ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ - ΑΛΓΟΡΙΘΜΟΙ ΑΝΑΖΗΤΗΣΗΣ

Στην εργασία έχουν υλοποιηθεί οι παρακάτω αλγόριθμοι αναζήτησης για την επίλυση ενός Puzzle διαστάσεων 3x3:

A) Depth-First Search (DFS)

B) Breadth-First Search (BFS)

C) Best-First Search (BestFS)

D) A*

Η υλοποίηση έχει γίνει σε C++ και ο κώδικας βασίζεται στο Maze Search που παρουσιάστηκε στο φροντιστήριο.

Η μοντελοποίηση του προβλήματος είναι αρκετά απλή. Κάθε κατάσταση του Puzzle αναπαρίσταται από έναν διδιάστο πίνακα χαρακτήρων 3x3 που απεικονίζει την τρέχουσα μορφή του Puzzle. Στόχος είναι να φέρουμε το Puzzle από μια αρχική κατάσταση στην ζητούμενη τελική, αντιμεταθέτοντας ουσιαστικά το κενό κελί με γειτονικούς αριθμούς. Οι επιτρεπτές κινήσεις είναι πάνω, κάτω, αριστερά και δεξιά. Οι κινήσεις αυτές υλοποιούνται από τις αντίστοιχες μεθόδους goUp(), goDown(), goLeft(), goRight(), στις οποίες γίνεται και ο έλεγχος εγκυρότητας των κινήσεων με βάση τα όρια του πίνακα.

Οι αλγόριθμοι δεν παρουσιάζουν κάποια τροποποίηση σε σχέση με αυτούς που χρησιμοποιήθηκαν στο πρόβλημα του Maze Search. Ο αλγόριθμος που προστέθηκε είναι ο A*, ο οποίος ωστόσο είναι ίδιος με τον BestFS, με μόνη διαφορά την ευριστική συνάρτηση.

Η ευριστική συνάρτηση h που χρησιμοποιείται στον αλγόριθμο BestFS είναι μια παραλλαγή της απόστασης manhattan. Η τιμή της είναι το άθροισμα της απόστασης manhattan του κάθε αριθμού από την τρέχουσα θέση του έως την σωστή θέση.

Αντίστοιχα, η ευριστική συνάρτηση που χρησιμοποιείται στον αλγόριθμο A* είναι η h + g, όπου h η συνάρτηση που περιγράφηκε παραπάνω και g το βάθος της κατάστασης.

Για το κλειστό σύνολο προτιμήθηκε η υλοποίηση με unordered map για γρηγορότερη αναζήτηση, οπότε χρησιμοποιείται μια μέθοδος getKey(), η οποία επιστρέφει το κλειδί της κάθε κατάστασης. Το κλειδί είναι ένας

9ψήφιος ακέραιος που αναπαριστά ουσιαστικά τον 3x3 πίνακα και επομένως είναι μοναδικός για κάθε κατάσταση του Puzzle.

Οι υπόλοιπες μέθοδοι παρουσιάζουν κάποιες αλλαγές οι οποίες δεν χρήζουν ιδιαίτερης αναφοράς, καθώς αφορούν δευτερεύοντες μεθόδους, όπως η υπερφόρτωση τελεστών κτλ.

Στον παρακάτω πίνακα φαίνονται τα αποτελέσματα εκτέλεσης του κάθε αλγορίθμου, τα οποία είναι σχετικά αναμενόμενα.

Αλγόριθμος	Βάθος Λύσης	Συνολικές Καταστάσεις στη Μνήμη	Καταστάσεις που εξετάστηκαν	Χρόνος (ms)
DFS	57972	101123	60105	281.865
BFS	10	1111	696	2.0018
BestFS	10	39	22	0.1251
A*	10	44	24	0.1174

Παρατηρούμε πως ο DFS δεν είναι καθόλου αποδοτικός για το συγκεκριμένο πρόβλημα αφού βρίσκει μια κακή λύση, χρειάζεται αρκετά περισσότερη μνήμη από τους υπόλοιπους αλγορίθμους και είναι αργός. Ο BFS βρίσκει την βέλτιστη λύση, χωρίς να έχει μεγάλες απαιτήσεις στη μνήμη και είναι σχετικά γρήγορος. Τέλος, οι ευρετικοί αλγόριθμοι BestFS και A* είναι πολύ αποδοτικοί αφού βρίσκουν πολύ γρήγορα την βέλτιστη λύση με ελάχιστες απαιτήσεις στην μνήμη. Τα αποτελέσματα αφορούν το συγκεκριμένο πρόβλημα με 3x3 Puzzle και τη συγκεκριμένη αρχική και τελική κατάσταση. Σε Puzzle διαφορετικών διαστάσεων και διαφορετικής αρχικής/τελικής κατάστασης τα αποτελέσματα διαφέρουν.