



Technical  
University  
of Crete



# Τεχνητή Νοημοσύνη

Αναφορά 1ης Εργαστηριακής άσκησης(Μέρος Α)

## Φοιτητές

Στέφανος Καλογεράκης

| 2015030064

## Διδάσκων

Χαλκιαδάκης Γ.



## Εισαγωγή

Στο πρώτο σκέλος της πρώτης προγραμματιστικής άσκησης ασχοληθήκαμε με τον πειραματισμό διαφόρων μεθόδων αναζήτησης και η απεικόνιση τους σε διαφορετικούς κόσμους. Πιο συγκεκριμένα, υλοποιήθηκαν οι αλγόριθμοι BFS (Breadth-first search), DFS (Depth-first search), A\* και online LTRA\* όπου για αυτούς τους κόσμους ζητήθηκε απεικόνιση του κόστους λύσης μονοπατιού, του κόστους αναζήτησης (κόμβοι που επεκτάθηκαν μέχρι να βρεθεί η λύση) και φυσικά η απεικόνιση του κόσμου.

## Υλοποίηση

Στις παρακάτω ενότητες εμφανίζονται για κάθε πιθανό συνδυασμό και κόσμο όπως ορίστηκε από την εκφώνηση όλα τα ζητούμενα κόστη και η απεικόνιση του κόστους λύσης (βέλτιστου μονοπατιού). Αξίζει να σημειωθεί ότι όλες οι λύσεις πραγματοποιήθηκαν με γνώμονα είναι όσο το δυνατόν πιο κατανοητές από τον αναγνώστη αλλά και ομοιόμορφες. Όπως θα γίνει αντιληπτό από την ανάγνωση του κώδικα αρκετά κομμάτια κώδικα χρησιμοποιήθηκαν σε όλες τις κλάσεις (κυρίως στα πλαίσια της απεικόνισης). Σαν main class μετά και από υπόδειξη της εκφώνησης χρησιμοποιήθηκε η GridGenerator, με αρχικό μενού όπως ζητήθηκε για την επιλογή της κάθε συνάρτησης. Παρακάτω υπάρχει επίσης ενότητα που επισημαίνονται ορισμένες αλλαγές σε σχέση με τις αρχικές κλάσεις που δόθηκαν.

### Αλλαγές στον αρχικό κώδικα

Αρχικά στην κλάση GridGenerator, που είναι και η main class έγινε η δημιουργία μενού, όπου ο χρήστης για τον κόσμο που έχει φορτώσει δύναται να επιλέξει κάποιον από τους αλγόριθμους που αναλύθηκαν. Το μενού τερματίζει μόνο όταν ο χρήστης δώσει ως επόμενη επιλογή τον αριθμό 9. Αξίζει να σημειωθεί ότι με την τωρινή υλοποίηση γίνεται αναπαράσταση του βέλτιστου μονοπατιού. Υπάρχει δυνατότητα να γίνει αναπαράσταση όλων των κόμβων που αναπτύχθηκαν με την μέθοδο `getAllStepsMatrix(1)`.

Η κλάση Drawing έχει μια μικρή αλλαγή στην γραμμή 118 του κώδικα για την αντιμετώπιση μιας ειδικής υπο περίπτωσης κατά την εκτύπωση του κόσμου.

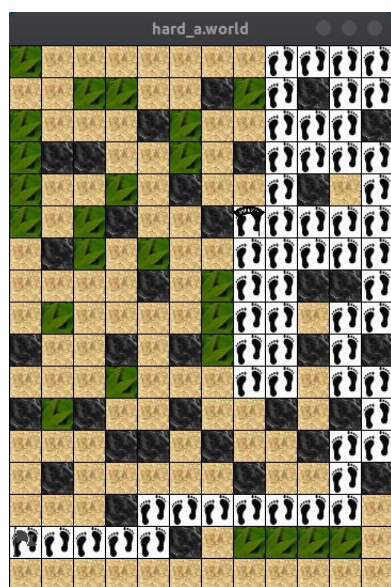
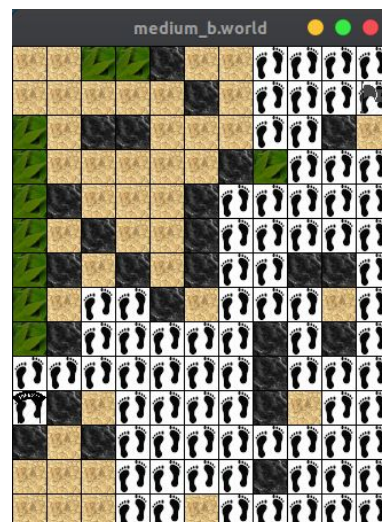
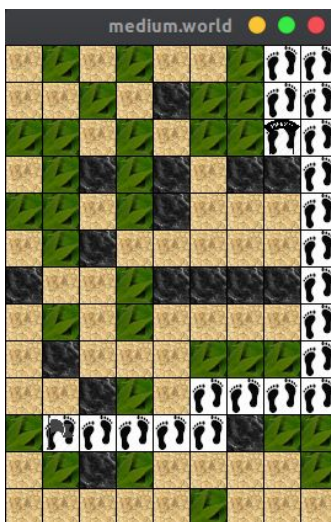
Στην κλάση Grid, προστέθηκε ένας πίνακας ίδιων διαστάσεων με το grid του εκάστοτε κόσμου που κρατάει την πληροφορία για τα κελιά που έχουν επισκεφτεί και έχει χρησιμοποιηθεί στις υλοποιήσεις μας.

## DFS

Η υλοποίηση βασίστηκε σε ψευδοκώδικα που αναφέρεται στην βιβλιογραφία. Όπως ήταν αναμενόμενο το κόστος στο γρασίδι δεν επηρεάζει την τελική βέλτιστη διαδρομή όπως φαίνεται και στις απεικονίσεις των κοσμων, μιας και είναι εξαντλητικός αλγόριθμος προς την αντίστοιχη κατεύθυνση που έχει ανατεθεί αλλιώς πραγματοποιεί backtracking αν δεν μπορεί να προχωρήσει. **Οι κατευθύνσεις αναζήτησης με την σειρά που έχουν ανατεθεί είναι πάνω, δεξιά, αριστερά, κάτω. Η ίδια κατεύθυνση έχει χρησιμοποιηθεί και σε όλες τις υπόλοιπες υλοποιήσεις.** Στην αρχή έγινε προσπάθεια για προσέγγιση με υλοποίηση αναδρομής, αλλά δημιουργούσε προβλήματα στον τελικό υπολογισμό του κόστους που απαιτούνταν. Παρακάτω παρουσιάζονται τα κόστη και οι απεικονίσεις των κόσμων

DFS										
Κόσμος	easy		medium		medium_b		hard_a		hard_b	
Κόστος-Δασος	2	10	2	10	2	10	2	10	2	10
Κόστος Αναζήτησης	143	439	26	82	159	471	93	221	206	590
Κόστος Λύσης	135	423	26	82	105	337	66	162	102	294

## Απεικόνιση για κόστος Δάσος 2-10

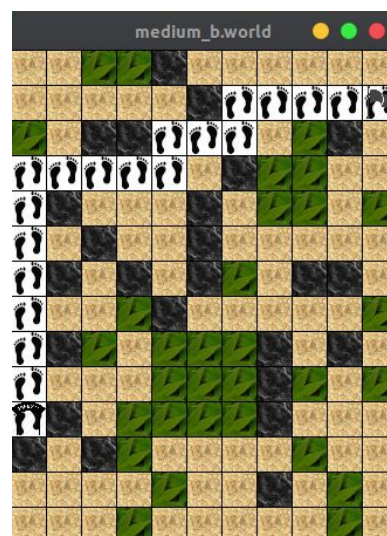
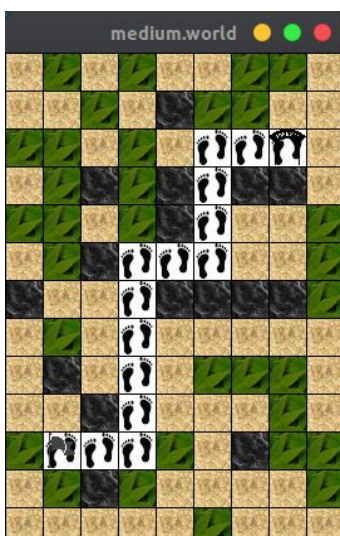
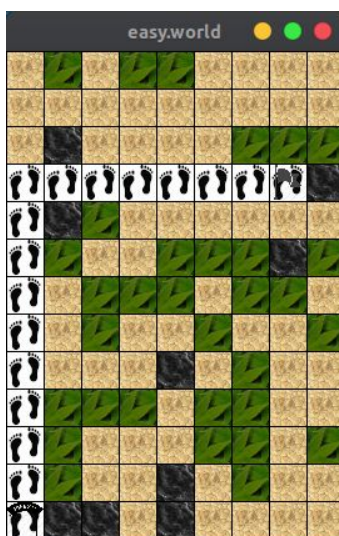


## BFS

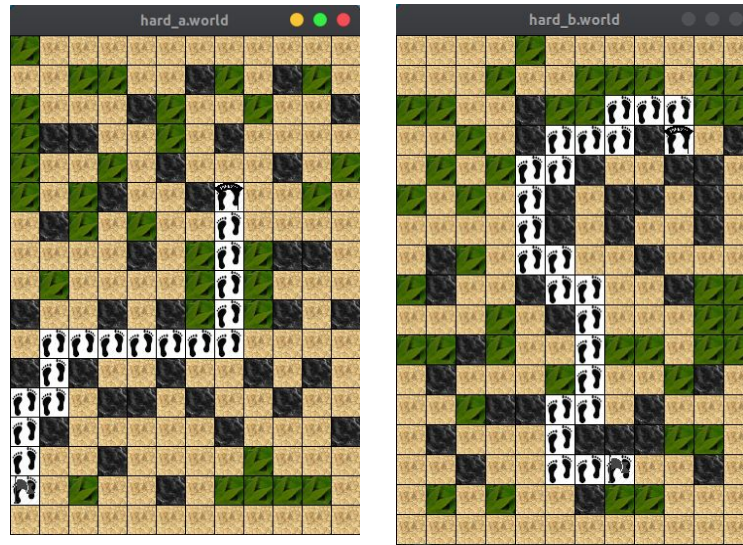
Και σε αυτή την περίπτωση η υλοποίηση δεν έχει κάτι ιδιαίτερο, αποτελεί μια κλασσική υλοποίηση του αλγορίθμου. Και εδώ όπως και περιμέναμε το κόστος δεν θα παίζει ρόλο στην τελική απεικόνιση και το βέλτιστο μονοπάτι αφού ο λαβύρινθος μας εξαπλώνεται ομοιόμορφα ανά επίπεδο μέχρι να βρεθεί η λύση. Ο κώδικας περιέχει επεξηγηματικά σχόλια για κάθε διευκρίνιση όπου απαιτείται

BFS										
Κόσμος	easy		medium		medium_b		hard_a		hard_b	
Κόστος-Δάσος	2	10	2	10	2	10	2	10	2	10
Κόστος Αναζήτησης	143	439	126	406	157	461	146	354	207	575
Κόστος Λύσης	22	62	21	69	28	92	23	63	32	96

### Απεικόνιση για κόστος Δάσος 2-10





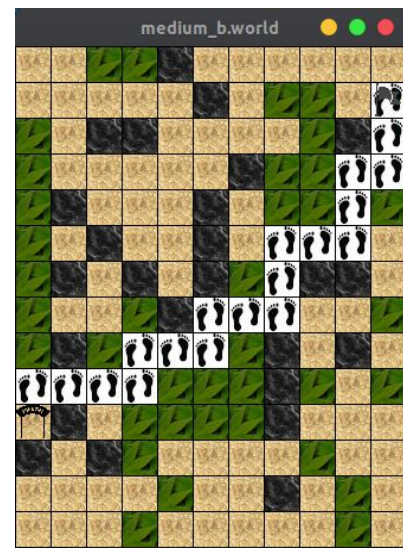
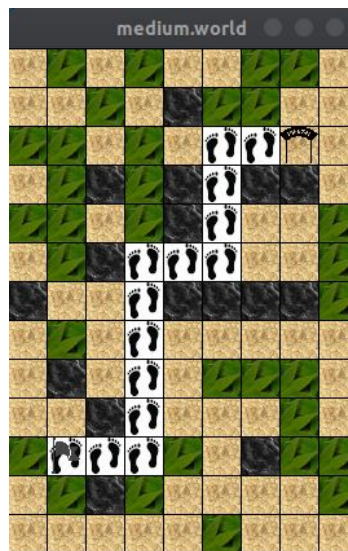
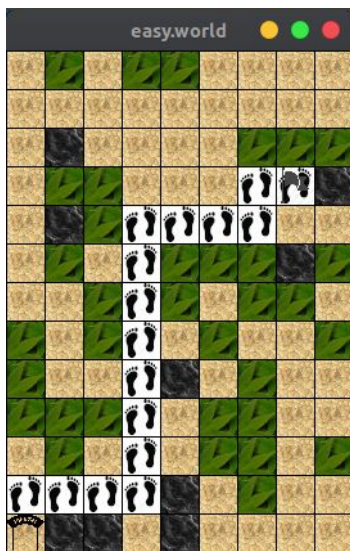


## A\*

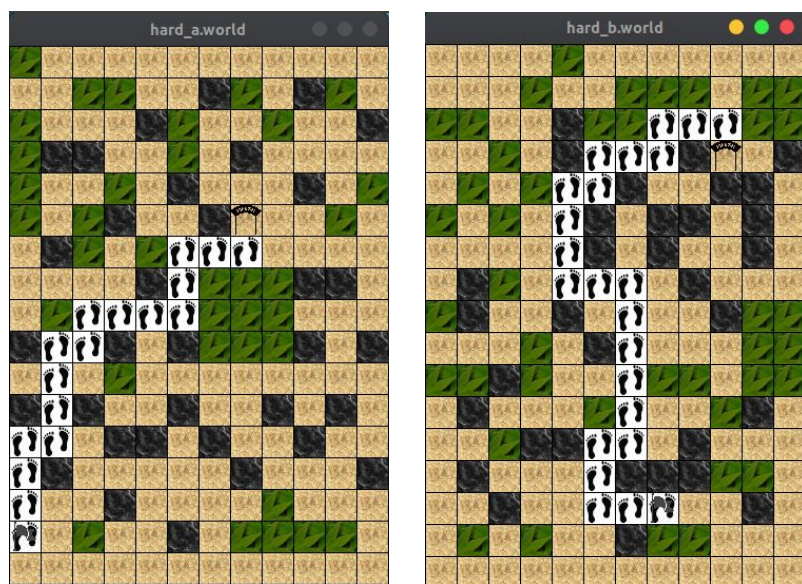
Η υλοποίηση μας βασίστηκε σε ψευδοκώδικα που αναφέρεται στην βιβλιογραφία. Σε αυτή την περίπτωση έχουμε διαφορετικά μονοπάτια για διαφορετικές κόστους. Αυτό οφείλεται στην χρήση heuristic συνάρτησης και αξιολόγησης της καλύτερης επιλογής. Στην υλοποίηση μας χρησιμοποιήσαμε Manhattan Distance (άθροισμα των απόλυτων διαφορών των συντεταγμένων  $x,y$ ). Οι ευριστικές τιμές του εκάστοτε κελιού αποθηκεύτηκαν σε πίνακα διαστάσεων του grid για να είναι εύκολα διαθέσιμες ανάλογα με την θέση. Επιπλέον για την υλοποίηση του αλγορίθμου, χρησιμοποιήθηκαν δύο λίστες closedSet και openList. Η closedset περιλαμβάνει κόμβους που έχουν ήδη αξιολογηθεί από τον αλγόριθμο, ενώ η openList περιέχει κόμβους που θα αξιολογηθούν στην συνέχεια. Αξίζει να αναφερθεί ότι για την τελευταία, χρησιμοποιήθηκε η δομή PriorityList της Java, για να μεταβαίνουμε στην επόμενη καλύτερη επιλογή σαν βήμα, αφού εξασφαλίστηκε ότι δεν υπάρχουν διπλότυπα.

A*										
Κόσμος	easy		medium		medium_b		hard_a		hard_b	
Κόστος-Δάσος	2	10	2	10	2	10	2	10	2	10
Κόστος Αναζήτησης	100	439	97	350	66	352	50	178	184	545
Κόστος Λύσης	20	41	21	46	22	38	19	27	32	63

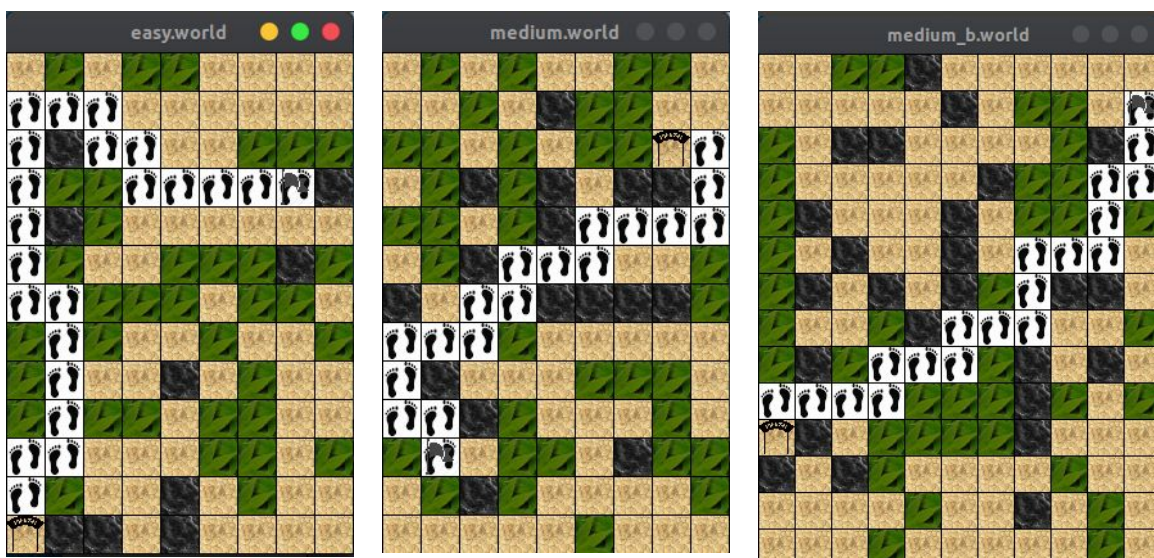
### Απεικόνιση για κόστος Δάσος 2



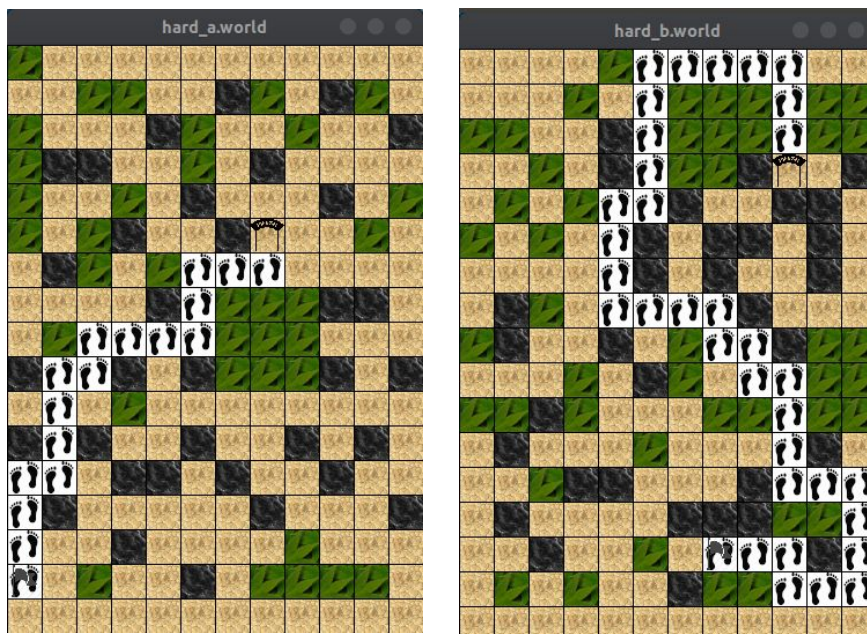




### Απεικόνιση για κόστος Δάσος 10







## Online LTRA\*

Ο LTRA\* αλγόριθμος είναι αρκετά παρόμοιος με τον A\* που αναλύθηκε παραπάνω με την διαφορά ότι σε αυτή την περίπτωση, η απόφαση θα λαμβάνεται σύμφωνα με τις τιμές των γειτονικών κελιών, οι οποίες ανανεώνονται καθώς μετακινούμαστε στον χώρο(δεν είναι γνωστές οι τιμές εκ των προτέρων). Η υλοποίηση βασίστηκε και πάλι σε πηγή που επισημαίνεται στην βιβλιογραφία και στις διαλέξεις του μαθήματος. Και εδώ η ευρετική που χρησιμοποιήθηκε ήταν η ίδια με τον A\* παραπάνω, αλλά τα πράγματα ήταν πιο απλά, αφού αρχικά μεταβαίναμε σε όλους τους γείτονες, και τελικά συνεχίζαμε σε αυτόν με το μικρότερο  $f(f(x) = g(x)+h(x))$ . Το κόστος του δάσους σε κάθε περίπτωση ήταν 10.

Online LTRA*					
Κόσμος	easy	medium	medium_b	hard_a	hard_b
Κόστος Αναζήτησης	287	415	196	170	409
Κόστος Λύσης	47	115	39	28	93



## Αξιολόγηση

Εν συντομία, από άποψη αξιολόγησης τα αποτελέσματα που λάβαμε ήταν αναμενόμενα. Παρατηρούμε ότι σε κάθε περίπτωση βέλτιστος αλγόριθμος ήταν ο  $A^*$ , το οποίο περιμέναμε αφού χρησιμοποιήθηκε και ευρετική συνάρτηση κόστους για την ελαχιστοποίηση του κόστους. Οι αλγόριθμοι DFS, BFS δεν λαμβανουν υποψη τα



ευριστικά κόστη, με το κόστος μονοπατιού σε κάθε περίπτωση να εξαρτάται από που βρίσκεται η αρχική θέση σε σχέση με την τελική. Τέλος, και το κόστος του αλγορίθμου LTRA\* εξαρτάται και αυτό από τις θέσεις αρχικής και τελικής θέσης, ενώ είναι ανεμενόμενο όλοι οι απαιτούμενοι κόμβοι που πρέπει να αναπτυχθούν να είναι αρκετοί αφού σε κάθε επανάληψη επισκεπτόμαστε όλους τους γειτονικούς κόμβους.

## Βιβλιογραφία

<https://www.hackerearth.com/practice/algorithms/graphs/depth-first-search/tutorial/>

<https://www.geeksforgeeks.org/breadth-first-search-or-bfs-for-a-graph/>

<https://gist.github.com/damienstanton/7de65065bf584a43f96a>

[http://turing.cs.pub.ro/blia\\_2003/Real-time\\_search\\_1.htm](http://turing.cs.pub.ro/blia_2003/Real-time_search_1.htm)