Assignment - 1 (Theory)

①

(a) For my entry number, $T_1 T_2 = 99$

$T_3 T_4 = 19$

So, the question becomes,

out of 118 Rounds what is the probability that Alice wins 99 rounds and Bob wins 19 given they both play aggresively only.

Assuming that Alice wins the first round and Bob wins the second, we can have all possible combinations of choosing 98 games (in which Alice wins) out of the remaining 116 games.

BUT! The probability of Alice winning any game is dependent on the number of games Bob has won, so we can only apply brute force in this part to calculate the required probability via all possible paths.

I have used the recursive algorithm given as :

$$\underbrace{dp}_{}[\underbrace{a}_{\text{No of wins of Alice}}][\underbrace{b}_{\text{No of wins of Bob}}] = \underbrace{\left(\frac{b}{a+b-1}\right)}_{\text{Probability of Alice winning when she has won (a-1) and Bob has won (b) rounds.}} * dp[a-1][b] + \underbrace{\left(\frac{a}{a+b-1}\right)}_{\text{Probability of Bob winning when alice has won 'a' times and bob has won 'b-1' times}} * dp[a][b-1]$$

Here $dp[x][y]$ represents the probability that alice has won 'x' rounds and bob has won 'y' rounds out of $x+y$ rounds.

We are given $dp[1][1] = 1$ (Given)

$dp[i][0] = 0 \ \forall i$

$dp[0][j] = 0 \ \forall j$

(It is given Alice & Bob have won one game each)

Now we calculate $dp[99][19]$ iteratively.

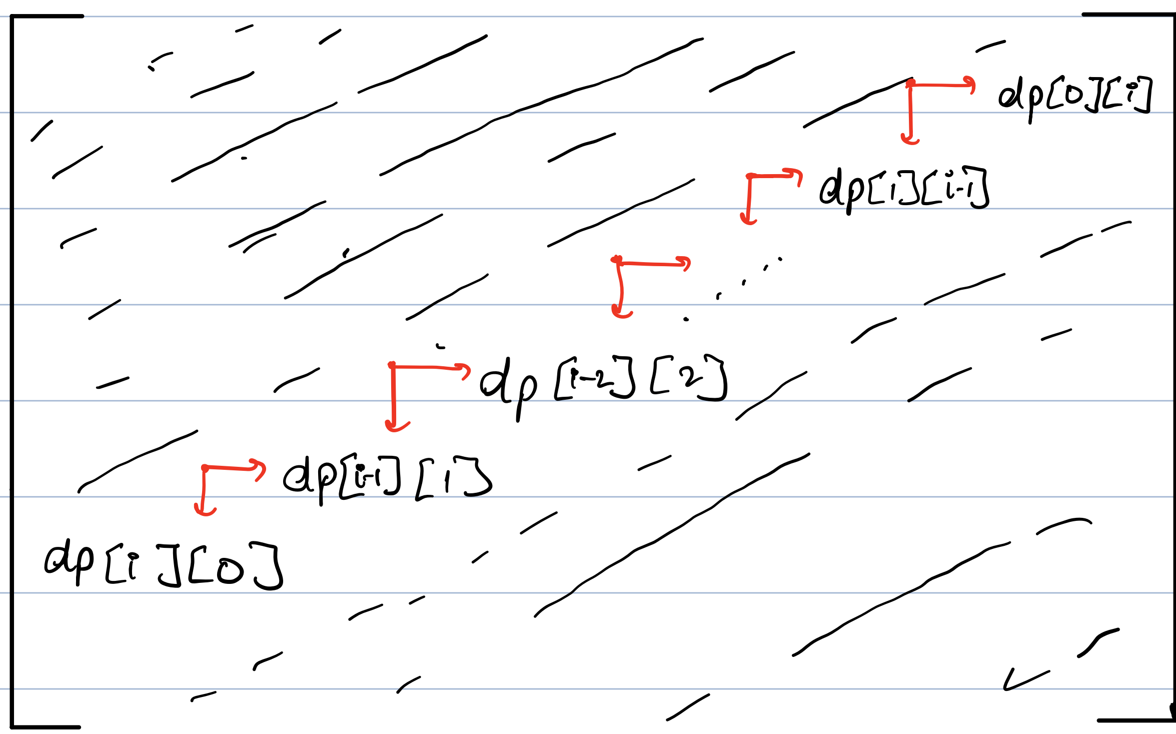
(b) Expectation:

For a given 't', the required expectation becomes,

$$E(X_1 + X_2 + \cdots X_t) = E(X_1) + E(X_2) + \cdots E(X_t)$$

(Using Linearity of Expectation)

Now we need to calculate expectation of X_i, to compute $E(X_i)$ let's look at the dp matrix we made earlier,



For each x_k we look at all elements $dp[i][j]$ s.t. $i+j=k$.

Note that they all lie along a diagonal.

So, we are required to calculate, (for x_k)

$1 \times$ Prob of kth round being won by Alice
$+ \ 0 \times$ Prob of kth round being a draw
$- \ 1 \times$ Prob of kth round being won by Bob

So, for each element, we calculate prob of reaching it from above (Alice wins) and from the left (Bob wins), and then calculate the required probability iteratively.

NOTE: $E(\sum_t x_i)$ comes out to be '0' for all values of t because probability of Alice & Bob winning is uniform s.t. they cancel out each other.

Variance : For a given T, we know

$E(\sum_T x_i) = 0$ from above, so

$Var(\sum_T x_i) = E((\sum_T x_i)^2)$

Let $Y = (\sum_T X_i)$

Lets take an example, $T = 10$

Now Possibilities after 10th round are:

$\times$ Not Possible

(10,0) ✓(9,1) ✓(8,2) ✓(7,3) ✓(6,4) ✓(5,5)

✓(4,6) ✓(3,7) ✓(2,8) ✓(1,9) (0,10)

$\times$ Not Possible

$\therefore$ The general representation can be

$(T-i, i)$ s.t $(1 \le i \le T-1)$ $i \in \mathbb{N}$

So, to reach this round, Alice needs $(T-i)$ wins and Bob needs i wins,

$$\therefore Y(T-i, i) = (X_1 + X_2 + \dots X_T)$$

$$= (\underbrace{T-i}_{\text{Alice won}} - \underbrace{i}_{\text{Bob won}})$$

$$Y(T-i, i) = (T-2i)$$

$$\therefore E(Y^2) = \sum_{1 \le y \le T-1} y^2 \times P(Y = y)$$

$$\Rightarrow E(Y^2) = (T-2i)^2 \times P(\underbrace{T-i, i}_{\text{Alice wins } T-i \text{ and Bob wins } i})$$

$$= (T-2)^2 \times P(T-1, 1) +$$

$$(T-4)^2 \times P(T-2, 2) +$$

$T-2(T-1)$

$$\vdots$$

$$(2-T)^2 \times P(1, T-1)$$

We calculate this value iteratively to get the answer for $\boxed{T = 19}$

2 (a)

As we know how Bob will play, we will apply Greedy algorithm in order to try to maximize points scored by Alice in each round.

<u>Case 1:</u> Bob drew the last game $\Rightarrow$ He opts balanced strategy

Alice \ Bob	Balanced (Bob)
Attack	$\left(\frac{7}{10}, 0, \frac{3}{10}\right)$
Balanced	$\left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right)$
Defense	$\left(\frac{1}{5}, \frac{1}{2}, \frac{3}{10}\right)$

The points scored on an average from each strategy of Alice will be :

(i) Attack : $\frac{7}{10} + \frac{1}{2}(0) + 0 \times \frac{3}{10}$

$= 7/10$

(ii) Balanced : $\frac{1}{3} + \frac{1}{2}\left(\frac{1}{3}\right) + 0\left(\frac{1}{3}\right)$

$= \frac{1}{2}$

(iii) Defense : $\frac{1}{5} + \frac{1}{2}\left(\frac{1}{2}\right) + 0\left(\frac{1}{3}\right)$

$= 9/20$

$\therefore$ ATTACK IS THE BEST OPTION

Case 2: Bob lost the last game $\Rightarrow$ He opts aggressive strategy

Alice	Aggressive (Bob)
Attack	$\left(\frac{n_B}{n_A+n_B}, 0, \frac{n_A}{n_A+n_B}\right)$
Balanced	$\left(\frac{3}{10}, 0, \frac{7}{10}\right)$
Defense	$\left(\frac{6}{11}, 0, \frac{5}{11}\right)$

The points scored on an average from each strategy of Alice will be :

(i) Attack : $\frac{n_B}{n_A+n_B} + \frac{1}{2}(0) + 0\left(\frac{n_A}{n_A+n_B}\right)$

(ii) Balanced : $\frac{3}{10} + \frac{1}{2}(0) + 0\left(\frac{7}{10}\right)$

$= 3/10$

(iii) Defense : $\frac{6}{11} + \frac{1}{2}(0) + 0\left(\frac{5}{11}\right)$

$= 6/11$

$\therefore$ ATTACK OR DEFENSE IS THE BEST OPTION

Attack will be better when $\frac{n_B}{n_A+n_B} > \frac{6}{11}$

otherwise defense is better

Case 3 : Bob won the last game $\Rightarrow$ He opts Defensive strategy

Alice		Defense (Bob)
	Attack	$\left(\frac{5}{11}, 0, \frac{6}{11}\right)$
	Balanced	$\left(\frac{3}{10}, \frac{1}{2}, \frac{1}{5}\right)$
	Defense	$\left(\frac{1}{10}, \frac{4}{5}, \frac{1}{10}\right)$

The points scored on an average from each strategy of Alice will be :

(i) Attack : $\frac{5}{11} + \frac{1}{2}(0) + 0\left(\frac{6}{11}\right)$
$= 5/11$

(ii) Balanced : $\frac{3}{10} + \frac{1}{4} + \frac{1}{5}(0)$
$= \frac{11}{20}$

(iii) Defense : $\frac{1}{10} + \frac{2}{5} + 0\left(\frac{1}{10}\right)$
$= \frac{1}{2}$

$\therefore$ BALANCED IS THE BEST OPTION

Thus, we choose the strategy of Alice accordingly and after performing Monte Carlo Simulations for $\sim 10^5$ times, we get ~ 57000 wins for Alice & ~ 43000 wins for Bob.

2(c)

For the above mentioned greedy strategy, to calculate the expected value for Alice to win 'T' rounds, we perform Monte-Carlo Simulations for ~ 10^5 times to calculate various values of number of rounds required for Alice to get T wins and then take their mean.

3(a)

For Alice to maximize the number of points gained in the current round, we again apply Greedy.

There is a uniform probability that Bob chooses to play Attack, Balanced or Defense.

Suppose in 3 Rounds, Bob play Attack, defense and Balanced once. We will calculate the expectation of total points Alice scores via any single strategy.

Bob

Alice	Attack	Balanced	Defence
Attack	$\left(\frac{n_B}{n_A+n_B}, 0, \frac{n_A}{n_A+n_B}\right)$	$\left(\frac{7}{10}, 0, \frac{3}{10}\right)$	$\left(\frac{5}{11}, 0, \frac{6}{11}\right)$
Balanced	$\left(\frac{3}{10}, 0, \frac{7}{10}\right)$	$\left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right)$	$\left(\frac{3}{10}, \frac{1}{2}, \frac{1}{5}\right)$
Defense	$\left(\frac{6}{11}, 0, \frac{5}{11}\right)$	$\left(\frac{1}{5}, \frac{1}{2}, \frac{3}{10}\right)$	$\left(\frac{1}{10}, \frac{4}{5}, \frac{1}{10}\right)$

Case 1: Alice attacks,

Average Points $= \frac{n_B}{n_A+n_B} + \frac{7}{10} + \frac{5}{11}$

Case 2: Alice plays balanced

Average Points $= \frac{3}{10} + \frac{1}{3} + \frac{1}{6} + \frac{3}{10} + \frac{1}{4} = 1.35$

Case 3: Alice defends,

Average Points $= \frac{6}{11} + \frac{1}{5} + \frac{1}{4} + \frac{1}{10} + \frac{2}{5} = 1.495$

As Defending is better than playing balanced, we only compare Attacking and Defending.

Attacking is better if $\boxed{\dfrac{n_B}{n_A + n_B} > \dfrac{15}{44}}$

otherwise Defending is better

So, Accordingly, I have chosen Attacking or Defending, depending on whichever is better. To test the correctness, we run Monte Carlo simulations for $\sim 10^5$ times and get roughly $(53-54) \times 1000$ wins for Alice and $(46-47) \times 1000$ wins for Bob.

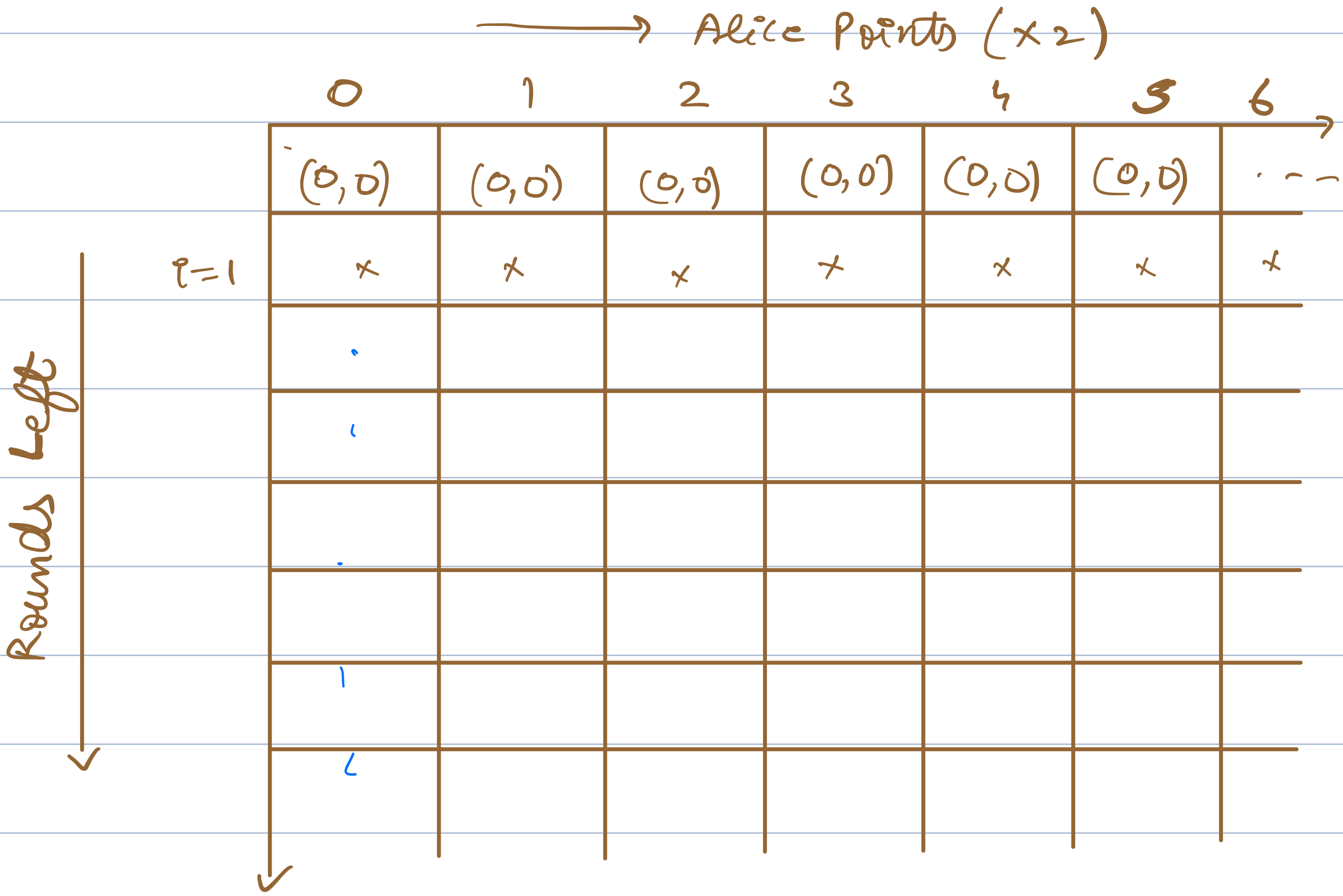
2 (b)

There does not exist any other strategy for Alice other than the greedy implemented in part (a) of question 2.

This can be verified as I have performed extensive testing trying various greedy strategies, for various initial conditions and also tested the code via applying dynamic programming (3-D) and none of the mentioned algorithms yielded better results.

③ (b)

In this part, we are applying dynamic programing where my dp array looks as follows:



We fill (i=1) row manually with greedy approach

And then we calulate dp[i][j] using

dp[i-1][j+2] → Alice wins

dp[i-1][j] → Alice loses

dp[i-1][j+1] → Draw

Our dp[i][j] is a tuple of 2 elements, where first is the strategy alice opts to maximize the expectation of it's points and second element is the maximum expectation before 'i' rounds are left

∴ We calculate the value of dp[total rounds][0] iteratively and report the corresponding answer.