

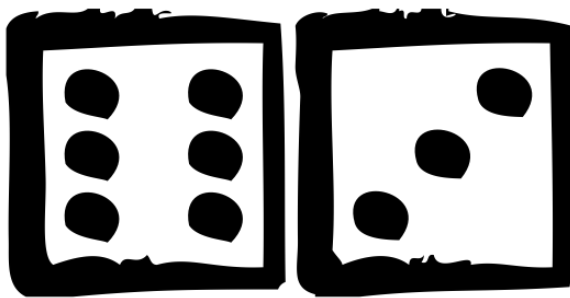
Problem Statement: The Doomed Dice Challenge

The below problems must be solved & implemented in Python/Java/Ruby/C++/Go

You are given two six-sided dice, Die A and Die B, each with faces numbered from 1 to 6.

You can only roll both the dice together & your turn is guided by the obtained sum.

Example: Die A = 6, Die B = 3. Sum = $6 + 3 = 9$



You may represent Dice as an Array or Array-like structure.

Die A = [1, 2, 3, 4, 5, 6] where the indices represent the 6 faces of the die & the value on each face.

Part-A (15-20 Minutes):

1. How many total combinations are possible? Show the math along with the code!
2. Calculate and display the distribution of all possible combinations that can be obtained when rolling both Die A and Die B together. Show the math along with the code!

Hint: A 6 x 6 Matrix.

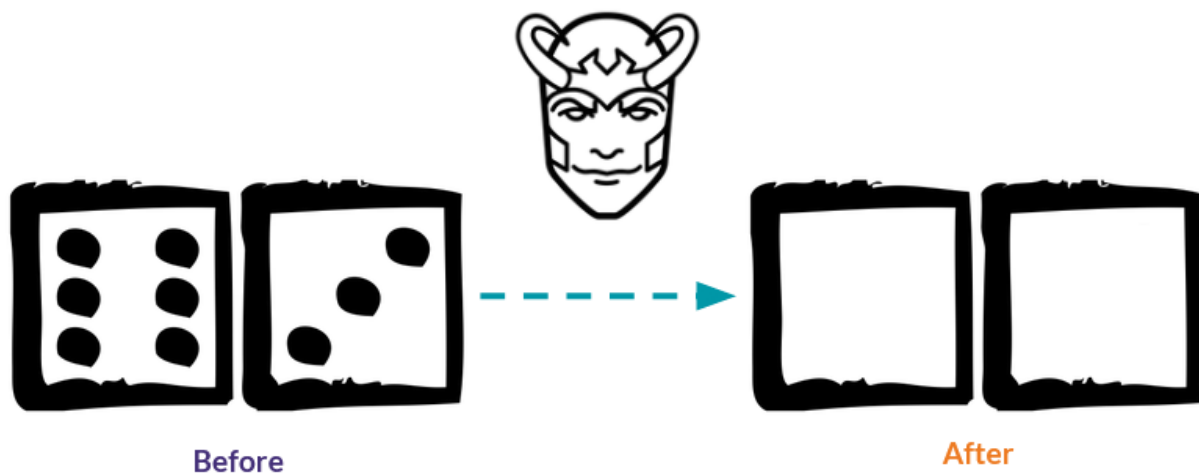
3. Calculate the Probability of all Possible Sums occurring among the number of combinations from (2).

Example: $P(\text{Sum} = 2) = 1/X$ as there is only one combination possible to obtain Sum = 2. Die A = Die B = 1.

Part-B (25-30 Minutes):

Now comes the real challenge. You were happily spending a lazy afternoon playing your board game with your dice when suddenly the mischievous Norse God [Loki](#) (You love Thor too much & Loki didn't like that much) appeared.

Loki dooms your dice for his fun removing all the "Spots" off the dice.



No problem! You have the tools to re-attach the "Spots" back on the Dice. However, Loki has doomed your dice with the following conditions:

- Die A cannot have more than 4 Spots on a face.
- Die A may have multiple faces with the same number of spots.
- Die B can have as many spots on a face as necessary i.e. even more than 6.

But in order to play your game, the probability of obtaining the Sums must remain the same!

So if you could only roll $P(\text{Sum} = 2) = 1/X$, the new dice must have the spots reattached such that those probabilities are not changed.

Input:

- Die_A = [1, 2, 3, 4, 5, 6] & Die B = Die_A = [1, 2, 3, 4, 5, 6]

Output:

- A Transform Function `undoom_dice` that takes (Die_A, Die_B) as input & outputs `New_Die_A = [?, ?, ?, ?, ?, ?]`, `New_Die_B = [?, ?, ?, ?, ?, ?]` where,
- No `New_Die A[x] > 4`