ΟΙΚΟΝΟΜΙΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ

ATHENS UNIVERSITY
OF ECONOMICS
AND BUSINESS

Θεωρία Παιγνίων και Αποφάσεων

Γεωργίου Αλέξιος-Λάζαρος(3180027)

Περιγραφή αρχείων:

RandomGenerator.py:

Αρχείο για τη δημιουργία των random αρχείων των παιχνιδιών.

Κάθε αρχείο παιχνιδιού έχει την εξής μορφή:

Τίτλος: "{Rule}_{Number}.txt" με Rule να είναι ένα από τα Plurality/Borda και Number ο αριθμός του αρχείου.

Έχω φτιάξει 360 αρχεία για κάθε κανόνα (από το 1 μέχρι το 360) για να έχω 30 αρχεία (στο περίπου) για κάθε συνδυασμό n,m τιμής. Τα πιθανά n είναι τα {5,6,7,8} και τα πιθανά m είναι τα {5,6,7}. Άρα έχουμε 4*3 = 12 συνδυασμούς

Στο εσωτερικό του αρχείου περιέχονται:

```
5  #Ο αριθμός παιχτών n
7 #Ο αριθμός επιλογών m
Plurality #Η συμβολοσειρά του κανόνα rule
7 1 6 3 2 5 4
5 2 1 6 4 7 3
6 2 4 1 3 5 7
1 7 5 2 3 4 6
7 6 3 5 1 4 2
#Ο πίνακας προτιμήσεων, κάθε γραμμή i έχει την σειρά προτίμησης του
παίχτη i, με σειρά επιλογής από αριστερά προς τα δεξιά οπότε έχω ένα
πίνακα με n γραμμές και m στήλες.
```

Main.py:

Στο αρχείο υλοποιείται η συνάρτηση bestResponseDynamics η οποία παίρνει το όνομα ενός αρχείου και εξάγει το αν βρήκαμε κάποιο σημείο ισορροπίας και σε ποιο γύρο σταμάτησε η εκτέλεση.

Στην συνέχεια με ένα απλό loop τρέχουμε την συνάρτηση για κάθε αρχείο και κρατάμε στατιστικά για την εύρεση ισορροπίας και τον γύρο.

Υλοποίηση bestResponseDynamics:

Στο πρώτο κομμάτι διαβάζουμε το αρχείο και εκχωρούμε σε μεταβλητές τα δεδομένα.

Οι προτιμήσεις αποθηκεύονται σε μια λίστα λιστών.

Λίστα λιστών playerspreferences [[1, 5, 3, 2, 4, 6], [2, 6, 1, 3, 4, 5],...]

Ανάλογα με τον κανόνα που έχει το αρχείο τρέχουμε τον εκάστοτε αλγόριθμο.

Αν ο κανόνας είναι **Plurality** τότε:

Φτιάχνουμε ένα πίνακα votedFor για να γνωρίζουμε την ψήφο κάθε παίχτη και να την προσαρμόζουμε εάν αυτή πρέπει να αλλάξει:

#Λίστα votedFor: [x1,x2,x3..,xn]

Φτιάχνουμε ένα leaderboarder το οποίο έχει τις επιλογές με την σειρά και πόσες ψήφους έχει η κάθε μια. Έχω προσθέσει και ένα πρώτο dummy στοιχείο [-1,-1] ώστε να υπάρχει παραλληλία με το id της επιλογής και την θέση του πίνακα (οι επιλογές ξεκινάνε από το 1).

#Λίστα λιστών leaderboarder [[-1,-1],[1, 0], [2, 0], [3, 0], [4, 0], [5, 0], [6, 0]]

Τώρα τρέχουμε την loop για την εύρεση του σημείου ισορροπίας μέχρι tMax γύρους:

```
while turns <= tMax and equilibriumFound == False:
    find current winner id preference
    for every player in players:
        if someone changed his vote in this loop
            break and go to next turn
        for every playerpreference of player:
            if changing vote to the preference cannot make it win
                continue to next preference
            if preference is the winner:
                then we do not change our vote since we could not
make previous preferences win so we do not have a reason to change
our vote
            else:
                we vote for the current preference
        if the player is the last one and we did not change any votes
during the loops
            we stop and declare the equilibrium found.
```

Ο αλγόριθμός είναι αυτός, στα σχόλια του κώδικα εξηγούνται περισσότερο οι υλοποιήσεις.

Είναι πιο εύκολη η κατανόηση του κώδικα αν έχουμε στο μυαλό μας την μορφή των δεδομένων/ λιστών.

Για το Plurality, τα στατιστικά για όλα τα 360 αρχεία (με όλους τους συνδυασμούς n, m) είναι:

```
Percentage of equilibrium convergence:  84.44444444444444
Average turns for equilibrium:  2.0
```

84% σύγκλιση και 2 γύρους μέσο όρο για την σύγκλιση σε σημείο ισορροπίας.

Δύο γύροι σημαίνει ότι στον πρώτο γίνεται κάποια αλλαγή (ένας παίχτης αλλάζει την ψήφο του) και στον δεύτερο ανιχνεύεται το σημείο ισορροπίας, άρα κατά μέσο όρο έχουμε μόνο μια αλλαγή ψήφου.

Αν ο κανόνας είναι **Borda** τότε:

Ακολουθούμε τον ίδιο αλγόριθμο με κάποιες μετατροπές.

Η Λίστα votedFor θα γίνει λίστα λιστών και θα έχει την ίδια μορφή με την playerspreferences. (Στην αρχή θα είναι ίσες)

Η προσαρμογή των ψήφων αλλάζει σύμφωνα με τον κανόνα Borda οπότε όπου προσαρμόζαμε τις ψήφους έχω βάλει ένα loop προσαρμογής αφού αλλάζουμε πολλές ψήφους. (Στην εξέταση ενός preference αν μπορούμε να κερδίσουμε αφαιρούμε όπως και στο plurality τις τωρινές ψήφους του παίχτη και ελέγχουμε άμα ισχύει η ανίσωση).

Στο Borda μπορούμε να δώσουμε m στην πρώτη μας επιλογή και 1 στην τελευταία (Κάτι σαν την ψηφοφορία της Eurovision άμα είχαμε μόνο 8 χώρες).

Οπότε ο έλεγχος αλλάζει στο αν η ψήφοι τις επιλογής που ελέγχουμε + m – 1 είναι μικρότεροι από τις ψήφους της νικητήριας τωρινής επιλογής χωρίς τις εκάστοτε ψήφους του παίχτη. (Σε κάθε επιλογή αφαιρούμε τις ψήφους του παίχτη για να ελέγξουμε και αν δεν μπορούμε να νικήσουμε τις ξανά προσθέτουμε στο leaderboarder)

Η αλλαγή της ψήφου σε περίπτωση που ο παίχτης μπορεί να κερδίσει κάτι καλύτερο γίνεται ως εξής.

```
# new_votes = [playerpreferenceid, usedtobefirstchoice,
usedtobesecondchoice, ... , winningchoice]
new_votes = votedFor[player]
new_votes.remove(winnerid)
new_votes.remove(playerpreferenceid)
new_votes.append(winnerid) #add it in last
new_votes.insert(0, playerpreferenceid) #add it in front
```

Αφαιρούμε την τωρινή νικητήρια επιλογή και την επιλογή που προτιμάμε από την λίστα και τις ξανά προσθέτουμε πίσω με την νικητήρια να είναι στο τέλος της λίστας και την προτιμώμενη στην αρχή.

Περισσότερα για τον κώδικα στα σχόλια.

```
Percentage of equilibrium convergence:  54.166666666666664
Average turns for equilibrium:  7.476923076923077
```

Τα στατιστικά για το Borda σε παρόμοια δεδομένα (360 παιχνίδια με τυχαία n,m).

*Παίρνοντας τυχαία n,m όπως και στο Plurality δεν αλλάζει πολύ το αποτέλεσμα αφού περίπου 30 θα είναι από κάθε συνδυασμό. Τα στατιστικά για κάθε συνδυασμό δεν περιμένω συγκριτικά με τον άλλον κανόνα ότι θα αλλάξουν ορισμένο ζευγάρι n,m, οπότε τα πήρα για όλα μαζί.

Φαίνεται ότι στο Borda υπάρχει λιγότερη πιθανότητα για σύγκλιση σε σημείο ισορροπίας (54% ενώ στο Plurality 84%) και οι γύροι που χρειάζονται για να βρεθεί είναι περίπου 6πλάσιοι αφού στο Plurality κατά μέσο όρο γίνετε μια αλλαγή στις ψήφους, ενώ εδώ 6.5 αλλαγές.

Αυτό συμβαίνει γιατί στο Borda οι παίχτες έχουν μεγαλύτερη δύναμη στην αλλαγή του αποτελέσματος, αφού μπορούν να «καταστρέψουν» πιο εύκολα μια επιλογή. Σκεφτείτε ότι κάποιος που έχει κίνητρο μπορεί να βάλει πρώτη επιλογή εκείνη που έχει κίνητρο ενώ αυτήν που πρέπει να χάσει τελευταία και έτσι μπορεί να «εκμεταλλευτεί» την επιλογή των άλλων παιχτών που μπορεί να έχουν ψηφίσει και αυτοί την πρώην νικητήρια επιλογή αλλά να προτιμούν την επιλογή που ο παίχτης κίνητρο να κερδίσει έστω και προτελευταία.

Επειδή τα παίγνια είναι πολλά θα σχολιάσω ένα από κάθε κανόνα:

(Επιλέγω κάποια μικρά παίγνια με λίγους γύρους για σύγκλιση (, για ευκολία)

Για το Borda:

Για πιο λεπτομερή αποτελέσματα για κάθε παίγνιο, αρκεί να βγάλετε τα σχόλια από τις γραμμές print στον κώδικα.

```
Running file Borda_18
===================
Got equilibrium at turn:  2
Voted for:  [[4, 3, 5, 2, 1], [4, 3, 2, 5, 1], [1, 5, 2, 4, 3], [5, 4, 3, 1, 2], [1, 3, 2, 4, 5], [1, 5, 4, 2, 3], [4, 1, 2, 3, 5]]
Leaderboarder:  [[1, 23], [2, 17], [3, 19], [4, 26], [5, 20]]
```

Το παιχνίδι 18 συγκλίνει με μια αλλαγή (υπάρχουν αρκετά που συγκλίνουν χωρίς αλλαγές και κάποια που χρειάζονται 40 γύρους και βέβαια κάποια που δεν συγκλίνουν ποτέ)

Το παίγνιο 18:

```
7
5
Borda
4 3 5 2 1
3 4 1 2 5
1 5 2 4 3
5 4 3 1 2
1 3 2 4 5
1 5 4 2 3
4 1 2 3 5
```

Βλέπουμε ότι με μια αλλαγή έχουμε σημείο ισορροπίας.

Βαθμοί κάθε επιλογής:

1: 1 + 3 + 5 + 2 + 5 + 5 + 4 = 25

2: 2 + 2 + 3 + 1 + 3 + 2 + 3 = 16

3: 4 + 5 + 1 + 3 + 4 + 1 + 2 = 20

4: 5 + 4 + 2 + 4 + 2 + 3 + 5 = 25

5: 3 + 1 + 4 + 5 + 1 + 4 + 1 = 19

Άρα κερδίζει η επιλογή 1 λόγω μικρότερου id (tiebreaker).

Οι αρχικοί βαθμοί επιλογής είναι και το κοινωνικό όφελος της επιλογής. Δηλαδή στο συγκεκριμένο παίγνιο το καλύτερο για το σύνολο είναι η επιλογή 1 (και η 4).

Ο παίχτης 1 ενώ κερδίζει η επιλογή 1 δεν μπορεί να κάνει κάτι αφού έχει ψηφίσει την 1 τελευταία και την 4 πρώτη. Δεν μπορεί να κερδίσει σε καμιά άλλη επιλογή ακόμα και αν αλλάξει την ψήφο του.

Ο παίχτης 2 δεν μπορεί να κάνει την 3 επιλογή να κερδίσει, μπορεί όμως να κάνει την 4 επιλογή. Θα ψηφίσει την 4 και θα ψηφίσει την 1 τελευταία, άρα

Ο παίχτης 2 έχει κίνητρο να αλλάξει την ψήφο του σε **4,3,2,5,1** από 3,4,1,2,5.

Στον επόμενο γύρο κανένας παίχτης δεν μπορεί να αλλάξει την κατάσταση σε κάτι καλύτερο για εκείνον, όπως βλέπουμε και από το log του αποτελέσματος.

Νικάει η επιλογή 4 με 26 πόντους.

Αν όλοι δήλωναν την αρχική τους προτίμηση θα κέρδιζε η 1 ενώ τώρα κερδίζει η 4.

Δεν υπάρχει πρόβλημα στο κοινωνικό όφελος στο συγκεκριμένο παίγνιο αφού και η επιλογή 4 έχει το ίδιο κοινωνικό όφελος με βάση τις προτιμήσεις των παιχτών (25).

Αυτό βέβαια δεν συμβαίνει πάντα, σε άλλα παίγνια η επιλογή που ισορροπείται το παίγνιο δεν είναι αυτή με το μεγαλύτερο κοινωνικό όφελος.

Για το Plurality:

```
Running file Plurality_1
===================
Got equilibrium at turn:  2
Voted for:  [7, 1, 6, 1, 7]
Leaderboarder:  [[1, 2], [2, 0], [3, 0], [4, 0], [5, 0], [6, 1], [7,
2]]
```

Το παίγνιο 1:

```
5
7
Plurality
7 1 6 3 2 5 4
5 2 1 6 4 7 3
6 2 4 1 3 5 7
1 7 5 2 3 4 6
7 6 3 5 1 4 2
```

Αρχική ψήφοι: [7, 5, 6, 1, 7] Σύνολο ψήφων: [[1, 1], [2, 0], [3, 0], [4, 0], [5, 1], [6, 1], [7, 2]]

Ο παίχτης 1 δεν έχει κίνητρο να αλλάξει αφού η πρώτη επιλογή του νικάει (7).

Ο παίχτης 2 χάνει ψηφίζοντας την 5, δεν μπορεί να κάνει την 2 να κερδίσει αφού 0 + 1 <= 2, Μπορεί όμως να κάνει την 1 να κερδίσει αλλάζοντας την ψήφο του σε 1, αφού 1 + 1 <= 2 (1<7 tiebreaker). Άρα θα αλλάξει την ψήφο του από 5 σε 1.

Στον επόμενο γύρο δεν έχει κίνητρο να αλλάξει κανένας παίχτης την ψήφο του.

(Όσοι δεν ψηφίζουν 7, έχουν την 1 σε μεγαλύτερη προτίμηση από την 7)

Άρα το παιχνίδι βρίσκετε σε ισορροπία με νικητή την επιλογή 1 με 2 ψήφους.

Το κοινωνικό όφελος αν το μετρήσουμε με την λογική του Plurality (1 αν είναι πρώτη επιλογή, 0 αν δεν είναι) τότε μειώνετε, αφού 2 άτομα ήθελα την 7 και στο τέλος επιλέγετε η 1 που είχε ένα άτομο με πρώτη προτίμηση. (Δεν έχει τόσο νόημα η μέτρηση του κοινωνικού οφέλους με αυτόν τον τρόπο εκτός αν στο παίγνιο οι παίχτες θα είναι χαρούμενοι μόνο με την πρώτη τους επιλογή, και η σειρά για τις υπόλοιπες δεν είχε σημασία)

Το κοινωνικό όφελος αν το μετρήσουμε με την λογική του Borda τότε:

1:6+5+4+7+3=25

2:3+6+6+4+1=20

3:4+1+3+3+5=16

4:1+3+5+2+2=13

5:2+7+2+5+4=20

6:5+4+7+1+6=23

7:7+2+1+6+7=23

Αν οι παίχτες ψήφιζαν με βάση την προτίμηση τους το κοινωνικό όφελος θα ήταν 23 (νικητής επιλογή 7), ενώ με την αλλαγή της στρατηγικής του παίχτη 2, τώρα νικάει η επιλογή 1 με κοινωνικό όφελος 25.

Άρα η αλλαγή της στρατηγικής του παίχτη 2, ωφέλησε το κοινωνικό σύνολο (25>23).

Είναι πολύ πιθανό στα παίγνια με κανόνα Plurality, να καταλήγουμε σε παίγνιο με μεγαλύτερο κοινωνικό όφελος, και πιθανών με μεγάλη διαφορά εφόσον οι προτιμήσεις έχουν επιλεγεί με τυχαίο τρόπο. Αυτό συμβαίνει αφού στο Plurality μετράει μόνο η πρώτη προτίμηση.