# Qishi Quiz 2 Solution

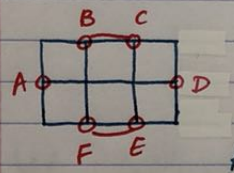
Liao Zhu

Qishi Quiz 2 Solution — Liao Zhu 朱樑

1. Chi-square test refers to any stats test with test statistic following a chi-square distribution under the null hypothesis. There are three main usages of Chi-square test:
   ① Test whether a sequence of observations comes from a theoretical distribution.
   ② Test whether two sequences of observations come from the same distribution.
   ③ Test independence of two variables.

   Test statistic: $$X_n^2 = \sum_{1 \le i \le n} \left( \frac{\text{observed}_i - \text{expected}_i}{\text{expected}_i} \right)^2 \xrightarrow{d} \chi_p^2$$

   For ~~two-way~~ a r row, c column contigency table, $p = (r-1)(c-1)$. □

2. 


   A connected graph is one-pass ⟺ There are 0 or 2 odd points.

   A, B, C, D, E, F are all odd points in the graph, so in the shortest path at least 4 points should be made even points by repeating edges. The shortest ~~paths~~ edges adding to the graph to make 4 points even is BC, FE. Now only two odd points A, D. So ~~add edges~~ the graph can be one-pass from A → D. The shortest route is 17 + length(BC) + length(EF) = 19. □

3. $X, Y \overset{iid}{\sim} N(0,1)$ so $X+Y \sim N(0,2)$

$$P(X \mid X+Y>0) = \frac{P(X, X+Y>0)}{P(X+Y>0)}$$

$$= \frac{P(Y>-X \mid X)\,P(X)}{P(X+Y>0)} \underset{X+Y\sim N(0,2)}{=\!=} 2\cdot P(Y>-X \mid X)\,P(X)$$

$$= 2\cdot \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{x^2}{2}\right\} \cdot \int_{-x}^{+\infty} \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{u^2}{2}\right\} du$$

$$= \frac{1}{\pi} e^{-\frac{x^2}{2}} \int_{-x}^{+\infty} e^{-\frac{u^2}{2}} du$$

$$= \sqrt{\frac{2}{\pi}}\, e^{-\frac{x^2}{2}} \left(1-\Phi(x)\right) \quad \text{where } \Phi(x) = P(Z \le x) \text{ where } Z \sim N(0,1)$$

□

4. (2) Define $P_i = P(S_n \text{ ever touched } i \text{ and } i \text{ is not a square number})$

$P_i = 0$ $(i \le 34$ or $i = 36)$ $\quad P_j = \sum_{i=1}^{6} \frac{1}{6} P_{j-i}$ $\quad (j \le 42)$

| $i$ | 35 | 36 | 37 | 38 | 39 | 40 | 41 |
|---|---|---|---|---|---|---|---|
| $P_i$ | 1 | 0 | $\frac{1}{6}$ | $\frac{1}{6}+\frac{1}{36}$ | $\frac{1}{6}+\frac{1}{6^2}+\frac{1}{6^3}$ | $\frac{1}{6}+\frac{1}{6^2}+\frac{1}{6^3}+\frac{1}{6^4}$ | ..... |

$Q_i = P(\text{stop at } i)$

$$Q_{43} = \sum_{i=1}^{6} \frac{1}{6} P_{43-i} > \sum_{i=1}^{5} \frac{1}{6} P_{43-i} = Q_{44} > \frac{1}{6} \sum_{i=1}^{4} P_{43-i} = Q_{45}$$

$$> \cdots\cdots > \frac{1}{6} P_{42} = Q_{48}$$

So the most probable amount you stop at is 43.

(1) $E(\text{profit} \mid \text{stop if reach 43, start from 35}) \ge \frac{5}{6} \times 43 = 35 + \frac{5}{6} > 35$

Page 2

Therefore, should not stop at 35.

(3). It can be shown that such number exist that you need to stop if you hit this. The intuition is for large numbers $P(\text{touch } x^2) \approx \frac{1}{6}$, so
$P(\text{gain } 0 \text{ after } n^2) \approx 1-(\frac{5}{6})^n \longrightarrow 1 \ (n \to \infty)$.

Consider a type of strategy: stop if the sum $\geq (x+1)^2-6$
Then $E(\text{gain} \mid x^2-1) \geq x^2-1$

$$\frac{5}{6}\left[(x+1)^2-1\right] \geq E(\text{gain} \mid x^2-1) \geq x^2-1$$

$$\Rightarrow \quad 5-\sqrt{19} < x < 5+\sqrt{19} < 10$$

Pick $x=9$. so $(x+1)^2-6 = 94$. Stop if sum $\geq 94$. □

5. Consider $(1, 0, 0)$
$(\frac{1}{2}, \frac{1}{2}, 0)$
$(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$

↓ Take average

$\frac{11}{18}, \frac{5}{18}, \frac{1}{9}$

The average size of the smallest one is $\frac{1}{9}$,
the average size of the largest one is $\frac{11}{18}$.
More generally, the $k^{th}$ mean of $n$ pieces is

$$\sum_{i=n-k+1}^{n} \frac{1}{i}$$

Still don't understand the proof. Explaination website included in the other document. □

Page 3

6. (1). $EY = E\left(\sum_{i=1}^{N} I_i\right) = \sum_{i=1}^{N} E(I_i) = \sum_{i=1}^{N} \frac{1}{N} = 1$

(2). $\mathrm{Var}Y = \mathrm{Var}\left(\sum_{i=1}^{N} I_i\right) = E\left[\left(\sum_{i=1}^{N} I_i\right)^2\right] - \left[E\left(\sum_{i=1}^{N} I_i\right)\right]^2$

$= \sum_{i=1}^{N} E(I_i) + 2 \sum_{1 \le i < j \le N} E(I_i I_j) - 1^2$

$= 1 - 1 + 2 \sum_{1 \le i < j \le N} \frac{1}{N(N-1)} = 2 \times \frac{N(N-1)}{2} \times \frac{1}{N(N-1)} = 1$

(3).

7. WLOG, let $Y'Y = X_1'X_1 = X_2'X_2 = 1$, $\bar{Y} = \bar{X_1} = \bar{X_2} = 0$

Denote $\rho_1 = X_1'Y$, $\rho_2 = X_2'Y$, $\rho = X_1'X_2$

So $\rho_1^2 = R_1^2 = 0.1$, $\rho_2^2 = R_2^2 = 0.2$, $\rho_1\rho_2 = 0.1\sqrt{2} = \frac{\sqrt{2}}{10}$

$$\det\left[\mathrm{Cov}(X_1, X_2, Y)\right] = \det\begin{pmatrix} 1 & \rho & \rho_1 \\ \rho & 1 & \rho_2 \\ \rho_1 & \rho_2 & 1 \end{pmatrix} \geq 0 \Rightarrow \rho \in \rho_1\rho_2 \pm \sqrt{\rho_1^2\rho_2^2 + (1-\rho_1^2-\rho_2^2)}$$

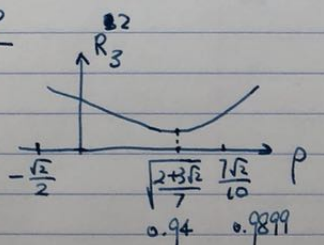
$$\in \left[-\frac{\sqrt{2}}{2}, \frac{7}{10}\sqrt{2}\right]$$

$$R_3^2 = 1 - Y'\left[I - X(X'X)^{-1}X'\right]Y \quad \text{where } X = (X_1, X_2)$$

$$= 1 - Y'Y + Y'X \cdot \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}^{-1} \cdot \left(\frac{1}{1-\rho^2}\right) X'Y$$

$$= \frac{1}{1-\rho^2}\left[\rho_1^2 + \rho_2^2 - 2\rho_1\rho_2\rho\right] = \frac{3-2\sqrt{2}\rho}{10-10\rho^2}$$

$$\frac{dR_3^2}{d\rho} = \frac{20(3-\sqrt{2})}{(10-10\rho^2)^2}\left[\rho^2 - \frac{2+3\sqrt{2}}{7}\right]$$




$$\begin{cases} R_3^2(\rho = -\frac{\sqrt{2}}{2}) = 1 \\ R_3^2(\rho = \sqrt{\frac{2+3\sqrt{2}}{7}}) \doteq 0.304 \end{cases} \Rightarrow R_3^2 \in [0.304, 1]$$

□

8. Define $g(T) = \begin{cases} 1 & \text{if } T = \text{Head} \\ 0 & \text{if } T = \text{Tail} \end{cases}$

$$g_k(T_1, T_2, \cdots, T_k) = (g(T_1), g(T_2), \cdots, g(T_k))_2$$

where $(\cdot)_2$ represents binary number.

For $\forall n \in \mathbb{N}^*$, define $k_n = \lceil \log_2 n \rceil$

$$f(T_1, T_2, \cdots, T_{k_n}) = \begin{cases} \text{Head} & \text{if } g_k(T_1, \cdots T_{k_n}) = 1 \\ \text{Tail} & \text{if } g_k(T_1, \cdots T_{k_n}) \in [2, n] \\ \text{Reject and toss again} & \text{otherwise.} \end{cases}$$

Page 5

The rejection rate $\leq \frac{2^{\lceil \log_2 n \rceil} - n}{2^{\lceil \log_2 n \rceil}} < \frac{1}{2}$. □

9. Define $f_i$ = The average of the number of bridges crossed till the first time he is on $i^{th}$ island.

$f_1 = 0$

$f_{i+1} = f_i + E\left[(X_i - 1)(i-1) + 1\right]$ where $X_i$ = Count of fix until (including initial) becomes strong

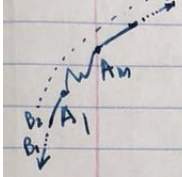
then $X_i \sim Geo(\frac{1}{2})$, $EX_i = 2$

So $f_{i+1} = f_i + (EX_i - 1)(i-1) + 1 = f_i + i$ so $f_n = \frac{(n-1)n}{2}$

So $f_{10} = \frac{9 \times 10}{2} = 45$

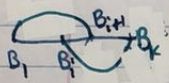
people $\Leftrightarrow$ vertex, friend AB $\Leftrightarrow$ edge AB

18. Pseudo Code:



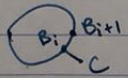

① Define "Expand $A_1 \cdots A_m$ to sub-longest path $B_1 B_2 \cdots B_K$":
Given ~~node~~ vertex A, extend "start" A and "end" A point until no longer extendable. $A^{(k_2)} \leftarrow \cdots \leftarrow A^{(1)} \leftarrow A_1 \cdots A_m \rightarrow A^{(1)} \rightarrow A^{(2)} \cdots \rightarrow A^{(k_1)}$
then rename them $B_1 \rightarrow B_2 \rightarrow \cdots A \cdots \rightarrow B_K$.
$A^{(k_2)}$ A $A^{(k_1)}$

② Define "Form cycle from sub-longest path $B_1, \cdots B_K$":
Traverse $B_1, B_2, \cdots B_K$. Since $B_1 \cdots B_K$ is the sub-longest path, by ① we know if edge $VB_1$ exist, then $V \in \{B_2, \cdots B_K\}$.
Otherwise can extend $B_1 \cdots B_K$ to $V, B_1 \cdots B_K$)




Similarly, if $\exists$ edge $VB_K$, then $V \in \{B_1, \cdots B_{K-1}\}$.
By pigeon hole rule, $\exists\ i \in \{1, 2, \cdots K-1\}$. s.t. $\exists$ edge $B_i B_{i+1}$, $B_i B_K$.
Return the cycle $B_1 \rightarrow B_2 \rightarrow \cdots \rightarrow B_i \rightarrow B_K \rightarrow B_{K-1} \rightarrow \cdots \rightarrow B_{i+1} \rightarrow B_1$

③ Algorithm:
(i) Expand Vertex 1 to sub-longest path $B_1 B_2 \cdots B_K$
(ii) Form cycle from $B_1 B_2 \cdots B_K \cdots B_1$
(iii) Traverse $B_1 B_2 \cdots B_1$, if find vertex $C \notin \{B_1, B_2 \cdots\}$ s.t. edge $CB_i$ exist $(B_i \in \{B_1 B_2 \cdots\})$




then expand $C \rightarrow B_i \rightarrow B_{i+1} \rightarrow \cdots \rightarrow B_{i-1}$ to new sub-longest path $B_1^{(2)} B_2^{(2)} \cdots B_K^{(2)}$.
Otherwise stop.
(iv) repeat (ii)(iii)

Each iteration increase cycle by at least one edge.
$O(n)$ step per iteration. So $O(n^2)$ time.
This algorithm gives a Hamiltonian Cycle. □

# quiz2_liao_zhu

June 9, 2019

Qishi quiz 2 code solution Liao Zhu

Qeustion 5. There is a great solution online, see the 4th answer in the link:

https://math.stackexchange.com/questions/13959/if-a-1-meter-rope-is-cut-at-two-uniformly-randomly-chosen-points-what-is-the-av/13967#13967

Question 10. This is a constant funtion with name "fun", taking an constant pointer p pointing to a constant int type as the paramter, and return a constant pointer to a constant int.

In [14]:
```
# Question 11. Delete in Single linked list

class list_node(object):
    def __init__(self, val=None, next=None):
        self.val = val
        self.next = next

def delect_node(head, val):
    if not head:
        return None
    if head.val == val:
        return head.next

    cur = head
    while cur.next:
        if cur.next.val == val:
            cur.next = cur.next.next
            return head
        cur = cur.next

    stop("No node with val found.")

def print_list(head):
    while head:
        print(head.val, head.next, "\n")
        head = head.next
```

In [17]:
```
l = list_node(3)
h = list_node(1, l)
print_list(h)
```

```
h = delect_node(h, 3)
print_list(h)
```

```
1 <__main__.list_node object at 0x000001DA21C065C0>

3 None

1 None
```

In [ ]: Question 12.

```
# include <iostream>
# include <string>

using namespace std;
class mat
{
    int nrow, ncol;
    float** m;

    mat(int row, int col){
        m = new float * [row];
        for (int p = 0; p < row; p++){
            m[p] = new float[col];
        }
    }

    float get(int i, int j){
        return m[i][j];
    }

    void set(int i, int j, float e){
        m[i][j] = e
    }

    // multiplication
    virtual mat*  multiply(mat* b) = 0;
}
```

Question 13
No. The constructor can not by virtual. Similar function can be achieved by a static function.
Question 14 It is leagal, but not recommended.

In [4]:
```
# Question 15. Given a string, return the longest palindrome subsequence

# My solution using dynamic programming
def longest_palin_subseq(string):
```

```
            if not string:
                return ""

            n = len(string)
            dp = [[("") for _ in range(n)] for __ in range(n)]

            for i in range(n):
                dp[i][i] = string[i]

            for end in range(n):
                for start in range(end - 1, -1, -1):
                    if string[start] == string[end]:
                        dp[start][end] = string[start] + dp[start + 1][end - 1] + string[end]
                    else:
                        pal1 = dp[start][end - 1]
                        pal2 = dp[start + 1][end]
                        dp[start][end] = pal1 if len(pal1) > len(pal2) else pal2

            return dp[0][n - 1]

        # The optimal algorithm for this is called Manacher's Algorithm, O(n) time, O(n) space.
```

```
In [9]: longest_palin_subseq("ACdCCddBA")
```

```
Out[9]: 'AdCCdA'
```

```
In [2]: # Question 16.

        def reverse_sentence(sent):
            return " ".join(sent.split(" ")[::-1])
```

```
In [5]: reverse_sentence("Hello world, tomorrow the brids will sing")
```

```
Out[5]: 'sing will brids the tomorrow world, Hello'
```

```
In [17]: # Question 17. Max profit

         def max_subarray(array):
             if not array:
                 return 0

             cur_sum = 0
             max_sum = 0

             for x in array:
                 cur_sum = max(0, cur_sum + x)
                 if cur_sum > max_sum:
                     max_sum = cur_sum
```

```
            return max_sum

        def max_profit(stocks):
            if len(stocks) <= 1:
                return 0

            diff = [stocks[i + 1] - stocks[i] for i in range(len(stocks) - 1)]
            return max_subarray(diff)
```

In [20]:
```
import numpy as np
d = [-1, 3, -2, 1, -5]
print(max_subarray(d))

stocks = np.cumsum(d)
print(max_profit(stocks))
```

```
3
3
```