A numerical study of fixation probabilities for strategies in the Iterated Prisoner's Dilemma

Marc Harper Vincent Knight

Abstract

The Iterated Prisoner's Dilemma is a well established framework for the study of emergent behaviour. In this paper an extensive numerical study of the evolutionary dynamics of this framework are presented.

Fixation probabilities for Moran processes are obtained for 172 different strategies. This is done in both a standard 200 turn interaction and a noisy setting.

To the authors knowledge this is the largest such study. It allows for insights about the behaviour and performance of strategies with regard to their survival in an evolutionary setting.

1 Introduction

Since the formulation of the Moran Process in [6], this model of evolutionary population dynamics has been used to gain insights about the evolutionary stability of strategies in a number of settings. Similarly since the first Iterated Prisoner's Dilemma (IPD) tournament described in [1] the Prisoner's dilemma has been used to understand the evolution of cooperative behaviour in complex systems.

The analytical models of a Moran process are based on the relative fitness between two strategies and take this to be a fixed value r [8]. This is a valid model for simple strategies of the Prisoner's Dilemma such as to *always cooperate* or *always defect*. This manuscript provides a detailed numerical analysis of **172** complex and adaptive strategies for the IPD. In this case the relative fitness of a strategy is dependent on the population distribution.

Further deviations from the analytical model occur when interactions between players are subject to uncertainty. This is referred to as noise and has been considered in the IPD setting in [3, 7, 10]. Noise is also considered here.

This work provides answers to the following questions:

1. What strategies are good invaders?
2. What strategies are good at resisting invasion?
3. How does the population size affect these findings?

$$\rho = \frac{1 - r^{-i}}{1 - r^{-N}}$$

and $\rho = 1/N$ if $r = 1$ (the neutral fixation probability).

This corresponds to a game matrix [[1, 1], [r, r]] (or [[r, r], [1, 1]]), which is of course not what we have – it's a little complicated because our "fitness" is not the payout from the game matrix, rather the sum of the total scores of all the interactions each round. So ALLC and TFT are neutral wrt to each other because they will have the same score each round, giving an effective fitness landscape $f(i, N-i) = A[i, N-i]^T$ given by the matrix $A = [[1, 1], [1, 1]]$. This means that noise and the number of turns per Moran round are significant parameters. I think we should fix the turns at 200; some recent authors run the turns to infinity (to reach stationarity on the sub-"Markov process" on the states (C, C), (C, D), (D, C), (D, D)) but we can't analytically compute the stationary distribution for strategies that use more than one round of memory (and it's not really a Markov process for more than one round of memory anyway). Plus it's unrealistic, and ultimately just amounts to a transform of the game matrix.

To see if one strategy is not neutral with respect to another, we want to empirically measure the fixation probability and compare to the neutral rate. To do this right we need a lot of counts, since we're estimating a binomial probability p with variance $p(1-p)/k$ and p is close to $1/N$. To get the variance small you need something like $k > 1000$ observations (we can work out the precise requirements).

Note we're not estimating r for each strategy (pair) since we're in a frequency dependent situation, so we need to look at the population states (1, N-1) and (N-1, 1) for every pair of strategies, i.e. we can't assume that we're in a $\rho \leftrightarrow 1-\rho$

symmetry. More precisely, $\rho_{(1,N-1)}! = 1 - \rho_{(N-1,1)}$ in general. However we can (for fun) compute r from ρ with Newton's method (it's not easily invertable for $N > 3$), or take a Bayesian approach on what the distribution of ρ is and then compute a distribution for r in the usual way.

A nice addition would be, for an interesting combination of strategies, to measure the fixation value for all $(i, N-i)$ and compare to the above formula for the value of r derived from the $(1, N-1)$ case. This would show how much we deviate from frequency independence.

The existing notebook attempts to get at 1 and 2 by looking at the distributions of fixation probabilities for each strategy – that's what the box plots for each N try to visualize for particular N, and the "Player Rankings by Median vs. Population Size" for how the cooperative strategies become more successful as N increases. That plot is the main takeaway IMO, and reinforces the "evolution of cooperation" narrative that's so popular. We can tie back to Press and Dyson here – yes, ZD strategies are good Head-to-Head and in small populations, but they aren't great when the population size gets bigger. How much bigger? Even at N=4 there is a dramatic decline for ZD-extort. Note that this goes against the claims of Stewart and Plotkin (they claimed that ZD strategies basically dominate the Moran process no matter how much memory you allow). This also matches our tournament results – ZD strategies win matches but not tournaments.

It would be great to see how the ensemble strategies (meta strategies) fare, if we don't mind burning the CPU cycles. I left them out of my initial analysis.

More future work: * Mutation – for mutation we no longer have fixation, rather a stationary distribution. This may require some more programming to compute efficiently (perhaps my stationary library). There's a lot of interesting work to do here.

Ip think we'd want to include a few of the heatmaps in the final section of the notebook for some interesting cases, like FoolMeOnce, EvolvedLookerUp, etc. Pushing N higher will make all the plots more interesting. How high we can get N? I'd really like to get it to N ¿= 11.

Structure:

- Overview of Moran processes;
- Review of the literature ([4, 8, 6]);
- Short discussion about the Axelrod library.

Some work has looked at evolutionary stability of strategies within the Prisoner's Dilemma [5] but this is not done in the more widely used setting of the Moran process but in terms of infinite population stability.

In [2] Moran processes are looked at in a theoretic framework for a small subset of strategies.

In [4] machine learning techniques are used to train a strategy capable of resisting invasion and also invade any memory one strategy.

2 Methodology

To carry out this large numerical experiment 172 strategies are used from [9]. These include 169 default strategies in the library at the time (excluding strategies classified as having a long run time) as well as the following 3 finite state machine strategies:

Appendix A shows all the players in question. More information about each player can be obtained in the documentation for [9]. The memory depth of the used strategies is shown in Table 1a.

Memory Depth	0	1	2	3	4	5	6	9	10	11	12	16	20	40	200	∞
Count	3	31	12	8	2	6	1	1	5	1	1	2	2	2	1	94

(a) Memory depth

Stochastic	Count
False	123
True	49

(b) Stochastic versus deterministic

Table 1: Summary of properties of used strategies

All strategies are paired and these pairs are used in 1000 repetitions of a Moran process assuming a starting population of $(N/2, N/2)$. This is repeated for even N between 2 and 14. The fixation probability is then estimated for each value of N.

Note that due to the high computational cost of these experiments, for any given interaction between two players within the Moran process the outcome is sampled from a pre computed cache of 10000 match outcomes. This is carried out using the approximate Moran process implemented in [9].

As an example, Figure 1 shows the scores between two players that over the 10000 outcomes gives 6817 different scores.

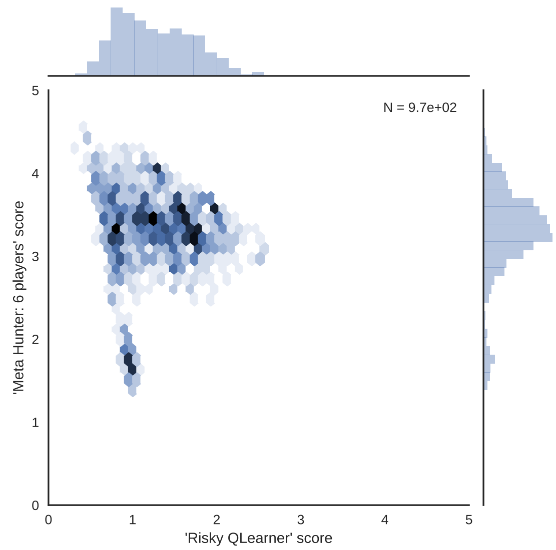


Figure 1: All possible scores for the pair of strategies that have the most different number of match outcomes

3 Validation

As described in [8] Consider the payoff matrix:

$$M = \begin{pmatrix} a, b \\ c, d \end{pmatrix} \tag{1}$$

The expected payoffs of i players of the first type in a population with $N - i$ players of the second type are given by:

$$F_i = \frac{a(i-1) + b(N-i)}{N-1} \tag{2}$$

$$G_i = \frac{ci + d(N-i-1)}{N-1} \tag{3}$$

With an intensity of selection ω the fitness of both strategies is given by:

$$f_i = 1 - \omega + \omega F_i \tag{4}$$

$$g_i = 1 - \omega + \omega G_i \tag{5}$$

The transitions within the birth death process that underpins the Moran process are then given by:

$$p_{i,i+1} = \frac{if_i}{if_i + (N-i)g_i} \frac{N-i}{N} \tag{6}$$

$$p_{i,i-1} = \frac{(N-i)g_i}{if_i + (N-i)g_i} \frac{i}{N} \tag{7}$$

$$p_{ii} = 1 - p_{i,i+1} - p_{i,i-1} \tag{8}$$

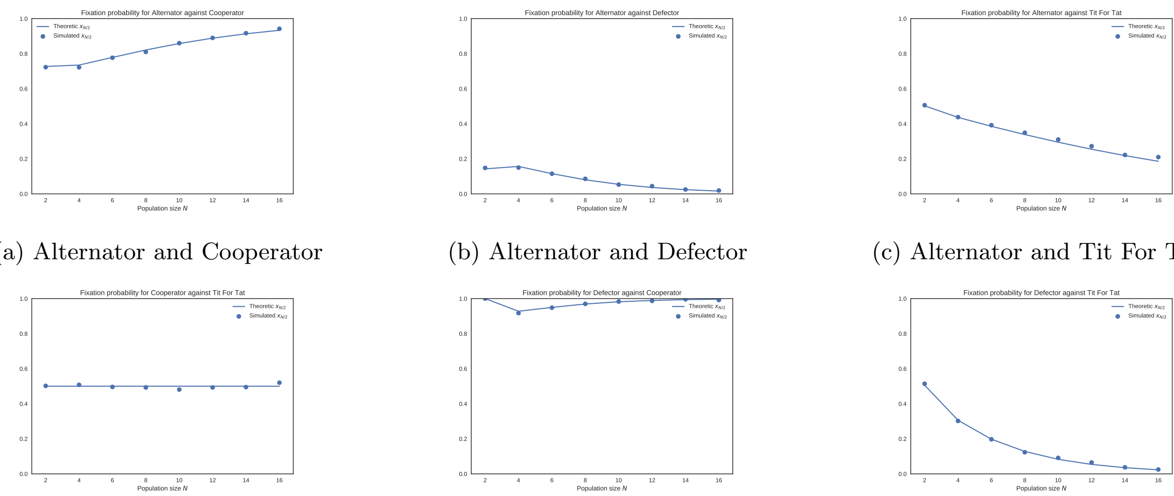
Using this it is a known result that the fixation probability of the first strategy in a population of i individuals of the first type (and $N - i$ individuals of the second. We have:

$$x_i = \frac{1 + \sum_{j=1}^{i-1} \prod_{k=1}^{j} \gamma_j}{1 + \sum_{j=1}^{N-1} \prod_{k=1}^{j} \gamma_j} \tag{9}$$

where:

$$\gamma_j = \frac{p_{j,j-1}}{p_{j,j+1}}$$

Using this comparisons of $x_{N/2}$ are shown in Figure 2. Note that these are all deterministic strategies and show a perfect match up between the expected value of (9) and the actual Moran process.



(a) Alternator and Cooperator

(b) Alternator and Defector

(c) Alternator and Tit For Tat

(d) Cooperator and Tit For Tat

(e) Defector and Cooperator

(f) Defector and Tit For Tat

Figure 2: Comparison of theoretic and actual Moran Process fixation probabilities for **deterministic** strategies

Figure 3 shows the fixation probabilities for stochastic strategies. These are no longer a good match which highlights the weakness of the analytical formulae the relies on the average payoffs. A detailed analysis of the 172 strategies considered will be shown in the next Section.

4 Numerical results

Structure:

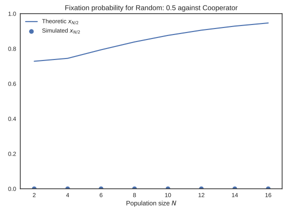
- General overview of the data obtained;
- Inclusion of most of the work in `Moran.ipynb`.

5 Conclusion

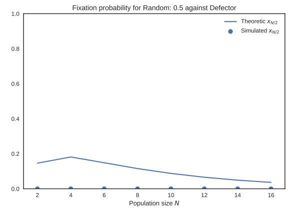
Beyond the raw data, we should try to estimate the strategies that are 1) most resistant to invasion 2) the best invaders 3) "most neutral"

as a function of N across the entire population of strategies. This can really open up if you want to say optimize a parameterized strategy to be most resistant to invasion (a topic of future work, perhaps) – for example Random(p) for what p is best?

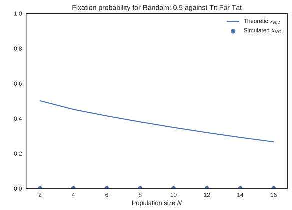
Further Variants (possible additions or future papers): * Noise * Spatial structure * More than two types in the population * Modified Moran processes (e.g. Fermi selection with the strength of selection coefficient) * Altered game matrices



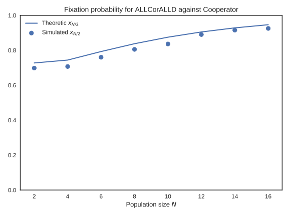
(a) Random and Cooperator



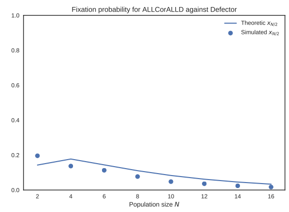
(b) Random and Defector



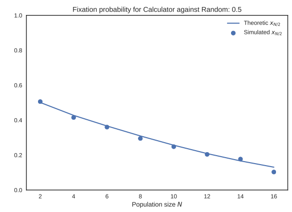
(c) Random and Tit For Tat



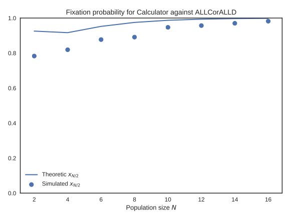
(d) All C or all D and Cooperator



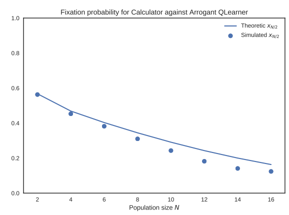
(e) All C or all D and Defector



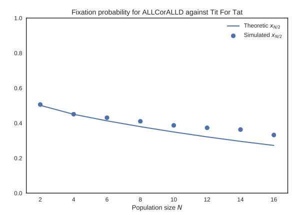
(f) Calculator and Random



(g) Calculator and All C or all D



(h) Calculator and Arrogant Q learner



(i) All C or all D and Tit For Tat

Figure 3: Comparison of theoretic and actual Moran Process fixation probabilities for **stochastic** strategies

Noise is especially interesting because a lot of the cooperative strategies are going to appear neutral to each other (since neither will cast a D unprovoked). A little bit of noise should shuffle the ranks around quite a bit, and show off the abilities of e.g. OmegaTFT. Might be worth including at least one of the "Player Rankings by Median vs. Population Size" plots for some value of noise (such as 0.05).

References

[1] R. Axelrod. "Effective Choice in the Prisoner's Dilemma". In: *Journal of Conflict Resolution* 24.1 (1980), pp. 3–25.

[2] Seung Ki Baek et al. "Comparing reactive and memory- one strategies of direct reciprocity". In: *Nature Publishing Group* (2016), pp. 1–13. DOI: `10.1038/srep25676`. URL: `http://dx.doi.org/10.1038/srep25676`.

[3] Jonathan Bendor. "Uncertainty and the Evolution of Cooperation". In: *The Journal of Conflict Resolution* 37.4 (1993), pp. 709–734.

[4] Christopher Lee, Marc Harper, and Dashiell Fryer. "The Art of War: Beyond Memory-one Strategies in Population Games". In: *Plos One* 10.3 (2015), e0120625. ISSN: 1932-6203. DOI: `10.1371/journal.pone.0120625`. URL: `http://dx.plos.org/10.1371/journal.pone.0120625`.

[5] Jiawei Li, Graham Kendall, and Senior Member. "The effect of memory size on the evolutionary stability of strategies in iterated prisoner ' s dilemma". In: X.X (2014), pp. 1–8.

[6] P.A.P. Moran. "Random Processes in Genetics". In: April (1957), pp. 60–71.

[7] M Nowak and K Sigmund. "A strategy of win-stay, lose-shift that outperforms tit-for-tat in the Prisoner's Dilemma game." In: *Nature* 364.6432 (1993), pp. 56–58. ISSN: 0028-0836. DOI: `10.1038/364056a0`.

[8] Martin A Nowak. *Evolutionary Dynamics: Exploring the Equations of Life.* Cambridge: Harvard University Press. ISBN: 0674023382. DOI: `10.1086/523139`.

[9] The Axelrod project developers. *Axelrod: v2.9.0.* Apr. 2016. DOI: `499122`. URL: `http://dx.doi.org/10.5281/zenodo.499122`.

[10] Jianzhong Wu and Robert Axelrod. “How to Cope with Noise in the Iterated Prisoner’s Dilemma”. In: *The Journal of Conflict Resolution* 39.1 (1995).

A List of players

1. Adaptive
2. Adaptive Tit For Tat: 0.5
3. Aggravater
4. ALLCorALLD
5. Alternator
6. Alternator Hunter
7. AntiCycler
8. Anti Tit For Tat
9. Adaptive Pavlov 2006
10. Adaptive Pavlov 2011
11. Appeaser
12. Arrogant QLearner
13. Average Copier
14. Better and Better
15. BackStabber: (’D’, ’D’)
16. Bully
17. Calculator
18. Cautious QLearner
19. Champion
20. CollectiveStrategy
21. Contrite Tit For Tat
22. Cooperator
23. Cooperator Hunter
24. Cycle Hunter
25. Cycler CCCCCD
26. Cycler CCCD
27. Cycler CCD
28. Cycler DC
29. Cycler DDC
30. Cycler CCCDCD
31. Davis: 10
32. Defector
33. Defector Hunter
34. Desperate
35. DoubleCrosser: (’D’, ’D’)
36. Doubler
37. EasyGo
38. Eatherley
39. Eventual Cycle Hunter
40. Evolved ANN
41. Evolved ANN 5
42. Evolved ANN 5 Noise 05
43. Evolved FSM 4
44. Evolved FSM 16
45. Evolved FSM 16 Noise 05
46. EvolvedLookerUp1_1_1
47. EvolvedLookerUp2_2_2
48. Evolved HMM 5
49. Feld: 1.0, 0.5, 200
50. Firm But Fair
51. Fool Me Forever
52. Fool Me Once
53. Forgetful Fool Me Once: 0.05
54. Forgetful Grudger
55. Forgiver
56. Forgiving Tit For Tat
57. Fortress3
58. Fortress4
59. GTFT: 0.33
60. General Soft Grudger: n=1,d=4,c=2
61. Soft Go By Majority
62. Soft Go By Majority: 10
63. Soft Go By Majority: 20
64. Soft Go By Majority: 40
65. Soft Go By Majority: 5
66. ϕ
67. Gradual
68. Gradual Killer: (’D’, ’D’, ’D’, ’D’, ’D’, ’C’, ’C’)
69. Grofman
70. Grudger
71. GrudgerAlternator
72. Grumpy: Nice, 10, -10
73. Handshake
74. Hard Go By Majority
75. Hard Go By Majority: 10
76. Hard Go By Majority: 20
77. Hard Go By Majority: 40
78. Hard Go By Majority: 5
79. Hard Prober
80. Hard Tit For 2 Tats
81. Hard Tit For Tat
82. Hesitant QLearner
83. Hopeless
84. Inverse
85. Inverse Punisher
86. Joss: 0.9
87. Knowledgeable Worse and Worse
88. Level Punisher
89. Limited Retaliate: 0.1, 20
90. Limited Retaliate 2: 0.08, 15
91. Limited Retaliate 3: 0.05, 20
92. Math Constant Hunter

93. Naive Prober: 0.1
94. MEM2
95. Negation
96. Nice Average Copier
97. Nydegger
98. Omega TFT: 3, 8
99. Once Bitten
100. Opposite Grudger
101. π
102. Predator
103. Prober
104. Prober 2
105. Prober 3
106. Prober 4
107. Pun1
108. PSO Gambler 1_1_1
109. PSO Gambler 2_2_2
110. PSO Gambler 2_2_2 Noise 05
111. PSO Gambler Mem1
112. Punisher
113. Raider
114. Random: 0.5
115. Random Hunter
116. Remorseful Prober: 0.1
117. Resurrection
118. Retaliate: 0.1
119. Retaliate 2: 0.08
120. Retaliate 3: 0.05
121. Revised Downing: True
122. Ripoff
123. Risky QLearner
124. SelfSteem
125. ShortMem
126. Shubik
127. Slow Tit For Two Tats
128. Slow Tit For Two Tats 2
129. Sneaky Tit For Tat
130. Soft Grudger
131. Soft Joss: 0.9
132. SolutionB1
133. SolutionB5
134. Spiteful Tit For Tat
135. Stalker: D
136. Stochastic Cooperator
137. Stochastic WSLS: 0.05
138. Suspicious Tit For Tat
139. Tester
140. ThueMorse
141. ThueMorseInverse
142. Thumper
143. Tit For Tat
144. Tit For 2 Tats
145. Tricky Cooperator
146. Tricky Defector
147. Tullock: 11
148. Two Tits For Tat
149. VeryBad
150. Willing
151. Winner12
152. Winner21
153. Win-Shift Lose-Stay: D
154. Win-Stay Lose-Shift: C
155. Worse and Worse
156. Worse and Worse 2
157. Worse and Worse 3
158. ZD-Extort-2: 0.1111111111111111, 0.5
159. ZD-Extort-2 v2: 0.125, 0.5, 1
160. ZD-Extort-4: 0.23529411764705882, 0.25, 1
161. ZD-GTFT-2: 0.25, 0.5
162. ZD-GEN-2: 0.125, 0.5, 3
163. ZD-SET-2: 0.25, 0.0, 2
164. e
165. Meta Hunter: 6 players
166. Meta Hunter Aggressive: 7 players
167. Meta Majority Memory One: 31 players
168. Meta Winner Memory One: 31 players
169. NMWE Memory One: 31 players
170. FSM Player: [(0, 'C', 7, 'C'), (0, 'D', 1, 'C'), (1, 'C', 11, 'D'), (1, 'D', 11, 'D'), (2, 'C', 8, 'D'), (2, 'D', 8, 'C'), (3, 'C', 3, 'C'), (3, 'D', 12, 'D'), (4, 'C', 6, 'C'), (4, 'D', 3, 'C'), (5, 'C', 11, 'C'), (5, 'D', 8, 'D'), (6, 'C', 13, 'D'), (6, 'D', 14, 'C'), (7, 'C', 4, 'D'), (7, 'D', 2, 'D'), (8, 'C', 14, 'D'), (8, 'D', 8, 'D'), (9, 'C', 0, 'C'), (9, 'D', 10, 'D'), (10, 'C', 8, 'C'), (10, 'D', 15, 'C'), (11, 'C', 6, 'D'), (11, 'D', 5, 'D'), (12, 'C', 6, 'D'), (12, 'D', 9, 'D'), (13, 'C', 9, 'D'), (13, 'D', 8, 'D'), (14, 'C', 8, 'D'), (14, 'D', 13, 'D'), (15, 'C', 4, 'C'), (15, 'D', 5, 'C')], 1, C
171. FSM Player: [(0, 'C', 13, 'D'), (0, 'D', 12, 'D'), (1, 'C', 3, 'D'), (1, 'D', 4, 'D'), (2, 'C', 14, 'D'), (2, 'D', 9, 'D'), (3, 'C', 0, 'C'), (3, 'D', 1, 'D'), (4, 'C', 1, 'D'), (4, 'D', 2, 'D'), (5, 'C', 12, 'C'), (5, 'D', 6, 'C'), (6, 'C', 1, 'C'), (6, 'D', 14, 'D'), (7, 'C', 12, 'D'), (7, 'D', 2, 'D'), (8, 'C', 7, 'D'), (8, 'D', 9, 'D'), (9, 'C', 8, 'D'), (9, 'D', 0, 'D'), (10, 'C', 2, 'C'), (10, 'D', 15, 'C'), (11, 'C', 7, 'D'), (11, 'D', 13, 'D'), (12, 'C', 3, 'C'), (12, 'D', 8, 'D'), (13, 'C', 7, 'C'), (13, 'D', 10, 'D'), (14, 'C', 10, 'D'), (14, 'D', 7, 'D'), (15, 'C', 15, 'C'), (15, 'D', 11, 'D')], 1, C
172. FSM Player: [(0, 'C', 0, 'C'), (0, 'D', 3, 'C'), (1, 'C', 5, 'D'), (1, 'D', 0, 'C'), (2, 'C', 3, 'C'), (2, 'D', 2, 'D'), (3, 'C', 4, 'D'), (3, 'D', 6, 'D'), (4, 'C', 3, 'C'), (4, 'D', 1, 'D'), (5, 'C', 6, 'C'), (5, 'D', 3, 'D'), (6, 'C', 6, 'D'), (6, 'D', 6, 'D'), (7, 'C', 7, 'D'), (7, 'D', 5, 'C')], 1, C