπτές δοκιμασίες δεν αντικατοπτρίζουν με πληρότητα και σαφήνεια ούτε την οικοδόμηση των γνώσεων του μαθητή οι οποίες είναι σχετικές με αυτές των πλουσίων εννοιολογικών μοντέλων της επιστήμης των υπολογιστών αλλά ούτε και των ικανοτήτων του να αξιοποιεί τις γνώσεις αυτές σε πραγματικές καταστάσεις (Ben-Ari, 2001). Προτιμήσαμε μια ήπια διαμορφωτική αξιολόγηση κατά την συγγραφή των προγραμμάτων, με ευθύνη κυρίως των ίδιων των μαθητών, ως ένα είδος αυτοαξιολόγησης, καθώς τόσο η ηλικία όσο και η κατεύθυνση σπουδών τους είναι τέτοια που και το επιτρέπει και το ενθαρρύνει.

Η παραπάνω ιδέα εφαρμόστηκε στην τάξη, σε δύο τμήματα Θετικής Κατεύθυνσης στο 5ο ΓΕΛ Βύρωνα κατά το σχολικό έτος 2017-2018 ως μία σειρά μαθημάτων υπό τον γενικό τίτλο “Algorithms are a science of cleverness!” και οι πρώτες εντυπώσεις ήταν θετικές. Στις επόμενες ενότητες παρουσιάζονται με περισσότερη λεπτομέρεια κάποιιοι από τους παραπάνω αλγορίθμους καθώς και παρατηρήσεις από την εργασία στην τάξη. Κυρίως, όμως, η παρούσα εργασία, έστω κι αν επικεντρώνεται στην διδασκαλία του προγραμματισμού σε μαθητές Θετικής Κατεύθυνσης, ευελπιστεί να συμβάλλει σε έναν γόνιμο διάλογο, ο οποίος έχει ήδη ξεκινήσει (Κανίδης et al., 2018), (Τζιμογιάννης, 2007) για την αναβάθμιση της πληροφορικής παιδείας στο ελληνικό εκπαιδευτικό σύστημα.

2. Οι αλγόριθμοι

2.1 Broadcasting Algorithms

Περιγραφή του προβλήματος: Το 15μελές σχεδιάζει να οργανώσει χοροεσπερίδα για όλους τους μαθητές της Γ τάξης. Ο πρόεδρος του 15μελούς αναλαμβάνει την ενημέρωση των μαθητών, ώστε να ενημερωθούν όλοι και να μην υπάρξουν παράπονα. Όλοι οι μαθητές διαθέτουν την ίδια λίστα τηλεφώνων όλων των συμμαθητών τους. Οι μαθητές διακρίνονται σε αξιόπιστους (απαντούν στο τηλέφωνο, ενημερώνουν άλλους αν χρειαστεί) και μη-αξιόπιστους (δεν απαντούν στο τηλέφωνο ή ξεχνούν να ενημερώσουν άλλους). Οι τελευταίοι είναι κατανεμημένοι με τυχαίο τρόπο μέσα στην λίστα.

Ζητούμενο: Να σχεδιαστεί η βέλτιστη στρατηγική που θα μπορούσε να ακολουθήσει ο πρόεδρος του 15μελούς ώστε να ενημερωθεί το μέγιστο δυνατό πλήθος μαθητών.

Προσέγγιση στην τάξη: Συζητήθηκαν διάφορες στρατηγικές ως προς τα πλεονεκτήματα και μειονεκτήματά τους. Αν για παράδειγμα ο πρόεδρος τηλεφωνήσει προσωπικά σε όλους, θα ενημερωθούν όλοι όσοι απαντούν στο τηλέφωνό τους, αλλά

η διαδικασία είναι κοστοβόρα και χρονοβόρα. Αν ο πρόεδρος τηλεφωνήσει στον 1ο μαθητή της λίστας, εκείνος στον 2ο, ο 2ος στον 3ο, κ.ο.κ., το κόστος μοιράζεται, η διαδικασία ως σειριακή παραμένει χρονοβόρα κι επιπλέον η ύπαρξη έστω ενός μη αξιόπιστου μαθητή αναιρεί την επιτυχία της ενημέρωσης. Μια τρίτη ιδέα θα ήταν η ανάθεση σε κάθε μαθητή της ενημέρωσης μιας μικρής ομάδας συμμαθητών, οι οποίοι βρίσκονται πιο μετά στην λίστα από τον ίδιο και μάλιστα με τρόπο ώστε κάθε μαθητής να δέχεται όχι ένα αλλά περισσότερα τηλεφωνήματα, ώστε να αυξηθεί η πιθανότητα επιτυχίας της ενημέρωσης.

Τελικά, το αρχικό πρόβλημα επαναδιατυπώθηκε με μεγαλύτερη ακρίβεια ως εξής: Έστω ότι κάθε μαθητής στην λίστα αριθμείται με έναν αύξοντα ακέραιο αριθμό. Κάθε μαθητής i ενημερώνει τους συμμαθητές του που βρίσκονται στις θέσεις $2i+1$ έως και $2i+2r$. Για ένα δεδομένο πλήθος x αναξιόπιστων μαθητών, οι οποίοι βρίσκονται διασκορπισμένοι με τυχαίο τρόπο μέσα στην λίστα, θέλουμε να υπολογίσουμε πόσο μεγάλο πρέπει να είναι το r ώστε η πιθανότητα να μην ενημερωθούν τελικά όλοι οι αξιόπιστοι μαθητές να είναι το πολύ p .

Ακολούθησε η μοντελοποίηση των δεδομένων του προβλήματος. Ενδεικτικά αναφέρουμε ότι ο χαρακτηρισμός “αξιόπιστος ή μη-αξιόπιστος μαθητής” αναπαραστάθηκε με χρήση ενός στατικού πίνακα $A[N]$ (όπου N το πλήθος των μαθητών το οποίο δηλώθηκε αργότερα ως σταθερά του προγράμματος), τα στοιχεία του οποίου αρχικοποιήθηκαν με 0 για κάθε αξιόπιστο και με $-r-1$ (ή $-N-1$) για κάθε αναξιόπιστο μαθητή. Τα τηλεφωνήματα στα οποία απαντά ένας μαθητής αναπαριστώνται με αύξηση (κατά 1 κάθε φορά) του αντίστοιχου κελιού του πίνακα A . Κάθε μαθητής i που ενημερώνεται (άρα η αντίστοιχη τιμή $A[i]$ είναι θετική) ενημερώνει τους επόμενους (αυξάνονται κατά 1 τα κελιά $A[2i+1]$ έως και $A[2i+2r]$). Στο τέλος του αλγορίθμου, στους αξιόπιστους μαθητές που δεν ενημερώθηκαν αντιστοιχεί τιμή 0 στον πίνακα A , ενώ στους μη-αξιόπιστους μαθητές κάποια αρνητική τιμή. Μέσω αυτής της πληροφορίας είναι δυνατή η αξιολόγηση της επιτυχίας της όλης διαδικασίας και άρα της τιμής r που επιλέχθηκε.

Ο αλγόριθμος διατυπώθηκε από κοινού σε ΓΛΩΣΣΑ και στη συνέχεια μετατράπηκε σε πρόγραμμα (σε γλώσσες C και Python). Τα προγράμματα εκτελέστηκαν για διαφορετικές τιμές του r καθώς και για διαφορετικές κατανομές των αρχικών αρνητικών τιμών μέσα στον πίνακα A . Οι μαθητές εργάστηκαν σε μικρές ομάδες των 2-3 ατόμων.

Τέλος, αφού ανακεφαλαιώθηκε η όλη διαδικασία και συζητήθηκαν τα αποτελέσματα και οι παρατηρήσεις κάθε ομάδας, προτάθηκε από τον εκπαιδευτικό η ιδέα της τυχαίας επιλογής των συμμαθητών που θα ενημερώσει κάθε μαθητής. Συζητήθηκε με ποιες δομές δεδομένων θα μπορούσε να μοντελοποιηθεί αυτή η στρατηγική και ποια θα ήταν η γενικότερη αλγοριθμική της προσέγγιση.

Παρατηρήσεις: Οι μαθητές δεν ήταν προφανώς εξοικειωμένοι με την μετατροπή των δεδομένων ενός προβλήματος τα οποία αρχικά δίνονται σε προφορικό ή γραπτό

λόγο, σε άλλη αφηρημένη μορφή, η οποία διατηρεί μεν όλη την αναγκαία πληροφορία, την απλοποιεί όμως ως αναπαράσταση ώστε να είναι επεξεργάσιμη από ένα πρόγραμμα. Η συζήτηση στην τάξη για το “πώς θα αναπαραστήσουμε τους αξιόπιστους και πώς τους αναξιόπιστους μαθητές” ήταν εξαιρετικά ενδιαφέρουσα και απαιτεί ετοιμότητα από πλευράς του εκπαιδευτικού, ώστε να μπορεί να καθοδηγήσει διακριτικά τη σκέψη των μαθητών από ιδέες του τύπου “να τους λέμε *αξιόπιστος* και *αναξιόπιστος*” σε πιο δομημένες ιδέες, αλλά και για να μπορεί να διακρίνει ποιες από τις ιδέες των μαθητών θα μπορούσαν όντως να αποτελέσουν βάση για έναν αλγόριθμο και ποιες όχι.

Μετά τις δοκιμές των προγραμμάτων, όταν συζητήθηκε η ιδέα της “τυχαίας επιλογής συμμαθητών προς ενημέρωση”, ήταν πολύ λιγότερες οι αδιέξοδες ιδέες και οι μαθητές έφθασαν πιο γρήγορα σε κοινά αποδεκτή μοντελοποίηση των δεδομένων.

Διδακτικό κέρδος: (α) Έγινε φανερή η σημασία της ακρίβειας στην διατύπωση του προβλήματος. Σε αντίθεση με την τελική αυστηρότερη επαναδιατύπωση, η αρχική ελεύθερη περιγραφή χαρακτηρίζεται από ασάφειες που δυσχεραίνουν την επίλυση του προβλήματος. (β) Έγινε φανερή η πληθώρα των διαφορετικών δυνατοτήτων προσέγγισης ενός προβλήματος, καθώς πολλές από τις προτάσεις των μαθητών ήταν όντως υλοποιήσιμες. (γ) Η συζήτηση στην τάξη, ο καταγισμός ιδεών και η ομότιμη αξιολόγησή τους (καθώς κάθε ιδέα αμφισβητούνταν ή γινόταν αποδεκτή μέσα από ελεύθερη συζήτηση) ήταν εκτός από μια ελκυστική διαδικασία, μια μικρογραφία του πώς προσεγγίζονται πραγματικά προβλήματα Πληροφορικής.

2.2 Majority Algorithm

Περιγραφή του προβλήματος: Για να αναδειχθεί ο πρόεδρος ενός συλλόγου διενεργούνται εκλογές μεταξύ των μελών ως εξής: κάθε μέλος του συλλόγου έχει μία ψήφο και μπορεί να ψηφίσει οποιονδήποτε επιθυμεί, ακόμη και τον εαυτό του. Νικητής της εκλογικής διαδικασίας είναι το μέλος που θα συγκεντρώσει περισσότερες από τις μισές ψήφους. Σε περίπτωση που κανείς δεν το επιτύχει, δεν υπάρχει στοιχείο πλειοψηφίας.

Ζητούμενο: Να σχεδιαστεί αλγόριθμος για την εύρεση του στοιχείου πλειοψηφίας σε ένα σύνολο N διακριτών, μη-μοναδικών στοιχείων και να βελτιστοποιηθεί κατά το δυνατόν.

Προσέγγιση στην τάξη: Οι αρχικές ιδέες των μαθητών περιστράφηκαν γύρω από τις τακτικές καταμέτρησης που εφαρμόζουν στις εκλογές των μαθητικών κοινοτήτων: εφόσον το όνομα πάνω στο ψηφοδέλτιο συναντάται για 1η φορά, προστίθεται στην λίστα των ψηφισθέντων και ο αντίστοιχος μετρητής γίνεται 1. Αν το όνομα υπάρχει ήδη στην λίστα, απλώς αυξάνεται ο αντίστοιχος μετρητής κατά 1. Μετά το τέλος της καταμέτρησης, ελέγχεται αν ο μέγιστος μετρητής είναι μεγαλύτερος από $N/2$. Η στρατηγική αυτή, καθώς ήταν οικεία στους μαθητές, αποτέλεσε χρήσιμη βάση για την εισαγωγή των μαθητών στην έννοια των βασικών πράξεων και του υπολογισμού

μέσω αυτών της πολυπλοκότητας ενός αλγορίθμου. Η στρατηγική αξιολογήθηκε μέσω συζήτησης και προέκυψαν τα συμπεράσματα ότι (α) παράγεται περισσότερη πληροφορία από όση απαιτείται, καθώς υπολογίζονται οι ψήφοι όλων, (β) οι συγκρίσεις είναι της τάξης $O(N^2)$.

Στη συνέχεια παρουσιάστηκε από τον εκπαιδευτικό μια πρώτη έκδοση του βέλτιστου αλγορίθμου εύρεσης στοιχείου πλειοψηφίας, όπως αυτός υπάρχει στην βιβλιογραφία:

- Χρησιμοποίησε έναν σωρό, αρχικά άδειο.
- Φάση 1- Για κάθε στοιχείο x από τα N κάνει τα εξής: Αν ο σωρός είναι άδειος, τοποθέτησε το x στην κορυφή του σωρού. Αν δεν είναι άδειος, σύγκρινε το x και το στοιχείο κορυφής του σωρού. Αν τα δύο στοιχεία είναι ίδια, τοποθέτησε το x στην κορυφή του σωρού. Διαφορετικά, αφαίρεσε το στοιχείο κορυφής από τον σωρό.
- Φάση 2 - Αν ο σωρός είναι άδειος, δεν υπάρχει στοιχείο πλειοψηφίας. Διαφορετικά, πάρε το στοιχείο κορυφής και μέτρησε πόσες φορές εμφανίζεται μέσα στο αρχικό σύνολο των στοιχείων. Αν εμφανίζεται περισσότερες από $N/2$ φορές είναι στοιχείο πλειοψηφίας. Διαφορετικά, δεν υπάρχει στοιχείο πλειοψηφίας.

Ο αλγόριθμος παρουσιάστηκε περιφραστικά και στη συνέχεια μέσω γραφικής προσομοίωσης εκτελέστηκε για τρία διαφορετικά σύνολα εισόδου. Η συζήτηση περιστράφηκε γύρω από: (α) την απόδειξη της ορθότητας του αλγορίθμου (αρχικά διαισθητικά και στη συνέχεια δια της εις άτοπον απαγωγής), (β) τον υπολογισμό των βασικών πράξεων (συγκρίσεων) οι οποίες είναι της τάξης $O(N)$.

Η απαιτούμενη δομή για την μοντελοποίηση των δεδομένων, η οποία στην γραφική προσομοίωση αλλά και περιφραστικά αναφέρθηκε ως σωρός, είναι ένας στατικός πίνακας σε συνδυασμό με μια απλή μεταβλητή-δείκτη του στοιχείου κορυφής. Ο αλγόριθμος δόθηκε σε ΓΛΩΣΣΑ και οι μαθητές τον υλοποίησαν σε C και Python.

Κλείνοντας, αναφέρθηκαν εφαρμογές του αλγορίθμου εύρεσης πλειοψηφίας (εκλογές, κρίσιμοι υπολογισμοί που γίνονται ταυτόχρονα από πολλούς ανεξάρτητους H/Y και ως σωστό γίνεται αποδεκτό μόνο το αποτέλεσμα που πλειοψηφεί, υπολογισμός συχνοτήτων εμφάνισης αν το $N/2$ αντικατασταθεί με N/k , παρακολούθηση κίνησης πακέτων διαδικτύου, κ.α.).

Παρατηρήσεις: Η αξιολόγηση των αλγορίθμων, δηλ. ο υπολογισμός της πολυπλοκότητάς τους, δεν είναι δύσκολο να εξηγηθεί όταν χρησιμοποιηθεί παράδειγμα μέσω του οποίου οι μαθητές μπορούν όντως να φανταστούν τους εαυτούς τους να επιτελούν τις βασικές πράξεις του αλγορίθμου (στην περίπτωσή μας τις συγκρίσεις των ψηφοδελτίων).

Ο βέλτιστος αλγόριθμος εύρεσης πλειοψηφίας παραξένεψε τους μαθητές και μόνο μέσω των παραδειγμάτων και της γραφικής προσομοίωσής τους μπόρεσαν να εξηγήσουν τελικά γιατί ο αλγόριθμος δουλεύει. Πρόκειται για έναν πραγματικά εντυπωσιακό αλγόριθμο, ακριβώς επειδή διαφέρει πολύ από την προφανή μέθοδο καταμέτρησης. Έγινε σαφές ότι η προφανής μέθοδος δεν είναι πάντα η πιο αποτελεσματική και ότι συχνά υπάρχει καλύτερος αλγόριθμος.

Διδακτικό κέρδος: (α) Οι μαθητές γνώρισαν μια νέα δομή δεδομένων, την στοίβα. Συζητήθηκαν οι διαφορές μεταξύ στοίβας και στατικού πίνακα, κι έγινε εμφανές ότι τα χαρακτηριστικά μιας δομής δεδομένων σχεδιάζονται κατά τα αρχικά στάδια επίλυσης του προβλήματος, υλοποιούνται όμως μέσω του αλγορίθμου, ο οποίος και πρέπει να διασφαλίσει ότι δεν θα παραβιαστεί ο αρχικός σχεδιασμός. (β) Ο υπολογισμός της πολυπλοκότητας ενός αλγορίθμου, έστω εμπειρικά και χωρίς την συνδρομή προχωρημένων μαθηματικών, εξασφαλίζει ότι οι μαθητές έχουν κατανοήσει πλήρως την λειτουργία του αλγορίθμου και της αντίστοιχης δομής δεδομένων, γι' αυτό και μπορούν να τον αξιολογήσουν. Ταυτόχρονα τους πείθει για την εφευρετικότητα και δημιουργική σκέψη που χαρακτηρίζει την Θεωρητική Πληροφορική.

2.3 Graph Algorithms

Περιγραφή του προβλήματος: Η διοργάνωση ενός τουρνουά μεταξύ των N ομάδων ποδοσφαίρου μιας περιοχής πρέπει να εκπληρώνει τις εξής προϋποθέσεις: (α) οι αγωνιστικές είναι $N-1$ ημέρες, (β) κάθε ομάδα πρέπει να παίξει εναντίον κάθε άλλης ομάδας σε αυτό το διάστημα, (γ) κάθε ομάδα παίζει εναντίον μίας και μόνο ομάδας σε κάθε αγωνιστική.

Ζητούμενο: Να απαντηθεί το ερώτημα εάν μπορεί να σχεδιαστεί ένα τέτοιο πλάνο για κάθε ζυγό αριθμό N , και αν ναι πως.

Προσέγγιση στην τάξη: Η διοργάνωση ποδοσφαιρικών τουρνουά έχει πολύ περισσότερες παραμέτρους από αυτές που τίθενται στο πρόβλημα. Συζητήθηκαν και συμφωνήθηκε να σχεδιαστεί αρχικά ένας αλγόριθμος για τις ελάχιστες αναγκαίες απαιτήσεις. Ο εκπαιδευτικός εισήγαγε την έννοια του γράφου, εξηγώντας απλώς τι ορίζεται ως κόμβος και τι ως ακμή. Χρησιμοποιώντας ως παράδειγμα την τιμή $N=6$, οι μαθητές ρωτήθηκαν αν θα μπορούσαν να αναπαραστήσουν τους αγώνες μεταξύ 6 ομάδων με έναν γράφο: πως θα φαίνεται ποια ομάδα παίζει εναντίον ποιας και σε ποια αγωνιστική. Το συμπέρασμα ήταν η σχεδίαση ενός πλήρους γράφου με 6 κορυφές, μια για κάθε ομάδα, όπου οι ακμές ενώνουν τις ομάδες που αγωνίζονται μεταξύ τους. Για να προστεθεί η πληροφορία σε ποια αγωνιστική γίνεται κάθε αγώνας, χρησιμοποιούνται 5 διαφορετικά χρώματα και οι ακμές χρωματίζονται ανάλογα (υπάρχουν 3 ακμές ίδιου χρώματος, για κάθε χρώμα). Η μοντελοποίηση, επομένως, των δεδομένων γίνεται με γραφικό, γεωμετρικό τρόπο.

Στη συνέχεια, τέθηκε το ερώτημα αν θα μπορούσαν να περιγραφούν βήματα κατασκευής του γράφου, ώστε για οποιονδήποτε ζυγό αριθμό N να παράγεται το πλάνο των αγώνων. Ο εκπαιδευτικός πρότεινε να αφαιρεθεί ένας κόμβος, ώστε να προκύψει ένα κανονικό πεντάγωνο. Όλες οι εξωτερικές πλευρές χρωματίστηκαν με διαφορετικό χρώμα. Για να ικανοποιούνται οι απαιτήσεις του αγωνιστικού πλάνου, οι μαθητές παρατήρησαν ότι κάθε εσωτερική ακμή πρέπει να χρωματιστεί με το χρώμα της γεωμετρικά παράλληλης της εξωτερικής πλευράς. Με το ίδιο χρώμα χρωματίζεται και η ακμή που συνδέει τον κόμβο του πενταγώνου που έχει περισσέψει με τον 6ο κόμβο που αφαιρέθηκε αρχικά.

Ο αλγόριθμος περιγράφηκε σε ΓΛΩΣΣΑ, χρησιμοποιώντας έναν διδιάστατο πίνακα για την αναπαράσταση του γράφου και υλοποιήθηκε σε C και Python.

Ως επέκταση του αλγορίθμου συζητήθηκαν οι εντός-εκτός έδρας αγώνες και διαπιστώθηκε ότι η εναλλαγή δεν είναι απόλυτα εφικτή. Ως πληροφορία ωστόσο, μπορούν να προστεθούν βέλη στις ακμές του γράφου (κατευθυνόμενος γράφος) έτσι ώστε “τα βέλη να δείχνουν προς την ομάδα που παίζει στην έδρα της”. Ανάλογα τροποποιείται και ο αλγόριθμος.

Κλείνοντας, υπολογίστηκε το δυνατό πλήθος γράφων ($18! = 6,4 \cdot 10^{15}$) που μπορεί να προκύψει για ένα τουρνουά μεταξύ 18 ομάδων, για την επεξεργασία των οποίων από έναν Η/Υ θα χρειαζόντουσαν περίπου 74 ημέρες. Το πεδίο αυτό είναι ακόμη ερευνητικά ανοιχτό.

Παρατηρήσεις: Εδώ η μοντελοποίηση των δεδομένων έγινε σε δύο βήματα: αρχικά χρησιμοποιήθηκε μια γεωμετρική αναπαράσταση (ο γράφος) και στη συνέχεια αυτή η σχηματική αναπαράσταση μετατράπηκε σε δομή δεδομένων (δισδιάστατος πίνακας). Η χρήση χρωμάτων στους γράφους είναι σύμφωνα με την βιβλιογραφία μια διαδεδομένη τεχνική στους σχετικούς αλγορίθμους. Ο τρόπος με τον οποίο χρωματίστηκαν οι ακμές για να προκύψει το αγωνιστικό πλάνο (παράλληλες ακμές, περιστροφή του γράφου κατά μια κορυφή, κλπ.) εντυπωσίασε τους μαθητές κυρίως διότι δεν ήταν αναμενόμενη ως τεχνική επίλυσης προβλημάτων πληροφορικής. Έγινε όμως εύκολα κατανοητή και αυτό φάνηκε στην ευκολία με την οποία τροποποίησαν τον αλγόριθμο όταν προστέθηκε η εναλλαγή των εντός-εκτός έδρας αγώνων.

Διδακτικό κέρδος: Οι μαθητές γνώρισαν τους γράφους ως εργαλεία για την μοντελοποίηση δεδομένων και μπόρεσαν να σκεφτούν και να περιγράψουν διάφορες άλλες περιπτώσεις όπου η χρήση γράφων θα ήταν ενδεδειγμένη.

2.4 Dynamic Programming Algorithms

Περιγραφή του προβλήματος: Στο τραπέζι μεταξύ δύο παικτών υπάρχουν N σπέρτα και οι παίκτες παίζουν εναλλάξ. Κάθε παίκτης μπορεί να τραβήξει 1, 2 ή 3 σπέρτα κάθε φορά που είναι η σειρά του. Χάνει όποιος παίκτης αναγκαστεί να τραβήξει το τελευταίο σπέρτο.

Ζητούμενο: Να σχεδιαστεί στρατηγική την οποία εφόσον ακολουθήσει ένας παίκτης να του εξασφαλίζει την νίκη.

Προσέγγιση στην τάξη: Παρουσιάστηκαν οι κανόνες του παιχνιδιού, οι μαθητές χωρίστηκαν σε ζεύγη και κάθε ζεύγος πήρε 18 σπάρτα. Τους ζητήθηκε να παίζουν μερικές φορές το παιχνίδι και να σκεφτούν ίσως κάποια καλή στρατηγική νίκης. Καμία ομάδα δεν κατάφερε να σκεφτεί κάποια σταθερή στρατηγική, παρά μόνο κάποιες καλές ιδέες όταν το πλήθος των σπάρτων γινόταν μικρό.

Στη συνέχεια, αναλύθηκαν διαδοχικά και διεξοδικά οι καταστάσεις $N=1$, $N=2$, $N=3$ μέχρι και $N=6$, ως προς το αν σε κάθε μια από αυτές υπάρχει στρατηγική νίκης. Ως στρατηγική νίκης εννοούμε να υπάρχει πλήθος σπάρτων τέτοιο που αν το τραβήξει ο ένας παίκτης, τότε να αναγκάζει τον αντίπαλο να χάσει (να τραβήξει το τελευταίο σπάρτο). Γρήγορα φάνηκε ότι για να συμπεράνουμε την ύπαρξη στρατηγικής νίκης στις καταστάσεις $N=4$ και εξής, θα πρέπει να γνωρίζουμε τι συμβαίνει στις προηγούμενες τρεις. Για λόγους οικονομίας χρόνου, παρουσιάστηκαν από το εκπαιδευτικό οι στρατηγικές νίκης μέχρι και για $N=18$ και επιβεβαιώθηκε από τους μαθητές ότι για να υπάρχει στρατηγική νίκης για μια κατάσταση N , θα πρέπει μια από τις τρεις προηγούμενες ($N-1$, $N-2$, $N-3$) να μην έχει στρατηγική νίκης.

Ο αλγόριθμος διατυπώθηκε σε ΓΛΩΣΣΑ και υλοποιήθηκε σε C και Python με χρήση ενός μονοδιάστατου πίνακα για την αποθήκευση της λογικής τιμής ΑΛΗΘΗΣ όταν υπάρχει στρατηγική νίκης ή ΨΕΥΔΗΣ όταν δεν υπάρχει. Στην περίπτωση αυτή δεν υπήρχε κάποια ιδιαίτερη απαίτηση ως προς την μοντελοποίηση των δεδομένων.

Στη συνέχεια ζητήθηκε από τους μαθητές να υπολογίσουν την πολυπλοκότητα του αλγορίθμου θεωρώντας ως βασικές πράξεις την εκχώρηση τιμής και την σύγκριση. Η πολυπλοκότητα προκύπτει πολυωνυμική $O(N)$. Χρησιμοποιώντας μαθηματικά γνωστά στους μαθητές από την ύλη της Β και Γ τάξης, παρουσιάστηκε από τον εκπαιδευτικό ο υπολογισμός της πολυπλοκότητας του αλγορίθμου ως προς τον αριθμό των bit που απαιτούνται για την αναπαράσταση του N . Η πολυπλοκότητα προκύπτει εκθετική.

Το συμπέρασμα αυτό πυροδότησε μια ενδιαφέρουσα συζήτηση σχετικά με τα P και NP προβλήματα της Πληροφορικής, πολλά από αυτά (πχ Knapsack Problem, Shortest Path Problem, Travelling Salesman Problem) εντάσσονται στο πεδίο του Δυναμικού Προγραμματισμού.

Παρατηρήσεις: Αν και το μάθημα ξεκίνησε με την διασκεδαστική προσέγγιση του προβλήματος (παιχνίδι με τα σπάρτα), το κλείσιμο χαρακτηρίστηκε από αυστηρή επιστημονική προσέγγιση (υπολογισμός εκθετικής πολυπλοκότητας) δίνοντας μια εικόνα στους μαθητές για τα μαθηματικά της Πληροφορικής. Ήταν κάτι που το χαρακτήρισαν μεν δύσκολο αλλά κι εντυπωσιακό. Η υλοποίηση του αλγορίθμου ήταν απλούστατη και υπήρξαν ομάδες μαθητών που τροποποίησαν με δική τους

πρωτοβουλία το πρόγραμμα (πχ να ρωτάει τον παίκτη πόσα σπίρτα έχει μπροστά του και να προτείνει πόσα να τραβήξει, να παίζει παίκτης εναντίον υπολογιστή, κλπ).

Διδακτικό κέρδος: (α) Οι μαθητές κατανόησαν την διαδικασία του Δυναμικού Προγραμματισμού ως μια τεχνική επίλυσης προβλημάτων μέσω της επίλυσης όλων των προβλημάτων μικρότερου μεγέθους από το αρχικό. (β) Γνώρισαν κάποια από τα μεγάλα ανοιχτά προβλήματα της Πληροφορικής στον τομέα αυτό. (γ) Έγινε μια ελάχιστη εισαγωγή στα μαθηματικά της Πληροφορικής κι αυτό κατά κάποιο τρόπο αποκατέστησε στα μάτια των μαθητών την Πληροφορική ως επιστήμη.

2.5 Eulerian Path Algorithms

Περιγραφή του προβλήματος: Είναι γνωστό το κλασικό παιχνίδι της σχεδίασης με μονοκοντυλιά: η σχεδίαση ξεκινάει και καταλήγει στο ίδιο σημείο, είναι συνεχής και κάθε γραμμή ζωγραφίζεται μία και μόνο φορά. Υπάρχουν σχήματα που δεν μπορούν να σχεδιαστούν με μονοκοντυλιά και άλλα για τα οποία υπάρχουν πολλοί τρόποι σχεδίασης, όπως για παράδειγμα το σπιτάκι για το οποίο υπάρχουν 44 διαφορετικές λύσεις!

Ζητούμενο: Δεδομένου ενός γεωμετρικού σχήματος, το οποίο αποτελείται από ένα πλήθος κορυφών N κι ένα πλήθος γραμμών M που ενώνουν τις κορυφές, να σχεδιαστεί αλγόριθμος που θα ανακαλύπτει αν το σχήμα μπορεί να ζωγραφιστεί με μονοκοντυλιά και αν ναι να προτείνει έναν τρόπο.

Προσέγγιση στην τάξη: Οι μαθητές παρακινήθηκαν να σχεδιάσουν μονοκοντυλιά ένα αστέρι 5 κορυφών (συνολικά 10 σημείων και 15 γραμμών) κι ένα καραβάκι επίσης 5 κορυφών (συνολικά 8 σημείων και 10 γραμμών). Το πρώτο επιτεύχθηκε αμέσως ενώ το δεύτερο αποδείχθηκε στην πράξη αδύνατο. Η αποτυχία αυτή οδήγησε στο ερώτημα “Πότε υπάρχει λύση;” και πράγματι οι μαθητές κατόρθωσαν να δουν ότι η μονοκοντυλιά απαιτεί να φτάνει σε κάθε σημείο ζυγό πλήθος γραμμών (ώστε όσες φορές κι αν φτάσουμε σε κάποιο σημείο πάντα να υπάρχει μια αχρησιμοποίητη γραμμή από την οποία να μπορούμε να φύγουμε).

Μετά από μια σύντομη αναφορά στις διάσημες γέφυρες του Koenigsberg και τον L. Euler, η μονοκοντυλιά μετονομάστηκε σε κύκλο του Euler και οι μαθητές προσπάθησαν να ανακαλύψουν μέσα από καταιγισμό ιδεών και συζήτηση έναν αλγόριθμο εύρεσης κύκλων του Euler. Ως παράδειγμα χρησιμοποιήθηκε ένας πλήρως διασυνδεδεμένος μη-κατευθυνόμενος γράφος 6 κόμβων. Υπάρχουν διάφορες υλοποιήσεις του αλγορίθμου του Euler και οι διαφορές στην πολυπλοκότητά τους οφείλονται στην χρήση διαφορετικών δομών δεδομένων. Ως απλούστερη και αποδοτική λύση χρησιμοποιήθηκαν τρεις λίστες: (α) λίστα του συνόλου των κόμβων του γράφου, (β) λίστα των ακμών ως ζεύγη κόμβων, (γ) λίστα των κόμβων του κύκλου Euler. Ο αλγόριθμος εκφράστηκε σε ΓΛΩΣΣΑ και υλοποιήθηκε σε C και Python.

Μια πολύ ενδιαφέρουσα εφαρμογή των κύκλων Euler και των γράφων γενικότερα συναντάται στην γονιδιωματική μεγάλης κλίμακας. Ο εκπαιδευτικός ζήτησε από τους μαθητές να καταθέσουν τις σχετικές γνώσεις τους από το μάθημα της Βιολογίας Θετικής Κατεύθυνσης και στη συνέχεια παρουσίασε επιστημονικές εργασίες οι οποίες αναφέρονται στην αναγκαιότητα της θεωρίας γράφων γενικότερα (Quiroz-Ibarrá, 2017) αλλά και των κύκλων του Euler ειδικότερα (Pevnzer et al., 2001) στον τομέα αυτό της Βιολογίας.

Παρατηρήσεις: Η χρησιμότητα των κύκλων του Euler στην αντιμετώπιση προβλημάτων γενετικής εξέπληξε ευχάριστα τους μαθητές. Έχοντας ήδη σχηματίσει μια στοιχειώδη κριτική ικανότητα για τις διάφορες αλγοριθμικές πολυπλοκότητες, μπόρεσαν εύκολα να διακρίνουν την πραγματική χρησιμότητα τόσο του αλγορίθμου του Euler όσο και γενικότερα των αλγορίθμων των γράφων στην γονιδιωματική μεγάλης κλίμακας. Κάποιοι μαθητές ενδιαφέρθηκαν για περαιτέρω μελέτη επιστημονικών εργασιών βιοπληροφορικής.

Η προγραμματιστική υλοποίηση του αλγορίθμου, ωστόσο, ήταν αρκετά απαιτητική και δεν επιτεύχθηκε από όλους τους μαθητές.

Διδακτικό κέρδος: (α) Γνωριμία με πιο σύνθετες δομές δεδομένων για την υλοποίηση γράφων. (β) Σύνδεση της Πληροφορικής με άλλες επιστήμες, όπως η Βιολογία. Οι μαθητές αναγνώρισαν την αναγκαιότητα όχι μόνο των όποιων υπολογιστικών συσκευών για την εκτέλεση πολύπλοκων επιστημονικών υπολογισμών, αλλά και των θεωρητικών μοντέλων της Πληροφορικής σε άλλες επιστήμες.

2.8 Άλλοι αλγόριθμοι

Εκτός των παραπάνω αλγορίθμων, προσεγγίστηκαν στην τάξη με παρόμοια μεθοδολογία (α) cake-cutting αλγόριθμοι (πως θα μοιραστεί ένα ανομοιογενές αγαθό σε N παίκτες με δικαιοσύνη) οι οποίοι αποτελούν ανοιχτό ερευνητικό πεδίο με πολλές εφαρμογές αλλά με προς το παρόν κακούς αλγορίθμους, (β) online αλγόριθμοι (όπου τα προς επεξεργασία μεγέθη δεν είναι εκ των προτέρων γνωστά αλλά η επεξεργασία τους πρέπει να γίνει άμεσα), με διασημότερο ίσως εκπρόσωπο το bin-packing problem. Οι αλγόριθμοι αυτοί δεν θα παρουσιαστούν αναλυτικά για λόγους υπέρβασης της επιτρεπόμενης έκτασης του άρθρου.

Σε μικρότερες τάξεις έχουν παρουσιαστεί ακολουθώντας τις αρχές του εποικοδομητισμού, ο Αλγόριθμος Βέλτιστης Διαδρομής του Dijkstra (Αλεξόπουλος & Ρόμπολα, 2012) στην Α' τάξη ΓΕΛ και ο Αλγόριθμος Κωδικοποίησης του Huffman (Ρόμπολα, 2013) στην Β' τάξη ΓΕΛ, χωρίς ωστόσο οι μαθητές να προχωρήσουν σε προγραμματιστική υλοποίηση.

3. Συμπεράσματα

Η παρούσα εργασία περιέγραψε μια πρώτη και αρκετά διστακτική προσπάθεια ένταξης νέων αλγορίθμων στην διδασκαλία της πληροφορικής, αναλύοντας ταυτόχρονα την ανταπόκριση των μαθητών. Έχουμε συγκεντρώσει πειστικές ενδείξεις ότι οι μαθητές του Γενικού Λυκείου είναι σαφώς σε θέση να συμμετάσχουν ενεργά σε μαθήματα αλγοριθμικής και θα ήταν ευχής έργον να είναι σε θέση να υλοποιούν με ευκολία τους αλγορίθμους που μελετούν. Η προγραμματιστική αυτή ευχέρεια, ωστόσο, απαιτεί προεργασία περισσότερων χρόνων και δεν αποκτάται αποκλειστικά στην τελευταία τάξη του Λυκείου. Μάλιστα είναι λυπηρό και θα έπρεπε ίσως να οδηγήσει σε προβληματισμό το γεγονός ότι τόσο η αλγοριθμική δημιουργική σκέψη όσο και η προγραμματιστική ευχέρεια αποτελούν εφόδια τα οποία οι μαθητές της κατεύθυνσης Οικονομίας-Πληροφορικής, για τους λόγους που αναλύθηκαν στην εισαγωγή του παρόντος άρθρου, δεν θα μπορέσουν να αποκτήσουν στο πλαίσιο του ΑΕΠΠ.

Παρατηρήθηκε επίσης ότι μέσω της αλγοριθμικής και της σφαιρικότητας που δνητικά προσδίδει στην διδασκαλία, η Πληροφορική αποκαθίσταται στα μάτια των μαθητών από χρήση εργαλείων ΤΠΕ, παιχνιδιών και μέσων κοινωνικής δικτύωσης, σε επιστήμη η οποία μάλιστα συνδράμει τις υπόλοιπες επιστήμες σε καίρια ζητήματα. Οι μαθητές της Θετικής Κατεύθυνσης αποτελούν το ιδανικό ίσως μαθητικό κοινό του οποίου τα επιστημονικά προβλήματα, η έρευνα πίσω από αυτά και οι μαθηματικά αποδεδειγμένες λύσεις εξάπτουν το ενδιαφέρον. Αποτελούν επίσης ένα μαθητικό κοινό το οποίο δικαιούται να εξερευνησει και να μάθει – διότι μπορεί – πολλά περισσότερα από όσα μέχρι στιγμής προσφέρει το ΑΠΣ του ΑΕΠΠ.

Αναφορές

Ben-Ari, M. (2001). Constructivism in Computer Science Education. *Journal of Computers in Mathematics and Science Teaching*, 20(1), 45-73.

Quiroz-Ibarra, J. E., Mallén-Fullerton, G. M., & Fernández-Anaya, G. (2017). DNA Paired Fragment Assembly Using Graph Theory. *Algorithms* 2017, 10, 36.

Hanley, S. (1994). *On Constructivism*. Ανάκτηση από το terpconnect.umd.edu/~toh/MCTP/Essays/Constructivism.txt

Pevnzer, P. A., Tang, H., & Waterman, M. S. (2001). *An Eulerian Path approach to DNA fragment assembly*. Ανάκτηση από το www.pnas.org/content/98/17/9748.

- Αλεξόπουλος, Κ., & Ρόμπολα, Ε. (2012). Μια πρόταση για την διδασκαλία του αλγορίθμου βέλτιστης διαδρομής του Dijkstra στο Γενικό Λύκειο. *4ο Συνέδριο Η Πληροφορική στην Εκπαίδευση*, Αθήνα.
- Κανίδης, Ε., Καραλιοπούλου, Μ., Αποστολάκης, Ι., & Τσιωτάκης, Π. (2018). Πλαίσιο για ένα Ενιαίο Πρόγραμμα Σπουδών Πληροφορικής Πρωτοβάθμιας και Δευτεροβάθμιας Εκπαίδευσης. *Έρκυνα, Επιθεώρηση Εκπαιδευτικών – Επιστημονικών Θεμάτων*, 14, 78-98.
- Μαστρογιάννης, Α. (2017). Η αλγοριθμική σκέψη ως θεμελιακή συνιστώσα της μαθηματικής σκέψης στην Υποχρεωτική Εκπαίδευση. *Νέος Παιδαγωγός, Διαδικτυακό περιοδικό εκπαιδευτικής αρθρογραφίας*, 8, 173-261.
- Μαστρογιάννης, Α. (2017β). Στοιχειώδεις, εφευρετικοί αλγόριθμοι για μαθητές Υποχρεωτικής Εκπαίδευσης. *23ο Διεθνές Συνέδριο Εκπαίδευση, Ετερότητα, Προσφυγικές Δομές και τα Ελληνικά ως δεύτερη ξένη γλώσσα*, Πάτρα, 96-111.
- Πολίτης, Π., & Κόμης, Β. (1999). Η Πληροφορική ως βασικό μάθημα της Γ' τάξης Τεχνολογικής Κατεύθυνσης του Ενιαίου Λυκείου: αλγοριθμική έναντι προγραμματιστικής προσέγγισης. *4ο Πανελλήνιο Συνέδριο με Διεθνή Συμμετοχή Διδακτική των Μαθηματικών και Πληροφορική στην Εκπαίδευση*, Ρέθυμνο, 344-348.
- Ρόμπολα, Ε. (2013). Μια πρόταση για την διδασκαλία του αλγορίθμου κωδικοποίησης του Huffman στο Γενικό Λύκειο. *5ο Συνέδριο Η Πληροφορική στην Εκπαίδευση*, Αθήνα.
- Τζελέπη, Σ., & Κοτίνη, Ι. (2012). Η Συμβολή της Υπολογιστικής Σκέψης στην Προετοιμασία του Αυριανού Πολίτη. *4ο Συνέδριο Η Πληροφορική στην Εκπαίδευση*, Αθήνα.
- Τζελέπη, Σ., & Κοτίνη, Ι. (2013). Ο εποικοδομητισμός ως μοντέλο διδασκαλίας της Πληροφορικής. *5ο Συνέδριο Η Πληροφορική στην Εκπαίδευση*, Αθήνα.
- Τζιμογιάννης, Α. (2007). Η διδασκαλία του προγραμματισμού στο Ενιαίο Λύκειο: προς ένα ολοκληρωμένο πλαίσιο με στόχο την ανάπτυξη δεξιοτήτων επίλυσης προβλημάτων. *2ο Συνέδριο ΤΠΕ στην Εκπαίδευση*, Σύρος.

Abstract

Teaching programming to 3rd Grade Senior High School students in Greece has the advantage of not being (University Entrance) Exam(s) oriented. A significant number of students attending this course orientate themselves towards Polytechnic and Computer Science university departments, where a basic, pre-existing knowledge of Algorithms and Data Structures is a valuable asset; such knowledge can be constructed within the framework of ADPE courses (Application Development in a Programming Environment), via carefully chosen algorithms of Informatics, as well as the latter's implementation using an array of different programming languages. In contrast to exam-oriented education, the study, and implementation of real algorithms, the discovery of the scientific research behind them as well as the evaluation of their complexity introduces authenticity to the student's algorithmic thinking.

Keywords: Algorithms, Data Structures, Programming.