

ΥΣ02 Τεχνητή Νοημοσύνη – Χειμερινό Εξάμηνο 2011-2012

Πρώτη Σειρά Ασκήσεων

(20% του συνολικού βαθμού στο μάθημα, Άριστα = 195 μονάδες)

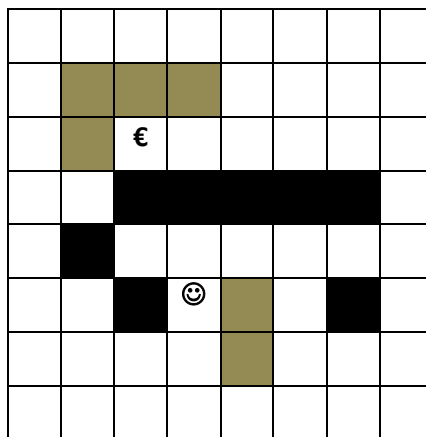
Ημερομηνία Ανακοίνωσης: 14/11/2011

Ημερομηνία Παράδοσης: 9/12/2011 σύμφωνα με τις οδηγίες που δίνονται στην ιστοσελίδα του μαθήματος. Δεν θα γίνουν δεκτές ασκήσεις που θα παραδοθούν μετά τις 9/12.

Αντιγραφή: Σε περίπτωση που προκύψουν φαινόμενα αντιγραφής, οι εμπλεκόμενοι θα βαθμολογηθούν στο μάθημα (όχι απλά στην άσκηση!) με βαθμό μηδέν.

Πρόβλημα 1:

Θεωρήστε το πρόβλημα της κίνησης σε ένα πλέγμα $n \times n$ με εμπόδια όπως αυτό στο παρακάτω παράδειγμα.



Το ανθρωπάκι (☺) ξεκινάει από ένα αρχικό τετραγωνάκι και θέλει να φτάσει στο τετραγωνάκι με τα ευρώ (€) ακολουθώντας τη διαδρομή με το ελάχιστο κόστος. Το ανθρωπάκι επιτρέπεται να κινηθεί οριζόντια ή κάθετα από ένα τετραγωνάκι τη φορά. Τα μαύρα τετραγωνάκια αποτελούν τείχος για το ανθρωπάκι και είναι αζεπέραστο εμπόδιο, ενώ τα καφέ τετραγωνάκια κρύβουν παγίδες οπότε έχουν το διπλάσιο κόστος διέλευσης από ένα κενό τετραγωνάκι. Το κόστος διέλευσης από ένα κενό τετραγωνάκι ορίζεται να είναι 1 ενώ από ένα καφέ 2.

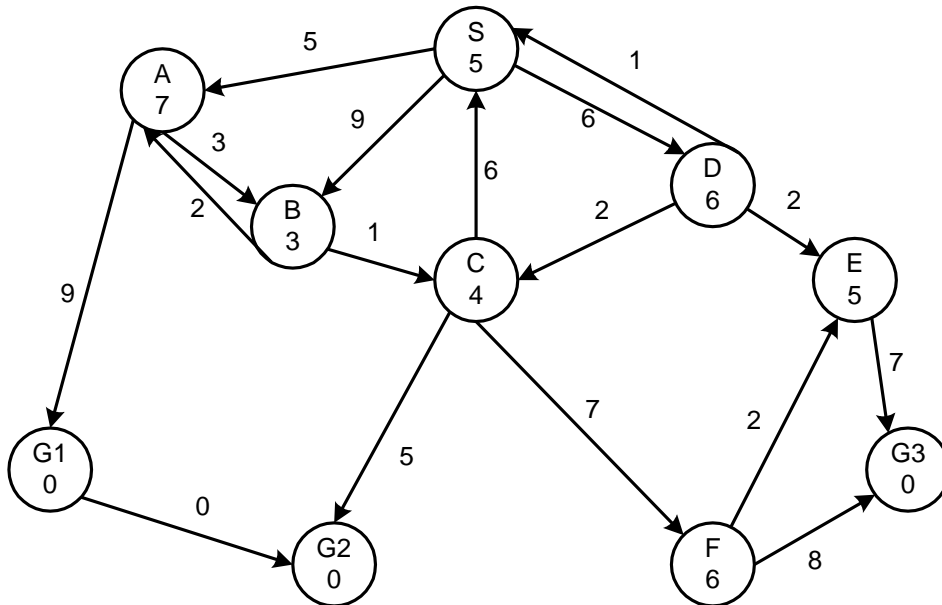
Να λύσετε το πρόβλημα αυτό με τον αλγόριθμο A* χρησιμοποιώντας κατάλληλες ευρετικές συναρτήσεις (heuristic functions). Η επιθυμητή λύση θα πρέπει να χρησιμοποιεί παραδεκτές (admissible) ευρετικές συναρτήσεις και να δουλεύει για πλέγματα $n \times n$ (αλλιώς θα πάρατε μόνο ένα μέρος των βαθμών). Η υλοποίησή σας θα πρέπει να βασιστεί

πάνω στον κώδικα Python που διατίθεται στη σελίδα <http://aima.cs.berkeley.edu/>. Ακόμα, αναμένεται να συγκρίνετε πειραματικά την απόδοση του αλγορίθμου για τις διαφορετικές ευρετικές συναρτήσεις που θα επινοήσετε, με βάση κατάλληλες παραμέτρους που θα ορίσετε. Τέλος, θα πρέπει να συμπεριλάβετε στην πειραματική αποτίμησή σας και εισόδους για τις οποίες είναι δύσκολο να βρεθεί η λύση.

(80 μονάδες)

Πρόβλημα 2:

Θεωρήστε τον παρακάτω γράφο που παριστάνει ένα χώρο αναζήτησης.



S είναι ο κόμβος που αντιστοιχεί στην αρχική κατάσταση και G1, G2, G3 είναι κόμβοι που αντιστοιχούν σε καταστάσεις στόχου. Οι ακμές του γράφου κωδικοποιούν τη συνάρτηση διαδόχων και δίνουν το κόστος κάθε μετάβασης από μια κατάσταση σε μια άλλη. Τέλος, κάθε κόμβος περιέχει ένα αριθμό που είναι η τιμή μιας ευρετικής συνάρτησης h που δίνει το εκτιμώμενο κόστος της φθηνότερης διαδρομής από τον κόμβο αυτό σε ένα κόμβο στόχου.

Για καθένα από τους αλγόριθμους

- Αναζήτηση πρώτα σε πλάτος
- Αναζήτηση πρώτα σε βάθος
- Αναζήτηση πρώτα σε βάθος με επαναληπτική εκβάθυνση
- Άπληστη αναζήτηση πρώτα στον καλύτερο

- A*

να δώσετε: (α) τον κόμβο στόχου στον οποίο φτάνει πρώτα ο αλγόριθμος, και (β) τη σειρά με την οποία βγαίνουν οι κόμβοι από την λίστα «σύνορο» (fringe). Να υποθέσετε ότι: (α) οι αλγόριθμοι έχουν υλοποιηθεί κατάλληλα ώστε να λειτουργούν σωστά σε χώρους αναζήτησης που είναι γράφοι και (β) όταν ο αλγόριθμος δεν μπορεί να «διακρίνει» δύο κόμβους τότε επιλέγει με αλφαβητική σειρά.

(20 μονάδες)

Πρόβλημα 3: Θεωρήστε ένα πρόβλημα αναζήτησης Π που το λύνουμε με τους ακόλουθους αλγόριθμους: αναζήτηση πρώτα σε πλάτος και αναζήτηση πρώτα σε βάθος. Έστω ότι το δένδρο αναζήτησης για το Π είναι πεπερασμένο, έχει βάθος d , έχει παράγοντα διακλάδωσης b , και ο κόμβος με το μικρότερο βάθος που αντιστοιχεί σε κατάσταση στόχου βρίσκεται σε βάθος $g \leq d$.

Ποιός είναι ο μικρότερος και ποιος ο μεγαλύτερος αριθμός κόμβων που μπορούν να δημιουργηθούν από τους παρακάτω αλγόριθμους:

1. Αναζήτηση πρώτα σε βάθος
2. Αναζήτηση πρώτα σε πλάτος
3. Αναζήτηση με επαναληπτική εκβάθυνση

Να εξηγήσετε με λεπτομέρεια τις απαντήσεις σας.

(30 μονάδες)

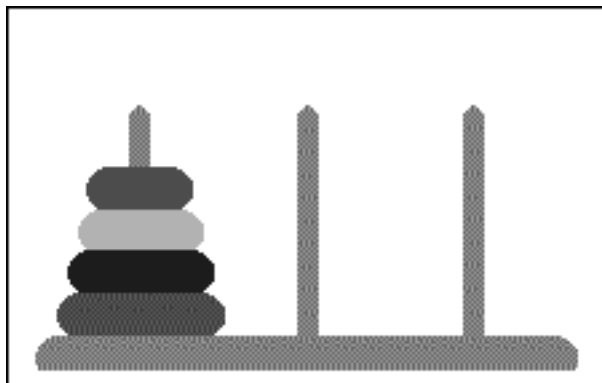
Πρόβλημα 4:

1. Υποθέστε ότι κάνουμε αναζήτηση σε γράφο χρησιμοποιώντας τον αλγόριθμο GRAPH-SEARCH. Έστω ότι είμαστε σε ένα κόμβο N που δεν αντιστοιχεί σε κατάσταση στόχου και δεν είναι στην λίστα closed. Τότε, με βάση τον αλγόριθμο, θα βάλουμε την κατάσταση του N στην λίστα closed και τους απογόνους του στο σύνορο (fringe). Θα ήτανε καλή ιδέα να αλλάξουμε τον αλγόριθμο ώστε να τσεκάρουμε πρώτα ότι οι απόγονοι του N δεν αντιστοιχούν σε καταστάσεις που είναι στη λίστα closed;
2. Ο αλγόριθμος A* ελέγχει αν ένας κόμβος αντιστοιχεί σε κατάσταση στόχου μόνο όταν ο κόμβος αυτός επιλεγεί από το σύνορο για έλεγχο και επέκταση. Θα ήτανε καλή ιδέα να αλλάξουμε τον αλγόριθμο ώστε κάθε φορά που επιλέγεται ένας κόμβος N να ελέγχουμε επίσης αν οι απόγονοι του N αντιστοιχούν σε κατάσταση στόχου;

Οι απαντήσεις σας να είναι ακριβείς και με παραδείγματα.

(20 μονάδες)

Πρόβλημα 5: Θεωρήστε το πρόβλημα των πύργων του Hanoi.



Έχουμε n δίσκους ($n=4$ στο παραπάνω σχήμα) τοποθετημένους σε έναν στύλο με τον τρόπο που φαίνεται παραπάνω (δεν υπάρχει μεγαλύτερος δίσκος πάνω σε μικρότερο). Υπάρχουν συνολικά τρεις στύλοι. Μια **κίνηση** σε αυτό το πρόβλημα σημαίνει να μετακινηθεί ο πάνω δίσκος από ένα στύλο σε ένα άλλο (αλλά δεν μπορεί να τοποθετηθεί ένας μεγαλύτερος δίσκος πάνω από έναν μικρότερο). Πως μπορούν να μετακινηθούν όλοι οι δίσκοι από τον αριστερότερο στο δεξιότερο στύλο χρησιμοποιώντας τον ενδιάμεσο στύλο σαν βοηθητικό;

Έχετε να απαντήσετε στα εξής ερωτήματα:

- Να ορίσετε το παραπάνω πρόβλημα με λεπτομέρεια σαν πρόβλημα αναζήτησης.
- Ποιος είναι ο παράγοντας διακλάδωσης του δένδρου αναζήτησης? (όσο το δυνατόν πιο μικρός, παρακαλώ!). Ποιο είναι το μέγεθος του χώρου καταστάσεων?
- Θεωρήστε τις παρακάτω ευρετικές συναρτήσεις που μπορούν να χρησιμοποιηθούν για τη λύση του προβλήματος:
 - Ο αριθμός των δίσκων σε ένα στύλο διαφορετικό από τον δεξιότερο.
 - Ο αριθμός των δίσκων πάνω στον μεγαλύτερο δίσκο.
 - $2^k - 1$ όπου k είναι ο μεγαλύτερος δίσκος που δεν είναι στη θέση του (υποθέτουμε ότι αριθμούμε τους δίσκους με $1, \dots, n$ από τον μικρότερο στον μεγαλύτερο).
 - 2^{k-1} όπου k είναι ο μεγαλύτερος δίσκος που δεν είναι στη θέση του.

Ποιες από τις παραπάνω συναρτήσεις είναι παραδεκτές και ποιες όχι; Για τις παραδεκτές, να αποδείξετε την απάντησή σας. Για τις μη παραδεκτές, να δώσετε αντιπαράδειγμα.

- **(Bonus)** Να λύσετε το πρόβλημα με χρήση του αλγόριθμου A* χρησιμοποιώντας τον κώδικα που αναφέραμε στο Πρόβλημα 1 και μια εκ των παραπάνω ευρετικών συναρτήσεων.

(40+40 μονάδες)

Πρόβλημα 6: Θεωρείτε ότι η επιλογή της Python ως γλώσσας προγραμματισμού για το μάθημα Τεχνητή Νοημοσύνη έκανε το μάθημα πιο ενδιαφέρον για σας; Θα προτιμούσατε

κάποια άλλη γλώσσα που γνωρίζετε ήδη; Θα συστήνατε και σε άλλους φοιτητές να μάθουν Python;

(5 μονάδες)