HW01 — Q1

Yue Xu

$\because a^{(1)} = h_1(W_1 a^{(0)} + b_1) = h_1(W_1 x + b_1) = \max(W_1 x + b_1, 0)$

$$\therefore W_1 x + b_1 = \begin{bmatrix} 1 & -2 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 3 \\ -1 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 4 \\ -1 \end{bmatrix}$$

$\therefore a^{(1)} = \max(W_1 x + b_1, 0) = [4; 0]^T$

$\because a^{(2)} = h_2(W_2 a^{(1)} + b_2) = W_2 a^{(1)} + b_2$

$$\therefore a^{(2)} = \begin{bmatrix} 2 & 2 \\ 2 & -3 \end{bmatrix} \begin{bmatrix} 4 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ -4 \end{bmatrix} = \begin{bmatrix} 8 \\ 8 \end{bmatrix} + \begin{bmatrix} 0 \\ -4 \end{bmatrix} = \begin{bmatrix} 8 \\ 4 \end{bmatrix}$$

$\therefore$ output $a^{(2)} = [8; 4]^T$

HW01 - Q2

(a) $\because f(x,y) = 4x^2 + y^2 - xy - 13x$

$\therefore \frac{\partial f}{\partial x} = 8x - y - 13$

$\frac{\partial f}{\partial y} = 2y - x$

(b) f is minimum when $\frac{\partial f}{\partial x} = \frac{\partial f}{\partial y} = 0$

$\therefore \begin{cases} \frac{\partial f}{\partial x} = 0 \Rightarrow 8x - y - 13 = 0 \Rightarrow y = 8x - 13 \\ \frac{\partial f}{\partial y} = 0 \Rightarrow 2y - x = 0 \Rightarrow y = \frac{x}{2} \end{cases}$

$\therefore 8x - 13 = \frac{x}{2}$

$15x = 26$

$\therefore x = \frac{26}{15}$

$\therefore y = \frac{x}{2} = \frac{13}{15}$

$\therefore f$ minimum when $(\frac{26}{15}, \frac{13}{15})$

HW01 — Q3

(a) $\because$ hyperplane $w^Tx+b=0$, w is the normal vector

$\therefore$ Suppose x' is the closest point to x_0 on the hyperplane

$\therefore$ x_0-x' must parallel to w

$\therefore$ Suppose $x_0-x'=\lambda w \Rightarrow x'=x_0-\lambda w$

$\therefore$ $w^T(x_0-\lambda w)+b=0$

$\therefore$ $w^Tx_0-\lambda w^Tw+b=0$

$\therefore$ $\lambda\|w\|^2=w^Tx_0+b \Rightarrow \lambda=\frac{w^Tx_0+b}{\|w\|^2}$

$\therefore$ $d=\|x_0-x'\|=\|\lambda w\|=|\lambda|\,\|w\|=\frac{|w^Tx_0+b|}{\|w\|}$ is the distance

(b) pick x_1 on the hyperplane $w^Tx+b_1=0$

$\therefore$ $w^Tx_1+b_1=0$

$\Rightarrow w^Tx_1=-b_1$

$\therefore$ the distance between x_1 and hyperplane

$w^Tx+b_2=0$ is $d=\frac{|w^Tx_1+b_2|}{\|w\|}$

$\therefore$ $d=\frac{|-b_1+b_2|}{\|w\|}=\frac{|b_2-b_1|}{\|w\|}$

HW01 — Q4

(a) ① $\because f(x)=x^2$

$\therefore f(\lambda x+(1-\lambda)y)-\lambda f(x)-(1-\lambda)f(y)$

$=[\lambda x+(1-\lambda)y]^2-\lambda x^2-(1-\lambda)y^2$

$=\lambda^2x^2+2\lambda(1-\lambda)xy+(1-\lambda)^2y^2-\lambda x^2-(1-\lambda)y^2$

$=\lambda(\lambda-1)x^2+(1-\lambda)(-\lambda)y^2+2\lambda(1-\lambda)xy$

$=\lambda(\lambda-1)(x^2+y^2-2xy)$

$=\lambda(\lambda-1)(x-y)^2$

$\because 0<\lambda<1$

$\therefore \lambda>0,\ (\lambda-1)<0,\ (x-y)^2\geq 0 \Rightarrow \lambda(\lambda-1)(x-y)^2\leq 0$

$\therefore f(\lambda x+(1-\lambda)y)-\lambda f(x)-(1-\lambda)f(y)\leq 0$

$\therefore f(\lambda x+(1-\lambda)y)\leq \lambda f(x)+(1-\lambda)f(y)$

$\therefore f(x)=x^2$ is a convex function

② $\because f(x)=x^3$

$\therefore$ suppose $x=-1,\ y=0,\ \lambda=\frac{1}{2} \Rightarrow f(x)=-1,\ f(y)=0$

$\therefore f(\lambda x+(1-\lambda)y)=f(\frac{1}{2}\times(-1)+(1-\frac{1}{2})\times 0)=f(-\frac{1}{2})=-\frac{1}{8}$

$\lambda f(x)+(1-\lambda)f(y)=\frac{1}{2}\times(-1)+(1-\frac{1}{2})\times 0=-\frac{1}{2}$

$\because -\frac{1}{8}>-\frac{1}{2}$

$\therefore f(x)=x^3$ is not a convex function

(b) $\because f(x)=x^TAx$

$$\therefore f(\lambda x+(1-\lambda)y) = (\lambda x+(1-\lambda)y)^T A(\lambda x+(1-\lambda)y)$$

$$=(\lambda x^T+(1-\lambda)y^T)A(\lambda x+(1-\lambda)y)$$

$$=\lambda x^T A\lambda x+\lambda x^T A(1-\lambda)y+(1-\lambda)y^TA\lambda x + (1-\lambda)y^TA(1-\lambda)y$$

$$=\lambda^2 x^TAx+\lambda(1-\lambda)x^TAy+\lambda(1-\lambda)y^TAx+(1-\lambda)^2y^TAy$$

$$=\lambda^2x^TAx+\lambda(1-\lambda)x^TAy+\lambda(1-\lambda)y^TAx+(1-2\lambda+\lambda^2)y^TAy$$

$$\lambda f(x)+(1-\lambda)f(y)=\lambda\cdot x^TAx+(1-\lambda)y^TAy$$

$$\therefore f(\lambda x+(1-\lambda)y)-\lambda f(x)-(1-\lambda)f(y)$$

$$=\lambda(\lambda-1)x^TAx+\lambda(1-\lambda)x^TAy+\lambda(1-\lambda)y^TAx+\lambda(\lambda-1)y^TAy$$

$$=\lambda(\lambda-1)(x^TAx-x^TAy-y^TAx+y^TAy)$$

$$=\lambda(\lambda-1)(x^T-y^T)A(x-y)$$

$$=\lambda(\lambda-1)(x-y)^TA(x-y)$$

$\because$ A is a positive semi-definite matrix

$\therefore (x-y)^TA(x-y)\geq 0$, $\lambda>0$, $(\lambda-1)<0$

$\therefore \lambda(\lambda-1)(x-y)^TA(x-y)\leq 0$

$\therefore f(\lambda x+(1-\lambda)y)-\lambda f(x)-(1-\lambda)f(y)\leq 0$

$f(\lambda x+(1-\lambda)y)\leq\lambda f(x)+(1-\lambda)f(y)$

$\therefore f(x)=x^TAx$ is a convex function

Q5 Simulate tossing a biased coin (a Bernoulli trial) where P[HEAD] = 0.70.

(a) Count the number of heads in 50 trials. Record the longest run of heads.

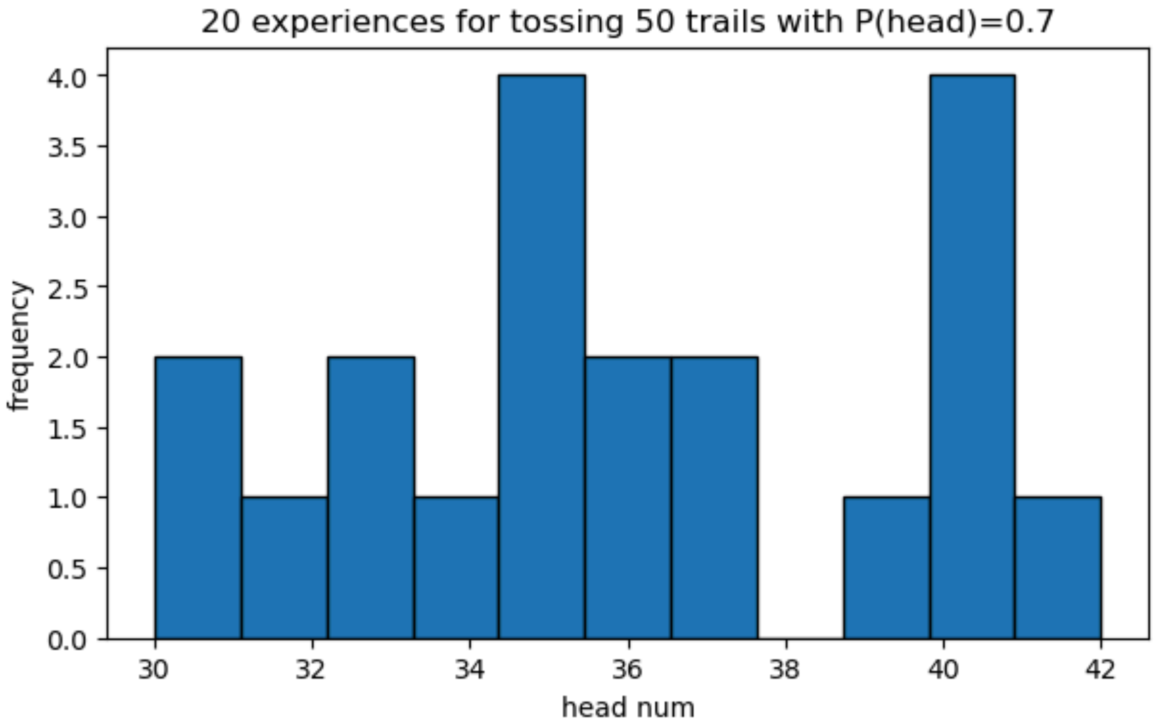
In [1]:
```
import random
import matplotlib.pyplot as plt

def coin_1_exp(head_prob, trail):
    #count the number of head
    num_head =0
    #longest run of head
    longest_head =0
    #record current run of head
    still_head =0
    for i in range(trail):
        a =random.random()
        #print(a)
        if a <=head_prob:
            #print("yes")
            num_head += 1
            still_head += 1
            longest_head = max(longest_head, still_head)
        else :
            #print("no")
            still_head = 0
        #check the last run of head
        longest_head = max(longest_head, still_head)
    return num_head, longest_head

#50 trails
num_head, longest_head = coin_1_exp(0.7, 50)
print("Number of Heads:", num_head)
print("Longest run of Heads:", longest_head)
```

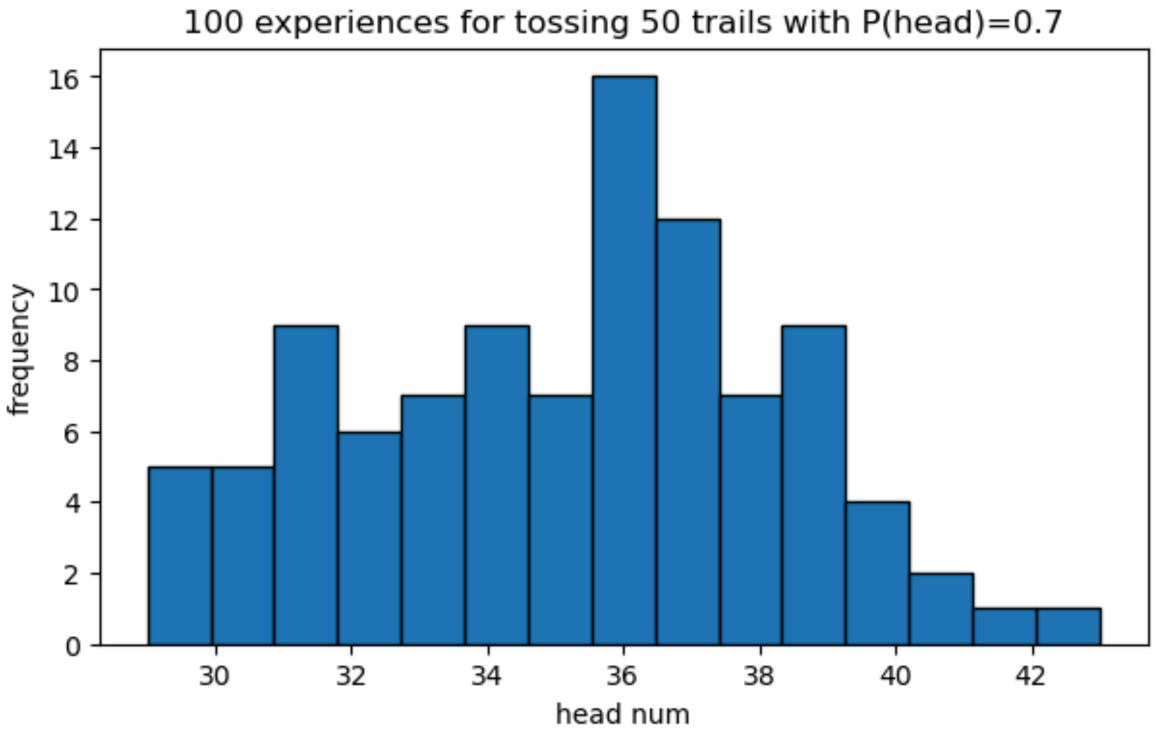
```
Number of Heads: 31
Longest run of Heads: 9
```

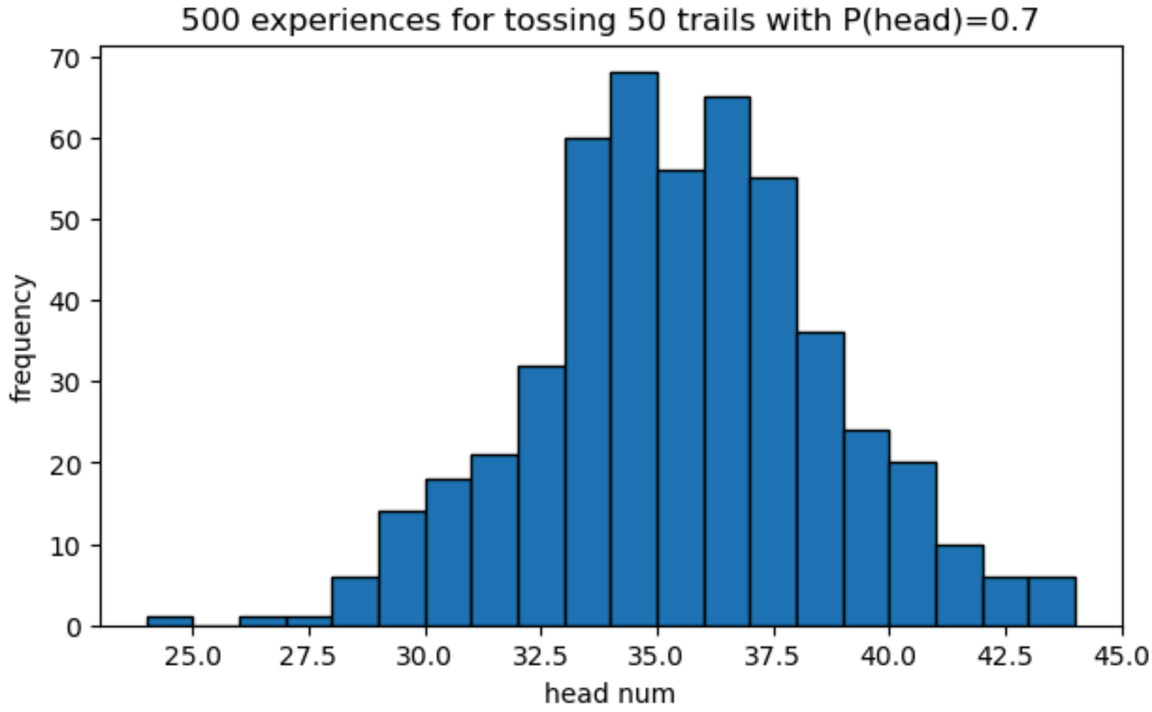
(b) Repeat the 50-flip experiment 20, 100, 200, and 1000 times. Use matplotlib to generate a histogram showing the observed number of heads for each case. Comment on the limit of the histogram.

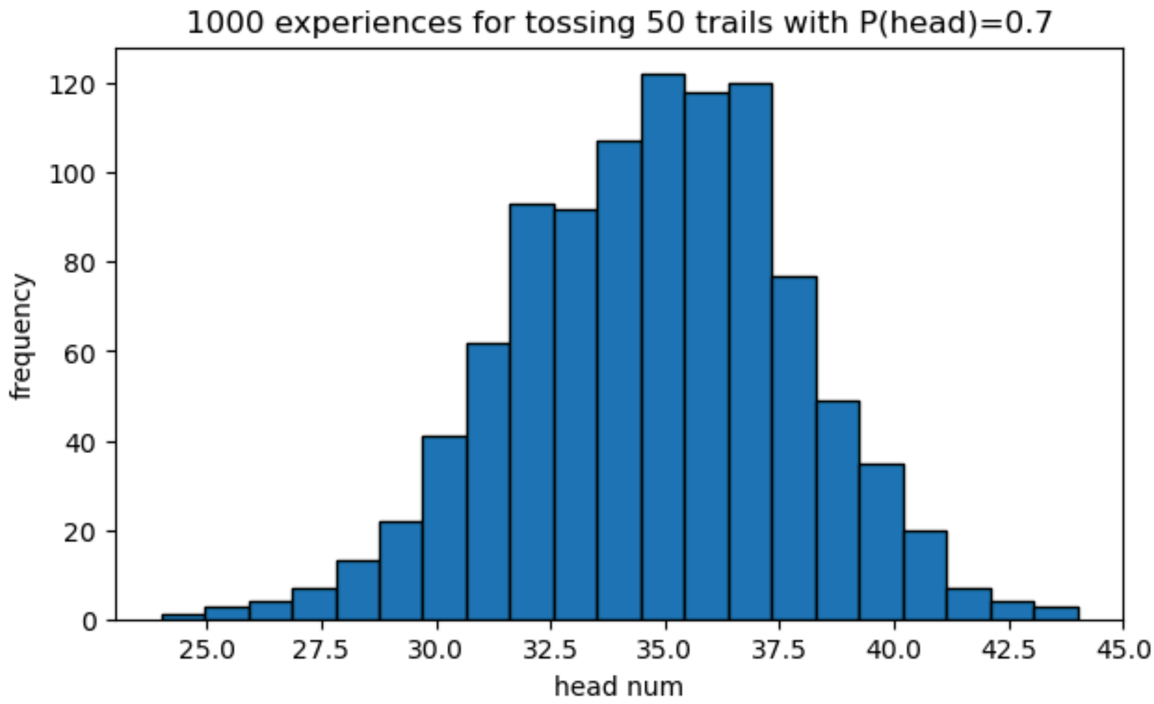
In [5]:
```
#repeat n times 50 trails
def repeat_exp(n_exp, head_prob, trail):
    #record every time's number of head
    exp_head = []
    for exp in range(n_exp):
        num_head, longest_head = coin_1_exp(head_prob, trail)
        exp_head.append(num_head)
    #draw the histogram
    plt.figure(figsize = (7,4))
    #set the number of bin = the number of different num_head
```

```
    num_bin = len(set(exp_head))
    plt.hist(exp_head, bins = num_bin, edgecolor = 'black')
    #plt.hist(exp_head, bins = trail, range = (0, trail), edgecolor = 'black')
    plt.title("%d experiences for tossing %d trails with P(head)=%3.1f" %(n_exp,
    plt.xlabel("head num")
    plt.ylabel("frequency")


exp_times = [20, 100, 500, 1000]
for i in exp_times:
    repeat_exp(i, 0.7, 50)
```







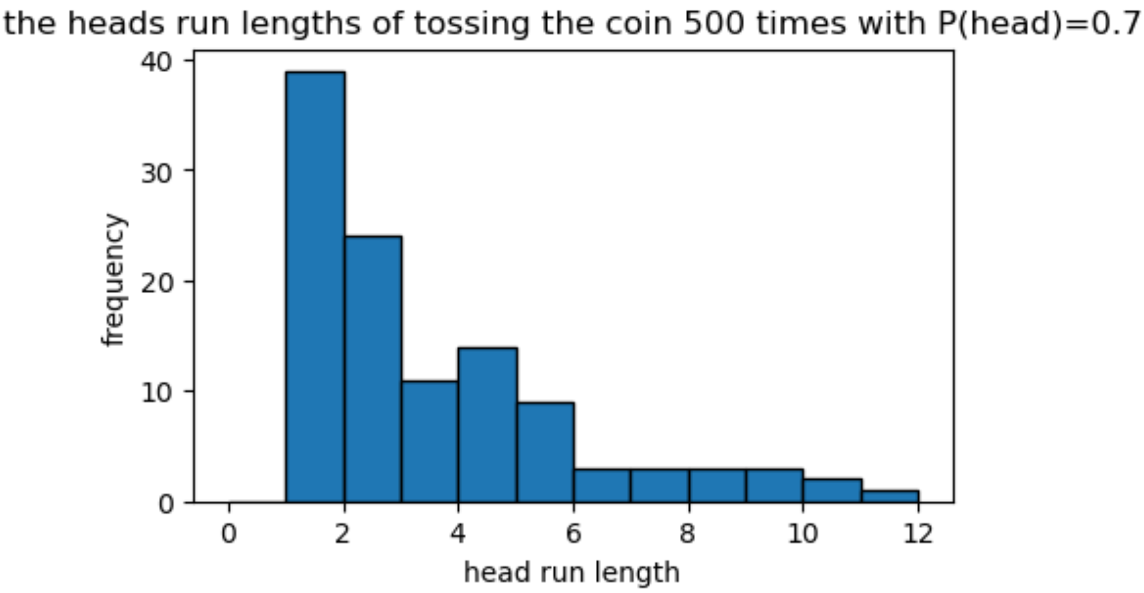


the limit of the histogram

1. As the number of trials increases, the histogram's shape gradually approaches a normal distribution. This trend is consistent with the Central Limit Theorem. So small sample sizes may lead to inaccuracies.
2. Different "bins" can affect the histogram's shape, impacting the interpretation of the data distribution.

(c) Simulate tossing the coin 500 times. Generate a histogram showing the heads run lengths.

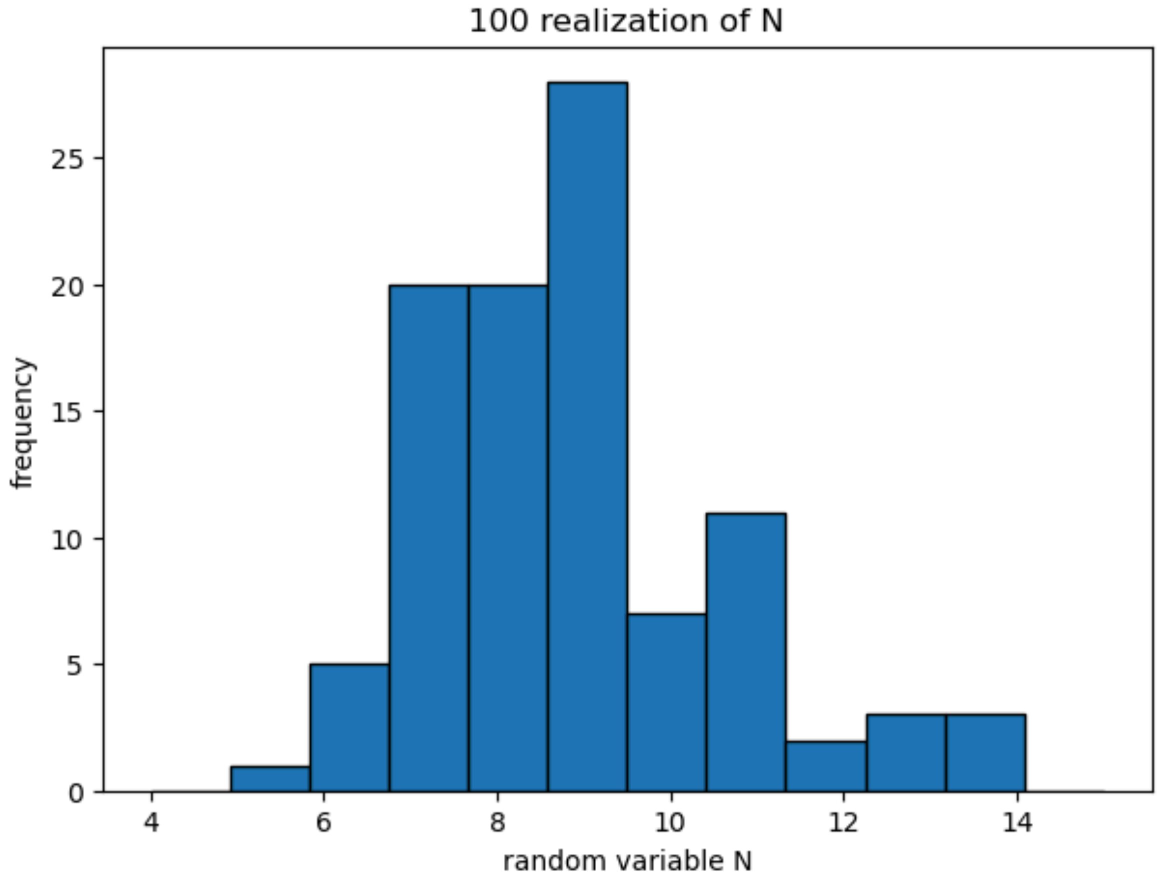
In [10]:
```
#record the length of every run of head
def run_length(head_prob, trail):
    #record length of run of head
    length_run = []
    #record current run of head
    still_head = 0
    for i in range(trail):
        a = random.random()
        if a <= head_prob:
            still_head += 1
        else :
            if still_head != 0:
                length_run.append(still_head)
            still_head = 0
    #record the min and max of N(to the range of x of histogram)
    max_longest_head=max(length_run)
    plt.figure(figsize = (5,3))
    plt.hist(length_run, bins = max_longest_head + 1, range = (0, max_longest_he
    plt.title("the heads run lengths of tossing the coin 500 times with P(head)=
    plt.xlabel("head run length")
    plt.ylabel("frequency")

run_length(0.7, 500)
```

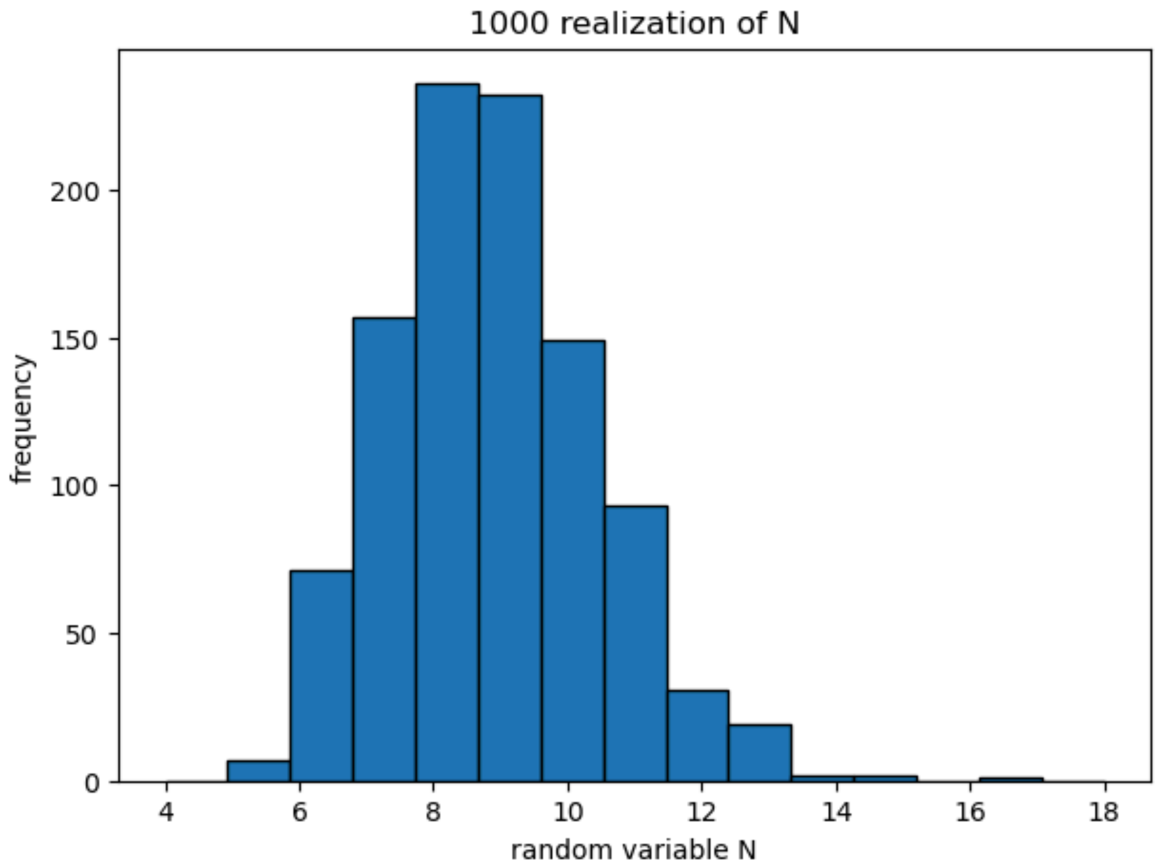


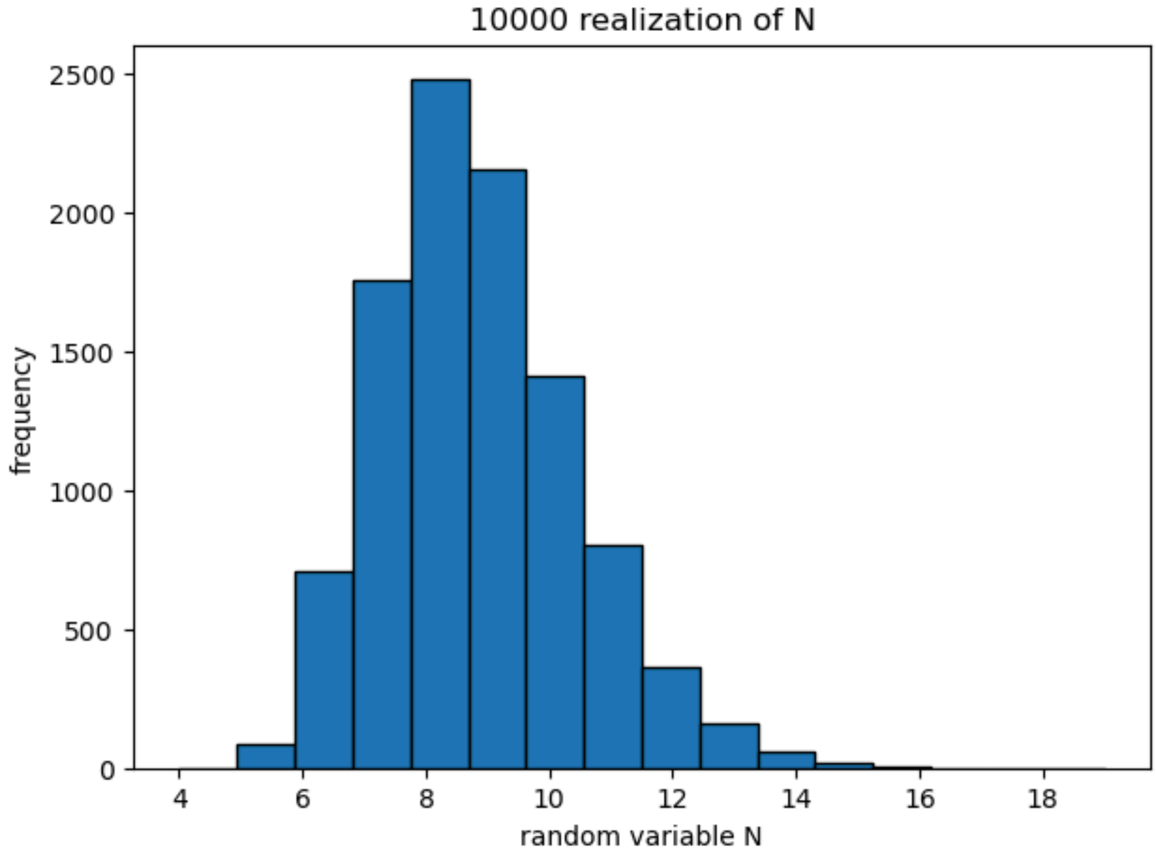
Q6

In [6]:
```
import numpy as np
import matplotlib.pyplot as plt
import random
#get a N
def a_random_variable(sum_limit):
    #N the number of random
    N = 0
    #record current sum of random
    sum_of_random = 0
    #as long as it's not >4, add another random number
    while sum_of_random <= sum_limit:
        sum_of_random += random.random()
        N += 1
    return N

#get more N
def more_random_variable(n_variable, sum_limit):
    #sum of all N(to calculate the expectatioin)
    sum_all_variable = 0
    #record all N
    list_N = []
    for i in range(n_variable):
        N = a_random_variable(sum_limit)
        list_N.append(N)
        sum_all_variable += N
    #record the min and max of N(to the range of x of histogram)
    max_N=max(list_N)
    min_N=min(list_N)

    #draw histograms
    plt.figure(figsize=(7,5))
    hist_range = max_N - min_N + 3
    plt.hist(list_N, bins = hist_range, range=(min_N - 1, max_N + 1), edgecolor
    plt.title("%d realization of N" %(n_variable))
    plt.xlabel("random variable N")
    plt.ylabel("frequency")

    #expectation
    expectation_N = sum_all_variable/n_variable
    print("the expectation of %d random variable is :E[N] = %f" %(n_variable, ex


n_variable = [100, 1000, 10000]
for i in n_variable:
    more_random_variable(i, 4)
```

```
the expectation of 100 random variable is :E[N] = 8.830000
the expectation of 1000 random variable is :E[N] = 8.743000
the expectation of 10000 random variable is :E[N] = 8.669800
```


100 realization of N
frequency
random variable N
0
5
10
15
20
25
4
6
8
10
12
14


1000 realization of N
frequency
random variable N
0
50
100
150
200
4
6
8
10
12
14
16
18


10000 realization of N
frequency
random variable N
0
500
1000
1500
2000
2500
4
6
8
10
12
14
16
18