ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
UNIVERSITY OF IOANNINA

ΠΑΝΕΠΙΣΤΗΜΙΟΥΠΟΛΗ ΙΩΑΝΝΙΝΩΝ
Τ.Θ. 1186, 45110 ΙΩΑΝΝΙΝΑ
Τ: 26510 08817 - F: 26510 08890

P.O. Box 1186
GR 45110 IOANNINA, GREECE
T: +30 26510 08817 - F: +30 26510 08890

Μάθημα: ΜΥΥ602 – Τεχνητή Νοημοσύνη
Ακαδημαϊκό έτος: 2023 – 2024
Διδάσκων: Α. Λύκας
Ημερομηνία παράδοσης: 19 Μαΐου 2024

ΟΝΟΜΑΤΕΠΩΝΥΜΑ, ΑΜ ΟΜΑΔΑΣ:

ΜΠΟΥΡΑΝΤΑΣ ΛΑΖΑΡΟΣ, 5303

ΜΠΟΛΟΤΣΗΣ ΣΤΥΛΙΑΝΟΣ, 5429

ΧΑΝΛΑΡΙΔΟΥ ΔΑΝΑΗ, 5386

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
UNIVERSITY OF IOANNINA

ΠΑΝΕΠΙΣΤΗΜΙΟΥΠΟΛΗ ΙΩΑΝΝΙΝΩΝ
Τ.Θ. 1186, 45110 ΙΩΑΝΝΙΝΑ
T: 26510 08817 - F: 26510 08890

P.O. Box 1186
GR 45110 IOANNINA, GREECE
T: +30 26510 08817 - F: +30 26510 08890

Εργαστηριακή Άσκηση 1 (Πρόγραμμα αναζήτησης με UCS και A*) (20%)

Αναζήτηση ομοιόμορφου κόστους (UCS)

Αλγόριθμος UCS

Αρχικοποίηση

➔ Δημιουργείται η ουρά προτεραιότητας και προστίθεται η αρχική κατάσταση.

Βρόχος αναζήτησης

➔ Αφαιρείται η κατάσταση με το χαμηλότερο κόστος από την ουρά.
➔ Αν η κατάσταση είναι η τελική, εκτυπώνεται η διαδρομή και τερματίζεται ο αλγόριθμος.

Διερεύνηση γειτονικών καταστάσεων

➔ Για κάθε δυνατή κίνηση (εξαιρώντας τις κινήσεις εκτός ορίων και τις μη έγκυρες κινήσεις), δημιουργείται μια νέα κατάσταση.
➔ Η νέα κατάσταση προστίθεται στην ουρά προτεραιότητας.

Επανάληψη

Η διαδικασία επαναλαμβάνεται μέχρι να βρεθεί η τελική κατάσταση ή να εξαντληθούν οι καταστάσεις στην ουρά.

Επικεφαλίδες Βιβλιοθηκών και Σταθερών

Αρχικοποιούνται οι βιβλιοθήκες και οι σταθερές όπου “N” είναι η διάσταση του πίνακα (3*3) και “MAX_STATES” είναι ο μέγιστος αριθμός καταστάσεων που μπορεί να εξεταστούν (9!)

Δομές Δεδομένων

- **boardState:** περιγράφει την κατάσταση του πίνακα με το τρέχον ταμπλό, τη θέση του κενού κελιού, το κόστος έχριτην κατάσταση αυτήν, την ευρετική συνάρτηση και τον δείκτη στον γονέα της κατάστασης.

- **PriorityQuene:** Υλοποιεί μια ουρά προτεραιότητας για την αποθήκευση των καταστάσεων του πίνακα.

Συναρτήσεις

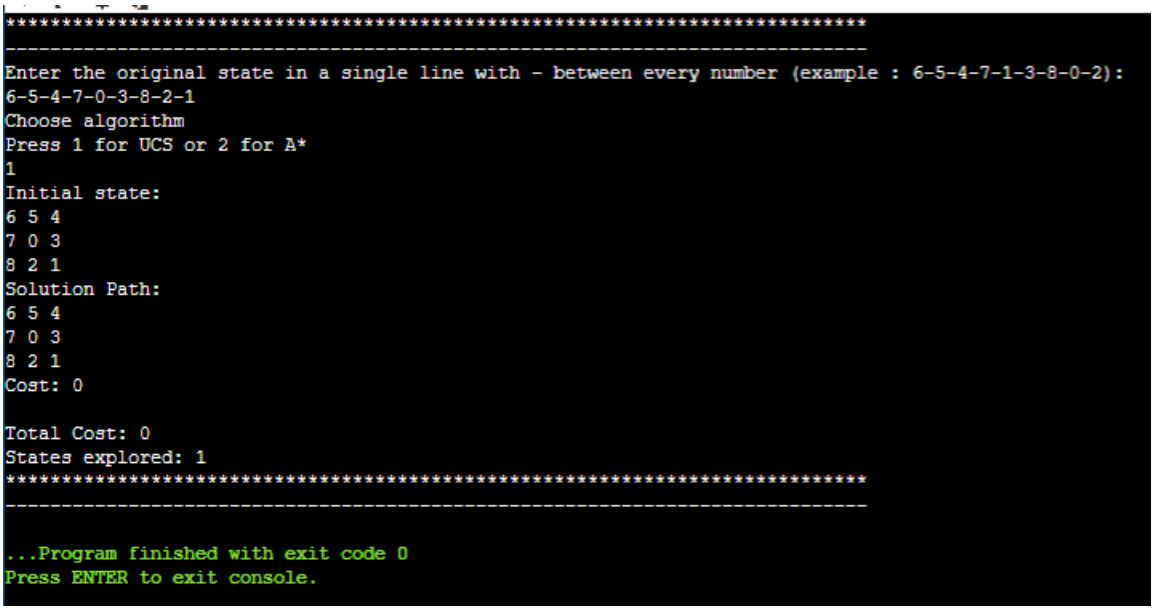
- **createBoardState:** δημιουργεί μια νέα κατάσταση του πίνακα υπολογίζει την ευρετική τιμή και επιστρέφει έναν δείκτη στην νέα κατάσταση.
- **areStateEqual:** ελέγχει αν δύο καταστάσεις είναι ίδιες συγκρίνοντας κάθε κελί του πίνακα με την χρήση βρόγχων.
- **isGoalState:** ελέγχει αν η τρέχουσα κατάσταση είναι η τελική συγκρίνοντας κάθε κελί του πίνακα με την χρήση βρόγχων.
- **perfomMove:** δημιουργεί μια νέα κατάσταση μετακινώντας το κενό κελί σε νέα θέση καλώντας την συνάρτηση swap.
- **misplacedTiles:** Υπολογίζει την ευρετική τιμή μετρώντας τα κελία που διαφέρουν από αυτά της τελικής κατάστασης.
- **printBoard:** εκτυπώνει την τρέχουσα κατάσταση του πίνακα.
- **printSolutionPath:** εκτυπώνει το μονοπάτι που ακολούθησε από την αρχική μέχρι την τελική κατάσταση.
- **createPriorityQuene:** δημιουργεί μια ουρά προτεραιότητας ώστε να αποθηκεύονται οι καταστάσεις του πίνακα που δημιουργούνται με δυναμική δέσμευση μνήμης.
- **isPriorityQueneEmpty:** ελέγχει την κενότητα της ουράς προτεραιότητας.
- **enquene:** εισάγει μια νέα κατάσταση στην ουρά προτεραιότητας.
- **dequene:** εξαγωγή μιας κατάστασης από την ουρά προτεραιότητας.
- **astarSearch:** υλοποιεί την αναζήτηση A* δηλαδή επεκτείνεται η κατάσταση του μετώπου αναζήτησης με το μικρότερο κόστος. Χρησιμοποιεί την ευρετική συνάρτηση ώστε να εκτιμήσει την απόσταση της τρέχουσας κατάστασης από την επιθυμητή τελική κατάσταση.

- **uniformCostSearch:** υλοποιεί την αναζήτηση ομοιόμορφου κόστους, δηλαδή επιλέγεται για επέκταση η κατάσταση του μετώπου αναζήτησης με το μικρότερο κόστος μονοπατιού μέχρι την επιθυμητή τελική κατάσταση.

Ο κώδικας προσπαθεί να βρει τον βέλτιστο τρόπο επίλυσης του παζλ 8, εκτελώντας κινήσεις και αναζητώντας την κατάσταση όπου ο πίνακας ταιριάζει με την τελική κατάσταση. Χρησιμοποιεί την ουρά προτεραιότητας για να εξασφαλίσει ότι πάντα εξετάζει την κατάσταση με το χαμηλότερο κόστος. Κάθε κίνηση κάθετη, οριζόντια ή διαγώνια έχει το ίδιο κόστος.

Επιλέγουμε 5 διαφορετικές αρχικές καταστάσεις και για τους δύο αλγορίθμους αναζήτησης:

- 6 5 4 7 0 3 8 2 1 (ΑΚ = ΤΚ)
- 5 7 4 6 1 3 8 0 2 (τυχαία ΑΚ)
- 6 5 4 7 2 3 8 0 1 (κάθετη κίνηση)
- 6 5 4 7 3 0 8 1 2 (οριζόντια κίνηση)
- 6 5 4 7 2 3 8 1 0 (διαγώνια κίνηση)



ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
UNIVERSITY OF IOANNINA

ΠΑΝΕΠΙΣΤΗΜΙΟΥΠΟΛΗ ΙΩΑΝΝΙΝΩΝ
Τ.Θ. 1186, 45110 ΙΩΑΝΝΙΝΑ
T: 26510 08817 - F: 26510 08890

P.O. Box 1186
GR 45110 IOANNINA, GREECE
T: +30 26510 08817 - F: +30 26510 08890

```
********************************************************************************
--------------------------------------------------------------------------------
Enter the original state in a single line with - between every number (example : 6-5-4-7-1-3-8-0-2):
5-7-4-6-1-3-8-0-2
Choose algorithm
Press 1 for UCS or 2 for A*

1
Initial state:
5 7 4
6 1 3
8 0 2
Solution Path:
5 7 4
6 1 3
8 0 2
Cost: 0

5 7 4
6 1 3
8 2 0
Cost: 1

5 7 4
6 0 3
8 2 1
Cost: 2

5 7 4
0 6 3
8 2 1
Cost: 3

5 0 4
7 6 3
8 2 1
Cost: 4

0 5 4
7 6 3
8 2 1
Cost: 5

6 5 4
7 0 3
8 2 1
Cost: 6

Total Cost: 6
States explored: 1182
********************************************************************************
--------------------------------------------------------------------------------

...Program finished with exit code 0
Press ENTER to exit console.
```

```
********************************************************************************
--------------------------------------------------------------------------------
Enter the original state in a single line with - between every number (example : 6-5-4-7-1-3-8-0-2):
6-5-4-7-2-3-8-0-1
Choose algorithm
Press 1 for UCS or 2 for A*
1
Initial state:
6 5 4
7 2 3
8 0 1
Solution Path:
6 5 4
7 2 3
8 0 1
Cost: 0

6 5 4
7 0 3
8 2 1
Cost: 1

Total Cost: 1
States explored: 4
********************************************************************************
--------------------------------------------------------------------------------

...Program finished with exit code 0
Press ENTER to exit console.
```

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
UNIVERSITY OF IOANNINA

ΠΑΝΕΠΙΣΤΗΜΙΟΥΠΟΛΗ ΙΩΑΝΝΙΝΩΝ
Τ.Θ. 1186, 45110 ΙΩΑΝΝΙΝΑ
Τ: 26510 08817 - F: 26510 08890

P.O. Box 1186
GR 45110 IOANNINA, GREECE
T: +30 26510 08817 - F: +30 26510 08890

```
********************************************************************************
--------------------------------------------------------------------------------
Enter the original state in a single line with - between every number (example : 6-5-4-7-1-3-8-0-2):
6-5-4-7-2-3-8-0-1
Choose algorithm
Press 1 for UCS or 2 for A*
1
Initial state:
6 5 4
7 2 3
8 0 1
Solution Path:
6 5 4
7 2 3
8 0 1
Cost: 0

6 5 4
7 0 3
8 2 1
Cost: 1

Total Cost: 1
States explored: 4
********************************************************************************
--------------------------------------------------------------------------------

...Program finished with exit code 0
Press ENTER to exit console.
```

```
********************************************************************************
--------------------------------------------------------------------------------
Enter the original state in a single line with - between every number (example : 6-5-4-7-1-3-8-0-2):
6-5-4-7-3-0-8-1-2
Choose algorithm
Press 1 for UCS or 2 for A*
1
Initial state:
6 5 4
7 3 0
8 1 2
Solution Path:
6 5 4
7 3 0
8 1 2
Cost: 0

6 5 4
7 0 3
8 1 2
Cost: 1

6 5 4
7 2 3
8 1 0
Cost: 2

6 5 4
7 2 3
8 0 1
Cost: 3

6 5 4
7 0 3
8 2 1
Cost: 4

Total Cost: 4
States explored: 436
********************************************************************************
--------------------------------------------------------------------------------

...Program finished with exit code 0
Press ENTER to exit console.
```

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
UNIVERSITY OF IOANNINA

ΠΑΝΕΠΙΣΤΗΜΙΟΥΠΟΛΗ ΙΩΑΝΝΙΝΩΝ
Τ.Θ. 1186, 45110 ΙΩΑΝΝΙΝΑ
T: 26510 08817 - F: 26510 08890

P.O. Box 1186
GR 45110 IOANNINA, GREECE
T: +30 26510 08817 - F: +30 26510 08890

```
********************************************************************************
--------------------------------------------------------------------------------
Enter the original state in a single line with - between every number (example : 6-5-4-7-1-3-8-0-2):
6-5-4-7-2-3-8-1-0
Choose algorithm
Press 1 for UCS or 2 for A*
1
Initial state:
6 5 4
7 2 3
8 1 0
Solution Path:
6 5 4
7 2 3
8 1 0
Cost: 0

6 5 4
7 2 3
8 0 1
Cost: 1

6 5 4
7 0 3
8 2 1
Cost: 2

Total Cost: 2
States explored: 25
********************************************************************************
--------------------------------------------------------------------------------

...Program finished with exit code 0
Press ENTER to exit console.
```

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
UNIVERSITY OF IOANNINA

ΠΑΝΕΠΙΣΤΗΜΙΟΥΠΟΛΗ ΙΩΑΝΝΙΝΩΝ
Τ.Θ. 1186, 45110 ΙΩΑΝΝΙΝΑ
Τ: 26510 08817 - F: 26510 08890

P.O. Box 1186
GR 45110 IOANNINA, GREECE
T: +30 26510 08817 - F: +30 26510 08890

Αναζήτηση Α με αποδεκτή συνάρτηση h(n)*

Αλγόριθμος Α*

Αρχικοποίηση

- ➔ Η αρχική κατάσταση προστίθεται στην ουρά προτεραιότητας.
- ➔ Δημιουργείται μια κενή ουρά προτεραιότητας και προστίθεται η αρχική κατάσταση.

Επανάληψη αναζήτησης

- ➔ Όσο η ουρά προτεραιότητας δεν είναι κενή, επαναλαμβάνεται η διαδικασία:
- ➔ Αφαιρείται η κατάσταση με το μικρότερο κόστος φ από την ουρά
- ➔ Αν η κατάσταση αυτή είναι η τελική, τότε εκτυπώνεται η λύση και η διαδικασία τερματίζεται.
- ➔ Εξετάζονται όλες οι δυνατές κινήσεις από τη τρέχουσα κατάσταση για να δημιουργηθούν νέες καταστάσεις.
- ➔ Κάθε νέα κατάσταση προστίθεται στην ουρά προτεραιότητας.

Επιλέγουμε τις ίδιες αρχικές καταστάσεις με πριν:

- 6 5 4 7 0 3 8 2 1 (ΑΚ = ΤΚ)
- 5 7 4 6 1 3 8 0 2 (τυχαία ΑΚ)
- 6 5 4 7 2 3 8 0 1 (κάθετη κίνηση)
- 6 5 4 7 3 0 8 1 2 (οριζόντια κίνηση)
- 6 5 4 7 2 3 8 1 0 (διαγώνια κίνηση)

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
UNIVERSITY OF IOANNINA

ΠΑΝΕΠΙΣΤΗΜΙΟΥΠΟΛΗ ΙΩΑΝΝΙΝΩΝ
Τ.Θ. 1186, 45110 ΙΩΑΝΝΙΝΑ
Τ: 26510 08817 - F: 26510 08890

P.O. Box 1186
GR 45110 IOANNINA, GREECE
T: +30 26510 08817 - F: +30 26510 08890

```
********************************************************************************
--------------------------------------------------------------------------------
Enter the original state in a single line with - between every number (example : 6-5-4-7-1-3-8-0-2):
6-5-4-7-0-3-8-2-1
Choose algorithm
Press 1 for UCS or 2 for A*
2
Initial state:
6 5 4
7 0 3
8 2 1
Final state is equal to goal state!
Total Cost: 0
States explored: 0
********************************************************************************
--------------------------------------------------------------------------------

...Program finished with exit code 0
Press ENTER to exit console.
```

```
--------------------------------------------------------------------------------
Enter the original state in a single line with - between every number (example : 6-5-4-7-1-3-8-0-2):
5-7-4-6-1-3-8-0-2
Choose algorithm
Press 1 for UCS or 2 for A*
2
Initial state:
5 7 4
6 1 3
8 0 2
Solution steps:
5 7 4
6 1 3
8 0 2
Cost: 0

5 7 4
6 1 3
8 2 0
Cost: 1

5 7 4
6 0 3
8 2 1
Cost: 2

5 7 4
0 6 3
8 2 1
Cost: 3

5 0 4
7 6 3
8 2 1
Cost: 4

0 5 4
7 6 3
8 2 1
Cost: 5

6 5 4
7 0 3
8 2 1
Cost: 6

Total Cost: 6
States explored: 1182
********************************************************************************
--------------------------------------------------------------------------------

...Program finished with exit code 0
Press ENTER to exit console.
```

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
UNIVERSITY OF IOANNINA

ΠΑΝΕΠΙΣΤΗΜΙΟΥΠΟΛΗ ΙΩΑΝΝΙΝΩΝ
Τ.Θ. 1186, 45110 ΙΩΑΝΝΙΝΑ
T: 26510 08817 - F: 26510 08890

P.O. Box 1186
GR 45110 IOANNINA, GREECE
T: +30 26510 08817 - F: +30 26510 08890

```
********************************************************************************
--------------------------------------------------------------------------------
Enter the original state in a single line with - between every number (example : 6-5-4-7-1-3-8-0-2):
6-5-4-7-2-3-8-0-1
Choose algorithm
Press 1 for UCS or 2 for A*
2
Initial state:
6 5 4
7 2 3
8 0 1
Solution steps:
6 5 4
7 2 3
8 0 1
Cost: 0

6 5 4
7 0 3
8 2 1
Cost: 1

Total Cost: 1
States explored: 4
********************************************************************************
--------------------------------------------------------------------------------

...Program finished with exit code 0
Press ENTER to exit console.
```

```
********************************************************************************
--------------------------------------------------------------------------------
Enter the original state in a single line with - between every number (example : 6-5-4-7-1-3-8-0-2):
6-5-4-7-3-0-8-1-2
Choose algorithm
Press 1 for UCS or 2 for A*
2
Initial state:
6 5 4
7 3 0
8 1 2
Solution steps:
6 5 4
7 3 0
8 1 2
Cost: 0

6 5 4
7 0 3
8 1 2
Cost: 1

6 5 4
7 2 3
8 1 0
Cost: 2

6 5 4
7 2 3
8 0 1
Cost: 3

6 5 4
7 0 3
8 2 1
Cost: 4

Total Cost: 4
States explored: 436
********************************************************************************
--------------------------------------------------------------------------------

...Program finished with exit code 0
Press ENTER to exit console.
```

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
UNIVERSITY OF IOANNINA

ΠΑΝΕΠΙΣΤΗΜΙΟΥΠΟΛΗ ΙΩΑΝΝΙΝΩΝ
Τ.Θ. 1186, 45110 ΙΩΑΝΝΙΝΑ
T: 26510 08817 - F: 26510 08890

P.O. Box 1186
GR 45110 IOANNINA, GREECE
T: +30 26510 08817 - F: +30 26510 08890

```
********************************************************************************
--------------------------------------------------------------------------------
Enter the original state in a single line with - between every number (example : 6-5-4-7-1-3-8-0-2):
6-5-4-7-2-3-8-1-0
Choose algorithm
Press 1 for UCS or 2 for A*
2
Initial state:
6 5 4
7 2 3
8 1 0
Solution steps:
6 5 4
7 2 3
8 1 0
Cost: 0

6 5 4
7 2 3
8 0 1
Cost: 1

6 5 4
7 0 3
8 2 1
Cost: 2

Total Cost: 2
States explored: 25
********************************************************************************
--------------------------------------------------------------------------------

...Program finished with exit code 0
Press ENTER to exit console.
```

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
UNIVERSITY OF IOANNINA
ΠΑΝΕΠΙΣΤΗΜΙΟΥΠΟΛΗ ΙΩΑΝΝΙΝΩΝ
Τ.Θ. 1186, 45110 ΙΩΑΝΝΙΝΑ
T: 26510 08817 - F: 26510 08890
P.O. Box 1186
GR 45110 IOANNINA, GREECE
T: +30 26510 08817 - F: +30 26510 08890

Ευρετική Συνάρτηση h(n):

Η ευρετική συνάρτηση h(n) που χρησιμοποιήσαμε είναι ο αριθμός των λανθασμένων πλακιδίων, δηλαδή ο αριθμός των πλακιδίων που δεν βρίσκονται στη σωστή τους θέση σε σχέση με την κατάσταση στόχο. Για να θεωρείται μια ευρετική συνάρτηση αποδεκτή (admissible), πρέπει να ικανοποιεί τις εξής προϋποθέσεις:

1. Να είναι φραγμένη από κάτω: Η ευρετική συνάρτηση δεν πρέπει ποτέ να υπερεκτιμά το πραγματικό κόστος για να φτάσει από την τρέχουσα κατάσταση στην κατάσταση στόχο.

2. Να είναι συνεπής: Η συνάρτηση πρέπει να ικανοποιεί την ανισότητα του τριγώνου, δηλαδή, για κάθε κατάσταση n και κάθε γειτονική κατάσταση m, πρέπει να ισχύει ότι h(n) <= c(n, m) + h(m) όπου c(n, m) είναι το κόστος για να μεταβούμε από την κατάσταση n στην κατάσταση m.

Η συνάρτηση h(n) που χρησιμοποιήσαμε είναι **αποδεκτή** επειδή:

• Δεν υπερεκτιμά ποτέ το πραγματικό κόστος. Ο αριθμός των λανθασμένων πλακιδίων είναι πάντα μικρότερος ή ίσος με τον ελάχιστο αριθμό κινήσεων που απαιτούνται για να τοποθετηθούν όλα τα πλακίδια στη σωστή τους θέση.
• Είναι συνεπής. Η μετακίνηση ενός πλακιδίου αυξάνει ή μειώνει τον αριθμό των λανθασμένων πλακιδίων κατά το πολύ ένα, διατηρώντας την ανισότητα του τριγώνου.

*Αποτελέσματα A**

• Λύση: Η λύση που βρέθηκε από τον A* ήταν ίδια με αυτή που βρέθηκε από τον UCS.

• Αριθμός επεκτάσεων: Ο αριθμός των επεκτάσεων για να βρεθεί η λύση ήταν σημαντικά μικρότερος σε σχέση με τον UCS. Αυτό οφείλεται στο ότι η ευρετική συνάρτηση h(n) κατευθύνει την αναζήτηση προς τις πιο υποσχόμενες καταστάσεις.

• Κόστος λύσης: Το συνολικό κόστος της λύσης ήταν το ίδιο και για τους δύο αλγορίθμους, επιβεβαιώνοντας ότι ο A* με την ευρετική h(n) είναι βέλτιστος.

Αποτελέσματα UCS

• Λύση: Η λύση που βρέθηκε από τον UCS ήταν ίδια με αυτή του A*, δεδομένου ότι ο UCS εξασφαλίζει την εύρεση της βέλτιστης λύσης.

• Αριθμός επεκτάσεων: Ο αριθμός των επεκτάσεων ήταν μεγαλύτερος σε σύγκριση με τον A*. Αυτό συμβαίνει γιατί ο UCS εξερευνά όλα τα μονοπάτια με το ίδιο κόστος πριν προχωρήσει σε μονοπάτια με μεγαλύτερο κόστος.

Συμπεράσματα

Από τα αποτελέσματα μπορούμε να εξάγουμε τα εξής συμπεράσματα:
• Η ευρετική συνάρτηση h(n) είναι αποδεκτή και βοηθά στη βελτίωση της αποδοτικότητας του αλγορίθμου A* μειώνοντας τον αριθμό των καταστάσεων που πρέπει να εξερευνηθούν.

• Ο αλγόριθμος A* με την ευρετική h(n) είναι βέλτιστος και συνεπής, εξασφαλίζοντας την εύρεση της βέλτιστης λύσης με μικρότερο αριθμό επεκτάσεων σε σύγκριση με τον UCS.

• Ο UCS, παρόλο που είναι επίσης βέλτιστος, είναι λιγότερο αποδοτικός από τον A* όταν χρησιμοποιείται μια κατάλληλη ευρετική συνάρτηση.
Η χρήση της ευρετικής συνάρτησης h(n) στον αλγόριθμο A* καταδεικνύει την αποδοτικότητα και την αποτελεσματικότητα της ευρετικής αναζήτησης σε σύγκριση με την τυπική αναζήτηση κόστους.