

Εργαστηριακή Άσκηση 1

(Πρόγραμμα αναζήτησης με UCS και A*)

ΑΘΑΝΑΣΙΟΣ ΑΚΡΙΩΤΗΣ-5077 ΓΕΩΡΓΙΟΣ ΒΛΕΤΣΑΣ-4924 ΑΠΟΣΤΟΛΟΣ ΜΑΡΜΑΓΚΑΣ-5030

- Αποτελέσματα για διαφορετικές αρχικές καταστάσεις:

1. Αρχική κατάσταση :

1	5	6
7	2	4
8		3

UCS αλγόριθμος

```
Initial Board:
1 5 6
7 2 4
8 0 3

1 5 4
0 6 3
8 7 2

Executing uniformCostSearch algorithm
Path from Initial State to Goal State:
1 5 6
7 2 4
8 0 3
6 5 4
1 0 3
8 7 2
1 5 6
7 0 4
8 2 3
6 5 4
1 7 3
8 0 2
1 5 0
7 6 4
8 2 3
6 5 4
0 7 3
8 1 2
1 5 4
7 6 0
8 2 3
6 5 4
7 0 3
8 1 2
1 5 4
7 6 3
8 2 0
6 5 4
7 0 3
8 1 2
Goal Found!
Cost of the Path: 11
Number of Expansions: 11
Time taken to solve: 543 milliseconds
```

A* αλγόριθμος

```
Path from Initial State to Goal State:
1 5 6
7 2 4
8 0 3

1 5 6
7 2 4
8 3 0

1 5 6
7 0 4
8 3 2

1 5 0
7 6 4
8 3 2

1 0 5
7 6 4
8 3 2

0 1 5
7 6 4
8 3 2

6 1 5
7 0 4
8 3 2

6 0 5
7 1 4
8 3 2

6 5 4
7 1 0
8 3 2

6 5 4
7 1 3
8 0 2

6 5 4
7 0 3
8 1 2

Goal Found!
Cost of the Path: 11
Number of Expansions: 11
Time taken to solve: 51 milliseconds
```

2. Αρχική κατάσταση :

4	2	3
1	5	
6	8	7

UCS αλγόριθμος

```
Initial Board:          5 4 3
4 2 3                  6 1 2
1 5 0                  8 7 0
6 8 7                  5 4 3
                        6 1 0
Executing uniformCostSearch algorithm 8 7 2
Path from Initial State to Goal State:
4 2 3                  5 4 0
1 5 0                  6 1 3
6 8 7                  8 7 2
                        5 0 4
4 0 3                  6 1 3
1 5 2                  8 7 2
6 8 7
                        0 5 4
0 4 3                  6 1 3
1 5 2                  8 7 2
6 8 7
                        6 5 4
5 4 3                  0 1 3
1 0 2                  8 7 2
6 8 7
                        6 5 4
5 4 3                  7 1 3
0 1 2                  8 0 2
6 8 7
                        6 5 4
5 4 3                  7 0 3
6 1 2                  8 1 2
0 8 7
                        Goal Found!
5 4 3                  Cost of the Path: 14
6 1 2                  Number of Expansions: 14
8 0 7                  Time taken to solve: 1220 milliseconds
```

A* αλγόριθμος

```
Executing A* algorithm
Path from Initial State to Goal State:
4 2 3
1 5 0
6 8 7

4 0 3
1 5 2
6 8 7

0 4 3
1 5 2
6 8 7

5 4 3
1 0 2
6 8 7

5 4 3
1 6 2
0 8 7

5 4 3
1 6 2
8 0 7

5 4 3
1 6 2
8 7 0

5 4 3
1 6 0
8 7 2

5 4 0
1 6 3
8 7 2

0 5 4
1 6 3
8 7 2

6 5 4
1 0 3
8 7 2

6 5 4
1 7 3
8 0 2

6 5 4
0 7 3
8 1 2

6 5 4
7 0 3
8 1 2

Goal Found!
Cost of the Path: 14
Number of Expansions: 14
Time taken to solve: 105 milliseconds
```

3. Αρχική κατάσταση :

8	6	4
1		2
7	5	3

UCS αλγόριθμος

```
Initial Board:      6 5 4
8 6 4              1 8 3
1 0 2              7 2 0
7 5 3

Executing uniformCostSearch algorithm
Path from Initial State to Goal State:
8 6 4              6 5 4
1 0 2              1 8 3
7 5 3              7 0 2

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

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

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

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

6 0 4              Goal Found!
1 8 5              Cost of the Path: 11
7 2 3              Number of Expansions: 11
6 5 4              Time taken to solve: 421 milliseconds
1 8 0
7 2 3
```

A* αλγόριθμος

```
Executing A* algorithm
Path from Initial State to Goal State:
4 2 3
1 5 0
6 8 7
4 0 3
1 5 2
6 8 7
0 4 3
1 5 2
6 8 7
5 4 3
1 0 2
6 8 7
5 4 3
1 6 2
0 8 7
5 4 3
1 6 2
8 0 7
5 4 3
1 6 2
8 7 0
5 4 3
1 6 0
8 7 2
5 4 0
1 6 3
8 7 2
5 0 4
1 6 3
8 7 2
0 5 4
1 6 3
8 7 2
6 5 4
1 0 3
8 7 2
6 5 4
1 7 3
8 0 2
6 5 4
0 7 3
8 1 2
6 5 4
7 0 3
8 1 2
Goal Found!
Cost of the Path: 14
Number of Expansions: 14
Time taken to solve: 48 milliseconds
```

4. Αρχική κατάσταση :

	1	2
7	5	6
4	3	8

UCS αλγόριθμος

```
Initial Board:          6 0 5
0 1 2                  1 4 2
7 5 6                  7 3 8
4 3 8

                        6 5 0
Executing uniformCostSearch algorithm 1 4 2
                                      7 3 8

Path from Initial State to Goal State:
0 1 2                  6 5 4
7 5 6                  1 0 2
4 3 8                  7 3 8

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

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

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

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

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

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

Goal Found!
Cost of the Path: 16
Number of Expansions: 16
Time taken to solve: 1916 milliseconds
```

A* αλγόριθμος

```
Executing A* algorithm
Path from Initial State to Goal State:
0 1 2
7 5 6
4 3 8

5 1 2
7 0 6
4 3 8

5 1 2
7 6 0
4 3 8

5 0 2
7 6 1
4 3 8

0 5 2
7 6 1
4 3 8

6 5 2
7 0 1
4 3 8

6 5 2
7 3 1
4 0 8

6 5 2
7 3 1
4 8 0

6 5 2
7 0 1
4 8 3

6 5 2
7 4 1
0 8 3

6 5 2
7 4 1
8 0 3

6 5 2
7 4 0
8 1 3

6 5 0
7 4 2
8 1 3

6 5 4
7 0 2
8 1 3

6 5 4
7 3 2
8 1 0

6 5 4
7 3 0
8 1 2

6 5 4
7 0 3
8 1 2

Goal Found!
Cost of the Path: 16
Number of Expansions: 16
Time taken to solve: 79 milliseconds
```


5. Αρχική κατάσταση :

8	7	6
5	4	3
2	1	

UCS αλγόριθμος

```
Initial Board:          7 6 4
8 7 6                  8 5 3
5 4 3                  0 2 1
2 1 0

Executing uniformCostSearch algorithm 7 6 4
                                         0 5 3
Path from Initial State to Goal State: 8 2 1
8 7 6
5 4 3                  0 6 4
2 1 0                  7 5 3
                        8 2 1

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

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

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

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

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

7 6 0
8 4 3
5 2 1

7 6 4
8 0 3
5 2 1

Goal Found!
Cost of the Path: 15
Number of Expansions: 15
Time taken to solve: 1635 milliseconds
```

A* αλγόριθμος

```
Executing A* algorithm
Path from Initial State to Goal State:
8 7 6
5 4 3
2 1 0

8 7 6
5 4 0
2 1 3

8 7 0
5 4 6
2 1 3

8 7 4
5 0 6
2 1 3

0 7 4
5 8 6
2 1 3

7 0 4
5 8 6
2 1 3

7 5 4
0 8 6
2 1 3

7 5 4
1 8 6
2 0 3

7 5 4
0 8 6
2 1 3

7 5 4
1 8 6
2 0 3

7 5 4
1 0 6
8 2 3

7 5 4
1 6 0
8 2 3

7 5 4
1 6 3
8 2 0

7 5 4
1 6 3
8 0 2

7 5 4
0 6 3
8 1 2

0 5 4
7 6 3
8 1 2

6 5 4
7 0 3
8 1 2

Goal Found!
Cost of the Path: 15
Number of Expansions: 15
Time taken to solve: 76 milliseconds
```

- ο Γενικά συμπεράσματα :

- i. **Επιλογή ευρετικής συνάρτησης:**

Για τον αλγόριθμο A* χρησιμοποιήθηκαν ως ευρετικές συναρτήσεις η Ευκλείδεια απόσταση και η Chebyshev απόσταση (ή μετρική L^∞). Μετά απο δοκιμές παρατηρήθηκε ότι με την Chebyshev ως ευρετική ο αλγόριθμος ήταν ταχύτερος στις περισσότερες περιπτώσεις και γι' αυτό το λόγο επιλέχτηκε.

Η Chebyshev απόσταση αντιστοιχεί στον ελάχιστο αριθμό κινήσεων που απαιτούνται για να φτάσουμε από ένα πλακίδιο σε ένα άλλο, λαμβάνοντας υπόψη και τις διαγώνιες κινήσεις. Αυτό επιτρέπει στον αλγόριθμο να εστιάζει σε μονοπάτια με ελάχιστο μήκος, μειώνοντας τον χρόνο αναζήτησης.

Έτσι δίνεται πάντα μια ακριβή εκτίμηση του πραγματικού κόστους για την επίτευξη του στόχου.

- ii. **Σύγκριση με UCS:**

Ο αλγόριθμος A* λειτούργησε ταχύτερα από τον UCS σε όλες τις περιπτώσεις, όπως ήταν αναμενόμενο. Αυτό οφείλεται στην πιο αποτελεσματική αναζήτηση που επιτυγχάνεται με τη χρήση ευρετικής συνάρτησης.

Παρόλο που ο A* λειτούργησε ταχύτερα, ο αριθμός των επεκτάσεων και το συνολικό κόστος ήταν ίδια με τον UCS. Αυτό οφείλεται στο ότι και οι δύο αλγόριθμοι εξερευνούν το ίδιο σύνολο κόμβων, αλλά ο A* με Chebyshev κάνει πιο "έξυπνες" επιλογές για την επόμενη επέκταση, οδηγώντας σε ταχύτερη επίλυση.