ΥΣ02 Τεχνητή Νοημοσύνη – Χειμερινό Εξάμηνο 2013-2014
Πρώτη Σειρά Ασκήσεων

(Υποχρεωτική, 25% του συνολικού βαθμού στο μάθημα)

Ημερομηνία Ανακοίνωσης: 23/12/2013

Ημερομηνίες Παράδοσης:

8/1/2013 στην θυρίδα του διδάσκοντα μέχρι τις 12:00 το πρωί: τα προβλήματα 1(α), 2 (α)-(γ), 4(α) και 5.

2/2/2014 σύμφωνα με τις οδηγίες που θα δοθούν στην ιστοσελίδα του μαθήματος: τα υπόλοιπα προβλήματα.

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

Πρόβλημα 1 :

Θεωρήστε το puzzle AlienTiles που περιγράφεται στην ιστοσελίδα http://www.alientiles.com/.

(α) Να εκφράσετε το puzzle σαν πρόβλημα αναζήτησης (θεωρήστε τον πρώτο στόχο που τίθεται από την παραπάνω ιστοσελίδα).

(β) Να προτείνετε παραδεκτές ευρετικές συναρτήσεις για το παραπάνω πρόβλημα και να τις συγκρίνετε θεωρητικά μεταξύ τους.

(γ) Να λύσετε το πρόβλημα με τον αλγόριθμο A* χρησιμοποιώντας τις παραπάνω ευρετικές συναρτήσεις. Η υλοποίηση σας θα πρέπει να βασιστεί πάνω στον κώδικα Python που διατίθεται στη ιστοσελίδα http://aima.cs.berkeley.edu/.

(δ) Να συγκρίνετε πειραματικά την απόδοση του A* για τις διαφορετικές ευρετικές συναρτήσεις που επινοήσατε, με βάση κατάλληλες παραμέτρους που θα ορίσετε. Να συμπεριλάβετε στην πειραματική αποτίμηση σας και εισόδους για τις οποίες είναι δύσκολο να βρεθεί η λύση.

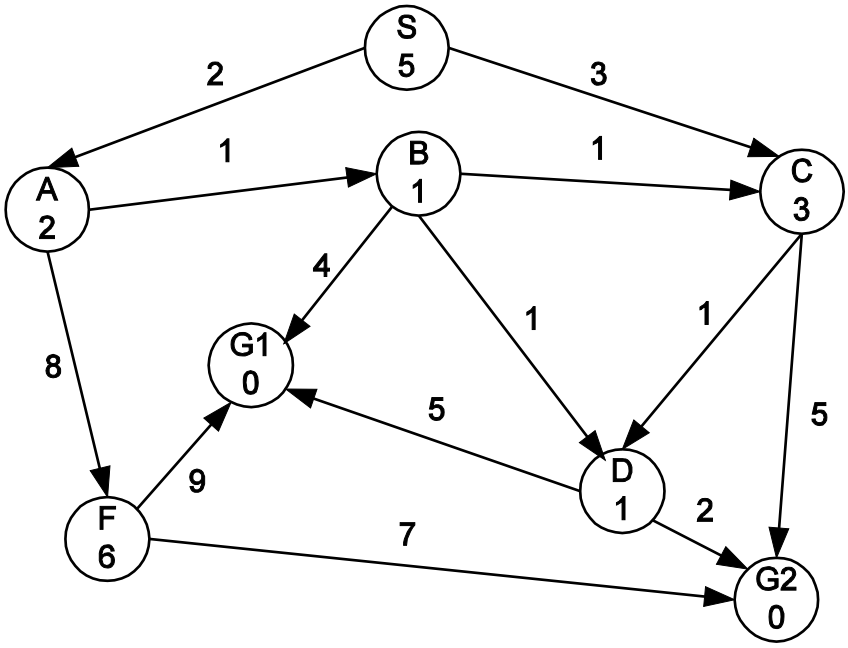
(100 μονάδες)

(ε) **25 Βαθμοί bonus.** Να φτιάξετε ένα γραφικό user interface για το πρόβλημα.

(στ) **Βαθμοί bonus**. Να λύσετε το puzzle και για άλλους στόχους από αυτούς που αναφέρονται στην ιστοσελίδα του.

Πρόβλημα 2:

Θεωρήστε τον παρακάτω γράφο που παριστάνει ένα χώρο αναζήτησης.



S είναι ο κόμβος που αντιστοιχεί στην αρχική κατάσταση και G1, G2 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

(α) Αναζήτηση πρώτα σε πλάτος
(β) Αναζήτηση πρώτα σε βάθος
(γ) Αναζήτηση πρώτα σε βάθος με επαναληπτική εκβάθυνση
(δ) Άπληστη αναζήτηση πρώτα στον καλύτερο
(ε) 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:

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.

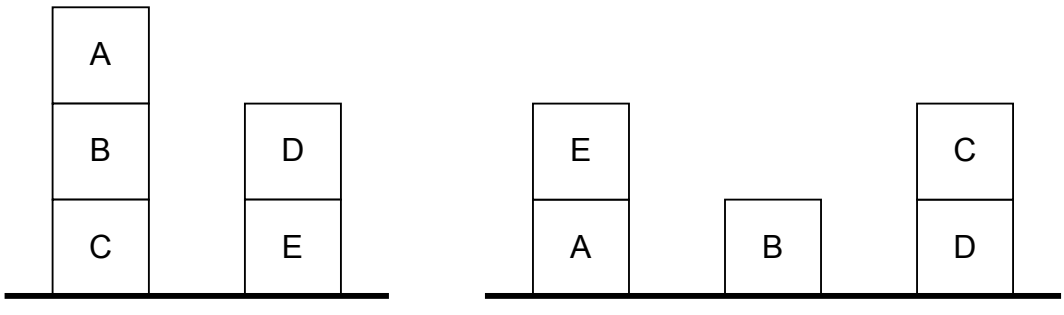
(20 μονάδες)

Πρόβλημα 4:

Θεωρήστε το πρόβλημα των κύβων (blocks world problem) που ορίζεται παρακάτω.

Έχουμε στη διάθεση μας ένα πεπερασμένο αριθμό ομοιόμορφων κύβων και ένα τραπέζι αρκετά μεγάλο για να τους χωρέσει όλους (αυτά είναι τα μόνα αντικείμενα του προβλήματος). Κάθε αντικείμενο βρίσκεται πάνω σε κάποιο άλλο αντικείμενο (τραπέζι ή κύβο). Για κάθε κύβο ισχύει το εξής: είτε είναι ελεύθερος, είτε κάποιος άλλος κύβος βρίσκεται πάνω του. Υπάρχει μια μόνο ενέργεια στη διάθεση του πράκτορα που λύνει το πρόβλημα: η μετακίνηση ενός ελεύθερου κύβου από ένα άλλο κύβο στο τραπέζι ή από ένα αντικείμενο (κύβο ή τραπέζι) σε ένα ελεύθερο κύβο. Η αρχική κατάσταση και η κατάσταση στόχου περιγράφονται δίνοντας με ακρίβεια την θέση κάθε κύβου. Το κόστος κάθε μετακίνησης είναι 1. Μια λύση στο πρόβλημα είναι μια ακολουθία μετακινήσεων που μας επιτρέπει να φτάσουμε από μια δοσμένη αρχική κατάσταση σε μια κατάσταση στόχου. Η βέλτιστη λύση του προβλήματος είναι αυτή που έχει το ελάχιστο κόστος (δηλαδή, αυτή που απαιτεί τις λιγότερες μετακινήσεις).

Παράδειγμα:



Αρχική Κατάσταση Κατάσταση Στόχου

το παραπάνω παράδειγμα η βέλτιστη λύση είναι:

1. Μετακίνησε D στο τραπέζι
2. Μετακίνησε A στο τραπέζι
3. Μετακίνηση E πάνω στο A
4. Μετακίνησε B στο τραπέζι
5. Μετακίνησε C πάνω στο D

(α) Να ορίσετε το παραπάνω πρόβλημα σαν πρόβλημα αναζήτησης.

(β) Να προτείνετε μία ή περισσότερες παραδεκτές ευρετικές συναρτήσεις για το πρόβλημα και να τις συγκρίνετε μεταξύ τους θεωρητικά.

(20 μονάδες)

(γ) Υλοποιήστε σε python και πειραματιστείτε όπως στο Πρόβλημα 1 (γ) και (δ).

(μονάδες bonus)

Πρόβλημα 5:

Να αποδείξετε με λεπτομέρεια τα φράγματα πολυπλοκότητας χώρου και χρόνου που δώσαμε για τον αλγόριθμο «αναζήτηση πρώτα σε βάθος με επαναληπτική εκβάθυνση» (iterative deepening search, IDS).

(30 μονάδες)