



ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ

Ακαδημαϊκό Έτος 2023-2024

1^η Εργαστηριακή Άσκηση (10/11/2023)

Μέρος Α: Παζλ 8 πλακιδίων

Επισκεφθείτε την ιστοσελίδα <https://tristanpenman.com/demos/n-puzzle/> και εξοικειωθείτε με την εφαρμογή. Δημιουργήστε το εξής στιγμιότυπο αρχικής και τελικής κατάστασης (στόχος).

Initial state:	Goal state:																		
<table><tr><td>2</td><td>8</td><td>1</td></tr><tr><td>4</td><td>6</td><td>3</td></tr><tr><td>-</td><td>7</td><td>5</td></tr></table>	2	8	1	4	6	3	-	7	5	<table><tr><td>1</td><td>2</td><td>3</td></tr><tr><td>8</td><td>-</td><td>4</td></tr><tr><td>7</td><td>6</td><td>5</td></tr></table>	1	2	3	8	-	4	7	6	5
2	8	1																	
4	6	3																	
-	7	5																	
1	2	3																	
8	-	4																	
7	6	5																	

Εκτελέστε τους αλγορίθμους αναζήτησης που είναι διαθέσιμοι:

- Breadth-first search: Αναζήτηση πρώτα κατά πλάτος
- Depth-first search: Αναζήτηση πρώτα κατά βάθος
- Iterative deepening search: Επαναληπτική εκβάθυνση (με DFS)
- Greedy search: Αναζήτηση πρώτα στο καλύτερο (BestFS)
- A*

Για τους δύο τελευταίους αλγορίθμους μπορούν να χρησιμοποιηθούν οι ευρετικές:

- Euclidean distance: άθροισμα απόστασης σε ευθεία γραμμή (ευκλείδειας) των λάθος πλακιδίων από τη σωστή θέση
- Manhattan distance: άθροισμα απόστασης Manhattan των λάθος πλακιδίων από τη σωστή θέση
- Tiles out-of-place: αριθμός λάθος πλακιδίων

1) Για κάθε αλγόριθμο καταγράψτε το μήκος της λύσης που βρίσκει και το κόστος που χρειάστηκε. Ως κόστος μπορείτε να θεωρήσετε τον αριθμό των κόμβων που έχουν επεκταθεί (για τον IDS, να λάβετε υπόψη τις επαναλήψεις). Συμπληρώστε τον παρακάτω πίνακα:

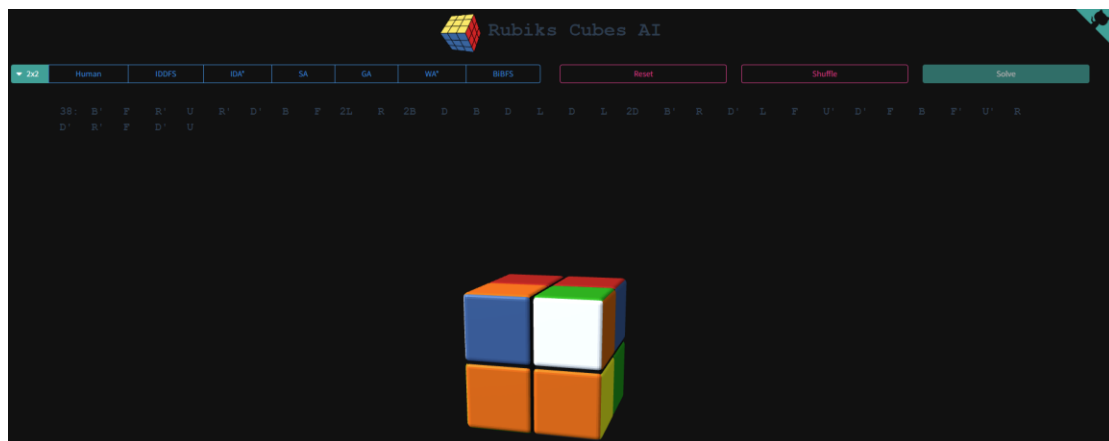
Αλγόριθμος	Μήκος Λύσης	Είναι βέλτιστη;	Κόστος
BFS			
DFS			
IDS			
Greedy, Euclidean			
Greedy, Manhattan			
Greedy, out-of-place			

A*, Euclidean			
A*, Manhattan			
A*, out-of-place			

- 2) Να εξηγήσετε ποιος/ποιοι αλγόριθμοι δεν δίνουν βέλτιστη λύση και γιατί.
- 3) Υπάρχει αλγόριθμος που μπορεί να μην είναι πλήρης; Γιατί;
- 4) Σχολιάστε την απόδοση των αλγορίθμων ως προς την ποιότητα της λύσης και την ταχύτητά τους. Ποιος είναι ταχύτερος και γιατί; Ποιος είναι και βέλτιστος και ταχύτερος; Επιβεβαιώστε θεωρητικά.
- 5) Να συγκρίνετε τους αλγορίθμους A* και Greedy στο συγκεκριμένο στιγμιότυπο. Δίνουν και οι δύο βέλτιστη λύση; Για ποιο λόγο;
- 6) Να συγκρίνετε τις τρεις ευρετικές για την απόδοση των αλγορίθμων A* και Greedy στο συγκεκριμένο στιγμιότυπο. Επηρεάζουν την ποιότητα της λύσης που βρίσκεται κάθε φορά ή/και την ταχύτητα του αλγορίθμου;

Μέρος Β: Κύβος του Ρούμπικ (mini-cube)

Επισκεφθείτε την ιστοσελίδα <https://virgs.github.io/rubiks-cubes-ai/> στην οποία δίνεται μια οπτικοποίηση της επίλυσης ενός 2X2 κύβου του Ρούμπικ με διάφορους αλγορίθμους ΤΝ.



Ο κύβος αυτός αποτελείται από 8 υποκύβους (γωνίες), καθένας από τους οποίους φέρει 3 αυτοκόλλητα διαφορετικού χρώματος στις ορατές πλευρές του (συνολικά 24 αυτοκόλλητα). Για την επίλυσή του μπορεί να γίνει μια ακολουθία από τις παρακάτω επιτρεπτές ενέργειες: U, F, R, που συμβολίζουν περιστροφή κατά 90° δεξιόστροφα της πάνω (Up), μπροστινής (Front) και δεξιάς (Right) πλευράς του κύβου αντίστοιχα, καθώς και U', F', R' αριστερόστροφα¹. Για το συγκεκριμένο κύβο, ο μέγιστος αριθμός κινήσεων που αρκεί για επίλυση από κάθε αρχική διάταξη (*αριθμός του θεού*) είναι 14.

Θεωρήστε την επίλυση του κύβου ως πρόβλημα αναζήτησης. Υποστηρίζονται οι παρακάτω αλγόριθμοι:

- IDDFS: Επαναληπτική Εκβάθυνση με DFS
- IDA*: Επαναληπτική Εκβάθυνση με A*. Ως ευρετική συνάρτηση $h(n)$ χρησιμοποιείται ο αριθμός των λάθος αυτοκόλλητων, δηλαδή αυτών που δεν βρίσκονται στη σωστή θέση.

¹ Οι κινήσεις Left, Back, Down δεν χρειάζεται να εξεταστούν λόγω της συμμετρίας του κύβου.

- SA: Προσομοιωμένη Ανόπτηση με μέγεθος γειτονιάς 100.
- GA: Γενετικός Αλγόριθμος με μέγεθος πληθυσμού 100 και συνάρτηση κόστους τον αριθμό των λάθος αυτοκόλλητων.
- WA*: A* που χρησιμοποιεί βάρος 50 στις τιμές τις προηγούμενης ευρετικής ($50 \cdot h(n)$).
- BiBFS: Αμφίδρομη αναζήτηση με BFS

- 1) Ποιο είναι το μέγεθος του χώρου καταστάσεων του προβλήματος;
- 2) Εξηγείστε ποιοι αλγόριθμοι μπορούν εγγυημένα να βρουν βέλτιστη λύση και ποιοι όχι για το συγκεκριμένο πρόβλημα.
- 3) Εξηγείστε ποιοι αλγόριθμοι είναι πλήρεις και ποιοι όχι για το συγκεκριμένο πρόβλημα.
- 4) Για κάθε αλγόριθμο καταγράψτε τον αριθμό των κινήσεων (μήκος λύσης) και το χρόνο που χρειάστηκε (ms). Συμπληρώστε τον παρακάτω πίνακα²:

Αλγόριθμος	Μήκος Λύσης	Είναι βέλτιστη;	Ποσοστό απόκλισης	Χρόνος
IDDFS				
IDA*				
SA				
GA				
WA*				
BiBFS				

- 5) Μοντελοποιήστε μια κατάσταση και σχεδιάστε ενδεικτικά το δέντρο αναζήτησης για α) τον BiBFS και β) τον WA*. Αρκεί να σχεδιάσετε τις πρώτες 2-3 μεταβάσεις.
- 6) Σχολιάστε την απόδοση των αλγορίθμων ως προς την ποιότητα της λύσης και την ταχύτητά τους. Τι ποσοστό απέχει η λύση που βρίσκουν από τη βέλτιστη; Ποιος έχει τον καλύτερο λόγο απόδοσης προς χρόνο; Από τους αλγορίθμους που δεν είναι βέλτιστοι, σε ποια περίπτωση θα προτιμούσατε καθέναν από αυτούς;
- 7) Σχολιάστε ειδικά γιατί οι αλγόριθμοι IDA* και WA* δίνουν ή δεν δίνουν βέλτιστη λύση. Υπάρχει κάποιο πλεονέκτημα στη χρήση τους (π.χ. για κύβους μεγαλύτερης διάστασης);
- 8) Προσπαθήστε να προτείνετε μια παραδεκτή ευρετική για το πρόβλημα.

Παρατηρήσεις:

Η άσκηση είναι ατομική και πρέπει να παραδοθεί ως μία αναφορά σε μορφή pdf (ΑΣΚΗΣΗ1_ΕΠΙΘΕΤΟ_ΑΜ.pdf) μέσω e-class, **μέχρι τις 10/11/2023, 23:59**. Η άσκηση βαρύνει 10% στον τελικό βαθμό. Οι αναφορές ελέγχονται αυτόματα για κειμενική ομοιότητα και αυτές με υψηλό ποσοστό θα μηδενίζονται χωρίς άλλη ειδοποίηση. Για να ισχύσει η εργασία θα πρέπει επίσης να εκτελέσετε μια συγκεκριμένη άσκηση μέσω e-class που θα ανακοινωθεί.

² Για καλύτερα αποτελέσματα, εκτελέστε κάθε αλγόριθμο τουλάχιστον 3 φορές και αναφέρετε μέσες τιμές. Μπορείτε να εκτελέσετε περισσότερους από έναν αλγορίθμους ταυτόχρονα. Μην ξεχάσετε να κάνετε shuffle πριν από κάθε εκτέλεση!