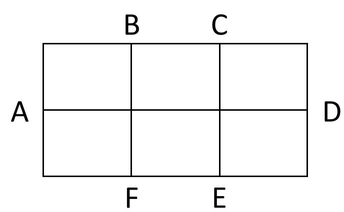
1.1 Q: What do you know about χ^2 test?
A: A chi-squared test, is any statistical hypothesis test where the sampling distribution of the test statistic is a chi-squared distribution when the null hypothesis is true. A chi-squared distribution can be constructed as sum of squared iid normal variables.

It is used to determine whether there is a significant difference between the expected frequencies and the observed frequencies in one or more categories. Assume there are k categories in total, and the probability from category i is p_i if null hypothesis is true. We observe n samples, and there are x_i observations in category i. Then,

$$X^2 = \sum_{i=1}^{k} \frac{(x_i - np_i)^2}{np_i} \xrightarrow{d} \chi^2$$

with degree of freedom of $k - 1$.

1.2 Q: Given a 2 by 3 grid (which has 6 blocks and 17 edges), what's the shortest route to visit all edges (assuming edge length is 1).
A: An Euler path in a graph is a walk through the graph which uses every edge exactly once. A graph has a Euler path if and only if there are at most two vertices with odd degrees.



A, B, C, D, E and F are all vertices with odd degrees. In the shortest path, at least 4 points have to be made even points by extra edges, thus, the shortest edges adding to the graph are BC and EF. Now the only two vertices with odd degrees are A and D. The Euler path would be A → D with length of 17 edges + length (BC) + length (EF) = 19.

1.3 Q: X, Y are iid $N(0,1)$, calculate P(X|X + Y > 0), try not use density function of joint distribution.
A:

$$\begin{aligned} P(X|X+Y>0) &= \frac{P(X=x, X+Y>0)}{P(X+Y>0)} \\ &= 2P(X=x, X+Y>0) \\ &= 2P(X)P(Y>-X|X) \\ &= \frac{2}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2} \int_{-x}^{+\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}u^2} du \\ &= \sqrt{\frac{2}{\pi}} e^{-\frac{1}{2}x^2} (1 - \Phi(x)) \end{aligned}$$

where $\Phi(x) = P(Z \leq x)$ and $Z \sim N(0,1)$.

1.4 Q: You have a six-sided dice, you can keep rolling the dice and you get the dollars equal to the amount of the sum. However, if at some point, the sum is a square number, you must stop and will get zero dollars. (1) If at some point, your sum is 35, should you stop or keep rolling? (2) If you choose to continue in the previous question and this is your strategy: you will keep

rolling until you reach 43, what is the most probable amount of dollars you would win when you stop? (3) Is there a best strategy for this game, any number that you should stop?
A: If the current sum is 35, the next square number is 36 followed by 49. Thus, if we pass 36, we could at least reach 43 before worrying about losing all money again.
(1) $E(payoff|stop\ if \geq\ 43) \geq 5/6 \times 43 = 35.83 > 35$, so you should keep rolling.
(2) The sum value before reaching 43 could be 37 to 42, and we use p_{37} to p_{42} to denote the density for each sum value. The stop value could be 43 to 48, and we use p_{43} to p_{48} to denote the corresponding density.

$$p_{43} = \frac{1}{6}(p_{37} + p_{38} + p_{39} + p_{40} + p_{41} + p_{42})$$
$$p_{44} = \frac{1}{6}(p_{38} + p_{39} + p_{40} + p_{41} + p_{42})$$
$$p_{45} = \frac{1}{6}(p_{39} + p_{40} + p_{41} + p_{42})$$
$$p_{46} = \frac{1}{6}(p_{40} + p_{41} + p_{42})$$
$$p_{47} = \frac{1}{6}(p_{41} + p_{42})$$
$$p_{48} = \frac{1}{6}(p_{42})$$

(3) There exists at least one state that we need to stop. Otherwise, we will never stop rolling the dice, in which case the expected payoff would be 0 since the probability of hitting some square number is 1.

1.5 Q: Given a stick, if randomly cut into 3 pieces, what's the average size of the smallest, of the middle-size, and of the largest sizes?
A: Given a stick, and randomly cut into n pieces, then average length of k^{th} longest segment $S_{(k)}$ is

$$E\left[S_{(k)}\right] = \frac{1}{n}\sum_{j=1}^{k}\frac{1}{n-j+1}$$

When $n = 2$, we have $1/4$ and $3/4$.
When $n = 3$, we have $1/9, 5/18,$ and $11/18$.
source: https://math.stackexchange.com/questions/13959/if-a-1-meter-rope-is-cut-at-two-uniformly-randomly-chosen-points-what-is-the-av

1.6 Q: At a party, N people throw their hats (all hats are different) into the center of room. The hats are mixed up and each people randomly selects one. Let Y be the number of people who select their own hats. Now ask (1) what is the expectation of Y? (2) what is the variance of Y? Now, the picking hats game rule is extended. For each hat pick round, the people choosing their own hats quit the game, while others (those picked wrong hats) put their selected hats back in the center of the room, mix them up, and then reselect. Also, suppose that this game continues until each individual has his own hat. Suppose N individuals initially join the game, let $R(N)$ be the number of rounds that are run and $S(N)$ be the total number of selections made by these N individuals, $(N > 1)$. (3) Find the expectation of $R(N)$. (4) Find the expectation of $S(N)$. (5) Find the expected number of false selections made by one of the N people.
A:

(1) We use I_k to denote the k^{th} person takes his own hat. Then we have

$$Y = \sum_{k=1}^{N} I_k$$

and we have

$$E[Y] = E\left[\sum_{k=1}^{N} I_k\right] = \sum_{k=1}^{N} E[I_k] = \sum_{k=1}^{N} \frac{1}{N} = 1$$

(2) According to above, we have

$$Y^2 = \left(\sum_{k=1}^{N} I_k\right)^2 = \sum_{k=1}^{N} I_k^2 + 2\sum_{i<j} I_i I_j$$

By $E[I_k^2] = 1/N$, and $E[I_i I_j] = 1/N(N-1)$, we know

$$E[Y^2] = N \times \frac{1}{N} + 2 \times \frac{N(N-1)}{2} \times \frac{1}{N(N-1)} = 1 + 1 = 2$$

which implies

$$Var(Y) = E[Y^2] - (E[Y])^2 = 2 - 1 = 1$$

(3, 4, 5)

1.7 Q: Consider linear regression of Y on features X_1, X_2. Model 1 - (Y, X_1), $R^2 = 0.1$; model 2 - (Y, X_2), $R^2 = 0.2$; model 3 - (Y, X_1, X_2), calculate the range of R^2 of model 3.
A: Without loss of generality, we assume $Y^T Y = 1, X_1^T X_1 = 1,$ and $X_2^T X_2 = 1$. We denote $\rho_1 = corr(X_1, Y)$, $\rho_2 = corr(X_2, Y)$ and $\rho = corr(X_1, X_2)$. R^2 of model 3:

$$R^2 = \frac{\rho_1^2 + \rho_2^2 - 2\rho\rho_1\rho_2}{1 - \rho^2}$$

Subject to the constraint $\rho_1^2 + \rho_2^2 - 2\rho\rho_1\rho_2 \le 1 - \rho^2$. By R^2 of Model 1 and 2, we have $\rho_1 = \sqrt{0.1}, \rho_2 = \sqrt{0.2}$. So by calculating the derivative with respect to ρ, we know the maximum and minimum of R^2 of Model 3 can be achieved at

$$\rho_{max}^* = \rho_1\rho_2 - \sqrt{(1-\rho_1^2)(1-\rho_2^2)}, \quad \rho_{min}^* = \frac{\rho_1}{\rho_2}$$

We conclude $R^2 \in [0.2, 1]$.

1.8 Q: Given a function for a fair coin, write a function for a biased coin that returns heads with probability $1/n$ (n is a param).
A: Sampling algorithm:
1. Find the smallest k, such that $2^k > n$.
2. Toss the coin for k times, and get a sequence of 0 and 1 as $a_1, a_2, \ldots, a_k$. We could form it as a binary number $v = a_1 a_2 \ldots a_k$, s.t. $0 \le v \le 2^k - 1$.
3. Return

$$toss = \begin{cases} Head, & \text{if v} = 0 \\ Tail, & \text{if v} \in [1, \text{n} - 1] \\ Reject\ and\ toss\ again, & \text{otherwise} \end{cases}$$

1.9 Q: 10 islands with 9 bridges. The bridges are either strong or weak (half half). Weak bridge falls if stepped on and the man is drifted to the 1st island, then all the bridges are miraculously fixed. To arrive the 10th island, how many bridges on average the man has to cross?
A: Geometric distribution?

1.11 Q: How do implement "delete" operation in a single-linked list?
A: LeetCode 237

```python
class Solution:
    def deleteNode(self, node):
        """
        :type node: ListNode
        :rtype: void Do not return anything, modify node in-place instead.
        """
        if node is None or node.next is None:
            return

        node.val = node.next.val
        node.next = node.next.next

        return
```

1.15 Q: Given a string, return the longest palindrome subsequence.
A: LeetCode 516

```python
class Solution:
    def longestPalindromeSubseq(self, s: str) -> int:
        if s is None or len(s) == 0:
            return 0

        n = len(s)
        f = [[0 for _ in range(n)] for _ in range(n)]

        # length == 1
        for i in range(n):
            f[i][i] = 1

        # length == 2
        for i in range(n - 1):
            f[i][i + 1] = f[i][i]
            if s[i] == s[i + 1]:
                f[i][i + 1] = 2

        # length >= 3
        for l in range(3, n + 1):
            # start position
            for start in range(n - l + 1):
                # end position
                end = start + l - 1
                f[start][end] = max(f[start][end - 1], f[start + 1][end])
                if s[start] == s[end]:
                    f[start][end] = max(f[start][end], f[start + 1][end - 1] + 2)

        return f[0][n - 1]
```

1.16 Q: How to inverse a string of sentence (without reverse the word)?
A: LeetCode 151

```python
class Solution:
    def reverseWords(self, s: str) -> str:
        return " ".join(s.strip().split()[::-1])
```

1.17 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 two transactions.
A: LeetCode 123

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

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

            # states 2, 4: f[day][state] = max(f[day - 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][state], 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
```

1.18 Q: The book problem: There is a group of N ($2 \leq N \leq 1000$) people which are numbered 1 through N, and every one of them has not less than $(N + 1)/2$ friends. A man with number 1 has the book, which others want to read. Write a program from the friend to the friend, and at last, has come back to the owner.
Note: if A is a friend of B then B is a friend of A. Input: First line of input contains a list of fri
A: DFS?

Note: Question 10, 12, 13, 14 are all C++ specific questions, which don't apply to me since I only use Python.