

Τεχνητή Νοημοσύνη
Εαρινό Εξάμηνο 2023
Διδάσκων: Α. Λύκας

- Ομάδες **δύο ή τριών** (κατά προτίμηση) **φοιτητών**.
- Φοιτητές που έχουν βαθμολογηθεί σε προηγούμενα έτη δεν δικαιούνται επανεξέτασης.
- Γλώσσα προγραμματισμού: **C ή Java** (όχι Python)
- Δεκτές για εξέταση γίνονται μόνο ασκήσεις που είναι ολοκληρωμένες, δηλ. τα προγράμματα μεταγλωττίζονται και εκτελούνται στους υπολογιστές του Τμήματος π.χ. opti3060ws03. Μην υποβάλετε κώδικα Java που απαιτεί το περιβάλλον eclipse για να μεταγλωττιστεί και να εκτελεστεί.
- **Δήλωση ομάδων:** e-mail στον διδάσκοντα με (ΑΜ, ονοματεπώνυμο) μελών της ομάδας **μέχρι 28 Απριλίου 2023** (αυστηρή προθεσμία).
- Η δήλωση συνεπάγεται υποχρέωση υποβολής των εργασιών.
- **Προθεσμία υποβολής εργασιών: 18 Μαΐου 2023** (αυστηρή προθεσμία).
- Η **υποβολή** θα γίνει με χρήση της εντολής **turnin** ως εξής:
turnin assignment@myy602 <your_filename>
- Θα δημιουργήσετε δύο καταλόγους, έναν για κάθε άσκηση. Κάθε κατάλογος θα περιλαμβάνει τον πηγαίο, τον εκτελέσιμο κώδικα της άσκησης και αρχείο κειμένου (pdf). Φυσικά θα πρέπει να υπάρχει πληροφορία για τα ονόματα και τα ΑΜ των μελών της ομάδας.
- Μην υποβάλετε συμπιεσμένα αρχεία .rar

Εργαστηριακή Άσκηση 1 (Τοποθέτηση κύβων σε επιθυμητό σχηματισμό) (20%)

Δίνονται $N=3*K$ αριθμημένοι (από 1 έως N) κύβοι (το K δίνεται από τον χρήστη του προγράμματος), και θέλουμε να τους τοποθετήσουμε σε **τρεις σειρές (η μία σειρά πάνω στην άλλη) με K κύβους η καθεμία σειρά**. Η θέση κάθε κύβου περιγράφεται με συντεταγμένες (x,y) όπου

- για $y=1$ (πρώτη σειρά που στηρίζεται σε ένα τραπέζι), οι δυνατές τιμές του x είναι $\{1,2,...,L\}$ ($L=4*K$) (1)
- για $y=2$ (δεύτερη σειρά που στηρίζεται στην πρώτη σειρά) και $y=3$ (τρίτη σειρά που στηρίζεται στη δεύτερη σειρά) οι δυνατές τιμές του x είναι $\{1,2,...,K\}$ (2)

Παρατηρείστε ότι πάνω στο τραπέζι (δηλ. $y=1$) έχουμε διαθέσιμες περισσότερες από K θέσεις ($L=4*K$). Οι πλεονάζουσες $3*K$ θέσεις (από $K+1$ έως $4*K$) μπορούν να χρησιμοποιηθούν για να τοποθετούνται προσωρινά κάποιοι κύβοι εκτός του βασικού σχηματισμού.

Η **τελική κατάσταση** του προβλήματος ορίζεται ως εξής:

- η πρώτη σειρά περιλαμβάνει του κύβους με αρίθμηση από 1 έως K (σε αύξουσα σειρά από αριστερά προς τα δεξιά) που στηρίζονται στο τραπέζι, δηλ. είναι τοποθετημένοι στις θέσεις $(1,1)$ έως $(K,1)$.
- η δεύτερη σειρά τους κύβους με αρίθμηση από $K+1$ έως $2K$ (σε αύξουσα σειρά από αριστερά προς τα δεξιά) που στηρίζονται πάνω στους κύβους της πρώτης σειράς, δηλ. είναι τοποθετημένοι στις θέσεις $(1,2)$ έως $(K,2)$.
- η τρίτη σειρά τους κύβους με αρίθμηση από $2K+1$ έως N (σε αύξουσα σειρά από αριστερά προς τα δεξιά) που στηρίζονται πάνω στους κύβους της δεύτερης σειράς, δηλ. είναι τοποθετημένοι στις θέσεις $(1,3)$ έως $(K,3)$.

Η κατάσταση του προβλήματος μπορεί να θεωρηθεί ως μια λίστα με N θέσεις όπου κάθε θέση περιέχει τις συντεταγμένες (x,y) του αντίστοιχου κύβου. Για παράδειγμα εάν $K=2$ (οπότε $N=6$), η τελική κατάσταση είναι η $[(1,1), (2,1), (1,2), (2,2), (1,3), (2,3)]$ (ο κύβος με αριθμό 1 στην θέση $(1,1)$, ο κύβος με αριθμό 2 στη θέση $(2,1)$, ο κύβος με αριθμό 3 στη θέση $(1,2)$ κλπ).

Για να είναι μια **κατάσταση** του προβλήματος **έγκυρη**, θα πρέπει α) να ικανοποιούνται οι περιορισμοί (1) και (2) για τη θέση (x,y) κάθε κύβου και β) καθένας από τους κύβους να είναι τοποθετημένος είτε στο τραπέζι είτε να στηρίζεται πάνω σε έναν άλλο κύβο. (Να φτιάξετε μια συνάρτηση που να ελέγχει εάν μια κατάσταση είναι έγκυρη).

Μετακίνηση κύβων: Ένας κύβος που βρίσκεται στη θέση (x,y) είναι **ελεύθερος** εάν δεν υπάρχει κύβος στη θέση $(x,y+1)$. (Θα πρέπει να φτιάξετε μια συνάρτηση που να βρίσκει εάν ένας συγκεκριμένος κύβος είναι ελεύθερος σε μια δοθείσα κατάσταση). Κάθε **ελεύθερος** κύβος που βρίσκεται σε μια θέση (x,y) μπορεί σε μια μετακινηθεί σε **κενή θέση (x',y')** , υπό την προϋπόθεση ότι η νέα κατάσταση που προκύπτει από τη μετακίνηση είναι **έγκυρη**. Οπότε ένας ελεύθερος κύβος μπορεί να μετακινηθεί:

- είτε σε επιτρεπτή κενή θέση στο τραπέζι (οπότε $y'=1$)
- είτε πάνω σε κάποιον άλλο ελεύθερο κύβο της πρώτης σειράς (οπότε $y'=2$).
- είτε πάνω σε κάποιον άλλο ελεύθερο κύβο της δεύτερης σειράς (οπότε $y'=3$).

Κόστος Μετακίνησης ενός κύβου από τη θέση (x,y) στη θέση (x',y') :

- Εάν προκαλείται ανύψωση του κύβου ($y'>y$), τότε το κόστος της ενέργειας είναι $y'-y$.
- Εάν προκαλείται κατέβασμα του κύβου ($y'<y$), τότε το κόστος της ενέργειας είναι $0.5*(y-y')$.
- Εάν η μετακίνηση μεταφέρει τον κύβο στην ίδια σειρά ($y'=y$), τότε το κόστος της ενέργειας είναι 0.75.

Θέλουμε να βρούμε την ακολουθία ενεργειών ελάχιστου κόστους για να μεταβούμε από έναν **αρχικό έγκυρο** σχηματισμό (AK) των N κύβων (που θα καθορίζεται στην αρχή του προγράμματος) στον τελικό σχηματισμό (TK) που αναφέρεται παραπάνω. 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 report.pdf) γιατί η συνάρτηση $h(n)$ που σκεφτήκατε είναι αποδεκτή.

Για κάθε πρόβλημα που έχετε να λύσετε και για την ίδια αρχική κατάσταση, να εφαρμόσετε τόσο την μέθοδο UCS όσο και την μέθοδο A^* για να μπορείτε να συγκρίνετε τις μεθόδους. Πιο συγκεκριμένα, για κάθε μέθοδο να τυπώνετε: α) το μονοπάτι από AK προς TK που βρήκατε, β) το κόστος του μονοπατιού αυτού, και γ) τον αριθμό των επεκτάσεων που έγιναν. Να αναφέρετε στο κείμενο (report.pdf) τα συμπεράσματά σας σχετικά με την αποδοτικότητα της A^* σε σχέση με τη UCS (ποιότητα λύσης, αριθμός επεκτάσεων). Να εξετάσετε διάφορες τιμές του K ($K=1,2,3,\dots$)

Εργαστηριακή Άσκηση 2 (Κατασκευή Παιγνίου) (10%)

Να κατασκευάσετε πρόγραμμα το οποίο θα παίζει ενάντια σε κάποιο χρήστη το παρακάτω παίγνιο δύο παικτών που παίζουν εναλλάξ. Υπάρχουν M όμοιες κάρτες τοποθετημένες σε ένα τραπέζι ο οποίες είναι χωρισμένες σε K ομάδες. Κάθε ομάδα O_i ($i=1,\dots,K$) έχει αρχικά $A_i \geq 2$ κάρτες. Προφανώς το άθροισμα των A_i είναι ίσο με M . Οι τιμές των M , K και A_i δίνονται κατά την έναρξη του παιχνιδιού.

Κατά τη διάρκεια του παιχνιδιού **κάθε παίκτης αφαιρεί μία ή περισσότερες κάρτες μίας μόνο ομάδας**.

Για κάθε ομάδα O_i ορίζεται κατά την έναρξη του παιχνιδιού ο μέγιστος αριθμός καρτών B_i ($B_i < A_i$) που μπορεί να αφαιρέσει ένας παίκτης κάθε φορά που παίζει.

Ο παίκτης που παίζει κάθε φορά i επιλέγει μια από τις ομάδες (έστω την O_i) με μη μηδενικό αριθμό καρτών ($N_i > 0$) και μπορεί να αφαιρέσει έναν αριθμό καρτών αυτής της ομάδας από 1 έως $\min(B_i, N_i)$ (N_i είναι ο αριθμός καρτών της ομάδας O_i τη στιγμή που παίζει ο παίκτης). **Νικητής είναι ο παίκτης που θα αφαιρέσει την τελευταία κάρτα από το τραπέζι.**

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. Δεν απαιτείται κλάδεμα α - β).

Να επισυνάψετε αρχείο κειμένου που να αναφέρει το πώς ορίζετε την κατάσταση του παιγνίου, πώς ορίζετε την αξία των τελικών καταστάσεων και να περιέχει μια σύντομη περιγραφή του κώδικα που αναπτύξατε.