ΠΑΝΕΠΙΣΤΗΜΙΟ ΜΑΚΕΔΟΝΙΑΣ
ΤΜΗΜΑ ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ
5ο εξάμηνο
Διδάσκων: Γιάννης Ρεφανίδης

ΕΡΓΑΣΙΑ #01

Το πρόβλημα του καταχωρητή

Έχουμε έναν επεξεργαστή με έναν καταχωρητή, ο οποίος αποθηκεύει ακέραιες μη αρνητικές τιμές. Επί του καταχωρητή ο επεξεργαστής μπορεί να εκτελέσει τις παρακάτω πράξεις, με το αντίστοιχο κόστος:

Πράξη	Αρχική τιμή	Τελική τιμή	Κόστος	Προϋπόθεση
Αύξηση κατά ένα	X	$X+1$	2	$X < 10^9$
Μείωση κατά ένα	X	$X-1$	2	$X > 0$
Διπλασιασμός	X	$2X$	$[X/2]+1$	$X > 0, 2X \leq 10^9$
Υποδιπλασιασμός	X	$\lfloor X/2 \rfloor$	$[X/4]+1$	$X > 0$
Τετράγωνο	X	X^2	$\left\lceil \frac{X^2 - X}{4} \right\rceil + 1$	$X^2 \leq 10^9$
Τετραγωνική ρίζα	X	$\sqrt{X}$	$\left\lceil \frac{X - \sqrt{X}}{4} \right\rceil + 1$	α) $X > 1$ β) X τέλειο τετράγωνο του $\sqrt{X}$

Θέλουμε να κατασκευάσουμε ένα πρόγραμμα στο οποίο θα δίνουμε την τρέχουσα (αρχική) τιμή του καταχωρητή και την τιμή στόχο και θα μας επιστρέφει τις πράξεις που πρέπει να εκτελεστούν επί του καταχωρητή, ώστε το περιεχόμενό του να γίνει ίσο με την τιμή στόχο, κατά προτίμηση με το λιγότερο δυνατό κόστος.

Θέμα 1ο : Αλγόριθμοι απληροφόρητης (ή τυφλής) αναζήτησης

Κατασκευάστε ένα πρόγραμμα που θα λύνει το πρόβλημα του καταχωρητή με τους αλγορίθμους πρώτα σε πλάτος (breadth-first search) και πρώτα σε βάθος (depth-first search). Το πρόγραμμα θα δέχεται ως παραμέτρους τη μέθοδο επίλυσης, την αρχική τιμή, την τιμή στόχο και το όνομα του αρχείου στο οποίο θα γραφεί η λύση. Για παράδειγμα, εάν το όνομα του προγράμματός σας είναι `register.exe` (μπορείτε φυσικά να το ονομάσετε όπως αλλιώς θέλετε), θέλετε να χρησιμοποιήσετε αναζήτηση κατά πλάτος, η αρχική τιμή είναι το 5, η τιμή στόχος το 18 και θέλετε η λύση να γραφεί στο αρχείο `solution.txt`, θα πρέπει να καλέσετε το πρόγραμμά σας με την εντολή:

```
register.exe  breadth  5 18 solution.txt
```

Εάν αντίθετα θέλετε να χρησιμοποιήσετε τον αλγόριθμο πρώτα σε βάθος, χρησιμοποιείστε τη λέξη `depth` αντί της λέξης `breadth` στην παραπάνω κλήση.

Σε σχέση με το αρχείο εξόδου `solution.txt`, το περιεχόμενό του θα πρέπει να είναι της εξής μορφής:

```
N, C        // N το πλήθος των εντολών της λύσης, C το συνολικό κόστος
instruction1 number1 cost1
instruction2 number2 cost2
```

…

`instructionN numberN costN`

όπου instruction1 έως `instructionN` οι N εντολές που αποτελούν τη λύση, `number1` έως `numberN` οι N αριθμοί επί των οποίων εκτελέστηκαν οι εντολές και `cost1` έως `costN` τα κόστη αυτών.

Κάθε εντολή θα είναι μια από τις:

- `increase`
- `decrease`
- `double`
- `half`
- `square`
- `root`

Το πρόγραμμά σας μπορεί να τυπώνει περιορισμένης έκτασης μηνύματα στην οθόνη, όπως π.χ. το χρόνο που χρειάστηκε για να λύσει το πρόβλημα, το πλήθος των βημάτων που έχει η λύση, ενδεχόμενα μηνύματα λάθους (π.χ. αδυναμία επίλυσης του προβλήματος μέσα σε συγκεκριμένα χρονικά όρια, π.χ. 60 seconds κλπ).

Δώστε ιδιαίτερη προσοχή στον τρόπο κλήσης του προγράμματός σας, καθώς και στη μορφή του αρχείου εξόδου, σύμφωνα με όσα περιγράφηκαν παραπάνω, ώστε το πρόγραμμά (συμπεριλαμβανομένων των λύσεων που αυτό παράγει) να μπορεί να **ελεγχθεί αυτόματα**.

Προσοχή: Για την αποφυγή ατέρμονων βρόχων (αφορά κατά κύριο λόγο την αναζήτηση πρώτα σε βάθος), το πρόγραμμά σας θα πρέπει να ελέγχει για κύκλους τουλάχιστον στο τρέχον μονοπάτι.

Θέμα 2ο : Αλγόριθμοι πληροφορημένης (ή ευρετικής) αναζήτησης

Επαυξήστε το πρόγραμμά σας ώστε να λύνει το πρόβλημα του καταχωρητή χρησιμοποιώντας τους αλγορίθμους αναζήτησης πρώτα στο καλύτερο και A*. Χρησιμοποιείστε τις λέξεις `best` και `astar` στην είσοδο για να τους δηλώσετε. Επιλέξτε μια παραδεκτή ευρετική συνάρτηση, λαμβάνοντας υπόψη τα κόστη των εντολών όπως περιγράφονται στην εκφώνηση.

Θέμα 3ο : Πειραματική αξιολόγηση των αλγορίθμων

Δοκιμάστε τους τέσσερις αλγορίθμους αναζήτησης που υλοποιήσατε σε μεγάλο πλήθος προβλημάτων που θα επιλέξετε εσείς. Συγκρίνετε τους χρόνους επίλυσης και το πλήθος των βημάτων σε κάθε λύση. Εκθέστε τα συμπεράσματά σας όσον αφορά (συγκριτικά) το χρόνο που απαιτείται από κάθε αλγόριθμο για να λύσει τα κάθε πρόβλημα και το μήκος της λύσης που αυτός βρίσκει. Συμπεριλάβετε πίνακες και διαγράμματα στο παραδοτέο σας.

Θα πρέπει να υποβάλλετε:

1. Τον κώδικα του προγράμματός σας και αρχεία με τις λύσεις των προβλημάτων που δοκιμάσατε να λύσετε (**50%**).
2. Αναλυτική αναφορά (**50%**) (pdf ή docx/doc ή odt) που θα περιλαμβάνει:
   - Εξώφυλλο
   - Περιεχόμενα
   - Περιγραφή του κώδικά σας, ιδίως της ευρετικής συνάρτησης που επιλέξατε να χρησιμοποιήσετε
   - Απαρίθμηση των προβλημάτων στα οποία δοκιμάσατε τον κώδικά σας
   - Screenshots από την εκτέλεση του κώδικα σε μερικά (όχι όλα) από τα προβλήματα που δοκιμάσατε
   - Αναλυτικά τα αποτελέσματα που πήρατε και συγκριτικός σχολιασμός των αλγορίθμων

Υποβολές που δεν θα συνοδεύονται από αναφορά θα μηδενίζονται. Επίσης θα ελέγχονται υποβολές με μεγάλη ομοιότητα μεταξύ τους, με αίτημα για εξηγήσεις των εμπλεκομένων.

Οι εργασίες σας μπορούν να υποβληθούν είτε στο Google Classroom είτε στο openeclass.uom.gr (**απαγορεύεται να υποβάλλετε την εργασία σας και στις δύο πλατφόρμες**). Εκπρόθεσμες υποβολές γίνονται δεκτές, βαθμολογούνται όμως με 20% χαμηλότερη βαθμολογία.

Υπόδειξη: Στο Google drive του μαθήματος μπορείτε να βρείτε παρόμοια εργασία παλαιότερων ετών που αφορά το Ν-παζλ, μαζί με τον κώδικα της ενδεικτικής της λύσης στην γλώσσα προγραμματισμού C. Μπορείτε να βασίσετε τη δική σας λύση στον κώδικα που σας δίνετε ή να γράψετε νέο κώδικα.