

Τεχνητή Νοημοσύνη
Εαρινό Εξάμηνο 2019
Διδάσκων: Α. Λύκας

- Ομάδες δύο ή τριών (κατά προτίμηση) φοιτητών.
- Εξέταση ασκήσεων: την τελευταία εβδομάδα μαθημάτων του εαρινού εξαμήνου (θα υπάρξει νεότερη ανακοίνωση για δηλώσεις ομάδων, υποβολή και εξέταση εργασιών)
- Γλώσσα προγραμματισμού: C ή Java
- Δεκτές για εξέταση γίνονται μόνο ασκήσεις που είναι ολοκληρωμένες (δηλ. τα προγράμματα μεταγλωττίζονται και εκτελούνται).

Εργαστηριακή Άσκηση 1 (Αναζήτηση σε λαβύρινθο)

Να κατασκευάσετε πρόγραμμα αναζήτησης για την επίλυση του ακόλουθου προβλήματος **πλοήγησης ρομπότ σε λαβύρινθο**:

Κατασκευή λαβύρινθου: Θεωρούμε ένα πλέγμα (πίνακα) με $N \times N$ κελιά, κάποια από τα οποία είναι ελεύθερα, ενώ τα υπόλοιπα περιέχουν εμπόδια και δεν μπορούμε να τα επισκεφθούμε. Κάθε κελί (x,y) χαρακτηρίζεται ως ελεύθερο ή όχι αποφασίζοντας ανεξάρτητα με πιθανότητα p . Τα N και p καθορίζονται στην αρχή του προγράμματος. Κάθε κελί έχει πλευρά μήκους 1.

Θέλουμε να ορίσουμε τη **διαδρομή** (εάν υπάρχει) που πρέπει να ακολουθήσει το ρομπότ ξεκινώντας από ένα αρχικό κελί (S) ώστε να φτάσει στο πλησιέστερο (με βάση το κόστος διαδρομής) από δύο συγκεκριμένα τελικά κελιά $G1$ και $G2$. Οι συντεταγμένες (x,y) των κελιών S , $G1$ και $G2$ δίνονται από τον χρήστη στην αρχή του προγράμματος. Το ρομπότ μπορεί κάθε φορά να μετακινείται **είτε οριζόντια είτε κατακόρυφα είτε διαγώνια** σε ένα **γειτονικό ελεύθερο** κελί. **Όλες οι κινήσεις έχουν κόστος 1.**

Για το παραπάνω πρόβλημα να υλοποιήσετε:

- i) αναζήτηση ομοιόμορφου κόστους (UCS)
- ii) αναζήτηση A^* χρησιμοποιώντας όσο το δυνατόν καλύτερη **αποδεκτή ευρετική** συνάρτηση $h(n)$. Θα πρέπει να εξηγήσετε γραπτώς (σε έγγραφο κειμένου pdf) γιατί η συνάρτηση $h(n)$ που σκεφτήκατε είναι αποδεκτή.

Για κάθε λαβύρινθο που θα εξετάζετε, να εφαρμόζετε τόσο την μέθοδο UCS όσο και την μέθοδο A^* για να μπορείτε να συγκρίνετε τις μεθόδους. Πιο συγκεκριμένα, για κάθε μέθοδο να τυπώνετε: α) τον λαβύρινθο, τα κελιά S , $G1$ και $G2$ και το μονοπάτι που βρήκατε, β) το κόστος του μονοπατιού αυτού, και γ) τον αριθμό των επεκτάσεων που έγιναν.

Να εξετάσετε διάφορες τιμές του N και του p .

Εργαστηριακή Άσκηση 2 (ανάπτυξη παιγνίου)

Να κατασκευάσετε πρόγραμμα το οποίο θα παίζει ενάντια σε κάποιο παίκτη το εξής παίγνιο δύο παικτών που παίζουν εναλλάξ:

Πάνω σε ένα τραπέζι υπάρχουν **αρχικά M_1 άσπροι κύβοι και M_2 μαύροι κύβοι** (τα M_1, M_2 παράμετροι που ορίζονται στην αρχή του προγράμματος, π.χ. $M_1=5, M_2=6$). Όταν έρχεται η σειρά του, κάθε παίκτης πρέπει να αφαιρεί από το τραπέζι **i) είτε ένα κύβο οποιουδήποτε χρώματος ii) είτε δύο άσπρους και ένα μαύρο κύβο iii) είτε δύο μαύρους και έναν άσπρο κύβο**. Το παίγνιο **τερματίζει** όταν **δεν απομείνει κανένας κύβος** πάνω στο τραπέζι.

Νικητής είναι ο παίκτης που θα αφαιρέσει τον τελευταίο κύβο (ή κύβους) από το τραπέζι, εάν στους κύβους από αφαιρούνται στην τελευταία κίνηση υπάρχει τουλάχιστον ένας άσπρος. Εάν στην τελευταία κίνηση **δεν αφαιρεθεί άσπρος κύβος**, το αποτέλεσμα θεωρείται **ισοπαλία**.

Θεωρείστε ότι 'παίκτης MAX' = πρόγραμμα, 'παίκτης MIN' = αντίπαλος και ότι παίζει πρώτος ο MAX.

Αφού πρώτα ορίσετε κατάλληλες τιμές για την αξία των τελικών καταστάσεων, να κατασκευάσετε το πρόγραμμα εκτέλεσης του παιγνίου στο οποίο **ο MAX πρέπει να παίζει βέλτιστα εκτελώντας τον αλγόριθμο MINIMAX με ρίζα την τρέχουσα κατάσταση για να αποφασίσει για την κίνηση που θα κάνει κάθε φορά.** (Η υλοποίηση του MINIMAX να γίνει με τη χρήση αναδρομής).