

CS4FN: Πληροφορική για Διασκέδαση

Υπολογιστική Σκέψη: Αναζητώντας τα Λόγια

- Βοηθώντας ανθρώπους με σύνδρομο εγκλεισμού
- Τί είναι η υπολογιστική σκέψη;
- Πώς βρίσκουν οι υπολογιστές αυτό που αναζητούν;
- Πώς μπορούμε να ξέρουμε ποιος αλγόριθμος είναι καλύτερος;

Αναζητώντας τα Λόγια

Μια από τις χειρότερες δυνατές παθήσεις είναι το σύνδρομο εγκλεισμού. Σε αφήνει εντελώς παράλυτο, ικανό μόνο να ανοιγοκλείνεις το ένα σου μάτι. Ένα ευφυές μυαλό παραμένει φυλακισμένο σ' ένα άχρηστο σώμα, ικανό να αισθανθεί τα πάντα, αλλά ανίκανο να επικοινωνήσει. Μπορεί να συμβεί ξαφνικά στον οποιονδήποτε, σαν αποτέλεσμα ενός εγκεφαλικού. Ένας προφανής τρόπος να βοηθήσετε ανθρώπους με σύνδρομο εγκλεισμού θα ήταν να γίνετε γιατροί ή νοσοκόμοι, αλλά πως θα μπορούσε να βοηθήσει ένας επιστήμονας της πληροφορικής;

Δεν υπάρχει θεραπεία για το σύνδρομο εγκλεισμού, οπότε δεν υπάρχουν πολλά πράγματα που μπορεί να κάνει το ιατρικό προσωπικό, εκτός από το να φροντίζει ώστε οι ασθενείς να αισθάνονται άνετα. Ένα μεγάλο πρόβλημα που θα πρέπει να αντιμετωπιστεί ωστόσο είναι πως θα μπορούσε κανείς να βοηθήσει αυτούς τους ανθρώπους να «μιλήσουν». Ένας επιστήμονας της πληροφορικής θα μπορούσε προφανώς να αναπτύξει κάποια νέα τεχνολογία για να βοηθήσει. Ωστόσο, με λίγη υπολογιστική σκέψη μπορούμε να δώσουμε μια πολύ καλύτερη απάντηση από ένα απλό «χρειαζόμαστε την τεχνολογία».

«Το Σκάφανδρο κι η Πεταλούδα» είναι ένα βιβλίο που εμπνέει απίστευτη δύναμη. Είναι η αυτοβιογραφία του Ζαν-Ντομινίκ Μπομπύ (Jean-Dominique Bauby), την οποία έγραψε αφού ξύπνησε σε ένα κρεβάτι νοσοκομείου, σε πλήρη παραλυσία. Στο βιβλίο περιγράφει πως είναι να ζεις με σύνδρομο εγκλεισμού. Όμως είχε έναν τρόπο να επικοινωνεί, όχι μόνο για να γράψει το βιβλίο, αλλά και με το ιατρικό προσωπικό, τους φίλους και τους συγγενείς. Το κατάφερε χωρίς να χρησιμοποιήσει οποιοδήποτε τεχνολογικό μέσο. Πώς;

Βάλτε τον εαυτό σας στη θέση του, ξυπνώντας στο κρεβάτι του νοσοκομείου. Πώς θα μπορούσατε να επικοινωνήσετε; Πώς θα μπορούσατε να γράψετε ένα ολόκληρο βιβλίο; Έχετε μόνο μια βοηθό με χαρτί και μολύβι για να καταγράψει τις «λέξεις» σας. Το μόνο που μπορείτε να κάνετε είναι να ανοιγοκλείσετε το ένα σας μάτι, να βλεφαρίσετε. Δεν μπορείτε να κινηθείτε με κανέναν άλλον τρόπο. Αυτό σημαίνει ότι δεν μπορείτε και να μιλήσετε. Μπορείτε να σκεφτείτε έναν τρόπο επικοινωνίας;

Μπορείτε να βρείτε το πρωτότυπο κείμενο στην ιστοσελίδα teachinglondoncomputing.org, μαζί με μια περισσότερη "γυαλιστερή" έκδοση και συνοδευτικές δραστηριότητες για την τάξη.

Απλό σαν την Αλφαβήτα

Αυτό που χρειάζεται είναι ένας τρόπος να μετατρέπονται τα βλεφαρίσματα σε γράμματα. Αρχικά, ίσως συμφωνήσετε ότι ένα βλεφάρισμα μπορεί να σημαίνει “Α”, δύο να σημαίνουν “Β”, κτλ. Η βοηθός απλά χρειάζεται να μετρά τα βλεφαρίσματα και να καταγράφει το αντίστοιχο γράμμα.

Αναπτύσσοντας μια τέτοια ιδέα, σκεφτόμαστε υπολογιστικά – είναι ο τρόπος με τον οποίο οι επιστήμονες της πληροφορικής λύνουν προβλήματα. Πρόκειται μάλιστα για ένα είδος υπολογιστικής σκέψης που ονομάζεται **«αλγοριθμική σκέψη»**. Ένας επιστήμονας της πληροφορικής αποκαλεί τον συμφωνημένο τρόπο επικοινωνίας **«αλγόριθμο»**: μια ακολουθία από βήματα που πρέπει να ακολουθηθούν με δεδομένη σειρά ώστε να επιτευχθεί ένας στόχος (εδώ ο στόχος είναι η μετάδοση γραμμάτων και λέξεων). Η αλγοριθμική σκέψη αφορά την κατασκευή αλγορίθμων για την επίλυση προβλημάτων.

Η ομορφιά των αλγορίθμων είναι ότι οποιοσδήποτε εκτελεί τα βήματά τους δεν χρειάζεται να καταλαβαίνει τι κάνει. Με τον δικό μας αλγόριθμο, είναι εύλογο ότι η βοηθός γνωρίζει τι κάνει και γιατί, όμως το βιβλίο θα μπορούσε να γραφτεί ακόμα κι αν δεν γνώριζε. Το μόνο που χρειάζεται να κάνει η βοηθός είναι να μετρά τα βλεφαρίσματα και να σημειώνει τα γράμματα. Θα μπορούσαμε ακόμα και να της δώσουμε έναν πίνακα με την αντιστοιχία πλήθος βλεφαρισμάτων και γραμμάτων ώστε να μη χρειάζεται καθόλου να σκεφτεί. Η ομορφιά των αλγορίθμων είναι ότι επιτρέπουν στους ανθρώπους να κάνουν πράγματα «μηχανικά» -- και αυτό βέβαια σημαίνει ότι και οι υπολογιστές μπορούν να ακολουθήσουν τυφλά αυτές τις οδηγίες.

Ο αλγόριθμός μας για την επικοινωνία ουσιαστικά αποτελείται από δύο σκέλη. Υπάρχει ένα σκέλος το οποίο πρέπει να ακολουθήσει ο Μπομπύ (να ανοιγοκλείνει το μάτι του όσες φορές πρέπει) και ένα σκέλος για την βοηθό (να μετρά τα βλεφαρίσματα και να σημειώνει το αντίστοιχο γράμμα όταν τα βλεφαρίσματα σταματήσουν). Στην πραγματικότητα, οι επιστήμονες της Πληροφορικής έχουν ένα ειδικό όνομα για αυτό το είδος αλγορίθμου που μεταφέρει πληροφορία ανάμεσα σε ανθρώπους ή υπολογιστές – ονομάζεται **«πρωτόκολλο»**. Αν και οι δύο πλευρές ακολουθήσουν το δικό τους μέρος του πρωτοκόλλου, τότε οι λέξεις που σκέφτεται ο Μπομπύ θα καταλήξουν γραμμένες σε ένα φύλλο χαρτί. Αν οποιοσδήποτε από τους δύο κάνει λάθος – για παράδειγμα αν χάσει το μέτρημα, οπότε δεν ακολουθεί πιστά το πρωτόκολλο – τότε το μήνυμα δεν θα μεταφερθεί. Το σπουδαίο πράγμα με τους υπολογιστές είναι ότι δεν κάνουν τέτοια σφάλματα. Ακολουθούν τις οδηγίες τους πιστά, κάθε φορά.

Η αλγοριθμική σκέψη είναι ένα είδος επίλυσης προβλημάτων, στο οποίο το ζητούμενο δεν είναι απλά μια απάντηση, όπως «42», αλλά μια σειρά βημάτων που οποιοσδήποτε μπορεί να ακολουθήσει (συμπεριλαμβανομένων και των υπολογιστών) για να πάρει απαντήσεις. Μόλις αναπτύξαμε μια τέτοια λύση για τον Μπομπύ. Δεν μας λέει απλά τι προσπαθεί να επικοινωνήσει την συγκεκριμένη στιγμή, αλλά μας παρέχει έναν τρόπο να υπολογίσουμε οποτεδήποτε τι θέλει να πει. Ακούγεται όμως αρκετά αργός. Ίσως να υπάρχει ένας καλύτερος τρόπος. Η προσπάθεια βελτίωσης των λύσεων αποτελεί επίσης μέρος της αλγοριθμικής σκέψης.

Πως το έκανε ο Μπομπύ

Ο Μπομπύ είχε πράγματι έναν καλύτερο τρόπο, έναν καλύτερο αλγόριθμο. Πρέπει να θυμόμαστε ότι η βοηθός μπορεί να μιλάει, οπότε αυτό μπορούμε να το εκμεταλλευτούμε. Στον αλγόριθμο που χρησιμοποιήθηκε, η βοηθός διάβαζε δυνατά το αλφάβητο «Α... Β... Γ...» Όταν εκφωνούσε το γράμμα που σκεφτόταν ο Μπομπύ, τότε αυτός βλεφάριζε. Η βοηθός κατέγραφε το γράμμα και ξεκινούσε και πάλι από την αρχή, γράμμα προς γράμμα. Δοκιμάστε το με έναν φίλο σας – επικοινωνήστε με αυτόν τον τρόπο τα αρχικά του ονόματός σας. Μετά σκεφτείτε πως θα ήταν αν δεν είχατε κανέναν άλλον τρόπο να μιλήσετε. Ελπίζω το όνομά σας να μην είναι Ωρίωνας ή Ωγυγία.

Όταν το δοκιμάσετε, ίσως συνειδητοποιήσετε ότι υπάρχουν και άλλα προβλήματα που θα πρέπει να λύσουμε για να δουλεύει πραγματικά αυτή η μέθοδος. Δοκιμάστε το μερικές φορές και πιθανώς να επινοήσετε κι άλλους τρόπους να βελτιώσετε τον αλγόριθμο. Τί άλλο μπορείτε να σκεφτείτε;

Τακτοποιώντας τις Λεπτομέρειες

Ένα πράγμα που μπορεί να εντοπίσατε είναι πως υπάρχουν περισσότερα σύμβολα που θα πρέπει να συμπεριλάβουμε, εκτός από τα 24 γράμματα. Χρειαζόμαστε κενά, αριθμητικά ψηφία, τελείες, κτλ. Κάτι ακόμα που πρέπει να σκεφτούμε είναι τι θα συμβεί αν υπάρξει ένα λάθος βλεφάρισμα. Χρειαζόμαστε έναν τρόπο να πούμε «αγνόησε το τελευταίο βλεφάρισμα και ξεκίνα τα γράμματα από την αρχή». Ένας τρόπος θα ήταν να συμφωνήσουμε ότι ένα γρήγορο διπλό βλεφάρισμα σημαίνει ακριβώς αυτό. Ίσως εσείς να σκεφτήκατε και άλλα προβλήματα που χρειάζεται να λυθούν.

Η αλγοριθμική σκέψη αφορά την ενασχόληση με όλα αυτά τα προβλήματα και την επίλυσή τους. Αφορά την συνειδητοποίηση ότι υπάρχουν πολλοί τρόποι να γίνει κάτι και μετά την εύρεση του καλύτερου τρόπου για την συγκεκριμένη περίπτωση. Παρατηρήστε επίσης ότι ένα από τα προβλήματα ήταν τι κάνουν οι άνθρωποι. Θεωρητικά, η λύση μας λειτουργεί: απλά ανοιγοκλείνεις το μάτι σου στην σωστή στιγμή! Θα μπορούσαμε να πούμε αλαζονικά ότι οι άνθρωποι θα πρέπει να εκτελέσουν σωστά τα βήματα και είναι δικό τους σφάλμα αν δεν το κάνουν. Στην πράξη, μερικές φορές θα βλεφαρίσουν σε λάθος στιγμή. Είναι καλύτερο να λύσουμε το πρόβλημα με έναν τρόπο που βολεύει τους ανθρώπους. Στο κάτω-κάτω, έναν άνθρωπο προσπαθούμε να βοηθήσουμε! Η υπολογιστική σκέψη σχετίζεται και με την **κατανόηση των ανθρώπων**.

Ακόμα Καλύτερα

Η διαδικασία μπορεί να επιταχυνθεί αν συνειδητοποιήσουμε ότι μερικές φορές μπορούμε να μαντέψουμε μια λέξη πριν ακόμα ολοκληρωθεί. Αν έχεις ήδη τα γράμματα «π-ο-δ-η-λ-» τότε μπορείς να στοιχηματίσεις με αρκετή σιγουριά ότι η λέξη είναι «ποδήλατο». Οπότε θα μπορούσαμε να αλλάξουμε τους κανόνες και να επιτρέπουμε στη βοηθό να μαντεύει. Ο Μπομπύ του μηνύματος θα χρειαζόταν έναν τρόπο να απορρίπτει τις λέξεις που μαντεύονται λανθασμένα. Ίσως ο κανόνας θα μπορούσε να ορίζει ότι αν μια λέξη είναι σωστή τότε ανοιγοκλείνει το μάτι ενώ αν δεν είναι τότε δεν κάνει τίποτα. Βέβαια έτσι δουλεύει και η σύνταξη κειμένου με πρόβλεψη (predictive texting), δηλαδή ο αλγόριθμος που χρησιμοποιούν τα κινητά τηλέφωνα όταν γράφετε μηνύματα. Αν το σκεφτήκατε, ίσως από εκεί να σας ήρθε η ιδέα. Αν το σκεφτήκατε, τότε χρησιμοποιήσατε άλλη μια ικανότητα της υπολογιστικής σκέψης: τον **μετασχηματισμό προβλημάτων**. Συχνά προκύπτει ότι ορισμένα προβλήματα είναι ουσιαστικά ίδια με άλλα που έχετε αντιμετωπίσει προηγουμένως, σε

διαφορετικές καταστάσεις. Αν διαθέτετε ήδη μια λύση για τα προηγούμενα προβλήματα τότε μπορείτε απλά να τη χρησιμοποιήσετε και στα νέα προβλήματα που αντιμετωπίζετε.

Οι αλγόριθμοι είναι ακριβώς ένας τέτοιος τρόπος να δίνει κανείς γενικές λύσεις. Ένα κινητό τηλέφωνο αντιμετωπίζει το ίδιο πρόβλημα του να προβλέπει ποιες λέξεις πληκτρολογούνται με την βοήθ που πρέπει να μαντέψει ποια λέξη σκέφτεται το άτομο με σύνδρομο εγκλεισμού. Μόλις το αντιληφθούμε αυτό, οποιαδήποτε λύση βρούμε για το ένα πρόβλημα μπορεί να χρησιμοποιηθεί για το άλλο.

Η βοήθ του Μπομπύ χρησιμοποίησε πράγματι μια εκδοχή της σύνταξης κειμένου με πρόβλεψη. Ο Μπομπύ κατάλαβε επίσης ότι ο αρχικός αλγόριθμος (ας τον ονομάσουμε αλγόριθμο ΑΒΓ) μπορούσε να βελτιωθεί και με διαφορετικό τρόπο. Πριν καθλωθεί σε εκείνο το κρεβάτι, ήταν ο αρχισυντάκτης του γαλλικού γυναικείου περιοδικού Elle, οπότε ήξερε πολλά πράγματα για τη γλώσσα. Γνώριζε ότι ορισμένα γράμματα είναι πιο συνηθισμένα από άλλα στις ανθρώπινες γλώσσες. Για παράδειγμα, στα γαλλικά και στα αγγλικά το Ε είναι το πιο κοινό γράμμα. Έβαλε λοιπόν τη βοήθ να διαβάσει τα γράμματα με σειρά ανάλογη του πόσο κοινά είναι, της συχνότητάς τους. Στα ελληνικά, η σειρά είναι «Α... Ο... Τ... Ε...». Στα γαλλικά, είναι «Ε... Σ... Α... Ρ...» Μιλούσε γαλλικά, οπότε αυτή την σειρά χρησιμοποίησε. Με τον τρόπο αυτό, η βοήθ έφτανε πιο γρήγορα στα πιο συνηθισμένα γράμματα.

Ένα παρόμοιο κόλπο έχει χρησιμοποιηθεί ανά τους αιώνες για να «σπάσουν» μυστικούς κώδικες. Ο αλγόριθμος που χρησιμοποιεί τις συχνότητες των γραμμάτων εφευρέθηκε από Άραβες λόγιους περισσότερα από χίλια χρόνια πριν. Μάλιστα η Μαίρη, η Βασίλισσα των Σκωτσέζων (Mary Queen of Scots) αποκεφαλίστηκε επειδή ο αρχικατάσκοπος της Βασίλισσας Ελισάβετ, ο σερ Φράνσις Γουόλζινγκχαμ (sir Francis Walsingham), ήταν καλύτερος στην υπολογιστική σκέψη από εκείνη. Αυτή, ωστόσο, είναι μια διαφορετική ιστορία. Η ιδέα του Μπομπύ να χρησιμοποιήσει την ανάλυση των συχνοτήτων των γραμμάτων είναι άλλο ένα παράδειγμα **μετασχηματισμού προβλημάτων**. Αν αναγνωρίσουμε ότι το σπάσιμο ενός κώδικα και το να μαντεύουμε γράμματα είναι παρόμοια προβλήματα, τότε θα δούμε ότι η λύση με τις συχνότητες των γραμμάτων που εφευρέθηκε για το ένα, μπορεί να χρησιμοποιηθεί για το άλλο.

Πόσο γρήγορο είναι αυτό;

Ας επιστρέψουμε στον αλγόριθμο του Μπομπύ. Σίγουρα έχουμε βελτιώσει τα πράγματα. Ο νέος τρόπος πρέπει να είναι καλύτερος από την αρχική μας ιδέα. Όμως μια προφανής ερώτηση είναι πόσο γρήγορος είναι πραγματικά. Πόσο χρόνο του πήρε να γράψει αυτό το βιβλίο; Είναι το καλύτερο που μπορούμε να κάνουμε ή θα μπορούσαμε να έχουμε σκεφτεί έναν ακόμα καλύτερο αλγόριθμο και να τον έχουμε βοηθήσει να γράψει το βιβλίο πιο εύκολα;

Χρειαζόμαστε έναν τρόπο να μετράμε πόσο καλός είναι ένας αλγόριθμος. Θα μπορούσαμε να το κάνουμε πειραματικά, να χρονομετρούμε πόσο χρειάζεται ο καθένας για να μεταδώσει ένα συγκεκριμένο μήνυμα. Θα μπορούσαμε να το κάνουμε πολλές φορές με διαφορετικούς ανθρώπους και να διαπιστώσουμε ποιος τρόπος είναι γρηγορότερος κατά μέσο όρο. Κάτι τέτοιο θα απαιτούσε πολύ χρόνο και κόπο. Υπάρχει καλύτερος τρόπος.

Μπορούμε να σκεφτούμε **αναλυτικά**. Θα χρησιμοποιήσουμε απλά μαθηματικά για να υπολογίσουμε μια απάντηση. Πρώτον, αντί να σκεφτόμαστε το χρόνο, ας σκεφτούμε τη δουλειά που απαιτείται για την μετάδοση ενός γράμματος. Αν μετρήσουμε πόσα γράμ-

ματα του αλφαβήτου χρειάζεται να πει η βοηθός μέχρι να βρει το γράμμα που σκέφτεται ο Μπομπύ, τότε μπορούμε να μετατρέψουμε αυτήν την ποσότητα σε χρόνο, αν γνωρίζουμε πόσο χρόνο χρειάζεται για να ειπωθεί ένα γράμμα. Κάνουμε κάτι που ονομάζεται «αφαίρεση». Είναι άλλο ένα κομμάτι της υπολογιστικής σκέψης, το οποίο χρησιμοποιείται για την απλοποίηση προβλημάτων. Η αφαίρεση είναι απλά μια λέξη που σημαίνει «κρύβω κάποιες από τις λεπτομέρειες». Η ιδέα χρησιμοποιείται σε όλο το φάσμα της Πληροφορικής για να διευκολύνει την επίλυση προβλημάτων. Εδώ, χρησιμοποιούμε το πλήθος των γραμμάτων που εκφωνούνται από τη βοηθό σαν μια αφαίρεση του πραγματικού χρόνου που απαιτείται.

Οπότε, πως υπολογίζουμε το πλήθος των γραμμάτων που πρέπει να ειπωθούν; Υπάρχουν αρκετές ερωτήσεις που μπορούμε να ρωτήσουμε. Η απλούστερη είναι: ποια είναι η καλύτερη περίπτωση; Ποιο είναι το μικρότερο πιθανό πλήθος γραμμάτων που θα χρειαστεί να πει η βοηθός για να γραφτεί το βιβλίο; Θα μπορούσαμε επίσης να εξετάσουμε τη χειρότερη περίπτωση. Αν είμαστε άτυχοι, πόσο άσχημα θα μπορούσαν να είναι τα πράγματα; Τέλος, μπορούμε να κοιτάξουμε την μέση περίπτωση. Αυτή θα μας παρέχει μια ρεαλιστική εκτίμηση σχετικά με το πόση δουλειά χρειάστηκε.

Η καλύτερη και η χειρότερη

Ας αρκεστούμε στην περίπτωση κατά την οποία μεταδίδονται μόνο γράμματα, χωρίς αριθμητικά ψηφία και σημεία στίξης. Θα αναλύσουμε τον απλό αλγόριθμο ABΓ, όπου η βοηθός λέει “Α... Β... Γ...” και ο πομπός του μηνύματος βλεφαρίζει στο σωστό γράμμα.

Στην καλύτερη περίπτωση, όλο το βιβλίο θα αποτελούνταν αποκλειστικά από Α: «ΑΑΑΑΑΑ» (πιθανώς εκφράζοντας τον πόνο που αισθάνεται). Για να μεταδοθεί ένα Α, η βοηθός εκφωνεί απλά ένα γράμμα (μια ερώτηση) και έχουμε την απάντηση. Πολλαπλασιάστε αυτό με το πλήθος των γραμμάτων στο βιβλίο και έχουμε την καλύτερη περίπτωση για το κόστος συγγραφής του βιβλίου.

Στη χειρότερη περίπτωση, πιθανώς μια ιστορία με συνεχή επιφωνήματα, το «ΩΩΩΩΩ» χρειάζεται 24 ερωτήσεις για να φτάσει κανείς σε κάθε γράμμα. Αυτά είναι τα όρια σχετικά με το πόσο κοστίζει να μεταφέρει κανείς οποιοδήποτε μήνυμα. Είναι πάντα το λιγότερο 1 και το πολύ 24 ερωτήσεις ανά γράμμα.

Μια ακριβέστερη εκτίμηση θα ήταν το μέσο πλήθος ερωτήσεων ανά γράμμα: η μέση περίπτωση. Αυτό είναι εύκολο να υπολογιστεί. Σε ένα μεγάλο μήνυμα, για κάθε Α θα υπάρχει κατά μέσο όρο και ένα Ω κάπου αλλού μέσα στο μήνυμα. Για κάθε Β θα υπάρχει ένα Ψ και ούτω καθεξής. Αυτό σημαίνει ότι, κατά μέσο όρο σε όλο το βιβλίο, περίπου 12 ερωτήσεις θα γίνουν για κάθε γράμμα που υπαγορεύεται. Πολλαπλασιάστε το πλήθος των γραμμάτων του βιβλίου με το 12 και έχετε μια εκτίμηση για το πόσος κόπος χρειάστηκε για να γραφτεί. Πολλαπλασιάστε αυτό με τον μέσο χρόνο που χρειάζεται η βοηθός για να πει ένα γράμμα και έχετε τον χρόνο που χρειάστηκε για να γραφτεί το βιβλίο.

Η τροποποίηση του Μπομπύ, να ρωτάμε πρώτα για τα κοινά γράμματα, βελτιώνει κάπως τα πράγματα – πιθανώς ο μέσος όρος να μειωθεί στα 9 ή 10 εκφωνημένα γράμματα. Μπορούμε να κάνουμε τον υπολογισμό με μεγαλύτερη ακρίβεια, χρησιμοποιώντας τις συχνότητες των γραμμάτων. Οπότε είναι μια βελτίωση, αλλά η χειρότερη περίπτωση για ένα γράμμα είναι και πάλι 24.

Ωστόσο, όπως γνωρίζει κάθε επιστήμονας της πληροφορικής, μπορούμε να τα πάμε πολύ καλύτερα. Είναι δυνατόν να καταλήξουμε σε ένα γράμμα χρησιμοποιώντας μόνο 5 ερωτήσεις. Εγγυημένα! Αυτή δεν είναι η μέση περίπτωση, είναι η χειρότερη περίπτωση.

Μπορείτε να σκεφτείτε ποιες είναι οι 5 ερωτήσεις που χρειάζεται να κάνετε;

Κάν' το σε 5

Είτε βρήκατε την σωστή απάντηση είτε όχι, σας εγγυώμαι ότι ξέρετε το κατάλληλο είδος ερώτησης, αρκεί να εξετάσουμε ένα διαφορετικό πρόβλημα.

Ας παίξουμε το παιχνίδι των 20 ερωτήσεων. Είναι ένα παιδικό παιχνίδι στο οποίο ο ένας σκέφτεται ένα διάσημο πρόσωπο και οι υπόλοιποι προσπαθούν να μαντέψουν ποιο είναι κάνοντας ερωτήσεις. Όμως οι απαντήσεις μπορεί να είναι μόνο Ναι ή Όχι. Παίξτε το παιχνίδι με έναν φίλο σας και σκεφτείτε καθώς παίζετε τι είδους ερωτήσεις κάνετε.

Ας δούμε πως θα μπορούσε να εξελιχθεί ένα παιχνίδι.

“Είναι γυναίκα;”

Όχι.

“Έχει πεθάνει;”

Ναι.

“Είναι αστέρι του σινεμά;”

Όχι.

“Είναι από την Αγγλία;”

Όχι.

“Είναι από την Αμερική;”

Όχι.

“Είναι από την Ασία;”

Ναι.

“Είναι από την Ινδία;”

Ναι.

“Είναι πολιτικός;”

Ναι.

“Είναι ο Γκάντι;”

Ναι.

Το πιθανότερο είναι πως όταν παίξατε το παιχνίδι κάνατε παρόμοιες ερωτήσεις. Σχεδόν σίγουρα δεν ξεκινήσατε ρωτώντας “Είναι ο Τζέιμς Μποντ;”, “Είναι ο Μίκυ Μάους;”, “Είναι η Βασίλισσα της Αγγλίας;” Δεν θα φτάνατε ποτέ στην απάντηση σε 20 ερωτήσεις με αυτόν τον τρόπο. Τέτοιου είδους ερωτήσεις τις κάνετε μόνο στο τέλος, όταν είστε αρκετά σίγουροι. Πιθανότατα ξεκινήσατε με μια ερώτηση όπως “Είναι άντρας;” Γιατί αυτή είναι μια καλή ερώτηση; Επειδή αποκλείει τα μισά ενδεχόμενα, όποια κι αν είναι η απάντηση. Αν ρωτήσετε “Είναι ο Τζέιμς Μποντ;” τότε αποκλείετε εκατομμύρια ενδεχόμενα αν έχετε δίκιο, αλλά αν δεν έχετε (που είναι πολύ πιθανότερο) αποκλείετε μόνο ένα πρόσωπο. Θα πρέπει να είστε τυχερός όσο ο νικητής του λαχείου για να τα πάτε καλά με αυτόν τον τρόπο. Οπότε το μυστικό στο παιχνίδι των 20 ερωτήσεων είναι να κάνετε ερωτήσεις που αποκλείουν κάθε φορά τα μισά ενδεχόμενα.

Πόσο καλό είναι αυτό;

Πόσο καλό είναι αυτό; Λοιπόν, ας υποθέσουμε ότι αρχικά μπορεί να σκέφτομαι ένα εκατομμύριο ανθρώπους. Αν αποκλείω τους μισούς με κάθε ερώτηση, πόσες ερωτήσεις θα

χρειαστούν; Μετά από μια ερώτηση πέφτουμε στους 500.000, μετά από δύο στους 250.000, μετά στους 125.000, κλπ. Μετά από 10 ερωτήσεις έχουν μείνει περίπου 1000 άνθρωποι από το αρχικό ένα εκατομμύριο. Άλλη μια ερώτηση και μένουν 500, 250, ... κλπ και στην 20η ερώτηση η νίκη είναι εγγυημένη.

Επομένως, με τις κατάλληλες ερωτήσεις, στη χειρότερη περίπτωση χρειάζονται 20 ερωτήσεις για να βρει κανείς το άτομο που σκέφτομαι, μέσα από ένα εκατομμύριο ενδεχόμενα. Συγκρίνετε αυτό με το γεγονός ότι προηγουμένως είπαμε πως χρειάζονται 12 ερωτήσεις (και 24 στη χειρότερη περίπτωση) για να εντοπίσουμε ένα από τα 24 γράμματα του αλφαβήτου. Οι απαντήσεις με Ναι και Όχι δεν διαφέρουν σε τίποτα από το βλεφάρισμα ή όχι του ματιού. Όταν ρωτούσαμε “Είναι το Α; Είναι το Β;” κάναμε το αντίστοιχο του να ρωτάμε “Είναι ο Νέλσον Μαντέλα; Είναι ο Μίκυ Μάους;” Προσπαθούσαμε να εντοπίσουμε ένα από τα πολλά πιθανά ενδεχόμενα. Στην πραγματικότητα πρόκειται για το ίδιο πρόβλημα!

Ένας Νέος Αλγόριθμος

Εδώ εμφανίζεται και πάλι η ιδέα του **μετασχηματισμού προβλημάτων**. Αν πρόκειται για το ίδιο πρόβλημα, τότε βεβαίως η ίδια στρατηγική θα μας δώσει μια καλύτερη λύση από αυτές που δοκιμάσαμε μέχρι στιγμής. Ποιο είναι το ισοδύναμο της «διχοτόμησης» όταν ψάχνουμε για γράμματα του αλφαβήτου; Πρέπει να διχοτομούμε το αλφάβητο με κάθε ερώτηση. Η προφανής πρώτη ερώτηση είναι “Είναι πριν το Ν;” Η επόμενη ερώτηση εξαρτάται από την απάντηση στην προηγούμενη. Αν η απάντηση ήταν Ναι, τότε στη συνέχεια ρωτάμε “Είναι πριν το Η;”. Αν η απάντηση ήταν Όχι, τότε ρωτάμε “Είναι πριν το Τ;” κ.ο.κ. Με τον τρόπο αυτό, είμαστε σίγουρο ότι θα προσδιορίσουμε οποιοδήποτε γράμμα του αλφαβήτου σκέφτεται ο άλλος με 5 μόνο ερωτήσεις.

Μπορούμε μάλιστα να βελτιώσουμε ακόμα περισσότερο την κατάσταση χρησιμοποιώντας το τέχνασμα με την ανάλυση συχνότητας. Με 24 γράμματα θα μπορούσαμε, για παράδειγμα, να φτάσουμε στο γράμμα Α σε μόνο τρεις ερωτήσεις. Και μπορούμε και πάλι να χρησιμοποιήσουμε προβλέψεις για να μαντέψουμε λέξεις που έχουν σχηματιστεί μερικώς. Όλες αυτές οι λύσεις ισχύουν και εδώ.

Αλγόριθμοι αναζήτησης

Η λύση μας ήταν εφαρμόσιμη γιατί τα δύο προβλήματα ήταν ουσιαστικά τα ίδια. Είναι προβλήματα «αναζήτησης»: με δεδομένη μια σειρά αντικειμένων, βρες το ένα συγκεκριμένο αντικείμενο που αναζητούμε. Οι λύσεις σε αυτά τα προβλήματα ονομάζονται «**αλγόριθμοι αναζήτησης**». Είναι εγγυημένοι τρόποι να βρίσκουμε αυτό που ψάχνουμε. Η πρώτη προσέγγιση, να ελέγχουμε κάθε ενδεχόμενο με την σειρά (Είναι το Α; Είναι το Β; Είναι ο Νέλσον Μαντέλα; Είναι ο Τζέιμς Μποντ; ...) είναι ένας αλγόριθμος που ονομάζεται **σειριακή αναζήτηση**. Μερικές φορές είναι το καλύτερο που μπορείς να κάνεις. Για παράδειγμα, αν είστε μάρτυρες σε μια ληστεία και η αστυνομία στήσει τους υπόπτους στη σειρά για αναγνώριση, δεν θα μπορούσατε να κάνετε κάτι καλύτερο από την σειριακή αναζήτηση. Θα ελέγχατε κάθε πρόσωπο με την σειρά, μέχρι να εντοπίσετε τον ένοχο! Η σειριακή αναζήτηση δουλεύει καλά όταν τα αντικείμενα ανάμεσα στα οποία ψάχνετε δεν είναι διατεταγμένα με συγκεκριμένη σειρά. Αν αναζητάτε μια μπλουζα που θα μπορούσε να βρίσκεται σε οποιοδήποτε από τα συρτάρια σας, τότε ξεκινήστε από το πάνω συρτάρι και ελέγξτε τα με την σειρά.

Ο άλλος αλγόριθμός μας βασιζόταν σε ερωτήσεις διχοτόμησης: “Είναι πριν το Ν; Είναι γυναίκα;” Η εύρεση ερωτήσεων διχοτόμησης είναι μια γενική μέθοδος επίλυσης προβλη-

μάτων που ονομάζεται «**διαίρει και βασιλευε**». Αν αναπτύξετε μια τέτοια λύση σε ένα πρόβλημα, είναι πολύ πιθανόν να είναι γρήγορη, καθώς επαναλαμβανόμενες διχοτομήσεις οδηγούν σε μια απάντηση πολύ γρήγορα και πολύ πιο γρήγορα από το να ελέγχει κανείς μια λύση τη φορά. Ο απλούστερος αλγόριθμος αναζήτησης που βασίζεται στο διαίρει και βασιλευε ονομάζεται «**δυναδική αναζήτηση**». Φανταστείτε ότι τοποθετείτε σε συγκεκριμένη σειρά όλα τα πράγματα ανάμεσα στα οποία κάνετε αναζήτηση, τα μικρότερα σε μια άκρη και τα μεγαλύτερα στην άλλη. Με τη δυναδική αναζήτηση πηγαίνεις στη μέση και ελέγχεις αν αυτό που αναζητάς είναι πριν ή μετά. Απορρίπτεις λοιπόν το ένα μισό και συνεχίζεις με τον ίδιο τρόπο στο τμήμα που απομένει. Συνεχίζεις έτσι μέχρι να απομείνει μόνο ένα αντικείμενο, αυτό το οποίο αναζητάς. Αυτό είναι αρκετά παρεμφερές με την τακτική που ακολουθούμε όταν θέλουμε να βρούμε ένα όνομα σε έναν μεγάλο έντυπο τηλεφωνικό κατάλογο. Σίγουρα δεν θα ξεκινούσαμε από την πρώτη σελίδα και δεν θα ελέγχαμε κάθε όνομα με την σειρά, μέχρι να βρούμε αυτό που ψάχνουμε.

Υπάρχουν πολλοί περισσότεροι αλγόριθμοι αναζήτησης από αυτούς τους δύο. Για παράδειγμα, πως ψάχνει η Google μέσα σε κάθε ιστοσελίδα μέσα σε κλάσματα του δευτερολέπτου; Χρειάζεται έναν ακόμα καλύτερο αλγόριθμο!

Οι αλγόριθμοι αναζήτησης κάνουν χρήση άλλης μιας μορφής **αφαίρεσης**. Αφαιρούμε τις λεπτομέρειες του συγκεκριμένου προβλήματος και το αντιμετωπίζουμε απλά σαν ακόμα ένα πρόβλημα αναζήτησης. Τότε ο αλγόριθμός μας αποτελεί μια έτοιμη λύση για πολλά προβλήματα. Με άλλα λόγια, αν έχουμε σκεφτεί μια στρατηγική για να κερδίζουμε στο παιχνίδι των 20 ερωτήσεων, τότε μπορούμε να **γενικεύσουμε** αυτή τη λύση στην ιδέα του διαίρει και βασιλευε και να έχουμε μια γενική στρατηγική που λειτουργεί και για άλλα προβλήματα.

Βελτιώνοντας την ζωή του Μπομπύ

Επομένως, ο Μπομπύ έπρεπε να βάλει τη βοηθό του να κάνει ερωτήσεις διχοτόμησης. Σκεφτείτε το. Πέντε ερωτήσεις στην χειρότερη περίπτωση, αντί για 12 κατά μέσο όρο, πολλαπλασιασμένες επί όλα τα γράμματα στο βιβλίο του. Και φυσικά δεν είναι μόνο το βιβλίο, είναι και οι συνομιλίες με τους φίλους και τους συγγενείς του, τους γιατρούς και τις νοσοκόμες. Πόσο ευκολότερη θα ήταν η ζωή του αν ήξερε κάτι από Πληροφορική!

Η Αλγοριθμική Σκέψη Προηγείται

Αυτό που πρέπει να παρατηρήσει κανείς είναι πως δεν ασχοληθήκαμε καθόλου με την τεχνολογία. Όλα αφορούσαν δύο ανθρώπους που «συνομιλούν». Τώρα που έχουμε βρει έναν καλό τρόπο, έναν καλό αλγόριθμο, μπορούμε να σκεφτούμε πως θα τον αυτοματοποιήσουμε με την κατάλληλη τεχνολογία. Μπορούμε να αναπτύξουμε ένα σύστημα παρακολούθησης των ματιών (eye tracking) που ανιχνεύει βλεφαρίσματα ή ίσως έναν ειδικό σκούφο με ηλεκτρόδια (electrode cap) που μπορεί να ανιχνεύσει πότε σκέφτεται Ναι ή Όχι. Η ουσία είναι πως όποια τεχνολογία κι αν χρησιμοποιήσουμε, εσωτερικά θα χρειαζόταν έναν αλγόριθμο αναζήτησης. Αν διαλέξουμε έναν λανθασμένο αλγόριθμο τότε, όσο καλή κι αν είναι η τεχνολογία, η επικοινωνία θα είναι και πάλι αργή: 12 ερωτήσεις ανά γράμμα, αντί για 5. Δεν έχει καμία σημασία αν η βοηθός είναι ένας υπολογιστής ή ένας άνθρωπος. Αν δεν είχαμε πρώτα σκεφτεί τους αλγόριθμους θα είχαμε αναπτύξει ένα απογοητευτικά αργό σύστημα. Η Πληροφορική δεν αφορά μόνο την τεχνολογία, αλλά σχετίζεται και με την υπολογιστική σκέψη που χρησιμοποιείται για να αναπτυχθούν καλές λύσεις.

Η Κατανόηση των Ανθρώπων Προηγείται

Οπότε όλοι συμφωνούμε ότι με λίγη περισσότερη υπολογιστική σκέψη η ζωή του Μπομπύ θα μπορούσε να έχει βελτιωθεί. Αλλά περιμένετε μισό λεπτό. Ίσως να έχουμε κάνει λάθος. Ίσως να είχαμε εξασφαλίσει ότι το βιβλίο του δεν θα είχε γραφτεί ποτέ και η ζωή του θα ήταν μια κόλαση. Δεν ασχοληθήκαμε με την τεχνολογία, αλλά ξεκινήσαμε με την επιστήμη της Πληροφορικής. Ίσως θα έπρεπε να έχουμε ξεκινήσεις με τον άνθρωπο. Δώσαμε βαρύτητα στα σωστά πράγματα;

Σαν μέτρο της δουλειάς που χρειάζεται για να μεταδοθεί ένα γράμμα, χρησιμοποιήσαμε μια αφαίρεση: το πλήθος των ερωτήσεων που πρέπει να γίνουν από τη βοηθό. Αυτή η δουλειά της βοηθού μπορεί να είναι μονότονη, αλλά δεν είναι δύσκολη. Αν όμως τα βλεφαρίσματα απαιτούσαν μεγάλη προσπάθεια από τον Μπομπύ; Η λύση του χρειάζονταν μόνο ένα βλεφάρισμα ανά γράμμα. Ο δικός μας αλγόριθμος, που βασίζεται στο διαίρει και βασίλευε, χρειάζεται να βλεφαρίσει μέχρι και 5 φορές. Πολλαπλασιάστε αυτό με ένα ολόκληρο βιβλίο. Θα ήταν πέντε φορές πιο δύσκολο. Ίσως και πάλι το βλεφάρισμα να μην απαιτούσε και τόση προσπάθεια και ο αλγόριθμός μας πραγματικά να είναι καλύτερος. Δεν γνωρίζουμε την απάντηση γιατί δεν κάναμε την σχετική ερώτηση. Έπρεπε να την κάνουμε εξ αρχής. Έπρεπε να ξεκινήσουμε με τον άνθρωπο.

Επιπροσθέτως, η λύση του είναι εύκολη και κατανοητή από τον οποιονδήποτε. Η δική μας είναι πιο περίπλοκη και πιθανώς θα χρειαζόταν κάποιες εξηγήσεις για να την καταλάβει ένας επισκέπτης. Σίγουρα ο Μπομπύ δε θα ήταν σε θέση να εξηγήσει και πολλά. Είναι πολύ σημαντικό να σκεφτόμαστε τους ανθρώπους.

Για Εκείνον Πάντως Δούλεψε

Ένα πράγμα είναι σίγουρο για τη λύση του Μπομπύ: για εκείνον δούλεψε. Έγραψε ένα ολόκληρο βιβλίο με τον τρόπο αυτό. Ίσως η βοηθός να έκανε κάτι περισσότερο από το να καταγράψει απλά τα λόγια του. Ίσως άνοιγε τις κουρτίνες, του μιλούσε για τον έξω κόσμο ή απλά παρείχε λίγη καθημερινή ανθρώπινη ζεστασιά. Ίσως ο λόγος που έγραψε το βιβλίο ήταν να έχει μια δικαιολογία για να έχει συνέχεια δίπλα του έναν άνθρωπο να επικοινωνεί, πληρωμένο από τον εκδότη του! Ο αλγόριθμος επικοινωνίας δεν θα αφορούσε τότε το βιβλίο, αλλά θα ικανοποιούσε μια βαθύτερη ανάγκη για άμεση επικοινωνία με έναν άνθρωπο. Αντικαταστήστε τον άνθρωπο με την τεχνολογία και πιθανώς να αντικαταστήσετε το μόνο πράγμα που τον κρατούσε ζωντανό.

Από την άλλη μεριά, ίσως αν ήταν σε θέση να επικοινωνήσει μέσω υπολογιστή, θα μπορούσε να μεταφερθεί από το κρεβάτι του νοσοκομείου στον εικονικό κόσμο, να στέλνει e-mail σε φίλους, να tweet-άρει, να διατηρεί μια σελίδα στο facebook, να ελέγχει ένα avatar. Ίσως να έχουμε βελτιώσει τα πράγματα. Και πάλι, θα πρέπει να μάθουμε τι πραγματικά θέλει. Σε μια ακραία περίπτωση χρηστικότητα, όπως είναι αυτή, το σημαντικό είναι ο χρήστης να παραμείνει αναμεμειγμένος. Αυτό αποκαλείται χρηστοκεντρικός σχεδιασμός (user-centered design). Για την ακρίβεια, είναι η καλύτερη προσέγγιση όταν σχεδιάζουμε οποιοδήποτε σύστημα για ανθρώπους και όχι μόνο στις ακραίες περιπτώσεις. Οι χρήστες είναι αυτοί που τελικά πρέπει να προσαρμόσουν αυτό που τους είναι διαθέσιμο για να το κάνουν να λειτουργήσει προς όφελός τους, όχι μόνο τεχνικά, αλλά και συναισθηματικά και κοινωνικά. Ειδάλλως, μπορεί να επινοήσουμε μια λύση που είναι θαυμάσια στη θεωρία και κόλαση επί γης στην πράξη. Οι επιστήμονες της Πληροφορικής έχουν πολλά να σκεφτούν, εκτός από τους υπολογιστές.

Η Πληροφορική

Αλγόριθμοι Αναζήτησης

Με δεδομένο κάτι που πρέπει να αναζητηθεί (γνωστό και ως “κλειδί), ένας **αλγόριθμος αναζήτησης** εγγυάται ότι θα το βρει, αν υπάρχει. Το κλειδί θα μπορούσε να είναι ένα γράμμα που σκέφτεται κάποιος, ένας αριθμός μέσα σε έναν πίνακα, η ιστοσελίδα ενός αστέρα του σινεμά ή μια εγγραφή σε μια βάση δεδομένων με εργαζόμενους.

Ένας απλός αλγόριθμος αναζήτησης ονομάζεται «**γραμμική**», «σειριακή» ή «ακολουθιακή» αναζήτηση. Ελέγχει, με την σειρά, ένα προς ένα, τα αντικείμενα ανάμεσα στα οποία γίνεται η αναζήτηση. Αν βρεθεί το αντικείμενο που αναζητούμε τότε μπορούμε να σταματήσουμε. Αν σημειώσουμε και τη θέση στην οποία βρέθηκε, μπορούμε να το ξαναβρούμε άμεσα. Αν φτάσουμε στο τέλος χωρίς να έχουμε εντοπίσει το κλειδί, τότε γνωρίζουμε με βεβαιότητα ότι δεν υπάρχει.

Ένας ταχύτερος τρόπος αναζήτησης είναι η «**δυναμική**» αναζήτηση. Περιλαμβάνει την ταξινόμηση των αντικειμένων σε μια συγκεκριμένη σειρά, για παράδειγμα αριθμητική ή αλφαβητική. Αυτό μας επιτρέπει να κάνουμε ελέγχους που αποκλείουν την μισή λίστα σε κάθε βήμα. Ελέγχουμε την μεσαία καταχώρηση. Αν το κλειδί είναι πριν από την μεσαία καταχώρηση στην συγκεκριμένη διάταξη τότε το κλειδί θα πρέπει να βρίσκεται στο πρώτο μισό της λίστας (επειδή οι καταχωρήσεις είναι ταξινομημένες). Αν το κλειδί είναι μετά από την μεσαία καταχώρηση τότε θα πρέπει να βρίσκεται στο δεύτερο μισό. Αποκλείουμε έτσι την μισή λίστα και κάνουμε το ίδιο στο τμήμα που απομένει – επαναλαμβανόμενα, μέχρι να έχει απομείνει μόνο ένα αντικείμενο. Είτε θα πρόκειται για το κλειδί, είτε θα γνωρίζουμε ότι το κλειδί δεν υπάρχει στη λίστα.

Ανάλυση Αποδοτικότητας

Υπάρχουν πολλοί διαφορετικοί αλγόριθμοι αναζήτησης. Πώς μπορούμε να διαλέξουμε ανάμεσά τους; Ένας τρόπος είναι να βασιστούμε στο πόσο αποδοτικοί είναι. Μπορούμε να επιλέξουμε μια συγκεκριμένη σημαντική λειτουργία που να δίνει μια καλή αίσθηση του πόση δουλειά απαιτείται κάθε φορά για την αναζήτηση – όπως το πλήθος των ερωτήσεων που γίνονται ή το πλήθος των βλεφαρισμάτων που απαιτούνται. Μπορούμε στη συνέχεια να υπολογίσουμε πόσο συχνά γίνεται αυτή η λειτουργία στην καλύτερη, τη χειρότερη και την μέση περίπτωση.

Υπολογιστική Σκέψη

Η υπολογιστική σκέψη έχει να κάνει με την επίλυση προβλημάτων για ανθρώπους. Επομένως, οι άνθρωποι προηγούνται. Πρέπει να κατανοήσουμε το πρόβλημα από την δική τους σκοπιά, πριν αρχίσετε να ονειρεύεστε τις λύσεις. Ειδάλλως, η τρομερή τεχνική σας λύση θα είναι άχρηστη. Για να γίνετε ένας καλός επιστήμονας της Πληροφορικής, θα πρέπει να **καταλαβαίνετε τους ανθρώπους**.

Η **αλγοριθμική σκέψη** απαιτεί να επινοούμε ακριβείς τρόπους για να εκτελούμε συγκεκριμένες εργασίες, καλύπτοντας όλες τις δυνατές λεπτομέρειες. Με δεδομένη μια αλγοριθμική λύση, άλλοι άνθρωποι ή υπολογιστές μπορούν να ακολουθήσουν τις οδηγίες μηχανικά. Δεν είναι ανάγκη να λύσουν οι ίδιοι τα προβλήματα για να πάρουν απαντήσεις. Ακολουθήστε έναν αλγόριθμο αναζήτησης και θα βρείτε αυτό που ψάχνετε, ό,τι κι αν είναι αυτό.

Ένας τρόπος να βρείτε λύσεις είναι να εντοπίσετε πότε ένα πρόβλημα είναι ίδιο με ένα άλλο. Αν μπορούμε να **μετασχηματίσουμε ένα πρόβλημα** σε ένα άλλο που έχουμε αντιμε-

τωπίσει παλιότερα, τότε μπορούμε απλά να επαναχρησιμοποιήσουμε τη λύση. Αν έχουμε έναν καλό αλγόριθμο αναζήτησης, μπορούμε να τον προσαρμόσουμε σε πολλά διαφορετικά προβλήματα αναζήτησης.

Βλέποντάς το από την αντίθετη σκοπιά, αν έχουμε αναπτύξει μια στρατηγική επίλυσης ενός συγκεκριμένου προβλήματος, μπορούμε να **γενικεύσουμε** αυτή την λύση σε μια στρατηγική που θα δουλεύει και για άλλα προβλήματα. Οι ερωτήσεις διχοτόμησης σε ένα παιχνίδι μπορούν να γενικευθούν στην στρατηγική διαίρει και βασίλευε. Παρομοίως, μπορούμε να γενικεύσουμε έναν αλγόριθμο για ένα συγκεκριμένο πρόβλημα ώστε να προκύψει ένας αλγόριθμος αναζήτησης. Γενικεύοντας την ιδέα των ερωτήσεων “Είναι το A; Είναι το B;” προκύπτει ο αλγόριθμος σειριακής αναζήτησης που είναι εφαρμόσιμος σε κάθε πρόβλημα.

Μπορούμε να χρησιμοποιήσουμε **αναλυτική σκέψη**, η οποία μας παρέχει χειροπιαστούς τρόπους να συγκρίνουμε διαφορετικούς αλγόριθμους. Χρησιμοποιώντας την **αφαίρεση**, εστιάζουμε στις λεπτομέρειες που έχουν σημασία, αν και πρέπει να εξασφαλίσουμε ότι δεν θα αφαιρέσουμε λεπτομέρειες που έχουν σημασία! Η μέθοδος που είναι η καλύτερη για τον σκοπό μας μπορεί να είναι η ταχύτερη, αλλά ίσως να έχουν σημασία και άλλες ιδιότητες, όπως η απαιτούμενη μνήμη.

Teaching London Computing: teachinglondoncomputing.org

Computer Science for Fun: cs4fn.org

Αυτή η ιστορία γράφτηκε από τον Paul Curzon το 2014. Η ομάδα του cs4fn παρείχε υποστήριξη, ιδιαίτερα ο Jonathan Black. Η Zali Collymore-Hussain παρείχε σημαντικά σχόλια που οδήγησαν σε βελτιώσεις.

Για εκπαιδευτικούς πόρους που έχουν αναπτυχθεί για τους καθηγητές του Λονδίνου, οι οποίοι συμπεριλαμβάνουν τις δραστηριότητες για την τάξη και τις διαφάνειες που σχετίζονται με αυτό το φυλλάδιο, επισκεφθείτε την ιστοσελίδα teachinglondoncomputing.org, ένα κοινό πρόγραμμα ανάμεσα στο Queen Mary, University of London και το King’s College London, το οποίο χρηματοδοτείται από την Δημαρχία του Λονδίνου.

Για περισσότερο εκπαιδευτικό υλικό από το cs4fn, επισκεφθείτε την ιστοσελίδα cs4fn.org/teachers/.



Department
for Education

SUPPORTED BY

MAYOR OF LONDON

COMPUTING AT SCHOOL
EDUCATE · ENGAGE · ENCOURAGE

Η μετάφραση στα ελληνικά έγινε το καλοκαίρι του 2014 από τον Γιώργο Μπουκέα. Με χρήσιμα σχόλια και διορθώσεις συνεισέφερε ο Βασίλης Βασιλάκης και ο Πολύκαρπος Σκαπινάκης.