

```

%Assignment_1 Problem A
clear all;close all;clc;%close all of windows
n=50;%toss the coin for 50 times
m=0;%the toss index is begin at 0 for calculate the HEAD times
%in this condition, assume the diameter of one coin is 1cm
for i=1:n;%begin to toss the coin
    x=randperm(2)-1;
    y=x(1);
    if y==0;
        m=m+1;
    end
end
l_head=1*m;%calculate the total length of HEAD
fprintf('head shows: %f\n',m);
fprintf('the length of head in cm: %f\n',l_head);
%%
%%problem A (solution 2)
input=50;% toss a coin for 50 times
r=rand(1,50);% produce 50 numbers randomly between 0 to 1
%which used for produce probability
x=0;
for i=1:input
    if r(i)>0.5;%assume the random number value is bigger than 0.5,
then
        %we think it got HEAD
        x=x+1;%calculate the number of HEAD each time
    end
end
l_head=x*1;
fprintf('head shows: %f\n',x);
fprintf('the length of head in cm: %f\n',l_head);
%%
%%Problem B
clear all;close all;clc;
repetition=1000;%in this experiment we try for 1000 times
input=50;%every coin toss for 50 times in each experiment
heads=zeros(1,repetition)% total number of heads for each experiment
longests=zeros(1,repetition)%longest length of coin per/ex
head_1=zeros(1,repetition)% toss number until 1st head/ex
head_2=zeros(1,repetition)% toss number until 2st head/ex
head_3=zeros(1,repetition)% toss number until 3st head/ex
head_4=zeros(1,repetition)% toss number until 4st head/ex
head_2_sequence=zeros(1,repetition)% toss number until 2 head appear/ex
head_3_sequence=zeros(1,repetition)% toss number until 3 head appear/ex
head_4_sequence=zeros(1,repetition)% toss number until 2 head appear/ex
for i=1:repetition% run 1000 times
    head=0;% begin index/ex
    longest=0;% the longest length of heads/ex
    length=0;% current total length/ex
    for j=1:input % 50 times/ex and (1000*50)
        head_or_back=rand();%randomly produce number between 0 to 1
        if(head_or_back>0.5)% use >0.5 to replace the head
            head=head+1;%accumulate heads
            length=length+1;%accumulate length in one ex
            if(length==2&&head_2_sequence(i)==0)% 2 heads in sequence
                head_2_sequence(i)=j;%get 2 heads in per ex(50 toss), for
per run in 1000 times
            end
        end
    end
end

```

```

        end
        if (length==3&&head_3_sequence(i)==0)% 3 heads in sequence
            head_3_sequence(i)=j;
        end
        if (length==4&&head_4_sequence(i)==0)% 4 heads in sequence
            head_4_sequence(i)=j;
        end
    else
        if (length>longest)% keep the longest update with length
            longest=length;% give total length to longest, and make
sure always longest
        end
        length=0;%start new coin toss
    end
    if (head==1)% got 1 head
        head_1(i)=j;
    end
    if (head==2)% got 2 head
        head_2(i)=j;
    end
    if (head==3)% got 3 head
        head_3(i)=j;
    end
    if (head==4)% got 4 head
        head_4(i)=j;
    end

    if (length>longest)% keep the longest update with length
        longest=length;% this is the last length and end with
longest (total length)
    end
    heads(i)=head; %total heads number for each run (should be 1000)
    longests(i)=longest;%total heads length for each run (should be
1000)
end

%plot the frequency distrabution of 1000 runs/one run with 50 toss
figure(1)
hist(heads);
title('total heads of 1000 run');
figure(2)
hist(longests);
title('total length of 1000 run');
figure(3)
hist(head_1);
title('the number of tosses until 1st head for 1000 run');
figure(4)
hist(head_2);
title('the number of tosses until 2st head for 1000 run');
figure(5)
hist(head_3);
title('the number of tosses until 3st head for 1000 run');
figure(6)
hist(head_4);
title('the number of tosses until 4st head for 1000 run');
figure(7)
hist(head_2_sequence);

```

```

title('the number of tosses needed until there are 2 heads');
figure(8)
hist(head_3_sequence);
title('the number of tosses needed until there are 3 heads');
figure(9)
hist(head_4_sequence);
title('the number of tosses needed until there are 4 heads');

```

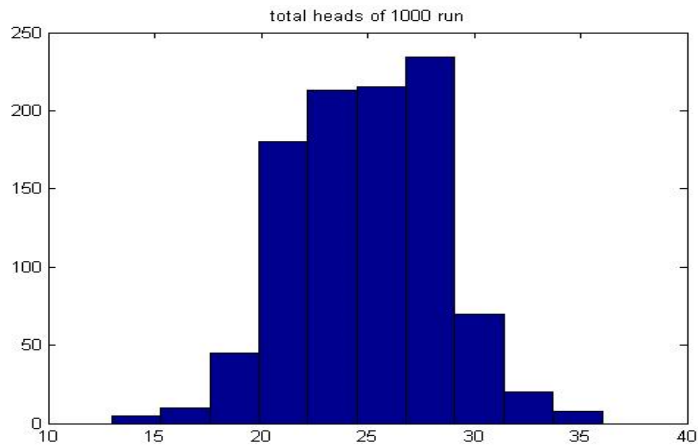


figure1

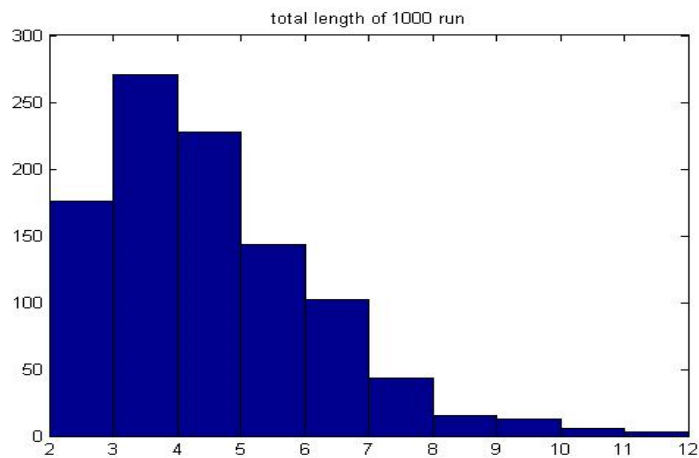


figure 2

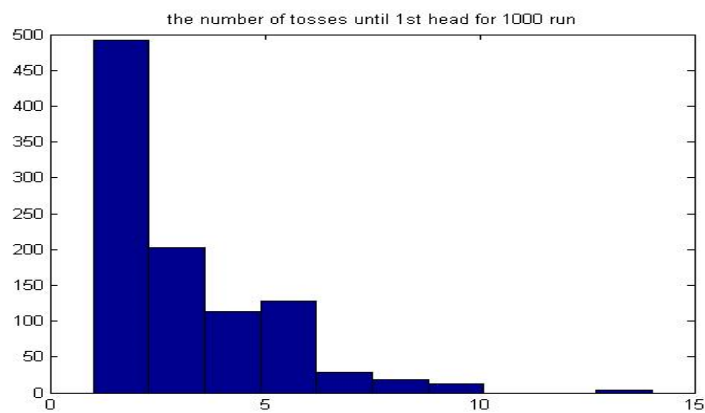


figure3

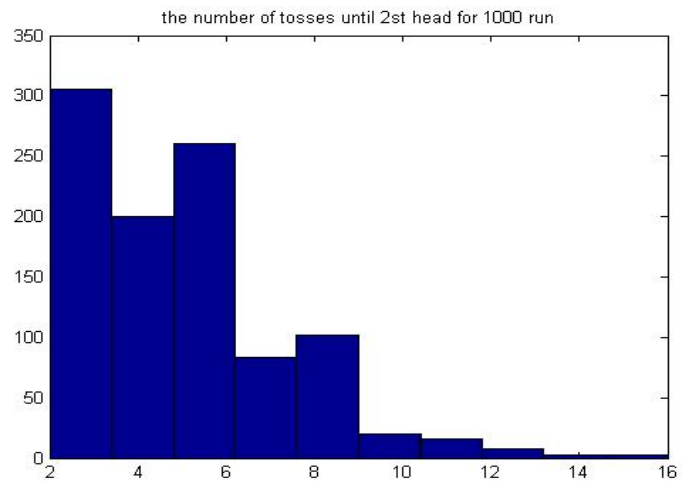


figure4

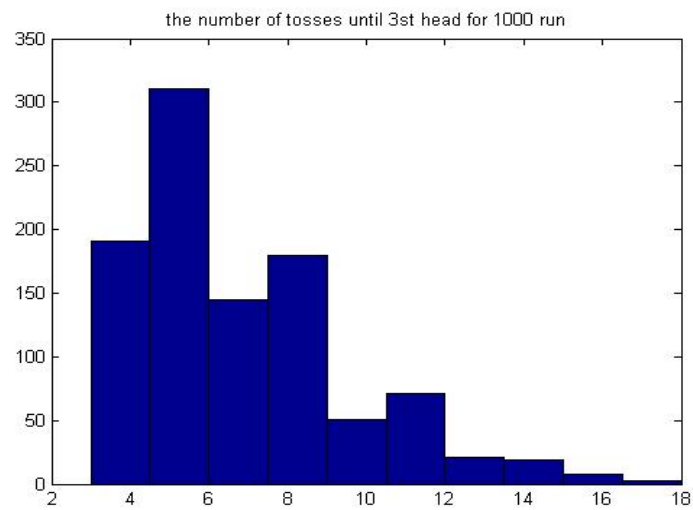


figure5

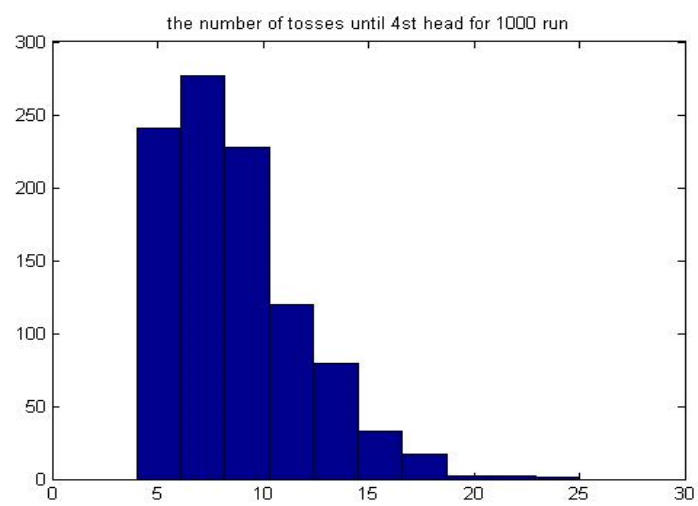


figure6

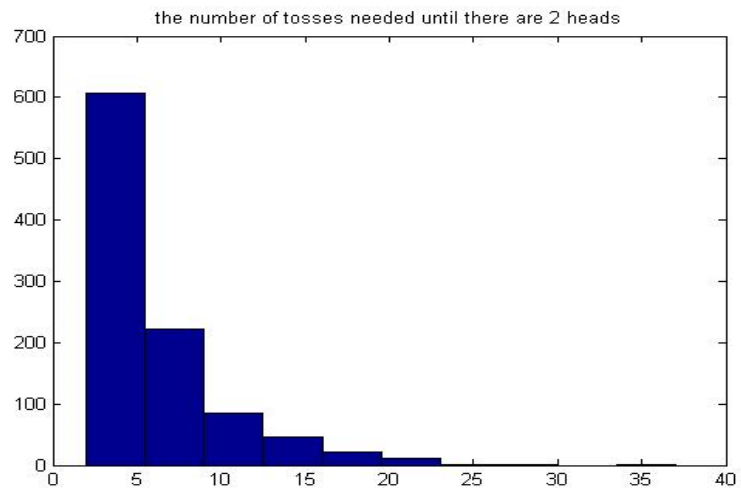


figure7

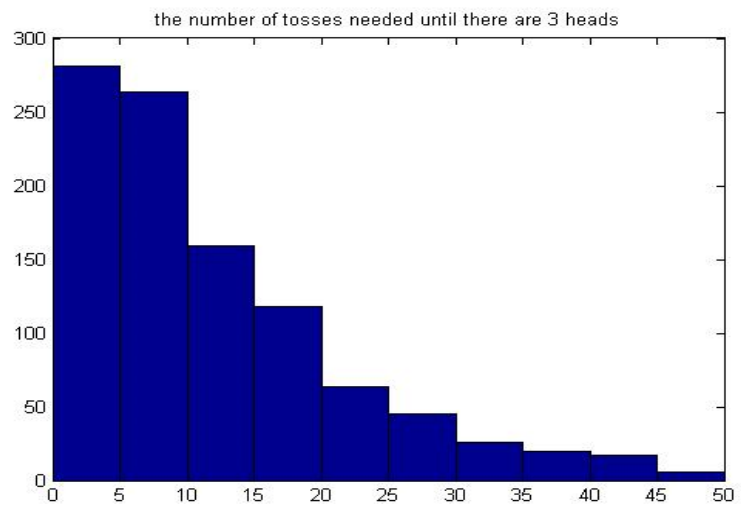


figure8

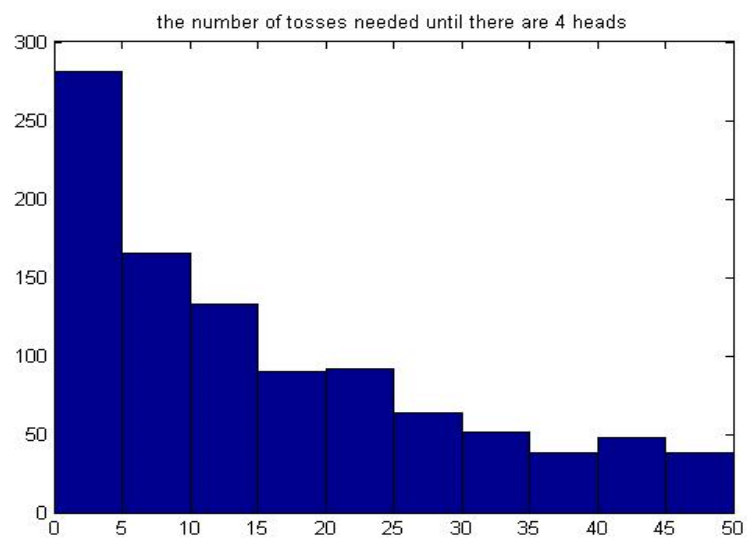


figure9