

1^η Εργασία

Αναπτύξτε προγράμματα σε Java, C++, Python (ή άλλη γλώσσα που θα σας επιτρέψουν οι βοηθοί των φροντιστηρίων) που να λύνουν **δύο από τα ακόλουθα τρία προβλήματα** (πρέπει να επιλέξετε ακριβώς δύο προβλήματα). *Ειδικά όσοι επιτρέπεται να παραδώσετε την εργασία ατομικά (δείτε το έγγραφο με τις γενικές οδηγίες των εργασιών), μπορείτε να ασχοληθείτε με μόνο ένα από τα προβλήματα ή προαιρετικά με δύο.*

α) **Το πρόβλημα των κανιβάλων και ιεραποστόλων**¹ χρησιμοποιώντας τον αλγόριθμο A^* με κλειστό σύνολο. Μπορείτε να χρησιμοποιήσετε ευρετικές συναρτήσεις παρόμοιες εκείνων της λύσης της άσκησης μελέτης 4.3 ή άλλες δικές σας (εξηγώντας γιατί τις επιλέξατε στο έγγραφο που θα υποβάλετε). Θεωρήστε ότι στην αρχική κατάσταση έχουμε N ιεραποστόλους στη μία όχθη και τον ίδιο αριθμό (N) κανιβάλων στην ίδια όχθη αλλά το N είναι παράμετρος που ορίζεται κατά την κλήση του προγράμματος. Η μέγιστη χωρητικότητα (σε άτομα) της βάρκας είναι M , όπου το M επίσης ορίζεται κατά την κλήση του προγράμματος. Κατά την κλήση του προγράμματος ορίζεται, επίσης, ο μέγιστος επιτρεπόμενος αριθμός διασχίσεων του ποταμού, έστω K . Το πρόγραμμά σας θα πρέπει να βρίσκει τη βέλτιστη λύση που δεν υπερβαίνει τις K διασχίσεις, αν υπάρχει τέτοια λύση. Μπορείτε να κάνετε δοκιμές με σχετικά μικρές τιμές των N , M (π.χ. 3, 4) και K (π.χ. ≤ 100). Θα πρέπει να αναφέρετε στο έγγραφο που θα υποβάλετε πόσο περίπου χρόνο (ανάλογα και με τον υπολογιστή που χρησιμοποιείτε) χρειάζεται το πρόγραμμά σας για να βρει λύση για τις τιμές των N , M , K που δοκιμάσατε, καθώς και τις αντίστοιχες λύσεις που βρίσκει.

β) **Το πρόβλημα των οκτώ βασιλισσών (4^η διάλεξη)**² γενικευμένο ώστε το μέγεθος της σκακίερας να είναι $N \times N$ και ο αριθμός βασιλισσών να είναι επίσης N . Μπορείτε να χρησιμοποιήσετε οποιονδήποτε αλγόριθμο αναζήτησης του μαθήματος (π.χ. γενετικό αλγόριθμο, αναρρίχηση λόφων, beam search). Μπορείτε να κάνετε δοκιμές με σχετικά μικρές τιμές του N (ακόμα και μικρότερες του 8). Θα πρέπει να αναφέρετε στο έγγραφο που θα υποβάλετε πόσο περίπου χρόνο (ανάλογα και με τον υπολογιστή σας) χρειάζεται το πρόγραμμά σας για να βρει τελική κατάσταση (όπου δεν αλληλο-απειλούνται βασίλισσες) για τις τιμές του N που δοκιμάσατε, καθώς και τις αντίστοιχες τελικές καταστάσεις που βρίσκει.

γ) **Reversi** (ή την παραλλαγή του Othello). Μπορείτε να βρείτε τους κανόνες του παιχνιδιού στην παρακάτω διεύθυνση:

¹ Βλ. και https://en.wikipedia.org/wiki/Missionaries_and_cannibals_problem. Ο διδάσκων θα εκτιμούσε προτάσεις σας για καλύτερο όνομα και αφήγημα του παιχνιδιού, που θα διατηρούν τα μαθηματικά του χαρακτηριστικά, χωρίς να υπονοούν βία ή ρατσιστικές αντιλήψεις.

² Βλ. και https://en.wikipedia.org/wiki/Eight_queens_puzzle.

<http://en.wikipedia.org/wiki/Reversi>

Το πρόγραμμά σας θα πρέπει να επιτρέπει στον χρήστη να παίζει Reversi (ή Othello) με αντίπαλο τον υπολογιστή. Για την επιλογή των κινήσεων του υπολογιστή, το πρόγραμμά σας πρέπει να χρησιμοποιεί τον αλγόριθμο MiniMax, κατά προτίμηση με πριόνισμα α - β . Κατά την έναρξη του παιχνιδιού, ο χρήστης θα πρέπει να μπορεί να επιλέξει το μέγιστο βάθος αναζήτησης του αλγορίθμου MiniMax. Ο χρήστης θα πρέπει να μπορεί να επιλέξει, επίσης, αν θα παίζει πρώτος ή όχι. Το πρόγραμμα πρέπει να απορρίπτει κινήσεις που παραβιάζουν τους κανόνες του παιχνιδιού. Αν ο παίκτης του οποίου είναι η σειρά να παίζει δεν μπορεί να τοποθετήσει πουθενά νέο πούλι χωρίς να παραβιάσει τους κανόνες, το πρόγραμμα πρέπει να εμφανίζει αυτόματα σχετικό μήνυμα και να ζητά να παίζει ο άλλος παίκτης. Μετά από κάθε κίνηση, το πρόγραμμα θα πρέπει να δείχνει την κατάσταση του παιχνιδιού (π.χ. τυπώνοντας κενά, X και O). Δεν μας ενδιαφέρει σε αυτό το μάθημα η διεπαφή χρήστη (π.χ. μην αφιερώσετε χρόνο στην ανάπτυξη γραφικής διεπαφής).

Η προθεσμία παράδοσης της εργασίας θα ανακοινωθεί στο e-class. **Διαβάστε προσεκτικά και το έγγραφο με τις γενικές οδηγίες των εργασιών του μαθήματος** (βλ. e-class).