Τμήμα Μηχανικών Η/Υ και Πληροφορικής Πανεπιστημίου Πατρών
NE509: *Οικονομική Θεωρία & Αλγόριθμοι*
Ακαδημαϊκό Έτος 2022-23
Διδάσκων: Σπύρος Κοντογιάννης

1ο Σετ Ασκήσεων: Συνδυαστικά Παιχνίδια

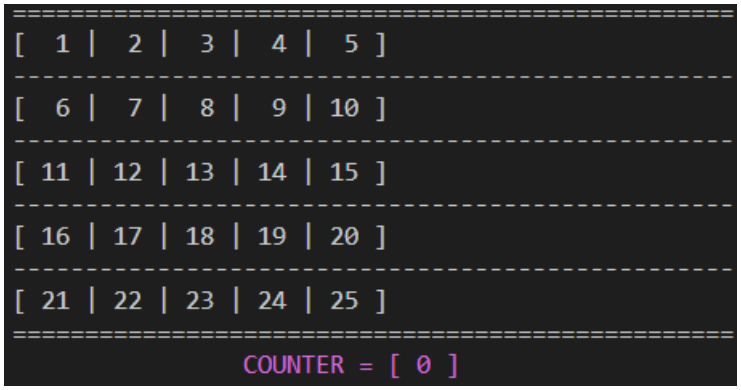
Ανακοίνωση: Παρασκευή, 24 Μαρτίου 2023 **Παράδοση:** Πέμπτη, 20 Απριλίου 2023
Τελευταία Ενημέρωση: Παρασκευή, 24 Μαρτίου 2023

1. Σύνοψη Εργασίας

Στόχος της παρούσας εργασίας είναι η εξοικείωση με τα συνδυαστικά παιχνίδια, και συγκεκριμένα, με μια διδιάστατη εκδοχή του παιχνιδιού NIM το οποίο μελετήθηκε στο πλαίσιο του μαθήματος. Θα πρέπει, αφού μελετήσετε προσεκτικά το συγκεκριμένο παιχνίδι, να είστε σε θέση να κατανοήσετε, για μικρές παρτίδες του N, πότε είναι εφικτό να νικήσετε και πότε όχι. Στη συνέχεια, θα πρέπει να υλοποιήσετε πρόγραμμα που εκτελεί μια παρτίδα ανάμεσα σε έναν άνθρωπο και τον υπολογιστή, επιτρέποντας μόνο συγκεκριμένα είδη κινήσεων (και για τους δύο). Θα πρέπει, επίσης, για τον υπολογιστή να υλοποιήσετε ρουτίνες που υλοποιούν συγκεκριμένες στρατηγικές, ενώ ο άνθρωπος θα μπορεί να επιλέγει αυθαίρετα τις κινήσεις του, αρκεί να υπακούει στους κανόνες του παιχνιδιού. Στόχος είναι να κατανοήσετε (και να εξηγήσετε) πότε ακριβώς ο υπολογιστής μπορεί να κερδίσει μια παρτίδα (ανάλογα με το αν ξεκινά πρώτος ή όχι, και με το μέγεθος της παλέτας). Εφόσον ο υπολογιστής έχει μπροστά του μια παρτίδα νίκης, θα πρέπει να είναι σε θέση να την κερδίσει, διαφορετικά να παίζει με συγκεκριμένη στρατηγική (πχ, τυχαία, ή firstfit, ή copycat) μέχρι να τελειώσει το παιχνίδι ή να αναγνωρίσει μια τετριμμένη παρτίδα που ΞΕΡΕΙ ότι μπορεί να την κερδίσει (και να παίξει αναλόγως).

2. Περιγραφή του Διδιάστατου NIM

Ως είσοδος δίνεται μια $N \times N$ παλέτα ($10 \geq N \geq 1$), της οποίας τα κελιά απαριθμούνται από επάνω-αριστερά (με την τιμή 1), μέχρι κάτω-δεξιά (με την τιμή N^2). Για παράδειγμα, για N=5, η παρτίδα εκκίνησης είναι η εξής:



Ο μετρητής στο κάτω μέρος της παλέτας απλά σημειώνει πόσα κελιά έχουν ήδη χρησιμοποιηθεί στο παιχνίδι μέχρι στιγμής (προφανώς, αρχικά είναι μηδέν). Θεωρούμε ότι κάποιος άνθρωπος (η **παίκτρια**) καλείται να παίξει ενάντια στον **υπολογιστή**.

2.1. Κανόνες του Παιχνιδιού

Οι κανόνες του παιχνιδιού είναι οι εξής:

1. Η παίκτρια αρχικά καλείται να επιλέξει χρώμα [**R**(ed) | **G**(reen)] και μέγεθος της παλέτας (η τιμή του N). Προφανώς, ο υπολογιστής λαμβάνει το άλλο χρώμα.
2. Επιλέγεται εντελώς τυχαία ποιος/α από τους δύο (η παίκτρια ή ο υπολογιστής) θα κάνει την πρώτη κίνηση.
3. Όποιος/α παίζει κάθε φορά, καλείται να επιλέξει ως κίνηση **τουλάχιστον ένα** και **το πολύ τρία ΚΕΝΑ κελιά**, στα οποία θα γράψει το δικό του/της χρώμα. Για τα κελιά που επιλέγονται θα πρέπει να ισχύουν τα εξής:
   a. Αν το πρώτο κελί που επιλέγεται βρίσκεται στην κύρια διαγώνιο (πχ, κάποιο από τα κελιά **1**, **7**, **13**, **19**, ή **25** στο προηγούμενο παράδειγμα), τότε δεν δίνεται δυνατότητα άλλης επιλογής, και το παιχνίδι συνεχίζεται με την επόμενη κίνηση του/της αντιπάλου.
   b. Αν το πρώτο κελί είναι μη-διαγώνιο, τότε δίνεται η δυνατότητα επιλογής ακόμη ενός ή δύο κελιών (ίσως όμως και κανενός), τα οποία θα πρέπει:
      i. Να είναι επίσης μη-διαγώνια.
      ii. Να βρίσκονται όλα στην ίδια γραμμή με το πρώτο στοιχείο, ή να βρίσκονται όλα στην ίδια στήλη με το πρώτο στοιχείο.
      iii. Να είναι συνεχόμενα στη συγκεκριμένη γραμμή ή στήλη.

   Για παράδειγμα, αν η παίκτρια είχε επιλέξει το κόκκινο χρώμα, και είχε επιλεγεί να κάνει αυτή την πρώτη κίνηση στην αρχικά κενή παλέτα, θα μπορούσε να ζητήσει για ως πρώτη κίνηση τα κελιά **{4,5}** (αλλά όχι τα **{2,4}**, ούτε τα **{1,2,3}**, ούτε τα **{1,7}**), κληροδοτώντας την εξής επόμενη παρτίδα στον υπολογιστή:

```
=================================================
[  1 |  2 |  3 |  R |  R ]
-------------------------------------------------
[  6 |  7 |  8 |  9 | 10 ]
-------------------------------------------------
[ 11 | 12 | 13 | 14 | 15 ]
-------------------------------------------------
[ 16 | 17 | 18 | 19 | 20 ]
-------------------------------------------------
[ 21 | 22 | 23 | 24 | 25 ]
=================================================
                COUNTER = [ 2 ]
=================================================
```

4. Το παιχνίδι συνεχίζεται μέχρι τη στιγμή που κάποιος από τους δύο συμμετέχοντες γράψει και στο τελευταίο κενό κελί της παλέτας.
5. Ηττημένος/η είναι όποιος/α κληθεί να κάνει κίνηση σε γεμάτη παλέτα. Για παράδειγμα, μετά από μια σειρά εναλλασσόμενων κινήσεων η προηγούμενη παρτίδα θα μπορούσε να είχε ολοκληρωθεί με νίκη της παίκτριας:

```
!!! You won this game !!!
        =================================================
        [  R |  R |  G |  R |  R ]
        -------------------------------------------------
        [  G |  R |  G |  R |  G ]
        -------------------------------------------------
        [  R |  R |  R |  G |  G ]
        -------------------------------------------------
        [  G |  G |  R |  G |  G ]
        -------------------------------------------------
        [  G |  R |  R |  R |  G ]
        =================================================
                        COUNTER = [ 25 ]
        =================================================
```

3. Δομή Δεδομένων για Αποτύπωση Εξέλιξης Παρτίδας

Για την καταγραφή της εξέλιξης του παιχνιδιού, θα πρέπει να χρησιμοποιήσετε μια απλή λίστα της Python με το όνομα $nimBoard$, με ακριβώς $N^2 + 1$ κελιά. Το κελί $nimBoard[0]$ είναι ο μετρητής των ήδη χρησιμοποιημένων κελιών στο παιχνίδι. Κάθε άλλο κελί αρχικοποιείται με την συμβολοσειρά ‘ ‘ (δηλαδή, ένας χαρακτήρας - κενό), κι εκπροσωπεί ένα συγκεκριμένο κελί της $N \times N$ παλέτας. Κατά την εξέλιξη του παιχνιδιού, τα κελιά του $nimBoard$ ενημερώνονται κατάλληλα ώστε κάθε φορά να καταγράφουν τα περιεχόμενα της παλέτας. Πχ, στο παραπάνω παράδειγμα, μετά την πρώτη κίνηση της παίκτριας, θα πρέπει να ισχύουν τα εξής:

- $nimBoard[0] == 2$
- $nimBoard[4] ==$ ‘R’
- $nimBoard[5] ==$ ‘R’
- $\forall k \notin \{0,4,5\}, theBoard[k] ==$ ‘ ’

Για διευκόλυνσή σας, θα σας δοθεί έτοιμη ρουτίνα, $drawNimPalette(board, N)$, για αποτύπωση της τρέχουσας κατάστασης του παιχνιδιού με τη μορφή της $N \times N$ παλέτας.

4. Ζητούμενα Εργασίας

Θα πρέπει να κάνετε στα εξής, στο πλαίσιο της παρούσας εργασίας:

4.1. Θεωρητικό Μέρος

Να εξετάσετε όλες τις (υπο)παρτίδες με το πολύ 5 ελεύθερα κελιά, σε μια $N \times N$ παρτίδα ($N \geq 5$) και για καθεμιά από αυτές να εξηγήσετε αν είναι παρτίδα νίκης ή ήττας για τον συμμετέχοντα που πρέπει να κάνει κίνηση. Φροντίστε να τεκμηριώσετε επαρκώς τον ισχυρισμό σας. Λάβετε υπόψη σας όλες τις μη-συμμετρικές υποπαρτίδες, ανάλογα με το αν κάποια από τα ελεύθερα κελιά ανήκουν ή όχι σε κοινές γραμμές / στήλες). Για παράδειγμα, αν έχουμε 2 κενά κελιά, υπάρχουν μόλις δύο μη συμμετρικές περιπτώσεις: Να είναι διαδοχικά σε κάποια γραμμή ή σε κάποια στήλη, ή να μην είναι διαδοχικά.

4.2. Προγραμματιστικό Μέρος 1: Υλοποίηση Παιχνιδιού

Να υλοποιήσετε πρόγραμμα που υλοποιεί τη διεξαγωγή μιας παρτίδας του διδιάστατου NIM, ανάμεσα στην παίκτρια και τον υπολογιστή. Το πρόγραμμά σας θα πρέπει:

- Να εκτελεί **σειρές κίνησης** (turns), όπου οι συμμετέχοντες εναλλάσσουν κινήσεις του παιχνιδιού.
- Να διαβάζει, σε κάθε σειρά, τις επιλογές κελιών του υπολογιστή ή της παίκτριας που έχει τη σειρά κίνησης.
- Να ελέγχει και να επιβάλλει την επιλογή μόνο επιτρεπτών κινήσεων, τόσο για την παίκτρια, όσο και για τον υπολογιστή.
- Να ελέγχει, μετά από κάθε κίνηση, αν η παρτίδα έχει ήδη τερματίσει, με ξεκάθαρο νικητή, οπότε να αποτυπώνει την τελική κατάσταση της (γεμάτης πλέον) παλέτας.
- Να ελέγχει, μετά από κάθε κίνηση, αν υπάρχου ακόμη κενά κελιά στην παλέτα, οπότε θα δίνεται η σειρά κίνησης στον/στην αντίπαλο και θα τυπώνεται η τρέχουσα κατάσταση της παλέτας.

Μια κίνηση της παίκτριας θα προσδιορίζεται από την (αναγκαία) πρώτη επιλογή ενός κενού κελιού, και στη συνέχεια από την επιλογή (από 0 έως 2) επιπρόσθετων κενών κελιών που, μαζί με το πρώτο κελί, απαρτίζουν νόμιμη κίνηση του παιχνιδιού. Αντίστοιχα, και μια κίνηση του υπολογιστή θα πρέπει να γίνει με συγκεκριμένη στρατηγική (βλ. επόμενη ενότητα).

Επίσης, σας παρέχονται έτοιμες ρουτίνες για τα εξής:

- $getRowAndColumn(move, N)$: Επιστρέφει τη γραμμή και τη στήλη του κελιού move, εντός της $N \times N$ παλέτας.
- $whoGoesFirst()$: Προσδιορίζει ποιος/ποια θα εκκινήσει την παρτίδα, γίνεται από το ίδιο το πρόγραμμα, με τη ρίψη ενός κέρματος.
- $drawNimPalette(nimBoard, N)$: Εκτυπώνει την τρέχουσα κατάσταση της παλέτας, για συγκεκριμένο περιεχόμενο του $nimBoard$.

4.3. Προγραμματιστικό Μέρος 2: Στρατηγικές για τον Υπολογιστή

Για μια επιλογή κίνησης, ο υπολογιστής θα πρέπει να επιλέγει με βάση κάποια στρατηγική. Ζητείται να υλοποιήσετε τρεις διαφορετικές στρατηγικές για τον υπολογιστή:

- $getComputerMove\_random(board, N)$: Επιλέγει ως πρώτο κενό κελί της επόμενης κίνησης του υπολογιστή, εντελώς τυχαία μεταξύ των κενών κελιών στην παλέτα. Αν είναι διαγώνιο κελί, τότε η κίνηση του υπολογιστή είναι πλήρης. Διαφορετικά, επιλέγει (ισοπίθανα) γραμμή ('row') ή στήλη ('column') και κάνει τις υπόλοιπες (από 0 έως 2) επιλογές κελιού, μεταξύ κελιών που είναι στην ίδια γραμμή, ή στην ίδια στήλη, με το πρώτο κελί, και είναι επίσης μη διαγώνια και είναι όλα μαζί διαδοχικά. Δεν είναι ανάγκη να δίνονται από το μικρότερο προς το μεγαλύτερο, ούτε να είναι όλα πριν ή μετά το πρώτο κελί. Αρκεί, τελικά, να είναι μεταξύ τους διαδοχικά.
- $getComputerMove\_firstfit(board, N)$: Επιλέγει ως πρώτο κενό κελί της επόμενης κίνησης του υπολογιστή, το πρώτο μη-κενό κελί που θα συναντήσει στο board. Αν είναι διαγώνιο κελί, τότε η κίνηση του υπολογιστή είναι πλήρης. Διαφορετικά, επιλέγει (ισοπίθανα) γραμμή ('row') ή στήλη ('column') και κάνει τις υπόλοιπες (από 0 έως 2) επιλογές κελιού, μεταξύ κελιών που είναι στην ίδια γραμμή, ή στην ίδια στήλη, με το πρώτο κελί, και είναι επίσης μη διαγώνια και είναι όλα μαζί διαδοχικά. Δεν είναι ανάγκη να δίνονται από το μικρότερο προς το μεγαλύτερο, ούτε να είναι όλα πριν ή μετά το πρώτο κελί. Αρκεί, τελικά, να είναι μεταξύ τους διαδοχικά.
- $getComputerMove\_copycat(nimBoard, opponentMove, N)$: Λαμβάνοντας ως είσοδο μια συλλογή από (1 έως τρία) κελιά (αυτά της τελευταίας κίνησης της παίκτριας), ο υπολογιστής επιλέγει ως κίνησή του εκείνη τη συλλογή κελιών που είναι συμμετρικά με αυτά, ως προς την κύρια διαγώνιο. Πχ, για την κίνηση **{<1,2>,<1,3>}** της παίκτριας, ο υπολογιστής θα πρέπει να (δοκιμάσει να) επιλέξει την κίνηση **{<2,1>,<3,1>}.** Αν αυτό δεν είναι εφικτό για οποιονδήποτε λόγο, χρησιμοποιείται μια επιλογή τυχαίας κίνησης με το πολύ το ίδιο πλήθος κελιών, πχ με τη ρουτίνα $getComputerMove\_firstfit(nimBoard, N)$. Ειδικά για την περίπτωση που η παίκτρια είχε επιλέξει ένα (μοναδικό) διαγώνιο κελί, πχ, το <k,k>, επιλέγεται ως κίνηση του υπολογιστή το επίσης διαγώνιο κελί <(N-k)+1, (N-k)+1>. Αν κάτι τέτοιο δεν είναι εφικτό, τότε επιλέγεται τυχαία κάποιο κενό διαγώνιο κελί. Αν η διαγώνιος είναι ήδη γεμάτη, τότε επιλέγεται το πρώτο μη-κενό κελί στο $nimBoard$.

5. ΠΑΡΑΔΟΣΗ ΕΡΓΑΣΙΑΣ

Δημιουργήστε τα εξής αρχεία:

(α) ΠΗΓΑΙΟΣ ΚΩΔΙΚΑΣ: Ένα συμπιεσμένο (ZIP) αρχείο, που να περιέχει όλα τα αρχεία του πηγαίου κώδικα που δημιουργήσατε για τις ανάγκες της εργασίας σας. Χρησιμοποιήστε την εξής ονοματολογία για το ZIP αρχείο που θα παραδώσετε: **NE509_LAB1_<PROVIDE YOUR AM HERE>_SOURCE-FILES.zip**.

(β) ΓΡΑΠΤΗ ΑΝΑΦΟΡΑ: Για τη γραπτή αναφορά σας (σε DOCX ή/και σε PDF μορφή), χρησιμοποιήστε την εξής ονοματολογία: **NE509_LAB1_<PROVIDE YOUR AM HERE>_REPORT.docx** ή **NE509_LAB1_<PROVIDE YOUR AM HERE>_REPORT.pdf**.

Στη γραπτή αναφορά σας θα πρέπει:

- να απαντάτε στο θεωρητικό μέρος όσο πιο εμπεριστατωμένα και τεκμηριωμένα μπορείτε, και
- να περιγράφετε αναλυτικά τις μεθόδους που υλοποιήσατε για το προγραμματιστικό μέρος της εργασίας (σαν ένα εγχειρίδιο χρήστη για το πρόγραμμά σας, δίνοντας τη σύνταξη, την είσοδο, την έξοδο και το σκεπτικό της υλοποίησής σας, αλλά όχι τον ίδιο τον κώδικα, δεν χρειάζεται στην αναφορά).