1.1 Q: 9 boys and 7 girls sit in a circle, what's the expectation of the number of boy-girl neighbors?
A:
A boy can have either 0, 1 or 2 girls as his neighbors and the corresponding probabilities are:
0 girls as neighbors: 8/15 * 7/14 = 4/15
2 girls as neighbors: 7/15 * 6/14 = 3/15
1 girl and 1 boy as neighbor: 1 – 4/15 – 3/15 = 8/15
The expected number of girls next to any boy is: 0 * 4/15 + 1 * 8/15 + 2 * 3/15 = 14/15.
Thus, the expected number of girls for all 9 boys are: 9 * 14/15 = 42/5.

1.2 Q: What's the difference between uncorrelated and independent variables. Which statement is stronger? Can you give an example?
A: Independence implies uncorrelation, but uncorrelation doesn't imply independence. So, independence is stronger. For example, for $X \sim U(-1,1)$, X^2 and X are uncorrelated, but they are not independent.

1.3 Q: You are allowed to toss a dice for up to three times. You can decide to stop after each toss, and you will get the same value of dollars as the number shown on your last toss. What is the value of the game?
A: This can be solved reversely.

$$E_3 = \frac{1}{6}\sum_{i=1}^{6} i = \frac{7}{2}$$
$$E_2 = \frac{1}{6}(4+5+6) + \frac{1}{2}E_3 = \frac{17}{4}$$
$$E_1 = \frac{1}{6}(5+6) + \frac{2}{3}E_2 = \frac{14}{3}$$

1.4 Q: If a rod with length L is cut randomly into N pieces, what is the distribution of the longest piece?
A:
https://math.stackexchange.com/questions/14190/average-length-of-the-longest-segment

1.5 Q: Given three random variables X, Y and Z, $\rho(X,Y) = \rho(Y,Z) = r$, $\rho(X,Z) = 0$, calculate the range of r.
A: The correlation matrix has to be positive semi-definite, with below:

$$corr = \begin{pmatrix} 1, & r, & 0 \\ r, & 1, & r \\ 0, & r, & 1 \end{pmatrix}$$

Thus, we have

$$1 - r^2 \geq 0$$
$$1 - 2r^2 \geq 0$$

So we can conclude $r \in [-\sqrt{2}/2, \sqrt{2}/2]$.

1.6 Q: A drunk man sits 1 step to the left door and 99 steps to the right door. He randomly walks left and right. As soon as he reaches an open door, he will go back home. What is the expected

number of steps he takes to go home: a) if both the right and left doors are open; b) if the left door is locked and the right door is open.
A: Martingale??

1.7 Q: What is ridge regression?
A: The ridge regression coefficient estimates $\hat{\beta}^R$ are the values that minimize

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_j\right)^2 + \lambda\sum_{j=1}^{p}\beta_j^2 = RSS + \lambda\sum_{j=1}^{p}\beta_j^2$$

where $\lambda \geq 0$ is a tuning parameter. Solution to ridge regression:

$$\beta = (X^T X + \lambda I)^{-1} X^T Y$$

1.8 Q: What is lasso regression?
A: The lasso regression coefficient estimates $\hat{\beta}^L$ are the values that minimize

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_j\right)^2 + \lambda\sum_{j=1}^{p}|\beta_j| = RSS + \lambda\sum_{j=1}^{p}|\beta_j|$$

where $\lambda \geq 0$ is a tuning parameter. Computationally, solving ridge is faster than solving lasso, since ridge has a closed form solution, while lasso has to be solved by a solver.

1.9 Q: Play a coin toss game, you win 2^i dollars if the i-th toss is head. How much would be pay for this game? (Spoiler: the expected payoff is infinity, but are you really willing to pay 1 million dollars for this game?)
A: The expected payoff of this game:

$$E[X] = 2 \times \frac{1}{2} + 2^2 \times \frac{1}{2^2} + \cdots = \infty$$

https://en.wikipedia.org/wiki/St._Petersburg_paradox

1.11 Q: Implement a thread safe singleton.
A:
https://gist.github.com/werediver/4396488

1.12 Q: Say you have an array for which the i-th element is the price of a given stock on day i. Design an algorithm to find the maximum profit. You may complete at most k transactions. Note: You may not engage in multiple transactions at the same time (i.e. you must sell the stock before you buy again).
A: LeetCode 188

```python
class Solution:
    def maxProfit(self, k: int, prices: List[int]) -> int:
        if prices is None or len(prices) == 0:
            return 0

        num_days = len(prices)
        num_states = 2 * k + 1

        if num_states >= num_days:
            result = 0
            for day in range(1, num_days):
                result += max(prices[day] - prices[day - 1], 0)
            return result

        f = [[0 for _ in range(num_states + 1)] for _ in range(num_days + 1)]
        for state in range(2, num_states + 1):
            f[0][state] = -math.inf

        for day in range(1, num_days + 1):
            # state 1, 3, 5: f[price][state] = max(f[price - 1][state], f[price - 1][state - 1] + prices[price - 1] - prices[price - 2])
            for state in range(1, num_states + 1, 2):
                f[day][state] = f[day - 1][state]
                if state > 1 and day > 1:
                    f[day][state] = max(f[day - 1][state], f[day - 1][state - 1] + prices[day - 1] - prices[day - 2])

            # states 2, 4: f[price][state] = max(f[price - 1][state - 1], f[price - 1][state] + prices[price - 1] - prices[price - 2])
            for state in range(2, num_states + 1, 2):
                f[day][state] = f[day - 1][state - 1]
                if day > 1:
                    f[day][state] = max(f[day - 1][state - 1], f[day - 1][state] + prices[day - 1] - prices[day - 2])

        result = -math.inf
        for state in range(1, num_states + 1, 2):
            result = max(result, f[num_days][state])

        return result
```

1.13 Q: A vector of integers, every element appears twice except one. Find the integer which only appears once.
A: LeetCode 136

```python
class Solution:
    def singleNumber(self, nums: List[int]) -> int:
        a = 0
        for i in nums:
            a ^= i
        return a
```

1.15 Q: How to generate random numbers? How to evaluate the quality of a random number generator?
A: We usually use multiplication, addition and module arithmetic operations to generate pseudo-random numbers. Starting from a seed X_0 and generate random numbers by

$$X_{n+1} = (aX_n + b) \bmod m$$

which will generate random numbers from 0 to $m - 1$.

1.16 Q: How to simulate dices with a coin? How to maximize the efficiency?
A: To toss a coin i times, we have in total 2^i possible outcomes. To roll a dice j times, we have in total 6^j possible outcomes. So to increase the efficiency, it is equivalent to find

$$i, j = \underset{2^i \geq 6^j}{\operatorname{argmin}} \frac{2^i - 6^j}{2^i}$$

However, this value has a limit 0 when j goes to infinity, so there is no minimizer. One of the good choices of (i, j) is (3,1).

1.17 Q: Given two Robots, one of which is currently somewhere at negative axis and the other one positive axis. The two Robots can only take the following commands:
Go left: go left by 1.

Go right: go right by 1.
If at-zero(): return true if the Robot is at 0.
Write a program with the above commands and *ONLY* while() or if-else to make the two Robots meet. The meeting location is not important. Also, the two robots would execute this program at the same time. E.g. if you make go left in your program, the two robots will both go left.
A:

```
while not at zero:
        Go left
        Go right
        Go left

while True:
        Go left
```