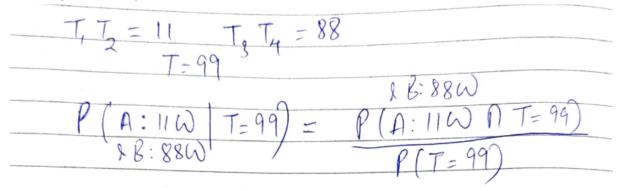
Assignment-1

Q.1

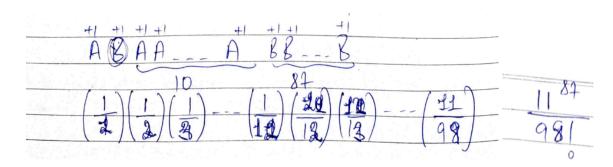
In this case, Alice wins 11 matches while Bob wins 88 matches. To calculate total probability, we have to sum all possible cases.

Initial case: Alice (A) wins first round and Bob (B) wins second round. Since both are attacking, probability of draw is 0.



1) B wins 1 time after A

For example, considering cases where a block of 10 consecutive wins by A is taken at once (AAAAAAAA),



Now, we take

1) fixed block of 9 consecutive wins (only) by A.

$${\sf S}$$
 = $\sum_{i=0}^{86} {(i+2)^9.2^{i+1}.11^{86-i}}$

2) fixed block of 8 consecutive wins (only) by A

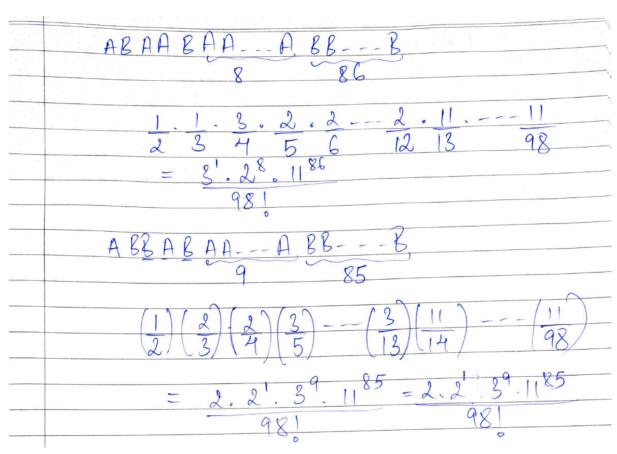
$$S = \sum_{i=0}^{86} 3^{i+1}.\,(i+2)^8.11^{86-i}$$

Therefore, we observe a pattern.

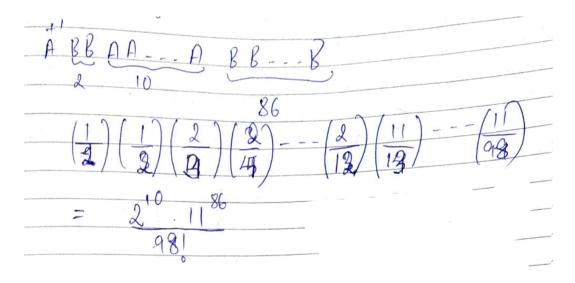
If we go till just 1 consecutive win by A.

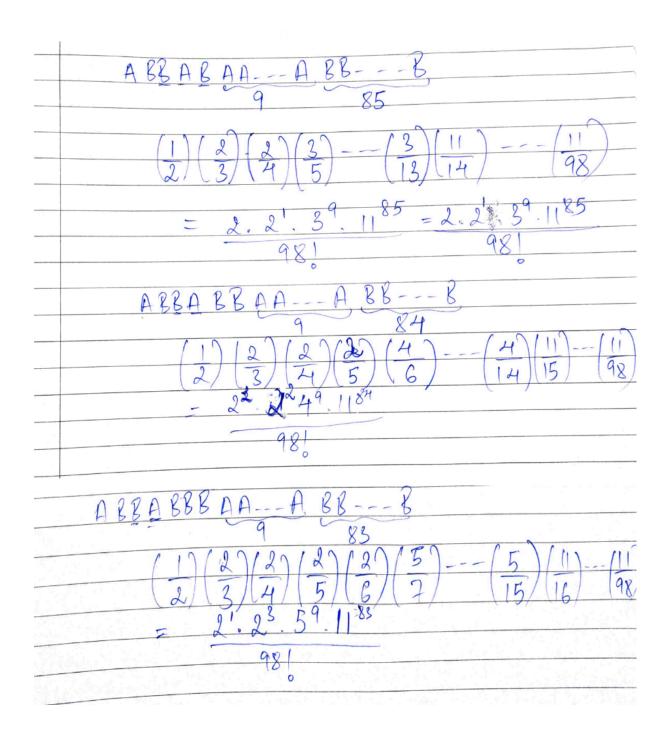
$$S = \sum_{j=2}^{10} \sum_{i=0}^{86} {(j)}^{i+1} (i+2)^{11-j}.11^{86-i}$$

By dividing S by 98!, we get probability of this specific case.

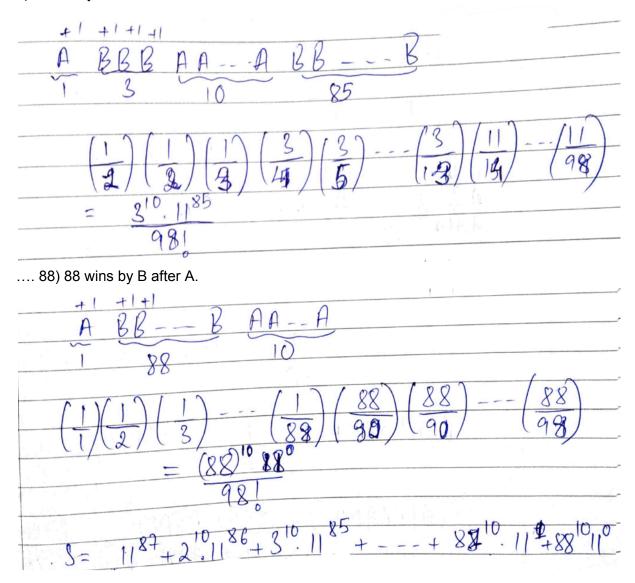


2) 2 wins by B after A Similarly, taking fixed blocks of 9, 8, 1 consecutive wins by A,





3) 3 wins by B after A



So, final probability would be S divided by 98!

The probability of each scenario can be generalised as i¹⁰ 11⁸⁸⁻ⁱ, where i signifies the number of matches won by B after A (second round onwards).

However, not all cases can be accounted by manual analysis.

Hence, we might simulate all ⁹⁹C₁₁ cases in the code. By computing all possible permutations and updating values of nA and nB for each case, the total probability may be calculated. However, this would take prohibitively long. Hence, we can simulate each round and compute approximate probability through Monte Carlo by iterating over 10⁴ possibilities.

1 b) We can sum conditional probability for each round, since

$$\begin{array}{c} x_i = 1 & A & \omega \\ x_i = 0 & D \\ \hline x_i = -1 & A \\ \hline E \begin{bmatrix} \frac{1}{2} & x_i \\ i = 1 \end{bmatrix} = \begin{array}{c} \frac{1}{2} & E & (x_i) = \frac{1}{2} & x_i P(x_i = x_i) \\ \hline i = 1 & \vdots = 1 \end{array}$$

$$\begin{array}{c} x_i = 1 & X_i & X_i = \frac{1}{2} & X_i & X_i & X_i = \frac{1}{2} & X_i & X$$

We assume that all events are independent. Hence,

$$P(A) = \sum_{i} P(A || B_i) P(B_i)$$

For example, at 6th round,

$$\frac{P(A|ABAAA) = P(A|ABAAA) = P(ABAAA|A)P(BAAAA)}{P(ABAAAA)} = P(A|BAAAA) = P(A|BAAAA)}$$

$$\frac{P(B_{J}|A) = P(A|B_{J})P(B_{J}) = P(B_{J}|A) = P(A|B_{J})}{P(B_{J})P(B_{J})} = P(A|B_{J})P(B_{J}) = P(A|B_{J})P(B_{J})}$$

$$\frac{P(B_{J}|A) = P(A|B_{J})P(B_{J}) = P(B_{J}|A) = P(A|B_{J})P(B_{J})}{P(B_{J})P(B_{J})} = P(A|B_{J})P(B_{J})$$

$$\frac{P(B_{J}|A) = P(A|B_{J})P(B_{J}) = P(B_{J}|A) = P(A|B_{J})P(B_{J})}{P(B_{J})P(B_{J})} = P(A|B_{J})P(B_{J})$$

$$\frac{P(B_{J}|A) = P(A|B_{J})P(B_{J}) = P(B_{J}|A) = P(A|B_{J})P(B_{J})$$

$$\frac{P(B_{J}|A) = P(A|B_{J})P(B_{J}) = P(B_{J}|A)$$

$$\frac{P(B_{J}|A) = P($$

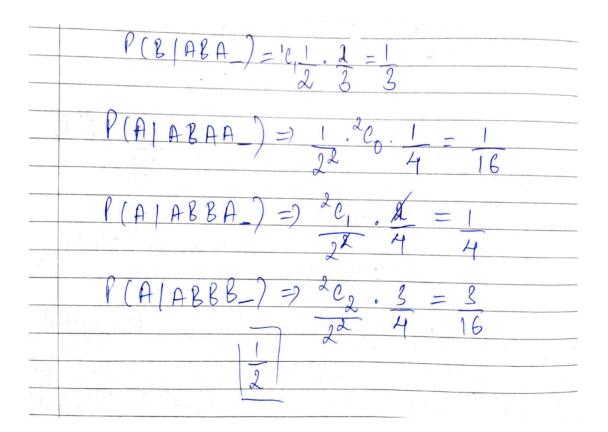
$$P(A|B_{J}) = P(A) P(A|B_{J}) = P(A) = \frac{1}{5}P(A|B_{1})P(B_{1})$$

$$= \frac{1}{5}P(A|B_{1}) P(B_{1})$$

$$= \frac{1}{5}P(A|B_{1})P(B_{1})$$

$$\frac{2^{4} - 2^{2}}{2^{4} - 2^{2}} = \frac{2^{4} - 2^{2}}{2^{5} - 4} + \frac{2^{4} - 2^{4} - 4}{2^{4} - 2^{4} - 4}$$

$$\frac{2^{6} (ABAAAA)^{2} (A ABAAAA)^{2} = \frac{1 \cdot ^{4} C_{1} \cdot ^{4} \cdot ^{4} \cdot ^{4} \cdot ^{2} \cdot ^{4} \cdot ^{4} \cdot ^{2} \cdot ^{4} \cdot ^{4}$$



Since all events are independent,

$$Cov(X_i, X_j) = 0$$

$$Var\left(\sum_{i}X_{i}\right)=\sum_{i}Var\left(X_{i}\right)=\sum_{i}\left(E\left(X_{i}^{2}\right)-E^{2}\left(X_{i}\right)\right)$$
 Hence,

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

Bob

We defined payoff_matrix:

prob={'Attack':([nA/(nA+nB), 0, nB/(nA+nB)], [0.7, 0, 0.3], [5/11, 0, 6/11]),

'Balanced': ([0.3, 0, 0.7], [1/3, 1/3, 1/3], [0.3, 0.5, 0.2]),

'Defensive': ([6/11, 0, 5/11], [0.2, 0.5, 0.3], [0.1, 0.8, 0.1])}

We could also define it as expectation value for Alice.

If Bob won the previous round, then he chooses to play defensively. If the previous round resulted in a draw, then he plays balanced. If he lost the previous round, then he plays aggressively.

Therefore, at every step, Alice may choose the step that maximises either expectation value, or the probability of winning.

- b) For non-greedy strategy, many mixed strategies can be chosen, for example Attack: 0.4, Balanced: 0.25, and Defence: 0.35, since Attack usually has highest probability of winning than Defence, followed by Balanced. However, we noticed that greedy strategy usually works more effectively.
- c) To estimate E[tau], we defined function simulate_round and iterated over it 10^5 times, where the expectation is incremented by 1 while the number of wins by Alice is less than 88.
- 3. a) To maximise her points, Alice may choose the strategy that maximises the expectation value i.e. 1*v[Bob().play_move()][0]+0.5*v[Bob().play_move()][1].
- b) For deciding optimal strategy, Alice can analyse the potential future expectation value and the current expectation value to select the one strategy that will yield the maximum points.

Based on the random choice that Bob makes, points can be incremented and the expectation value then analysed.