

**Τεχνητή Νοημοσύνη
Εαρινό Εξάμηνο 2020
Διδάσκων: Α. Λύκας**

- **Γλώσσα προγραμματισμού: C ή Java**

Εργαστηριακή Άσκηση 1 (Αναζήτηση)

Να κατασκευάσετε πρόγραμμα αναζήτησης για την επίλυση του ακόλουθου προβλήματος:

Έχουμε μια σειρά από $2N+1$ θέσεις (αριθμημένες από 1 έως $2N+1$) και ένα σύνολο από N άσπρες (A) και N μαύρες (M) σφαίρες οι οποίες τοποθετούνται στις θέσεις έτσι ώστε να υπάρχει μόνο μία σφαίρα σε κάποια θέση. Υπάρχει επομένως μία κενή θέση κάθε φορά στην οποία μπορεί να μεταφερθεί οποιαδήποτε από τις σφαίρες.

Στόχος είναι, ξεκινώντας από μια οποιαδήποτε αρχική τοποθέτηση των σφαιρών και, **μετακινώντας κάθε φορά μια σφαίρα στην κενή θέση**, να καταλήξουμε με το **μικρότερο κόστος ενεργειών** σε τελική κατάσταση στην οποία όλες οι μαύρες σφαίρες είναι τοποθετημένες αριστερά από τις άσπρες και υπάρχει μία άσπρη σφαίρα στην πιο δεξιά θέση.

Μια σφαίρα που η θέση της έχει απόσταση k θέσεις από την κενή θέση μπορεί να μετακινηθεί μόνο εάν $k \leq N$ και το κόστος της αντίστοιχης ενέργειας είναι ίσο k . Επίσης 'κυκλικές' κινήσεις απαγορεύονται. Το N καθορίζεται στην αρχή του προγράμματος.

Π.χ. για $N=3$ μια αρχική κατάσταση είναι η: AMA-MMA και μια τελική κατάσταση είναι η: MMMAA-A (η παύλα υποδηλώνει την κενή θέση).

Για το παραπάνω πρόβλημα να υλοποιήσετε:

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

Ως είσοδο θα δίνετε μια αρχική κατάσταση και θα τυπώνετε την παραγόμενη βέλτιστη ακολουθία καταστάσεων από την αρχική κατάσταση προς την τελική κατάσταση.

Να λύσετε το πρόβλημα για διάφορες τιμές του N . Προκύπτει κάποια διαφορά στην επίδοση (αριθμό επεκτάσεων) ανάμεσα στις δύο μεθόδους αναζήτησης;

Εργαστηριακή Άσκηση 2 (ανάπτυξη παιγνίου 'SOS')

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

Παίγνιο SOS σε ένα πλέγμα 3x3 ([https://en.wikipedia.org/wiki/SOS_\(game\)](https://en.wikipedia.org/wiki/SOS_(game)))

Κάθε παίκτης, όταν έρθει η σειρά του, μπορεί να τοποθετήσει είτε ένα S είτε ένα O σε οποιαδήποτε κενή θέση του πλέγματος. Αν με την τοποθέτησή του αυτή σχηματίζεται η τριάδα SOS είτε οριζόντια είτε κατακόρυφα είτε διαγώνια σε διαδοχικές θέσεις του πλέγματος, τότε το παίγνιο τερματίζει και ο παίκτης αυτός θεωρείται ο νικητής. Επίσης το παίγνιο τερματίζει όταν γεμίσει το πλέγμα χωρίς να έχει σχηματιστεί τριάδα SOS (αποτέλεσμα ισοπαλία).

Θεωρείστε ότι στην αρχική κατάσταση του παιγνίου όλες οι θέσεις είναι κενές εκτός την αριστερή ή τη δεξιά θέση στη μεσαία γραμμή του πλέγματος που θα πρέπει να περιέχει το O.

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).